ASSESSING THE ROBUSTNESS OF SYMMETRIC CIPHERS AND LSB STEGANOGRAPHIC TECHNIQUE UNDER PROPOSED NOVEL BITSUM ATTACK

A Thesis Submitted

to

LOVELY PROFESSIONAL UNIVERSITY

For the Award of

DOCTOR OF PHILOSOPHY
in
COMPUTER APPLICATIONS

By
AMANDEEP

Guide
Dr. G. Geetha

Faculty of Technology and Sciences
Lovely Professional University
Punjab(India)
April, 2016

Certificate

I certify that Ms. Amandeep has prepared her thesis entitled **'Assessing the robustness of Symmetric Ciphers and LSB Steganographic Technique under Proposed Novel Bitsum Attack'** for the award of Ph.D. Degree of Lovely Professional University under my guidance. She has carried out the work at the Department of Computer Applications, Lovely Professional University.

Advisor

Dr. G. Geetha

Professor and Associate Dean

School of Computer Applications

Lovely Professional University,

Punjab.

Date:

Declaration

I declare that this thesis entitled **'Assessing the robustness of Symmetric Ciphers and LSB Steganographic Technique under Proposed Novel Bitsum Attack'** has been prepared by me under the guidance of Dr. G. Geetha, Professor and Associate Dean of Lovely Professional University. No part of this thesis has formed the basis for the award of any degree or fellowship previously.

Amandeep

School of Computer Applications,

Lovely Professional University,

Punjab.

Date:

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Dr. G. Geetha for the continuous support through her immense knowledge, motivation and her unflinching patience during this study. She was always there with her guidance, sagacious advice and encouragement during the research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D study.

A special thanks to members of the panel of SOTA and Research Degree Cell for providing critical suggestions on the progress reports and help me in keeping my research work on track.

Mr. Rajeev Sobti, is one of the best teachers and a great mentor that I have had in my life. I am indebted to him for his continuous encouragement and guidance.

I am grateful to the Management of LPU - Mr. Ashok Mittal, Ms. Rashmi Mittal, and my seniors - Dr. Lovi Raj Gupta and Dr. Sanjay Modi - for all the support and encouragement they have been rendering from so long.

I am also thankful to Dr. Tirlochan Singh Sangha for encouraging the use of correct grammar and consistent notation in my writings and for carefully reading and commenting on countless revisions of this manuscript.

I would like to acknowledge Mr. Amandeep Singh Sandhu, Mr. Puneet Kaushal, Mr. Rahul Saha and Mr. Gulshan for numerous discussions and practical advices on related topics that helped me improve my knowledge in the area.

I am thankful to Mr. Rishi Chopra and Ms. Rajni Bhalla to support me in handling my administrative responsibilities on work.

A special thanks to my family. Words cannot express adequately how grateful I am to my mother-in law, father-in-law, my mother, and my father for all of the sacrifices that they have made on my behalf. Their prayers for me were always there to sustain me thus far. I would also like to thank all of my friends who motivated me to strive towards my goal and supported me in seeing through this study and writing.

Last but not the least, I express my heartfelt thanks to my husband, Mr. Amit Chauhan, and my daughter, Ms. Ishita Chauhan, for their unconditional love, support and unaccounted sacrifices, without which, it could never have been possible for me to accomplish this task.

Contents

Abstract

Digitization has brought in unprecedented exchange, transfer and teleportation of information across globally placed senders, receivers and accessors of information. This involves a serious concern to ensure that the information is accessed by the authorized users and receivers and without any eaves dropping or tampering of the information during transfer. Information security as well as network security depends on the three pillars: confidentiality, integrity and availability. The very first aspect of confidentiality is provided by different cryptographic algorithms. This domain of cryptography, which introduces the concepts of block ciphers, stream ciphers, symmetric and asymmetric key ciphers has invited serious attention including a number of researchers. Each of the algorithms has their own degree of strength. To analyze the strength of a cipher different cryptanalysis algorithms have been introduced so far. These algorithms tend to uncover the weaknesses in these algorithms for further improvements. Developing the security ciphers is very important, but developing the cryptanalytic algorithms is equally important so the adversaries could not harm the important information.

The first objective of this thesis is to develop an algorithm to check the robustness of these security systems. Looking at the other techniques of cryptanalysis, this study aims to introduce a new cryptanalysis method which uses the bit values of the keys, cipher texts and plain texts. The proposed algorithm under the present study namely Bitsum algorithm[1] is designed to exploit the fundamental property of the ciphers to check whether there exists any correlation between the key and the cipher text, plain text and cipher text.

The second objective of this thesis is to check the applicability of Bitsum algorithm on XOR cipher[2].

---

[1] Amandeep, G. Geetha, "Bit Sum Attack", The Security Journal, vol.35, pp.21-22, Fall 2011.

[2] G. Geetha, Amandeep, "Implication of Bit Sum Attack on XOR", in Proc. 2nd National Conference on Emerging Trends In Computer Application, pp. 47-50, Feb-2012.

XOR cipher was chosen to conduct this first experiment because more than 60% of the ciphers uses XOR as a part of their basic operation. During this experimentation, it was found that Bitsum Algorithm poses a threat on this cipher. There exists a correlation between Bitsum of the Key and Bitsum of the Ciphertext. Our experiment showed that in 72% of the cases correlation exists. So, this became the basis to conduct the experiments to achieve the objective of testing the applicability of bit sum algorithm on chosen block ciphers such as reduced TEA, TEA, XTEA, FEAL, Blowfish and AES.

The third objective to implement the proposed algorithm on chosen ciphers to find the correlation to discover their weaknesses, if any was carried out. The chosen symmetric ciphers under study were tested for following properties

a. Confusion Property
b. Diffusion Property
c. Constant Factor coefficient
d. Effect of ciphertext Bitsum alone on plaintext Bitsum
e. Some pattern of the ciphertext with respect to the plaintext and key.

Reduced key Tiny Encryption Algorithm[3] showed that if the Bitsum of the Key is either less than 14 or greater than 51, then the Bitsum of the ciphertext remains constant. The second result from the same experimentation showed that reduced key TEA is secure only if the Bitsum of the key lies between 14 and 51.

TEA, XTEA, FEAL, BLOWFISH and AES were also analyzed through Bitsum algorithm. From the analysis of TEA and XTEA, a particular set of weak keys were found. FEAL, BLOWFISH and AES were able to withstand Bitsum attack.

[3] Amandeep and G. Geetha, " On the Security of Reduced Key Tiny Encryption Algorithm" in Proc. International Conference on Computing Sciences, Punjab, 2012,pp.323-326. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6391697&isnumber=6391635

The data generated during the experiments were thoroughly analyzed in SPSS[4] and value of the correlation factors were calculated. Results and analysis of the properties under testing revealed the strength of the chosen ciphers.

In the present research it has been found that Blowfish and FEAL are secure algorithms as they hold confusion and diffusion properties and withstand Bitsum attack. TEA and XTEA are less secure in this regard. The versions of AES, by virtue of their confusion and diffusion property, are proved to be strong enough to provide security. Out of all the versions of AES, AES-256 found to be the strongest candidate[5].

There are various cryptanalytic techniques which discover various interesting properties of the ciphers. The present research identified some interesting pattern of the output(Ciphertext) with respect to the input(Plaintext) and Key. Key pattern theorems[6] are developed during the analysis of TEA and XTEA. Particularly, a specific pattern of the keys produce a particular pattern of Bitsum of ciphertexts.

The fourth objective of this thesis deals with the analysis of the applicability of Bitsum algorithm on LSB steganographic technique. The following interesting results reveal that Bitsum algorithm poses a threat to LSB steganography technique. It is found that there is correlation between Bitsum of original image and Bitsum of stego – image, the value of correlation coefficient decreases when Bitsum of the secret message is increased and the variation in the value of correlation coefficient is less in case of complicated images whereas single shade images shows more variation when we change the Bitsum of the secret message[7].

---

[4] Amandeep and G. Geetha, "Analysis of Bitsum Attack on Block Ciphers", Discrete Journal of Mathematical Sciences and Cryptography. [Accepted for Publication].

[5] Amandeep and G. Geetha, "On the Complexity of Algorithms affecting the security of TEA and XTEA", Far East Journal of Electronics and Communication. [Accepted for Publication].

[6] Amandeep and G. Geetha, "Implications of Bitsum attack on tiny encryption algorithm and XTEA", Journal of Computer Science, Vol. 10, pp. 1077-1083 , 2014.

[7] Amandeep and G. Geetha, "Analysis the applicability of Bitsum algorithm on LSB steganography technique", [submitted for publication].

In future, it is planned to test the applicability of Bitsum on other symmetric ciphers, hardware implementation of Bitsum algorithm and testing the applicability of Bitsum algorithm on other steganographic techniques.

List of Tables

List of Figures

Chapter 1: Introduction

"The beginning is the most important part of the work."

— Plato

1.1 Cryptology

Cryptology is a wider term that covers two disciplines i.e. Cryptography and Cryptanalysis [1]. Cryptography is an art that has evolved into the science of hiding message in such a way that only the intended receiver who has certain information or the key to decode the message to read/listen it [2]. The exchange of communication/messages between sender and a receiver is prone to eavesdropping and tampering. To ensure that the information a sender is sending, should be communicated to the actual receiver, even in the presence of some adversary [3]. Cryptanalysis is a technique by which some weaknesses are found in the cryptosystems to deduce the plaintext [4]. This technique is used to make the cryptosystems secure.

1.2 Cryptography

Cryptography [5] is an ancient art. It was used to hide and protect the vital message/war plan from the enemies during wars. Cryptography consists of two main techniques: Encryption and Decryption. Plaintext refers to the message to be sent, which is encrypted using a key(secret or public) to make it a ciphertext (the encoded text to be sent). Further, cryptography has three types:

- Symmetric Cryptography: In this a single secret key is used for encryption and decryption. The sender encrypts the message using this secret key and the receiver decrypts that message using the same key. This means that both the sender and the receiver should know the complete information about the key. Now, sharing of this secret key between two communicating parties, who are away from each other and others from knowing it is a big issue.

Symmetric cryptography involves/makes use of two types of ciphers i.e. either block ciphers or stream ciphers. Block ciphers operate on different modes: ECB - Electronic Codebook mode, CBC - Cipher Block Chaining mode, CFB - Cipher Feedback mode, OFB - Output Feedback mode.

- Asymmetric Cryptography: It was basically developed to deal with the problem of key distribution in symmetric key cryptography [6]. Another purpose to develop it was to devise a method of digital signatures for the authentication purpose. In this type of cryptography, two keys are used. One is used to encrypt the message and another is used to decrypt the message. One key is a secret key while the other is a public key. The public key is known to all.

  So, anyone may acquire the public key, encrypt the message and send it to the intended recipient. The person having the secret key can only decrypt the message. Even the sender can not decrypt the message sent by him. This solves the problem of key distribution.

  A digital signature is another application, and this can also be attained by asymmetric cryptography. The person, having the secret key, can encrypt the message using that key. This becomes his/her signatures. Anyone having the public key can decrypt it and can verify that this particular document is digitally signed by the owner of the secret key.
- Hash functions: This is a one way cryptographic technique [7]. No key is used to encrypt the message. When we use hash function with our message, it creates a message digest. Now this message digest can not be decrypted. This message digest can be sent along with the document. The sender can attach the message digest with the message. Now, the sender will encrypt the set of message and message digest.

  On decrypting the set, the receiver will separate the message and the digest and create one more message digest with the received message. The received message will be compared with the newly created message digest. If both are same, then the receiver will keep the message, otherwise he may discard the received document for it may not be the original. The applications of

hash function include password hashing, digital signatures, file integrity verification etc.

Figure 1 [8] summarizes the techniques explained above.

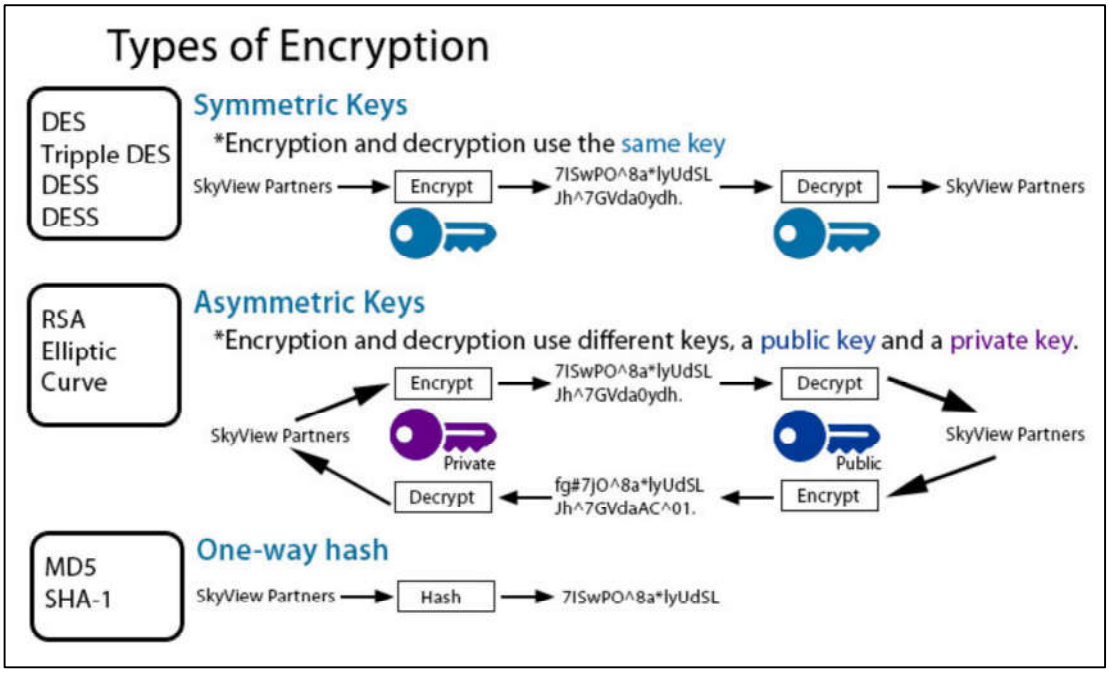

Figure 1: Symmetric Cryptography (a), Asymmetric Cryptography (b) and Hashing(c)

These techniques are used in cryptography to make the information secure. But the adversaries with the malicious intentions are also out trying to break the system. So, there has to be some other technique as well, which continuously makes the systems robust. One such technique is cryptanalysis.

1.3 Cryptanalysis

Cryptanalysis is the technique through which an adversary tries to break a cipher or get the key or gain any clue about the key [9]. One can get the plaintext by stealing it or purchasing it or access to it can be achieved through the cryptanalytic attacks. [10] But in academic research, the approach is different. It may also lead to finding a weakness in the cipher so that it can be exploited with the complexity less than brute force. So there exist so many types of cryptanalytic attacks. The broad classification of

cryptanalytic attacks is given below. The details of these attacks are discussed in Chapter 2: Literature Review.

1.4 Classification of Cryptanalytic Attacks

1.4.1 Classical Cryptanalysis

- Ciphertext only Attack
- Known Plaintext Attack
- Chosen Plaintext Attack.
- Chosen Ciphertext Attack
- Adaptive Chosen Plaintext and Adaptive Chosen Ciphertext Attacks
- Rubber Hose Attack (Cryptanalysis)
- Frequency Analysis

1.4.2 Cryptanalysis/Attack on Symmetric Algorithms

- Differential Cryptanalysis
- Linear Cryptanalysis
- Integral Cryptanalysis
- Statistical Cryptanalysis
- Mod-n Cryptanalysis
- Slide Attack

1.4.3 Other Cryptanalytic Attacks

- Side Channel Attacks
- Brute Force Attacks
- Meet-in-the-Middle Attack
- Birthday Attack
- Man in the Middle Attack
- Differential Power Analysis
- Cache Attack

1.5 An Overview of Block Ciphers

The present study is based on block ciphers, so this section is devoted to provide a wider view about block ciphers. There are various block ciphers which are still being used worldwide for the vigorous security. Block ciphers independently provide confidentiality. These may be the key components for providing integrity, authentication or even symmetric key digital signatures also. One block cipher, highly secure, may not be suitable for all applications, therefore many block ciphers are developed to meet the various types of requirements [11].

As block ciphers process data in the form of large blocks using the data in the form of bytes or words (e.g. 128 bit), so they can be easily standardized as per the standards of today's computer networks. Losing one block of data does not hamper the security of the entire data. However there is one disadvantage; these ciphers are not able to hide data patterns [12]. But this disadvantage can be overcome by using a different mode of operation of block ciphers.

1.5.1 The Block Cipher Modes of Operations

Modes of operation of block ciphers are the ways in which blocks of data are encrypted using the same cipher. Modes simply add some feature to the existing cipher like feedback etc., so that the data pattern in block ciphers could be concealed [2] [11]. There are four basic modes of operation:

1.5.1.1 Electronic Code Book (ECB)

This is the simplest mode of operation. In this mode each block is encrypted or decrypted individually. Same ciphertext will be generated with the same plaintext. This mode is to be used very carefully since this is vulnerable to known-plaintext attacks as there is the possibility of guessing the text. This mode of operation is explained in Figure 2 [13].

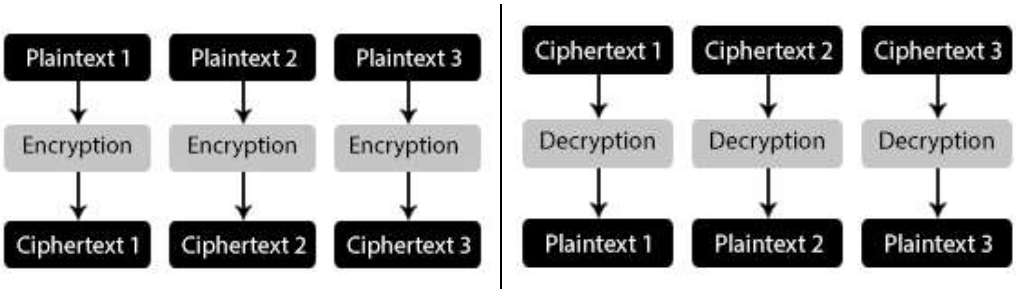

Figure 2 : ECB Encryption (a) and Decryption (b)

1.5.1.2 Cipher-Block Chaining

This mode eliminates the problem of ECB mode by incorporating the chaining technique. The output of the previous block will be XOR'ed with the plaintext of the next block. There will be no problem of pattern in the final ciphertext. Initialization Vector is used to give input to the first block, since the first block will not have any input from the previous one. This mode is explained in Figure 3 [13].

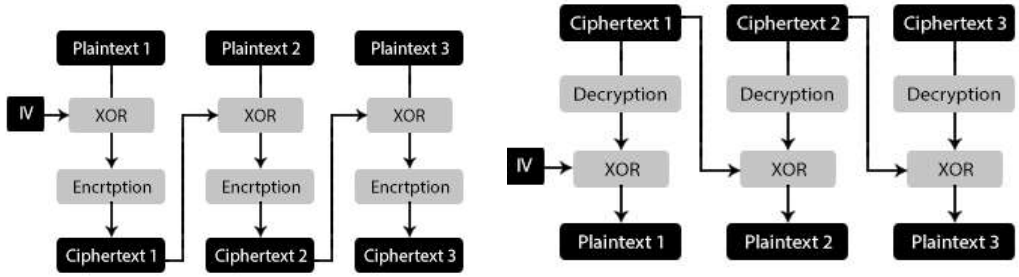

Figure 3: CBC Encryption (a) and Decryption (b)

1.5.1.3 Cipher Feedback (CFB)

In this mode, IV (initialization vector) is encrypted first, and then it is XOR'ed with plaintext to make it the ciphertext block. This ciphertext block then becomes the IV for the next block of data.

This mode exhibits the proper chaining of the plaintext and ciphertext. Noticeable feature of this mode is that if one bit of any block is damaged, the subsequent blocks will also get damaged. The operation is explained in Figure 4 [14].

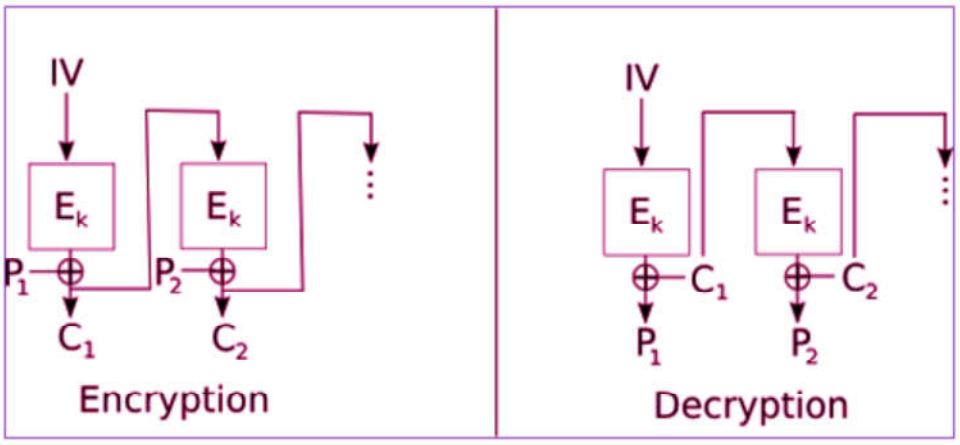

Figure 4: CFB Encryption (a) and Decryption (b)

1.5.1.4 Output Feedback (OFB)

In this mode of operation, error propagation is evaded. IV (initialization vector) is encrypted and XORed with plaintext to produce ciphertext block. But it is fed to the next block without XORing with the ciphertext. We may have to use digital signatures to overcome the problem of plaintext alteration. The operation is explained in Figure 5 [14].

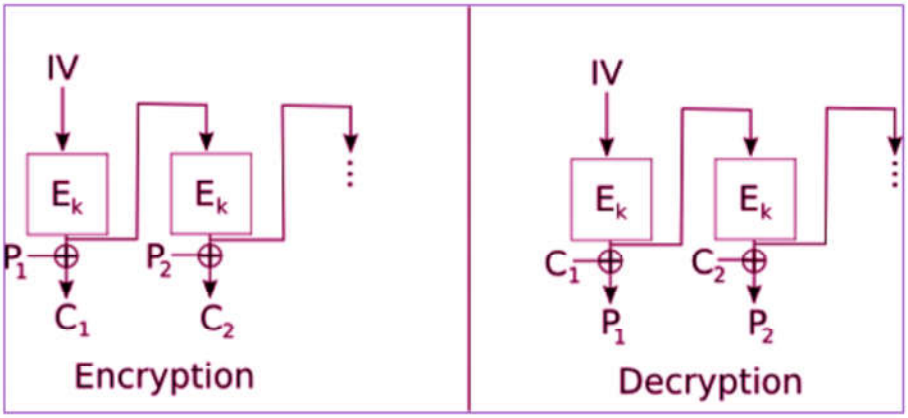

Figure 5: OFB Encryption (a) and Decryption (b)

1.5.1.5 Random Key Chaining (RKC)

Random key Chaining [15] is an authenticated encryption mode for block ciphers that achieve authentication over both plaintext and ciphertext. The unique feature of this mode is that it uses a unique key for every block of plaintext making it secure against

differential attacks. This mode facilitates parallel encryption and decryption along with the authentication which makes it smoking fast. RKC can also have pre-processed key stream for encryption and decryption. RKC can encrypt messages of 264 bits length when authentication is not required. However, when authentication is not a prerequisite, it can encrypt data of any length.

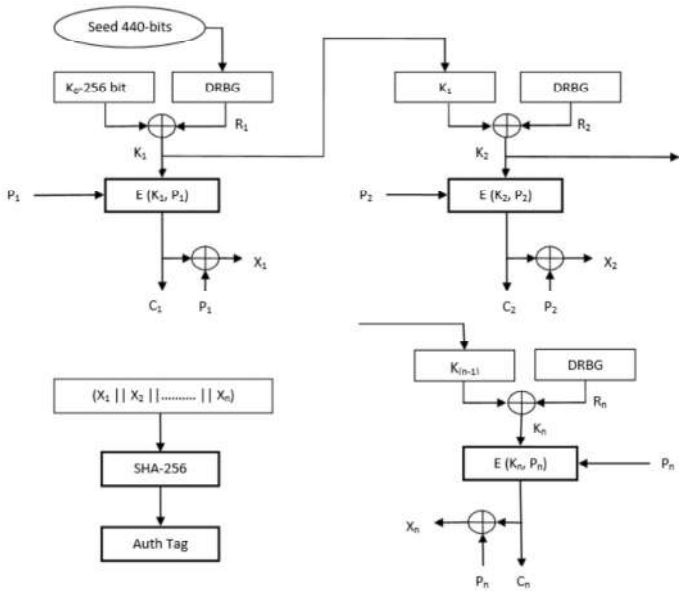

Figure 6: RKC Encryption Operation

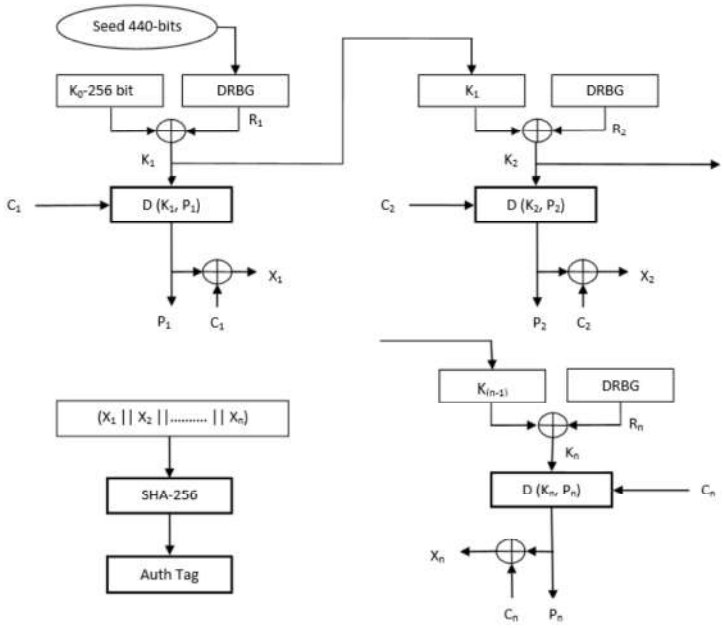

Figure 7: RKC Decryption Operation

RKC has security strengths of 2128+440, 2192+440 and 2256+440 depending upon the underlying variant of AES used in block cipher mode. Figure 6 and Figure 7 [15] summarize the encryption and decryption operation of RKC mode respectively.

1.5.2 Some Concepts Related to Block Ciphers

of the many block ciphers that have been proposed till date, many have left a strong impact in the world of information security. Some of them are still in use for maintaining security of data/information in various fields. It is worth taking note of two block ciphers, DES and AES, have ruled the security world because of their high security standards.

1.5.2.1 Shannon's Theory

Shannon that developed the theory of secrecy systems [16] [17], has given some concepts like entropy, redundancy of language, information required to break a cipher, definition of unconditional and computational secure ciphers. Key size should be decided so that cipher becomes computationally secure [18]. Brute force or exhaustive key search should not be easy.

1.5.2.2 Substitution- Permutation Networks

The theory of substitution – permutation [16], the basis of present block ciphers, substitution operation is simply the replacement of a binary word with another binary word. This substitution function makes a key for the cipher. This operation is called S-boxes. Permutation operation is the reordering of the bits of binary word. This reordering arrangement becomes the key. This operation is called P- boxes. Combination of these two operations makes an S-P network. S- Boxes provide confusion property and P- boxes provide diffusion property in the block ciphers.

1.5.3 Broad Classification of Block Ciphers

Broadly, block ciphers can be categorized into two categories.

1.5.3.1 Feistel cipher structure

Horst Feistel developed the structure for the block ciphers. This is also known as Feistel network. It takes data block and Key as input. Partitioning the data block into two halves left (L) and right(R) followed by a number of rounds. During each round L goes through an operation that depends on a round key and R while R does not change. Feistel operation is shown in Figure 8 [19].

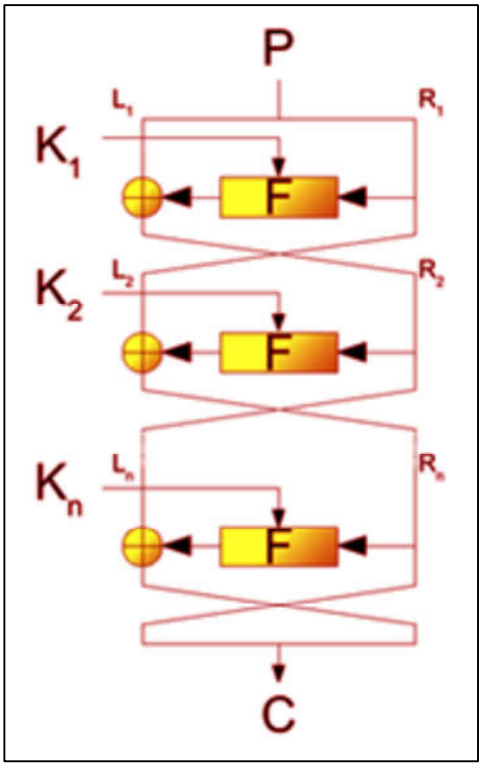

Figure 8: Feistel Function

1.5.3.2 Substitution – Permutation (SPN) ciphers

Theory of Substitution – permutation [16] given by Shannon is the basis of all the modern block ciphers such as AES, SAFER, SHARK etc.

1.5.4 Brief Summary of Block Ciphers

Below is the overview of some important block ciphers:

1.5.4.1 Lucifer

Lucifer [20] is demonstration of substitution- permutation network. Lucifer is based on Feistel structure and it uses 128 bit block as well as 128- bit key generating a 128 bit ciphertext block. It was submitted to NBS as a candidate for the standard of the commercial security. Later on it became DES after modifications.

1.5.4.2 DES

IBM and NBS made modifications in Lucifer to make it a security standard. DES [21] uses 64 bit data blocks to be encrypted with a key of 56 bits. It was the first standard algorithm for the security of online public utility services like financial transfers/transactions etc. The whole encryption process involves 16 rounds and can operate in 4 different modes. The decryption process is exactly reverses the process of encryption.

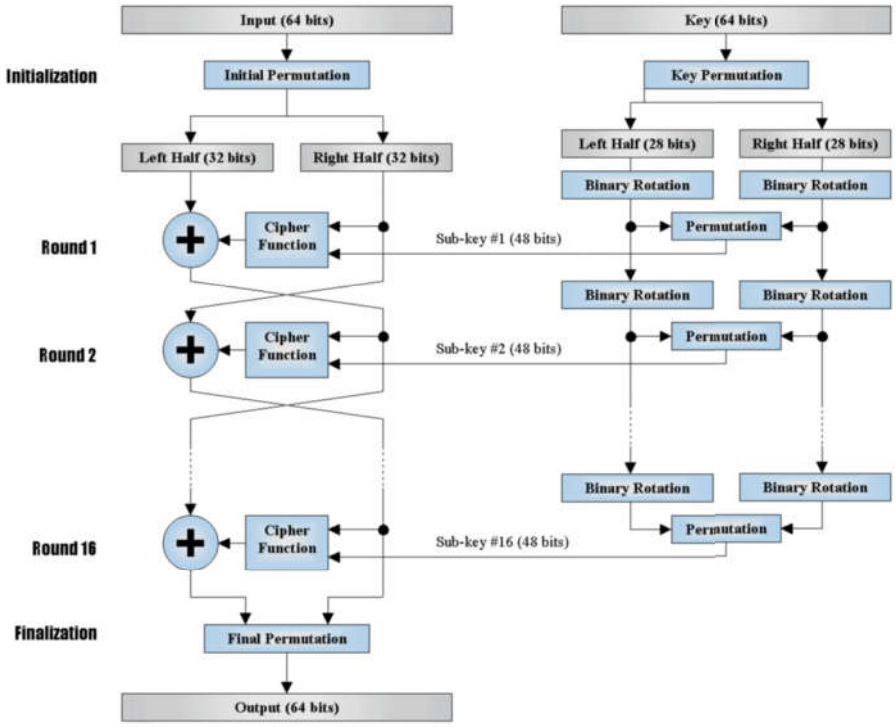

Figure 9: Block Diagram of DES [22]

1.5.4.3 Tiny Encryption Algorithm (TEA)

This is a small and fast algorithm originally designed [23] for resource constrained devices [24]. It can be efficiently used by smart cards and other embedded systems. It is designed for the easy implementation. It uses 64 bit data block and 128 bit key. It uses 32 cycles to complete the process of ciphertext generation. It is based on Feistel structure and uses bit shifting, XOR, and addition.

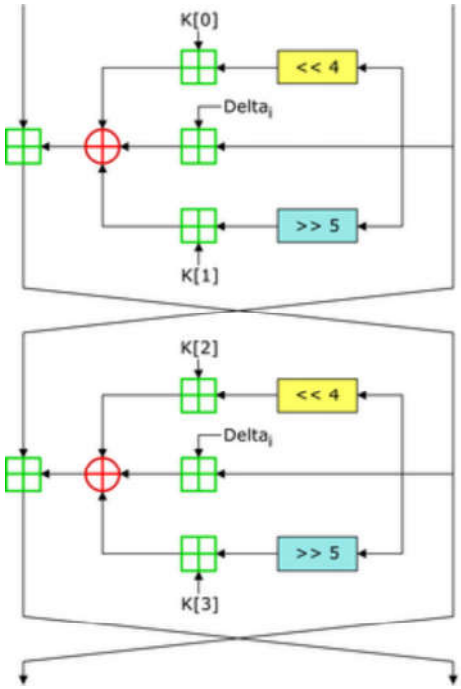

Figure 10:Two Feistel Rounds (One Cycle) of TEA [25]

1.5.4.4 TEA- Extensions (XTEA)

XTEA is the extension to Tiny Encryption Algorithm and it was developed to overcome the weaknesses of TEA. It also uses 64 bit data block and 128 bit key. It is based on Feistel structure but with variable amount of rounds. Some adjustments were made to to take care of the weaknesses of TEA. Rearrangement of the bit - shifts, XORs, additions and a more complex key schedule was introduced in this cipher. These simple modifications repaired the weaknesses in XTEA but retaining its inherent simplicity [26].

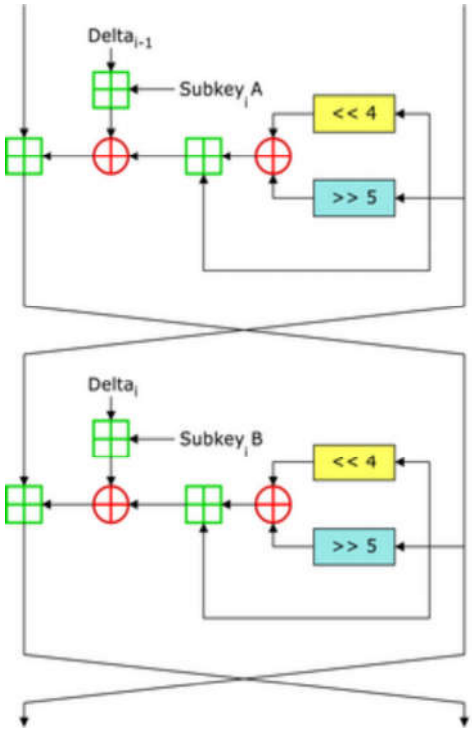

Figure 11:Two Feistel Rounds (One Cycle) of XTEA [27].

1.5.4.5 Fast Data Encipherment Algorithm - FEAL

FEAL [28] is a block cipher designed for faster processing, especially for microprocessors.

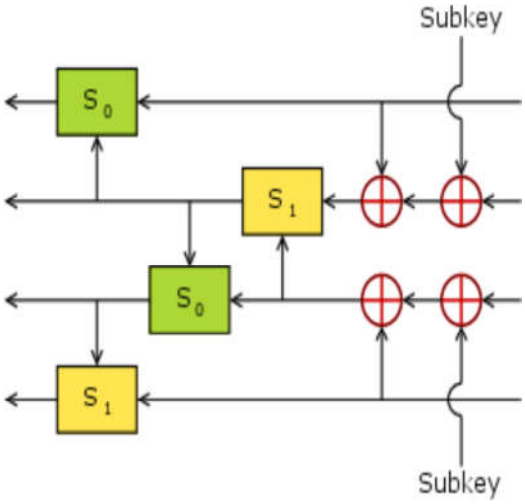

Figure 12:The FEAL Feistel Function

Figure 12 depicts the Feistel function of FEAL. It uses 64 bit block size and 64 bit key. FEAL is generally called FEAL-N where N is the number of rounds it takes for the processing e.g. FEAL-4, FEAL-8 or FEAL-32 etc. It was developed by NTT in 1987 as a replacement of DES with a faster speed. Its first version FEAL-4 was found to be quite weak. So its further improved versions were developed.

1.5.4.6 Blowfish

Blowfish is a 64-bit block cipher [29] with a variable-length key, 64-bit block cipher. It is fast, compact, simple, and secure cipher. The algorithm comprises of two parts: one part is data- encryption and the other part is key-expansion. Key expansion transforms a key of at most 448 bits into various subkey arrays totalling 4168 bytes. Its encryption is also based on 16-round Feistel network. This cipher is suitable for the systems where key does not change frequently.

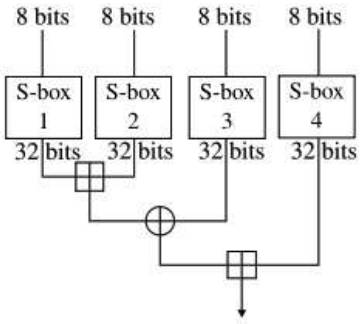

Figure 13:The Round Function (Feistel Function) of Blowfish

1.5.4.7 AES-Advanced Encryption Standard

AES is a symmetric block cipher developed to replace DES and 3DES. It was clear at that time that DES needed replacement since it was prone to so many attacks and 3DES was quite slow in processing. Block size and key size were required to be increased at that time. So, National Institute of Standards and Technology (NIST) in 2001 published Advanced Encryption Standard (AES).

Originally developed by Dr. Joan Daemen and Dr.Vincent Rijmen and named Rijndael, AES was accepted as a standard for a varied range of applications. It was named as AES in October 2000 after NIST selected and proposed it. Its block size is 128 bit and

key size is variable, it can be 128, 192, or 256 bits. It is an iterative cipher and not a Feistel cipher. Figure 14 [30] summarizes encryption and decryption operation of AES.

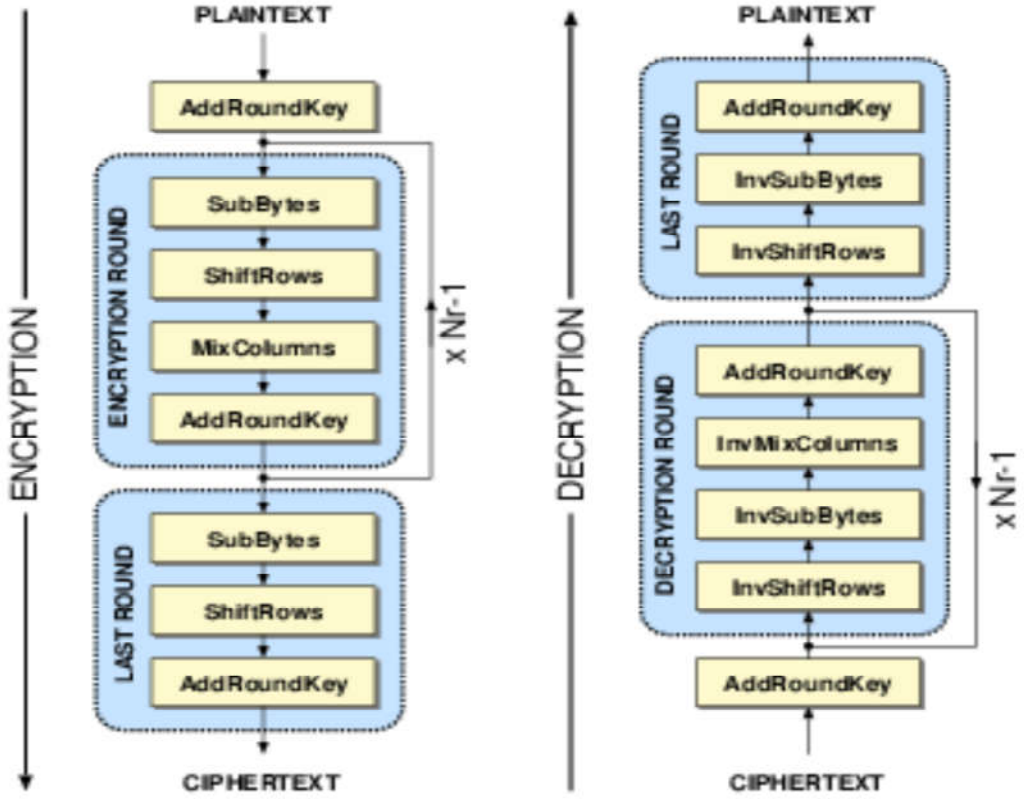

Figure 14: Basic Structure of the AES Algorithm: Encryption (left), Decryption (right).

1.6 Motivation for our Research

The rapid increase in computer networks/ networking, the risk of data security also initiated. After the birth of computer networks, the technology has changed drastically and rapidly. Starting from PSTN(Public Switched Telephone Network) to today's high speed networks, the role of computer networks is visibly massive. We generally communicate over insecure networks, but our data needs high security. In today's scenario, starting from a small e-mail communication to global financial transactions, everything is being done online. What is the security of our data and financial transactions in this online world? The answer to this very important question is the use of security ciphers. These security ciphers are designed in such a way that, breaking

them is near to impossible. The research to develop new security ciphers is an on-going process.

Cryptanalysis plays an important role in this research. Cryptanalysts are always on the look out for devising attack on the existing ciphers by finding flaws in them so that more and more secure ciphers could be developed in due course of time to maintain the level of security.

Block ciphers play a very important role in the security horizon. These are used for maintaining the confidentiality of communications. Not only this, but these ciphers are also building blocks of computer security, network security and message authentication codes. So, it is of prime importance to assess the robustness of these ciphers.

Where cryptography maintains the security of the information, there's another technique which conceals the fact that message is sent or not. This technique is known as steganography. This technique not only provides information security, but also anonymity and privacy. Apart from the benefits it possess, this technique can also be proved as a threat to the individuals and even society. The tradeoff between the threats and the benefits brings a lot of challenges. So it is also important to analyze steganography technique.

Objectives of the thesis are formulated on the basis of this motivation.

1.7 Objectives of the Thesis

1. The main objective of this thesis is to design and develop an algorithm to check the robustness of block ciphers.
2. The second objective is check the applicability of the developed algorithm on XOR, which is the building block for several ciphers.
3. To implement the developed algorithm on the secure ciphers like reduced TEA, TEA, XTEA, FEAL, BLOWFISH and AES to find the correlation and to test the following properties:
    a. Confusion Property
    b. diffusion Property
    c. Constant Factor coefficient

d. Effect of ciphertext Bitsum alone on plaintext Bitsum
e. To find some pattern of the output(Ciphertext) with respect to the input(Plaintext) and Key

4. Analysing the applicability of Bitsum algorithm on LSB steganography technique.

1.8 Contributions of this Thesis as per ACM Classification

Security and privacy is a comprehensive term and it has an important place in the tree structure of ACM computing classification systems. Cryptography, the branch of ACM computing classification of security and privacy has further two branches naming "Symmetric Cryptography and hash functions" and "cryptanalysis and other attacks". Symmetric cryptography is the old technique being used to build the cryptosystems. It is also called the secret key cryptography. On the other hand, cryptanalysis is the science of breaking the cryptosystems without knowing the key. This thesis aims to present the research work done in these two fields. Cryptanalysis of Block ciphers, which is part of the symmetric cryptography, is presented to assess their robustness. Figure 15 shows contribution of this thesis with respect to ACM classification of computing.

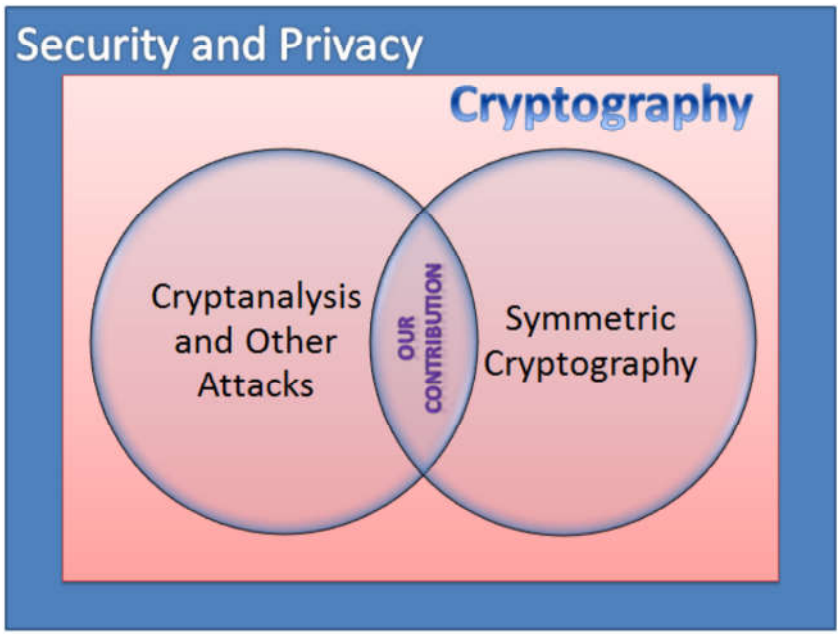

Figure 15: Contribution of Thesis as per ACM Classification

1.9 Summary

This chapter provides an overview of basic cryptology concepts. We talked about different types of cryptographic techniques and introduced cryptanalysis. Types of cryptanalytic attacks were listed which would be detailed in next chapter. The overview of block ciphers is given which further details about the mode of operation and the broad classification of block cipher. The framework of the major block ciphers i.e. TEA, XTEA, FEAL, BLOWFISH, AES is also discussed. Motivation of research is discussed before setting up the objectives of the thesis. Four major objectives of the thesis are listed in the later section of this chapter. The last section of this chapter has detailed about the contribution of this thesis as per the ACM classification of computing. Next chapter i.e. Chapter 2 is the detailed study of the review of literature related to the cryptanalysis of block ciphers.

Chapter 2: Literature Review

Good literature substitutes for an experience which we have not oursleves lived through.

Alexander Solzhenitsyn

This chapter is devoted to reviewing of the literature on cryptanalysis done on the block ciphers since 1981. The available literature discusses almost all the major cryptanalytic attacks done on the block ciphers. An overview of the various cryptanalytic methods of block ciphers is presented hereunder.

2.1 Cryptanalytic Methods

2.1.1 Some Regular Properties of DES

This theory started when a paper was presented in 1981 in CRYPTO conference explaining some unwanted properties of DES [31]. DES showed some regular features which made it vulnerable to the attacks. The paper suggested that cipher should be used carefully.

2.1.2 Linear Cryptanalysis of DES

In EUROCRYPT'93 [32], a method of linear cryptanalysis of DES was presented. It was a known plaintext attack(KPA). The basic purpose of this cryptanalysis was to build a linear approximate expression and for this, statistical linear path was constructed between input and output bits of S-boxes. Matsui succeeded in breaking 8-round, 12-round, and 16-round DES with a complexity of 2^{21}, 2^{33} and 2^{47} known-plaintexts respectively. Furthermore, Matsui also presented the experimental results of only cipher text attack. If ASCII coded plaintext representing English sentences constitutes the plaintext then 8-round DES can be broken with 2^{29} known cipher texts. And if random ASCII coded sentence represents the plaintext then 8-round DES is breakable with 2^{37} cipher texts. The only challenge the author tackled for attacking full 16-round DES was non-randomness originating in plaintext.

2.1.3 Experimental Cryptanalysis of DES

In CRYPTO'94 [33], Matsui presented the first experimental cryptanalysis of DES. This was an improvement of linear cryptanalysis and experiment of breaking full 16-round DES. The result shows that full 16-round DES is breakable with a complexity of 2^{43}. The author has obtained 26 key bits using two equations and then proceeds for rest 30-bits of the key and also suggests solving more equations to get more key bits before proceeding for the search of remaining key bits because if numbers of remaining key bits are lesser then it gives a scope for exhaustive search.

2.1.4 Differential-Linear Cryptanalysis

Hellman and Lanford acquainted the world with the concept of linear and differential cryptanalysis [34]. It is an 8 – round chosen-text attack and it performed efficiently on DES. On a specific workstation (SUN-4), this attack took less than 10 seconds.

2.1.5 Cryptanalysis using Related Key

Biham introduced new types of cryptanalytic attacks using related keys [35], and tested the influence of key scheduling algorithm on the robustness of block ciphers. This attack was applicable on Lucifer and LOKI, but not on DES since key scheduling of DES is quite complex as it involves different shift pattern in each of the round. Related key attack focuses on key scheduling algorithm and is applicable to those ciphers that have same algorithm for extracting the sub-key from the cipher key. This type of attack is totally independent of the number of rounds in the cipher and complexities of function used inside the round. If the algorithm used in each round for deriving the sub-key is common, then, by shifting the sub-key one round backward will result into a complete new set of valid key. To prevent this type of attack, the key scheduling algorithm of the cipher must be complex.

2.1.6 Cryptanalysis using Truncated and Higher Order Differentials

The ciphers which are secure against differential cryptanalysis, actually are vulnerable to the attacks using higher order or truncated differentials [36]. This kind of attack was tried on DES as well. Truncated differential is a concept involving only a part of the

cipher text-differentials, and higher order differential exploits the propagation of differentials in a larger set of plaintext. The author presented the method for testing the non-linearity of block cipher using higher order differentials and exploited small number of rounds in Feistel ciphers. The author presented truncated differential attack on 6 round DES with time complexity of 3500.

2.1.7 Cryptanalysis using Partial and Higher Order Differentials

The concept of partial and higher order differentials was presented by Knudsen [37]. DES was also examined under this attack. The ciphers which are secure against differential attack are not secure against the attack using partial and higher order differentials.

2.1.8 Non- Linear Cryptanalysis

In 1996, Knudsen and Robshaw [38] had implemented a simple attack on LOKI91 which demonstrated the effectiveness of non-linear techniques. Author has generalized Matsui's Linear Cryptanalysis method and possibly recovered additional seven bits of the key that too with 25% of the plaintext required earlier in linear cryptanalysis. Knudsen has shown that non-linear approximations can also bid some additions and should not be overlooked. This type of attack has brought flexibility in linear cryptanalysis and given a scope to modify the necessities according to the amount of processing capabilities and data available with cryptanalysis. Author also makes use of complicated multiple non-linear approximations and demonstrated that additional requirement of plaintext can be saved. It is a trade-off between amount of plaintext required and number of key-bits to be recovered which is crucial for cryptanalysis for deciding how much time the attacker wants to invest in attacking the cipher.

2.1.9 'SQUARE' – A Block Cipher

A new 128-bit block cipher 'SQUARE' which was supposed to be secure against differential cryptanalysis was presented [39]. A chosen plaintext attack was mounted against SQUARE which forced the designers to augment the number of rounds in this cipher. The Objective of designing this cipher was to prevent any possibility of

differential and linear attacks. However, the author himself has presented the attack that exploits some features of the cipher which were not secure and suggested not to use this cipher in any of the sensitive applications. It was a dedicated attack applicable to SQUARE up to 6-rounds and takes less time than exhaustive key search and the author was unable to find the 7-round square key search faster than exhaustive key search. The key schedule specifying the key derivation provides resistance against three types of attacks, which were: against related key attacks, case where cryptanalyst knows part of the cipher key, and case where cipher key entry is known.

2.1.10 Related-Key Cryptanalysis

In 1997, Wagner et.al. has presented related key attacks on various block ciphers [40]. In these attacks, it was assumed that the attacker learns the encryption with the original key as well as with the derived key (i.e. related key) also. A design principal to protect the ciphers against such attacks is also given in this paper. Using differential related key attack 3-WAY is breakable with a complexity of 2^{22} chosen plaintext and one related key query. DES-X a variant of DES proposed to withstand exhaustive key attack is not able to survive against related key differential attack. The attack uses plaintext differences as modulo 2, key difference as modulo 2^{64}, and needs only 64 chosen key relations to get the key. For defending DES-X, the author recommends deriving of DES-X keys from a single key with the use of SHA-1 to prevent against related key attack. In case of CAST, with the use of 2^{17} chosen plaintext the author was able to recover the master key with 2^{48} computations and one related-key query. The differential techniques of Biham and Shamir when combined with special related key pair will break Biham-DES with a complexity of 2^{27}. RC2 is breakable with 2^{34}chosen plaintext and one related key query using technique specified in this paper. Three different related key differential attacks have been performed on TEA and an improved attack is demonstrated that also combines the idea from Biham key rotation attack and breaks TEA with 2^{23} chosen plaintexts.

2.1.11 Cryptanalysis using Impossible Differentials

A new technique of cryptanalysis using impossible differentials was introduced in this paper [41]. This attack was implemented on 31 rounds of Skipjack. Approach used by

the author is just the opposite of the one seen in differential cryptanalysis. In this technique, differentials refer to those differences which should not occur for their occurrence helps in eliminating all those keys which lead to the cipher text pairs that will decrypt to plaintexts with zero probability. Impossible differential is the term assigned to the differentials whose probability of occurrence is zero. For finding the impossible differential an automated approach is also described. The author has suggested that full 32 round skipjack cannot be attacked using this method.

2.1.12 The Collision Attack

The collision an attack was presented on 7- round Rijndael [42] in 2000. There exists a collision between 4 rounds of Rijndael, which allowed the attacker to create the distinguisher. But the complexity of this attack is very high which makes it impractical. By using the collisions between some of the byte inclined functions made in to the cipher, an efficient distinguisher is created among block spaces and inner rounds. Unlike statistical attacks, this paper has subjugated a new kind of cryptanalytic holdup that too with a limited number of plaintexts. This paper also supports Rijndael for becoming an AES finalist because the author does not have any strong arguments that endangers full round Rijndael with this attack.

2.1.13 The Rectangle Attack

Rectangle attack was implemented on serpent [43]. Attack was presented on 7, 8, and 10-round variants of Serpent. For 10 – round 256 bit key Serpent, the attack needed $2^{207.4}$ time, $2^{126.8}$ chosen plaintexts, and $2^{131.8}$ bytes of RAM. A variant of the attack requires 2^{205} time but 2^{196} bytes of RAM.

Differential attack was also implemented on 7- round serpent, differential attack on 8-round serpent with 256 bit key and a 8- round distinguisher was created to attack 256-bit 10 round serpent. The author did a trade-off between time and space requirement to speed up the attack and was able to reduce the time to 2^{205} memory accesses by increasing the memory to 2^{196} bytes.

2.1.14 Cryptanalysis with Over Defined System of Equations

Courtois and Pieprzyk explained a property of serpent and Rijndael [44] with which these ciphers can be described as the system of over defined quadratic equations over GF (2). Such systems can be easily solved if they are over defined [45]. The author has created a hypothesis that using a defined system of algebraic equations S-box can be described. Because of the small size of S-box in Serpent and due to unexpected algebraic properties in Rijndael, this paper validates the hypothesis for both the ciphers. This paper introduces new method to use the specific structures of the equations called as XSL. This paper also shows that growing the number of rounds in Rijndael and Serpent is not necessarily increasing their security against this type of attack with a very high probability. This is because of the unexpected properties of both of these ciphers. The author also suggests prevention from this type of attack by changing some of the S-boxes of the ciphers and introducing those which are not described by over defined multivariate equations.

2.1.15 Repetition Codes Cryptanalysis of Block Ciphers

A new cipher-text only cryptanalysis of the cryptosystems with repetition codes was demonstrated in [46]. These repetition codes are error correcting codes in some of the ciphers. This type of attack may become a threat to all the block ciphers with repetition codes.

2.1.16 Commutative Diagram Cryptanalysis

This paper, presented in 2004, explains cumulative diagram cryptanalysis [47]. It is just like a framework which can explain many previous attacks on product ciphers. Bivariate interpolation and generalized truncated differential cryptanalysis are also introduced in this paper. This paper presented the view called as commutative diagram which demonstrates a framework after years of different types of cryptanalysis on block ciphers. The approach involved in this technique is to first of all find out the local properties and then combine them together to find out the whole property of the cipher. The objective of this type of analysis is to figure out a framework which can exactly guide us while we are designing a cipher against known attacks. Because with the

growing methods of cryptanalysis there are many ways from where a cipher can be broken.

2.1.17 Statistical Cryptanalysis

for the security of block ciphers, understanding of some necessary mathematical conditions is essential. This thesis well demonstrated some old statistical techniques [48]. The concepts of generalized linear distinguisher, aggregate distinguisher, and sequential distinguisher were proposed in this thesis.

2.1.18 Cryptanalysis of SHACAL-1

Handschuh and Naccache proposed two different variants of SHACAL based on compression function of standardized Hash Function [49]. SHACAL-1 is one of them and it is a block cipher based on SHA-1.

Differential and rectangle attack on reduced rounds of SHACAL-1 were presented in [50]. Some better rectangle distinguishers and differential properties were exploited in this paper. However this paper presents better differential characteristics than known previously. First of all authors have created distinguishers for round 0 to 23 with probability 2^{-50} and then with a probability of $2^{-302.3}$ constructed a 38 round distinguisher. Using this 38-round distinguisher, 51 round SHACAL was attacked. Also with a probability of 2^{-148} and 2^{-154}, 34-round and 40-round ciphers differential characteristics are utilized to attack on 49 rounds of this cipher.

2.1.19 Cryptanalysis of Lightweight Block Ciphers

Analysis of the robustness of lightweight ciphers are becoming more popular for their extensive usage in ubiquitous devices. their robustness is also required. In this paper, the author presents cryptanalysis of two lightweight block ciphers i.e. PRESENT and HIGHT [51]. This paper presents related key attack on 128 bit PRESENT with time complexity of 2^{104} and with time complexity of $2^{119.53}$ HIGHT is attacked by 31-round related key impossible differential and 26 round impossible differentials.

2.1.20 Cryptanalysis of SMS4

SMS4 is a 128 – bit block cipher that uses 128 – bit key and 32 rounds for its operation. Its is the security cipher for wireless LANs in China and is being used in WAPI. Standardization of SMS4 in China for wireless LAN authentication has attracted many researchers. This paper [52] presents differential cryptanalysis of this cipher on 23-rounds. Author gives explanation on minimum number of differential S-boxes for 6, 7 and 12 rounds.

2.1.21 Cryptanalysis of TEA, XTEA and HIGHT

TEA, XTEA and HIGHT are ciphers with block size of 64 – bit and key size of 128 – bit, specially designed ciphers for resource constrained devices. This paper [53] presents impossible differential cryptanalysis of these lightweight ciphers on their reduced rounds. The author choses ciphers with three basic operations i.e., modular addition, shifting, and XOR. Approach used in this attack is to find an input differential which will not be able to produce an output differential. In this paper 23-round XTEA and 17 round TEA is attacked using impossible differential attack by finding 14 round and 13 round impossible differential of XTEA and TEA respectively. An improved impossible differential attack is proposed on 27 round variant of HIGHT and also proved to be faster than exhaustive key search.

2.1.22 Cryptanalysis of KASUMI

KASUMI is the cipher used for the security of 3G- mobile communications. In this paper [54] 5- round impossible differentials were designed on the basis of observations related to key schedule weakness and FL, FO functions to attack on 7- round KASUMI. The author has presented approximations on first and last 7-rounds of the cipher. This paper it is indicated that $2^{114.3}$ and $2^{115.8}$ encryptions are required along with $2^{52.5}$ and 2^{62} known plaintexts for implementing the attack for the first and the last 7-rounds respectively.

2.1.23 Biclique Cryptanalysis

Biclique cryptanalysis required very less amount of data to attack block ciphers. This paper [55] presented the framework of this cryptanalysis. A new concept of cutset was also explained to clarify the ideas of this attack. This attack was applied on HIGHT, Piccolo-80 and Piccolo-128. Biclique cryptanalysis found its way because it comes out to be an accelerated key search with increasing number of rounds where other methods of cryptanalysis fail. This is only why it can break full round ciphers. Although in all the attacks computational complexity is either slightly improved or not even improved making it little better than exhaustive key search. Also in some of the cases, along with computational complexity, the data complexity is also not practical as far as this attack is concerned.

2.2 List of Block Ciphers

Table 1: List of Block Ciphers

Algorithm	Year	Block Size	Key Size	Cryptanalysis on the Cipher
Lucifer [56]	1971	48, 32 or 128 bits	48, 64 or 128 bits	Differential Cryptanalysis [57], [58]
DES [21]	1975	64 bits	56 bits (+ 8 parity bits)	Linear Cryptanalysis [32], Partial and higher order Differential Cryptanalysis [37]
DESX [59]	1984	64 bits	184 bits	Related-key cryptanalysis [60]
FEAL [28]	1987	64 bits	64 bits	Known Plaintext Attack [61], Statistical Attack [62]

RC2 [63]	1987	64 bits	8–1024 bits, in steps of 8 bits; default 64 bits	Related-key cryptanalysis [60]
Khafre [64]	1989	64 bits	512 bits	Differential Cryptanalysis [57]
Khufu [64]	1989	64 bits	512 bits	Miss in the Middle Attacks [65]
FEALNX [66]	1990	64 bits	128 bits	Statistical Attack [62], Known Plaintext Attack [67]
LOKI [68]	1990	64 bits	64 bits	Differential Cryptanalysis [57] [69]
Redoc II [70]	1990	80- bits	160 bits	Differential Cryptanalysis [57]
IDEA [71]	1991	64 bits	128 bits	Narrow-Bicliques [72]
Blowfish [73]	1993	64 bits	32-448 bits	Differential Attack , Reflection attack [74]
Safer K-64 [75]	1993	64 bits	64 bits	Key-schedule cryptanalysis [76]
VINO [77]	1993	64 bits	128 bits	No cryptanalysis identified till date
GOST [78]	1994	64 bits	256 bits	Key Recovery Attack [79]
MacGuffin [80]	1994	64 bits	128 bits	Differential Cryptanalysis [81]
RC5 [82]	1994	32, 64 or 128 bits (64	0 to 2040 bits (128 suggested)	Differential and Linear Cryptanalysis [83]

		suggested)		
TEA [23]	1994	64 bits	128 bits	Related-key cryptanalysis [60]
Misty1 [84]	1995	64 bits	128 bits	Impossible differential and collision search [85]
Akelarre [86]	1996	128 bits	128 bits	Chosen Plaintext attack, Ciphertext only attack [87]
BEAR [88]	1996	On the order of 2^{13} to 2^{23} bits or more	160 or 128 bits	Meet in the middle attack [89]
CAST128 [90]	1996	64 bits	40 to 128 bits	Differential Cryptanalysis [91], Higher Order Differential Attack [92]
LION [88]	1996	On the order of 2^{13} to 2^{23} bits or more	160 or 128 bits	Analysis of Security features [89]
Shark [93]	1996	64 bits	128 bits	No cryptanalysis identified till date
ICE [94]	1997	64 bits	64 bits	Differential cryptanalysis [95]
Square [39]	1997	128 bits	128 bits	Biclique cryptanalysis [96]
XMX [97]	1997	Variable	Variable, equal to block	Multiplicative differentials [98]

			size	
AES [99]	1998	128 bits	128, 192 or 256 bits	Biclique Attack [100], Algebraic Cryptanalysis [101]
BKSQ [73]	1998	96 bits	96, 144, 192 bits	Independent – Biclique Attack on Full BKSQ-96, Independent – Biclique Attack on Full BKSQ-144, Independent – Biclique Attack on Full BKSQ-192 [102]
CAST256 [103]	1998	128 bits	128, 160, 192, 224, 256 bits	Differential cryptanalysis [104]
CS Cipher [105]	1998	64 bits	128 bits	Analysis on the basis of differential, linear and truncated differential cryptanalysis [106]
Crypton [107]	1998	128 bits	128, 192, 256 bits	Impossible differential cryptanalysis [108]
DEAL [109]	1998	128 bits	128, 192, 256 bits	Key-Schedule Cryptanalysis

				[110]
DFC [111]	1998	128 bits	128, 192, 256 bits	Differential Cryptanalysis [112]
E2 [113]	1998	128 bits	128, 192, 256 bits	Impossible Differential Cryptanalysis [114]
Frog [115]	1998	128 bits	128, 192, 256 bits	Differential Attack [116]
Hasty Pudding [117]	1998	variable	Variable	Equivalent Keys [118]
LOKI97 [119]	1998	128 bits	128, 192, 256 bits	Linear cryptanalysis for some keys [120]
Magenta [121]	1998	128 bits	128, 192, 256 bits	Chosen Plaintext Attack [122]
Mars [123]	1998	128 bits	128, 192, 256 bits	Preliminary Cryptanalysis of Reduced-Round MARS [124]
RC6 [125]	1998	128 bits	128, 192, or 256 bits	Linear Cryptanalysis [126]
Rijndael [73]	1998	128 bits	128, 192, or 256 bits	Related-key attack [127]
Safer+ [128]	1998	128 bits	128, 192, or 256 bits	Linear Cryptanalysis [129]
Serpent [130]	1998	128 bits	128, 192, or 256 bits	The Rectangle Attack [43]
Skipjack	1998	64 bits	80 bits	Cryptanalysis

[131]				using Impossible Differentials [132]
Twofish [133]	1998	128 bits	128, 192 or 256 bits	Related-Key Attacks [134]
Triple-DES [135]	1998	64 bits	168, 112 or 56 bits	A Differential Fault Attack [136]
UES [137]	1999	128 bits	128, 192 or 256 bits	No cryptanalysis identified till date
Khazad [138]	2000	64 bits	128 bits	Extension of the Square attack [139]
Anubis [140]	2000	128 bits	128 to 320 bits in steps of 32 bits	Square Attack [141], Collision Attack [142]
Camellia [143]	2000	128 BITS	128, 192, 256 bits	Impossible differential cryptanalysis [144]
DFCv2 [145]	2000	128 bits	128, 192, 256 bits	No cryptanalysis identified till date
Grand Cru [146]	2000	128 bits	128 bits	No Cryptanalysis identified till date
Hierocrypt L1 [147]	2000	64 bits	128 bits	Differential and Impossible Differential Related-Key Attacks [148]
Hierocrypt3 [149]	2000	128 bits	128, 192, 256 bits	Impossible Differential Cryptanalysis

				[150]
Kasumi [151]	2000	64 bits	128 bits	Related-Key Rectangle Attack [152]
Nimbus [153]	2000	64 bits	128 bits	Differential cryptanalysis [154]
Noekeon [155]	2000	128 bits	128 bits	Side Channel Cube Attacks [156]
NUSH [157]	2000	64, 128, or 256 bits	128, 192, or 256 bits	Linear cryptanalysis [158]
Q [159]	2000	128 bits	128, 192 or 256 Bits	No cryptanalysis identified till date
Safer++ [160]	2000	128 bits	128 or 256 bits	Multiset and boomerang attacks [161]
SC2000 [162]	2000	128 bits	128, 192, or 256 bits	Differential cryptanalysis [163]
SHACAL [164]	2000	160 bits (SHACAL-1), 256 bits (SHACAL-2)	128 to 512 bits	Differential and Rectangle Attacks [165]
PRESENT [166]	2007	64 bits	80 or 128 bits	Related-key cryptanalysis [51]
KATAN and KTANTAN [167]	2009	32, 48, or 64-bit	80 Bits	3-subset meet-in-the-middle attack [168]

KLEIN [169]	2012	64 bits	64, 80 and 96-bits	Biclique Cryptanalysis [170]
LED [171]	2012	64 bits	64 bits, 128 bits	Differential analysis [172]
LEA [173]	2014	128-bit	128, 192, or 256-bit	Power Analysis Attacks [174]
Simon and Speck [175]	2015	32, 48, 64, 96 or 128 bits	64, 72, 96, 128, 144, 192 or 256 bits	Differential Analysis [176]

2.3 Summary

This chapter gives an overview of the attacks implemented on block ciphers. Starting from linear cryptanalysis to differential cryptanalysis, this chapter aims to explain attacks such as rectangle attack, chosen plaintext attack, adaptive attacks, side channel attacks and biclique attack is also explained in this chapter. Table 1 summarized the block ciphers cryptanalysis.

We have listed 69 block ciphers out of which 55 are using XOR as a part their basic operation, which becomes near about 90% of the listed algorithms.

This becomes the basis of our next chapter which details the design and development of the algorithm.

Chapter 3: Design and Development of Bitsum Algorithm

Perhaps the most important principle for the good algorithm designer is to refuse to be content.

— *Alfred V. Aho*

This chapter is dedicated to the design and development of the algorithm. First section of this chapter details about the motivation and the design philosophy of the algorithm. Next section discusses the developed algorithm and complexity of the algorithm. Last section of this chapter discusses the applicability of this algorithm on XOR.

3.1 Motivation and Design Philosophy of the Algorithm

The motivation for the design of this algorithm is basically driven from the concept of hamming weights. Hamming weight of a strings is actually the count of symbols different from the zero symbols present in the string. If we take the hamming distance of the binary string, it is the number of "1" bits present in that string.

For example, the binary number 110011 is having the hamming weight 4. If this string is 0000, then its hamming weight would be 0.

Hamming weight is used to attack cryptographic hardware. This attack was presented on addition operation [177]. This concluded that the if the addition operation is not used carefully, it may lean the information about certain keys.

The correlation between the hamming weights and the power consumption to do a side channel attack is discussed in the thesis of William Hnath [178].

The use of number of "1" bits present in the binary strings can be used to find the information about the key or to derive a certain correlation also. With this motivation, we used the concept of number of "1" bits to develop the algorithm and named it as " Bitsum Algorithm".

3.2 Bitsum Algorithm

Correlation that plays a very important role in cryptanalysis, is the degree to which the two variables are consistently related to each other. This theory of correlation is observable in our daily lives as well. Deriving such correlations is consistent effort in the Crypto world.

Cryptanalysts also try to find some correlation between Plaintext-Key, Key-Ciphertext, and Plaintext-Ciphertext with an intention to extract the 'KEY' consistently. There exists a number of attacks that have been developed with such an intention.

A chosen plaintext attack designed with such an intention was "Bitsum Algorithm" which was shaped to exploit the fundamental property of the ciphers for checking whether there exists any correlation between the key and the cipher text. The other intention of designing this algorithm was to investigate the strength of the symmetric ciphers. The main feature of this algorithm is simplicity of it. Following is the algorithm:

The Algorithm

A. Choose the cipher to investigate

B.

Loop

- For the cipher under investigation, encipher a fixed message M with N different keys.
- Calculate the correlation between the Bitsum of the produced ciphertexts and the Bitsum of their corresponding keys.

End Loop

C. Track will be kept of the messages that yield the best correlation between Bitsum of the ciphertext and the key.

D. Conclusion would be drawn on the basis of this record.

Explanation of the Algorithm

It was considered to have a cipher (cryptographic algorithm) for which the key consists of zero's and ones. Without loss of generality, suppose the key is of length 64 bits, a brute force attack would test 2^{64} keys in total in a worst case scenario. Now, if somehow it was possible to determine the Bitsum of the key (that is, the numbers of one's present), the keyspace would be greatly reduced. The worst case scenario here would be the 32 ones, for which there would be ${}^{64}C_{32}$ (64 choose 32) possible keys.

Investigating a chosen plaintext attack was proposed with the hope was that there may be a specific message for which the Bitsum of the ciphertext would correlate with the Bitsum of the key. The correlation would likely not be perfect, so a suggested range of values for the Bitsum of the key may be produced. Still, this would be a great savings over brute force. This has been the basis of the algorithm that was used to investigate the strength of Symmetric ciphers.

3.3 Complexity of Bitsum Algorithm

For the analysis of an algorithm, it is important to study its complexity. This would give an opportunity to compare its performance with those of other algorithms. For estimating the complexity of an algorithm, the facts that must be considered are the least time it takes, the storage requirements, and the data it needs to be executed.

3.3.1 The Big O Notation

For defining the complexity of an algorithm, an understanding of the 'O' notation is a must. The O is the Order on which the rate of growth of a function is dependent. This concept can be better explained through the example of an equation, T(n)

$$T(n) = 6n^2 + 5n - 3$$

$T(n) = O(n^2)$, which means that T(n) grows at the rate of n^2.

Formal Definition: f(n) = O(g(n)) means that there are positive constants c and k, such that $0 \leq f(n) \leq cg(n)$ for all $n \geq k$. The values of c and k must be fixed for the function f and must not depend on n [179].

The complexity of the algorithm is represented through some common notations. Some of which are given below:

Table 2: Common Notations used for Representing the Complexity of the Algorithms

Notation	Name
O(1)	constant
O(log(n))	logarithmic
O(n)	linear
$O(n^2)$	quadratic
$O(n^c)$	polynomial
$O(c^n)$	exponential

3.3.2 The Time and Space Complexity of Bitsum Algorithm

The time and space complexity of Bitsum algorithm is in the order of O(n), which states that the order of growth of Bitsum algorithm is linear.

3.4 Illustration and Application of Bitsum Algorithm

The cipher considered for illustration is XOR. It is presumed that if the key is random and is equal to size of the message, the XOR cipher is more secure. Stream Cipher works on the key stream generated by a pseudo random number generator. If the key is truly random, the obtained one time pad is unbreakable.

Case 1:

If the bits of the chosen plaintext are all “Zeros”, the sum of the bits of the key is equal to the sum of the bits of the ciphertext.

Case 2:

If the bits of the chosen plaintext are all “Ones”, the sum of the bits of the key is equal to the difference between total number of bits of the chosen plaintext and the sum of the bits of the ciphertext.

A strong correlation between the ciphertext and the key has been found.

Further, for finding a key, first, we have been calculating the sum of the bits of the ciphertext. If the sum exceeded half the total number of bits of the key, the chosen plaintext with all "ones" is used; else the chosen plaintext with all "Zeros" was used. Thus the search space is considerably reduced.

Application of Bitsum on XOR

We have analyzed in Chapter 2 that almost 90% of block ciphers are using XOR as a part of their basic operation. So it becomes really important to check the applicability of Bitsum Algorithm on XOR.

XOR is easily reversible. Hence, it is generally used in Steganography. Bit sum Algorithm poses a threat to all steganography techniques which uses XOR operation.

First of all, it was decided to conduct an experiment by implementing the Bitsum Algorithm on XOR.

3.5 Applicability of Bitsum Algorithm on XOR

XOR (Exclusive OR) operation is most common binary operation being used in security ciphers because it produces an output based on both the inputs. When the message bits are exclusive- ORed with the key bits, the message bits are flipped if the key bits are 1. But when the key bits are 0, the message bit doesn't change [180]. Following example will clarify the concept:

Message: 1 0 1 0 1 0 0

Key: 0 1 1 0 0 1 1

Result: 1 1 0 0 1 1 1

As in cryptanalysis, the cryptanalyst always tries to find a correlation between the plaintext, key and the ciphertext, Bitsum Algorithm also tries to find the relationship between the ciphertext and the key under a chosen plaintext. This study shown that

Bitsum Algorithm poses a threat on XOR cipher. The subsequent section will demonstrate the practical application of Bitsum algorithm on XOR operation.

3.5.1 Importance of XOR in Security

XOR operation is the basic operation in most of the security ciphers. Even the most significant ciphers like DES [21], TEA [23], FEAL [28], AES [99], etc. use XOR as the elementary building block for encryption and decryption. Even almost 90% of the block ciphers use XOR as a part their basic operation. Following are some major factors which show how important is XOR to the security:

1. The output of XOR always depends on both the operands. Whereas in other binary operations like AND and OR, the output is dependent on one operand only. In AND operation, if one of the bit is '0', the output will always be '0'. In OR operation, if one of the input bit is '1', the output will be '1'. But in case of XOR, the result depends on both the bits, which makes the work of the cryptanalyst becomes very difficult. Now the cryptanalyst will not be able to take a decision on the basis of only one operand.
2. XOR is a reversible operation. This feature of XOR will be better understood by the following example:
   Message (Plaintext) is '100001' and the key is '111000'. When these are XORed, the result (Ciphertext) obtained will be '011001'. This is called encryption with a key. The decryption can also be done with the same key i.e. if the result(ciphertext) is XORed with the same key, t the original message(plaintext) '100001' will be obtained again.

Encryption		Decryption	
Message(Plaintext):	1 0 0 0 0 1	Result(Ciphertext):	0 1 1 0 0 1
Key:	1 1 1 0 0 0	Key:	1 1 1 0 0 0
---		---	
Result(Ciphertext):	0 1 1 0 0 1	Message(Plaintext):	1 0 0 0 0 1

So, this behaviour of XOR operation renders it suitable for use in both encryption and decryption algorithm. Addroundkey function in AES is an example of this operation.

3. Hardware implementation of XOR is easier since it can be realized using universal gates e.g. NAND with less number of transistors. Whereas other operations like ADD or MULTIPLY requires more complicated hardware.

These properties of XOR make it the favorite among security ciphers.

3.5.2 Generation of Pseudo Random Binary Sequence using Linear Feedback Shift register (LFSR) with a (2^N-1) period

Linear Feedback Shift Register generator produce Linear Recursive Sequence. Two factors affect the length of the sequence before repetition occurs,

1. The feedback taps and
2. The initial state.

If proper feedback taps are taken, an LFSR of N registers would produce (2^N-1) shifts (m- sequence).

This study used Galois Field to generate m - sequence feedback taps. LFSR is represented by a polynomial in X. Below is the generalized polynomial:

$G= g_mX^m + g_{m-1}X^{m-1} + g_{m-2}X^{m-2}+\ldots\ldots g_2X^2+ g_1X^1+ g_0$

For N=8

Berlekamp algorithm [181] using MAPLE software is used for determination of feedback taps.

The feedback taps for Galois form are:

[8, 7, 6, 1]

[8, 7, 5, 3]

[8, 7, 3, 2]

[8, 6, 5, 4]

[8, 6, 5, 3]

[8, 6, 5, 2]

[8, 7, 6, 5, 4, 2]

[8, 7, 6, 5, 2, 1]

The feedback taps for Fibonacci form are:

[8, 7, 2, 1]

[8, 5, 3, 1]

[8, 6, 5, 1]

[8, 4, 3, 2]

[8, 5, 3, 2]

[8, 6, 3, 2]

[8, 6, 4, 3, 2, 1]

[8, 7, 6, 3, 2, 1]

VB.Net was used to generate the PRBS using both Galois form and Fibonacci form implementation of LFSR for m- sequences. The next section shows screenshots of the output.

3.5.3 Implementation Results - Screenshots

A Linear Feedback Shift Register (LFSR) is a simple way to produce a long sequence of random numbers, given a non-zero seed. It can be implemented in two ways, Fibonacci

implementation and Galois Implementation. Both the implementations are explained below:

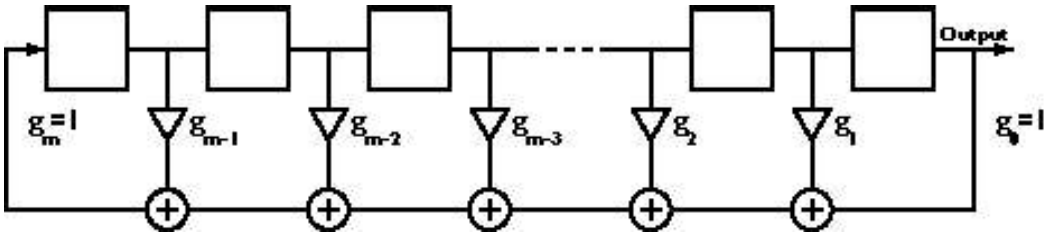

Figure 16:Fibonacci implementation of LFSR [182].

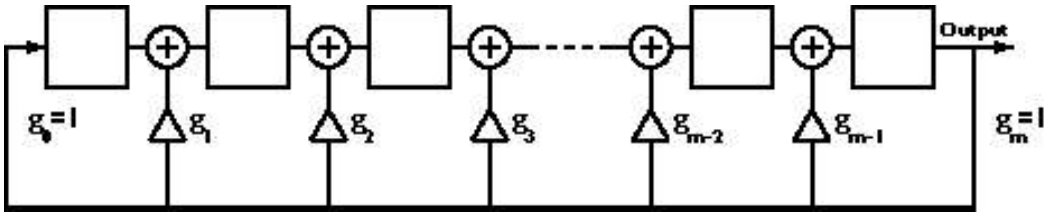

Figure 17: Galois implementation of LFSR [182].

The screenshots of implementation of LFSR for generating m- sequences are presented in this section. The number of register stages i.e. N is taken as 8.

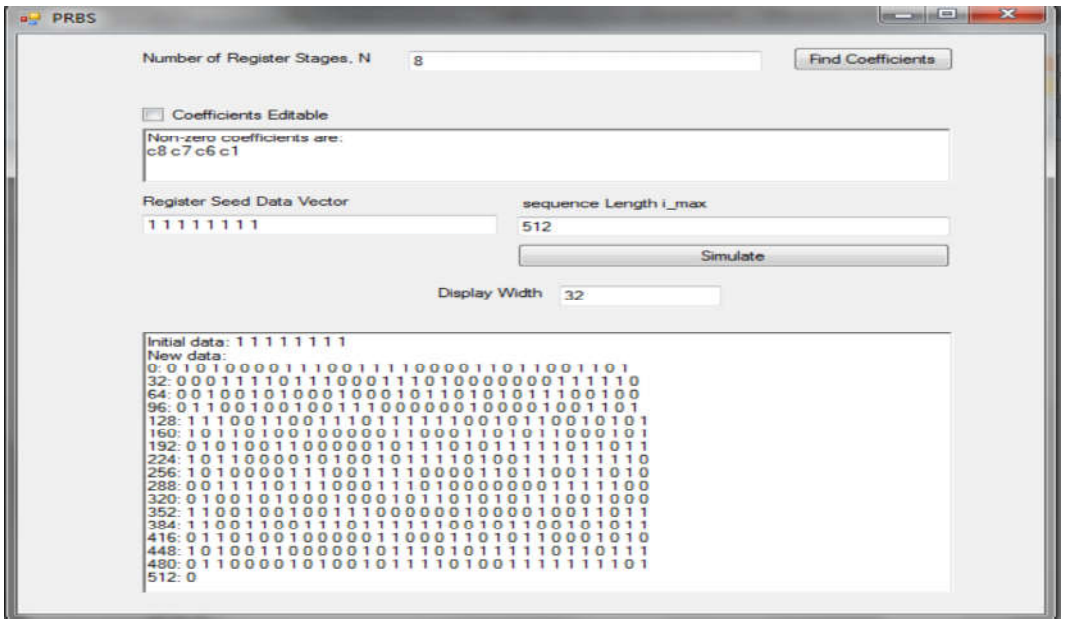

Figure 18 : Pseudo Random Binary Sequence using Galois Form Implementation of LFSR for m-sequence

For Galois Field the coefficients are 8 7 6 1 and the register seed data vector is 11111111. The sequence length is 512. Figure 18 shows the results of this implementation.

A case is similar with Fibonacci form implementation. The coefficients for Fibonacci are 8 7 2 1. The data for analysis is generated using these coefficients. **Figure 19** shows the results of this implementation.

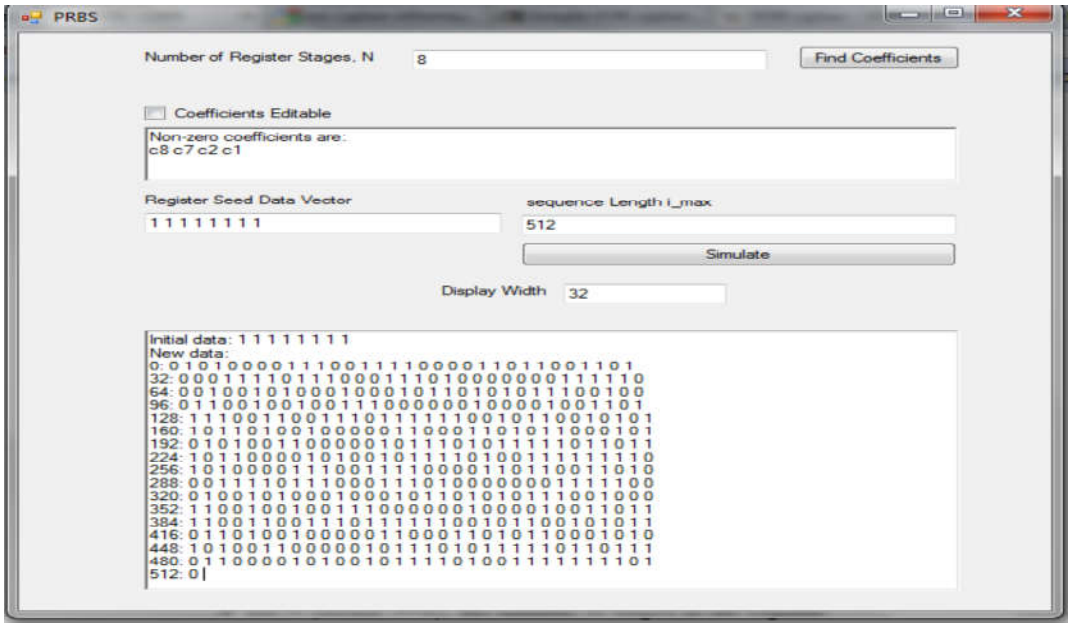

Figure 19: Pseudo Random Binary Sequence using Fibonacci Form Implementation of LFSR for m-sequence

The analysis of Bitsum Algorithm is shown in the following section.

3.5.4 Analysis of Bitsum Algorithm

For a chosen plaintext, the prime task is finding the relationship between two variables the Bitsum of the ciphertext and the Bitsum of the key. Relationships can be either linear or non-linear. Correlation is the measure of linear relationship between the variables. Linear relationships can be either positive (direct) or negative (inverse). The positive or direct relationship means that the values of both the variables will increase together or decrease together, i.e. if one increases the other will also increase; but if one decreases the other will also decrease. In a negative or inverse relationship, the changes

in the values of the variables would be in opposite directions. The correlation also describes the strength of the relationship. The relationship would be strong when the correlation is -1.0 to -0.5 or 1.0 to 0.5. It will be moderate when the correlation is -0.5 to -0.3 or 0.3 to 0.5.

The experiment is done with 8 bit pattern i.e. all 8- bit chosen plaintext would range from 00000000 to 11111111. All 256 combinations of key are used on every chosen plaintext and the correlation coefficient of Bitsum of key and Bitsum of ciphertext was computed. The formulated results are shown in Table 3.

Table 3: Deriving Relationship between Bitsum of Key and Bitsum of Ciphertext for a Chosen Plaintext

Chosen Plain Text	Correlation between Bitsum of key and Bitsum of ciphertext "R"	Relationship
00000000	1	Strong
00000010	0.75	Strong
00000011	0.5	Strong
00001011	0.25	Weak
00001111	0	NIL
00011111	-0.25	Weak
00111111	-0.5	Strong
01111111	-0.75	Strong
11111111	-1	Strong

Then we considered the output of PRBS (pseudo random binary sequence) for 512 bits in the above case, the output revealed unbiased results i.e. we got exactly 256 zeros and 256 ones. But when splitting them into 8 bit patterns was considered, for example the first 8 bits – 01010000, the sum of the key in this case is 2, which reveals that a strong

correlation exists between the Bitsum of the key and the Bitsum of the ciphertext. For the rest of the 504 bits, the results are tabulated in Table **4**.

Table 4: Deriving Relationship using Sample Data

Strong relationship	30%
Weak relationship	42%
No relationship	18%

3.5.5 Conclusion of this Experiment

The conclusion that this experiment brought forth was the fact that for a chosen plaintext, there exist a correlation between Bitsum of key and Bitsum of ciphertext. Since most of the cryptographic ciphers use XOR, the testing of Bitsum Algorithm on major cryptographic algorithms to determine their strength became essential.

3.6 Results and Discussions

This chapter has discussed the motivation and design philosophy of the algorithm to be developed. Algorithm was introduced in this chapter along with its complexity. The importance of XOR in the security algorithms were discussed, which further became a base to test the applicability of Bitsum algorithm on XOR. Implementation of Bitsum algorithm on XOR was done and analyzed. This experiment showed that in 72% of the cases correlation exists. In the later sections of this chapter, it was concluded that Bitsum Attack poses a threat on XOR. SO it becomes important to implement this attack on some of the ciphers which are using XOR as their basic operation.

Chapter 4 is based on the implications of Bitsum Algorithm on reduced TEA, TEA and XTEA.

Chapter 4: Implications of Bitsum Algorithm on Reduced Key Tiny Encryption Algorithm, TEA and XTEA

The achievement of one goal should be the starting point of another.

-- Alexander Graham Bell

This chapter is dedicated to the experimentation done after getting the positive results from previous chapter. The first section details the experiment done on reduced key Tiny Encryption Algorithm. When it was found that TEA with reduced key size is not secure under Bitsum attack, then TEA and XTEA were implemented and analyzed. Key pattern theorems were developed on the basis of analysis of data. These theorems are explained in the last sections of this chapter.

4.1 On the Security of reduced Key Tiny Encryption Algorithm

The motivation of this experiment was making of DES (Data Encryption Standard) [21]. NIST reduced the key size of Lucifer from 128-bit to 56- bit when it was made Data Encryption Standard. The same experiment [183] has been performed on Tiny Encryption Algorithm [23]. The key size of TEA was reduced from 128 – bit to 64 - bit to perform this experiment. TEA is a small algorithm which is being used in smart cards and embedded systems. If used and implemented carefully, it is a secure cipher.

The following sections show that the reduced version of TEA could not withstand Bitsum Algorithm.

4.4.1 Need of Information Security in Resource Constrained Devices

In this fast growing online culture, smart handheld devices are the new choice for doing any type of transaction. These small devices are resource constrained devices. The ciphers which are using 128-bit or more key size needs more space and computing power. If we use internet on these devices, we need more security for such devices. It

is not always true that, larger the key, more the security. The requirement of the security is different for different systems. Following situations will elucidate [18] the scenarios:

1. Security is required for brief period for online funds transfer.
2. When Strategic planning is done by the companies, these plans needs security for some years.
3. The formulas and designs of proprietary products have to be secure for their lifetime.
4. Any information related to the finances of the employees or the companies may need security for their lifetime.

In these types of situations, quick security solutions are imperative. The importance of security is discussed in [184]. Protecting the information may sometimes be quite expensive, but there may be some economical solutions also for that [185] .

4.4.2 Experimentation

Under this study implementation of reduced key TEA was undertaken using Bitsum Algorithm. For this, 1500 plaintexts were chosen and each message was enciphered using random 500 keys. All the results have been stored along with Bitsum of Plaintext, Key, and Ciphertext. The intention behind this was to find out the correlation between the Bitsum of plaintext, key, and ciphertext.

When 'r' is correlation coefficient and it gives us an idea of how strong the relation of two variables is.

'r' is calculated as

$$r= [N\Sigma XY - (\Sigma X)(\Sigma Y) / Sqrt([N\Sigma X^2 - (\Sigma X)^2][N\Sigma Y^2 - (\Sigma Y)^2])]$$

For each message, calculation was done to find/determine the correlation coefficient for Bitsum of Ciphertext produced with the Bitsum of the corresponding keys. The process was closely observed to find the best correlation coefficient. Some of the plaintexts formed very good correlation coefficient. The Bitsum of the plaintext were also

observed. Combinations for the plaintexts with that particular Bitsum were tried. Nearly all the combinations of that particular plaintext (with that Bitsum) generated good results.

It was observed that the Bitsum of the key could also find out some keys which could generate good results. A set of keys was found during this, which yielded the value of correlation coefficient almost equal to 1(0.99).

The cases which gave favourable results are listed hereunder.

Case 1: Bitsum of the Key Chosen at Random

Figure 20 shows the relation of Bitsum of the Key and Bitsum of the cipher text when the key was chosen at random.

From analysis of the data in the scatter chart it can be inferred that the Bitsum of the ciphertext remained constant for Bitsum of key ranging from 0 to 13 and 51 to 64.

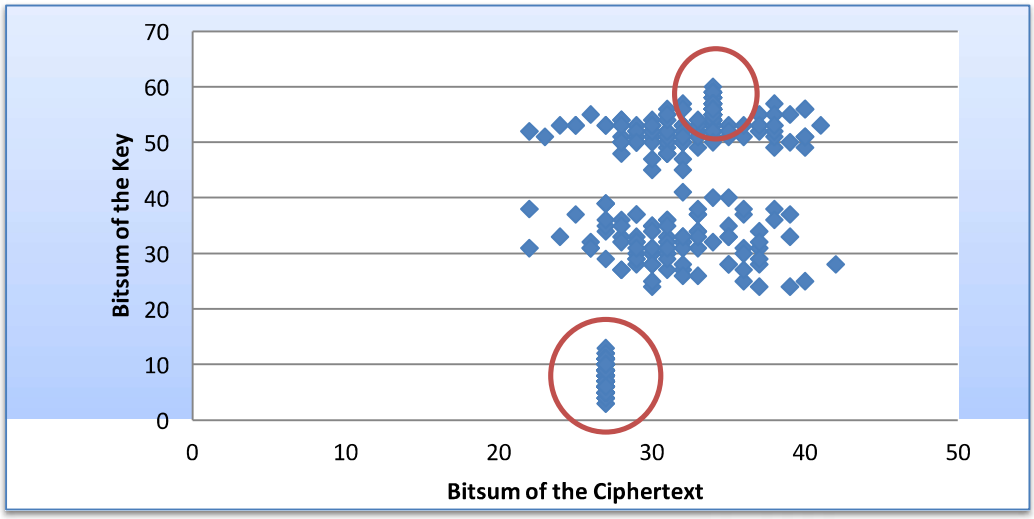

Figure 20:Relation between Bitsum of the Ciphertext with Bitsum of the Key Chosen at Random(Reduced Key TEA).

After analysing this graph, the constant values were further analyzed. Case 2 and Case 3 are discussed below.:

Case 2: Bitsum of the Key Less than 14

Figure 21 shows the relation of Bitsum of the Key and Bit sum of the cipher text when Bitsum of key is < 14.

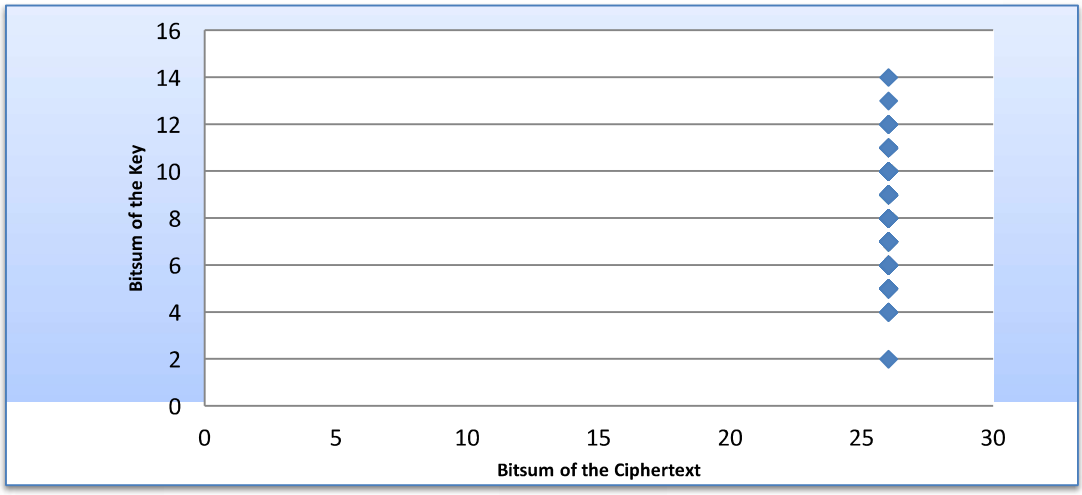

Figure 21: Relation between Bitsum of the Ciphertext with Bitsum of the Key<14(Reduced Key TEA).

It is visible from the scatter chart that whenever Bitsum of the key is less than 14, the Bitsum of the ciphertext remains constant.

Case 3: Bitsum of the Key Greater than 50

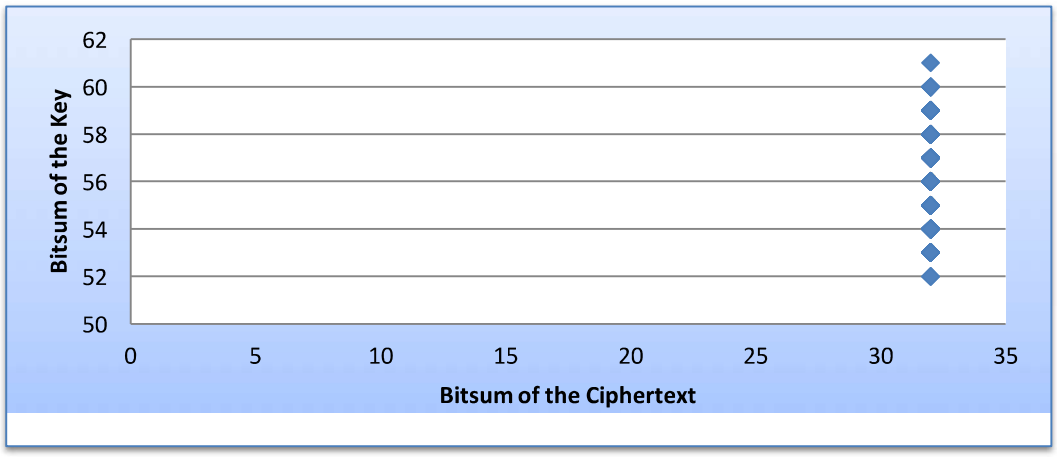

Figure 22: Relation between Bitsum of the Ciphertext with Bitsum of the Key >50 (Reduced Key TEA).

Figure 22 shows the relation of Bitsum of the key and Bitsum of the cipher text when Bitsum of key is > 50. Again it is visible form this scatter chart that if the chosen Bitsum of key is greater than 50, then the Bitsum of the ciphertext remains constant.

After finding these results, value of correlation coefficient was calculated. All these values are tabulated in Table 5. Observations showed that for some Bitsum of plaintext values, value of 'r' (correlation coefficient) is fairly high.

Table 5: Values of 'r'

Bitsum of Plaintext	Value of 'r'
1	0.690308985
2	-0.347327613
3	0.032906406
4	-0.547805863
5	0.54028605
6	0.137078721
7	0.129794285
8	0.209500805
9	-0.533315846
10	-0.528870777
11	-0.613067379
12	-0.695289267
13	0.650219494
14	-0.504409181
15	0.068242464
16	-0.563355275
17	0.652494041
18	0.022508143
19	0.191843891
20	0.251984003
21	-0.555625983
22	0.776741678
23	-0.131765042
24	0.764611686
25	-0.518213409
26	-0.109198641
27	-0.3383772
28	-0.80275502
29	0.570108404
30	-0.536051182
31	0.028415708
32	-0.538999656

33	-0.374552622
34	-0.478483858
35	-0.215823533
36	0.136820101
37	-0.2890425
38	-0.438684965
39	-0.455603721
40	0.775945384
41	-0.153504098
42	-0.797246664
43	-0.36052192
44	-0.152639835
45	0.160806228
46	0.385847639
47	-0.284381432
48	-0.531938352
49	-0.016846536
50	0.507177517
51	0.558967767
52	0.69059608
53	0.565680904
54	-0.357953916
55	0.302725029
56	-0.52556234
57	0.04049301
58	-0.257464525
59	0.380934763
60	-0.675337308
61	0.623425045
62	-0.457068802
63	-0.594698149
64	-0.726506258

The scatter chart in Figure 23 summarizes the results of this experimentation. This scatter chart shows the values of Bitsum of plaintext where value of ‘r’ is quite high. X-axis represents the value of Bitsum of the plaintext and y-axis represents the value of ‘r’ (correlation coefficient).

It is visible from the chart that some of the values of ‘r’ are approaching 1, signifying a strong correlation. The value of ‘r’ always lies between +1 and -1 [186].

When the value of ‘r’ approaches 0, which means there is poor correlation between the variables. But if value of ‘r’ approaches either +1 or -1, then it means that a strong correlation exists between the variables.

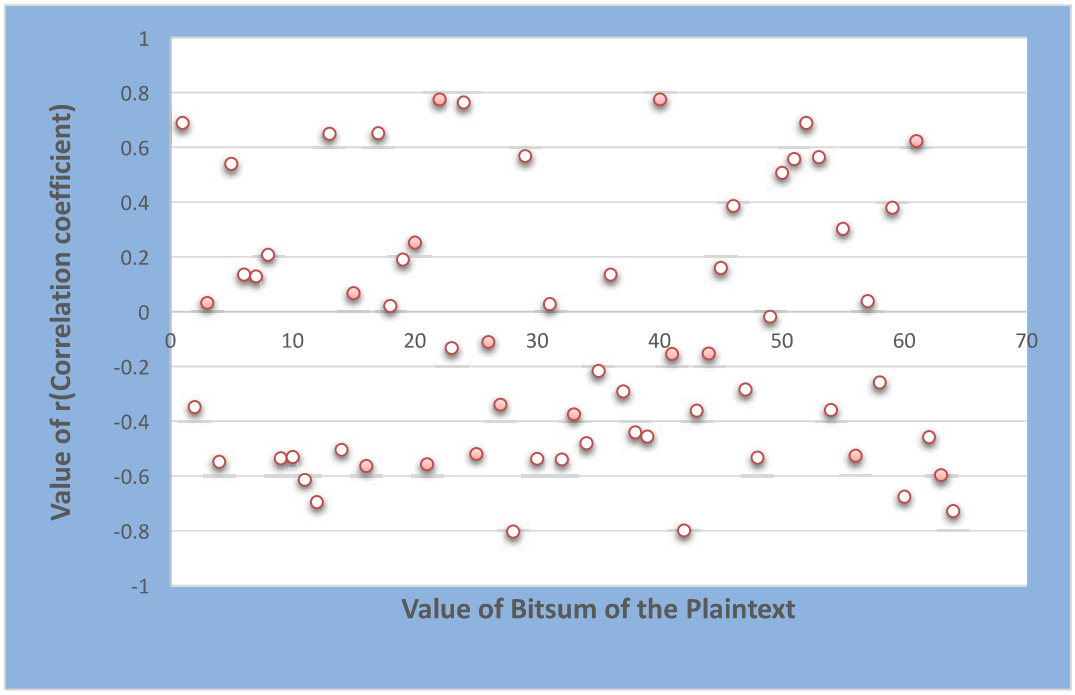

Figure 23: Relation between r (Correlation co-efficient) and Bitsum of the Plaintext (Reduced Key TEA).

The values in Table 6 summarize the strength of correlation between the two variables. These values are used to interpret the available data. On the basis of the analysis, two relationship theorems are developed which are defined in next section.

Table 6: Strength of Linear Relationship

Value of correlation coefficient	Strength of linear relationship
At least 0.8	Very Strong
0.6 to 0.8	Moderately strong
0.3 to 0.5	Fair
Less than 0.3	Poor

4.4.3 Deriving Relationships between Bitsum of Ciphertext, Bitsum of Key and Bitsum of Plaintext

Bitsum Theorem on Reduced TEA – I

If $K = \{k_1, k_2, k_3 \ldots\ldots k_n\}$, $k_i \in K$, $M = \{m_1, m_2, m_3 \ldots\ldots m_n\}$, $m_i \in M$ and

If (X<14 Or X>51),

On computing

$$C = TEA_e(m_i, k),$$

Y is constant.

Where K is the set of keys, $n= 2^{64}$, M is the set of plaintexts to be encrypted, X is he Bitsum of the key, Y is the Bitsum of the ciphertext, TEA_e is the encryption routine of Tiny encryption algorithm.

Bitsum Theorem on Reduced TEA – II

If $K = \{k_1, k_2, k_3 \ldots\ldots k_n\}$, $k_i \in K$, $M = \{m_1, m_2, m_3 \ldots\ldots m_n\}$, $m_i \in M$ and

If (X>=14 Or X<=51),

On computing

$$C = TEA_e(m_i, k),$$

Y lies between 20 and 45.

Where K is the set of keys, $n= 2^{64}$, M is the set of plaintexts to be encrypted, X is he Bitsum of the key, Y is the Bitsum of the ciphertext, TEA_e is the encryption routine of Tiny encryption algorithm.

Corollary: Reduced Key Tiny Encryption algorithm is secure if the Bitsum of the key lies between 14 and 51.

This attempt to check the security of TEA with a reduced key size was to test whether TEA is secure against Bitsum Algorithm or not. But it was found that TEA is not secure under these circumstances.

4.5 Implications of Bitsum Algorithm on TEA using SPSS

Objective: To examine the relation between the Bitsums of key, plain texts and cipher texts. Hypothesis had been created to conduct the tests.

H_0= Bitsums of cipher text and key are not correlated.

H_1= Bitsums of cipher text and key are correlated.

The correlation results shown below in Table 7 also suggest that there is a correlation between the Bitsum of cipher texts and the Bitsums of the keys in TEA algorithm as the significance value is .000 which is <.01 and we have rejected the null hypothesis.

Table 7: Correlation statistic between Bitsums of ciphertext and key For TEA

		C_Bitsum	K_Bitsum
C_Bitsum	Pearson Correlation	1	.143**
	Sig. (2-tailed)		.000
	N	1158	1158
K_Bitsum	Pearson Correlation	.143**	1
	Sig. (2-tailed)	.000	
	N	1158	1158
**. Correlation is significant at the 0.01 level (2-tailed)			

H_0= Bitsums of cipher text and plaintext are not correlated.

H_1= Bitsums of cipher text and plaintext are correlated.

Similar correlation statistics, as shown in Table 8, have been observed between the Bitsums of plaintexts and Bitsums of ciphertexts. We reject null hypothesis as significance value is 0 which is <.01. This suggests that there is no confusion and

diffusion property in TEA algorithm and as a result cryptanalysis is easier in this TEA algorithm.

Table 8: Correlation statistic between Bitsums of plaintext and ciphertext

		P_Bitsum	C_Bitsum
P_Bitsum	Pearson Correlation	1	.125**
	Sig. (2-tailed)		.000
	N	1158	1158
C_Bitsum	Pearson Correlation	.125**	1
	Sig. (2-tailed)	.000	
	N	1158	1158
**. Correlation is significant at the 0.01 level (2-tailed).			

Table 9:Coefficients for regression equation

	Unstandardized Coefficients		Standardized Coefficients		
Model	B	Std. Error	Beta	T	Sig.
1 (Constant)	15.104	2.062		7.325	.000
K_Bitsum	.036	.007	.153	5.255	.000
C_Bitsum	.230	.065	.103	3.529	.000
a. Dependent Variable: P_Bitsum					

The above data, shown in Table 9 provides us with a generalized regression equation which significantly provides this effect between Bitsums of plaintext, ciphertext and keys respectively. The equation is as follows.

$$P\_Bitsum = 15.104 + .036\ (K\_Bitsum) + .230\ (C\_Bitsum)$$

This means that one unit change in K_Bitsum will increase .036 unit in P-Bitsum and one unit change in C_Bitsum will increase .230 unit of P_Bitsum where the other constant factors are 15.104.

4.6 Implications of Bitsum Algorithm on TEA and XTEA when the Chosen key follows a specific pattern

First we chose TEA to investigate for its strengths and weaknesses. The results of reduced key size TEA had been produced in the previous section [187]. Now, in this section the results of cryptanalysis on 128- bit key TEA will be generated.

Implementation of TEA was done. Around 10000 plaintexts were enciphered and 50000 records were generated. Keys were chosen at random. This data was then analyzed, taking Bitsum of plaintext, key, and ciphertext under consideration. Again the intention was to find out some correlation between Bitsum of plaintext, key, or ciphertext. The value for correlation coefficient was also generated for every message. Figure 24 shows the scatter chart of the relation between Bitsum of key and Bitsum of ciphertext, when the key was chosen at random.

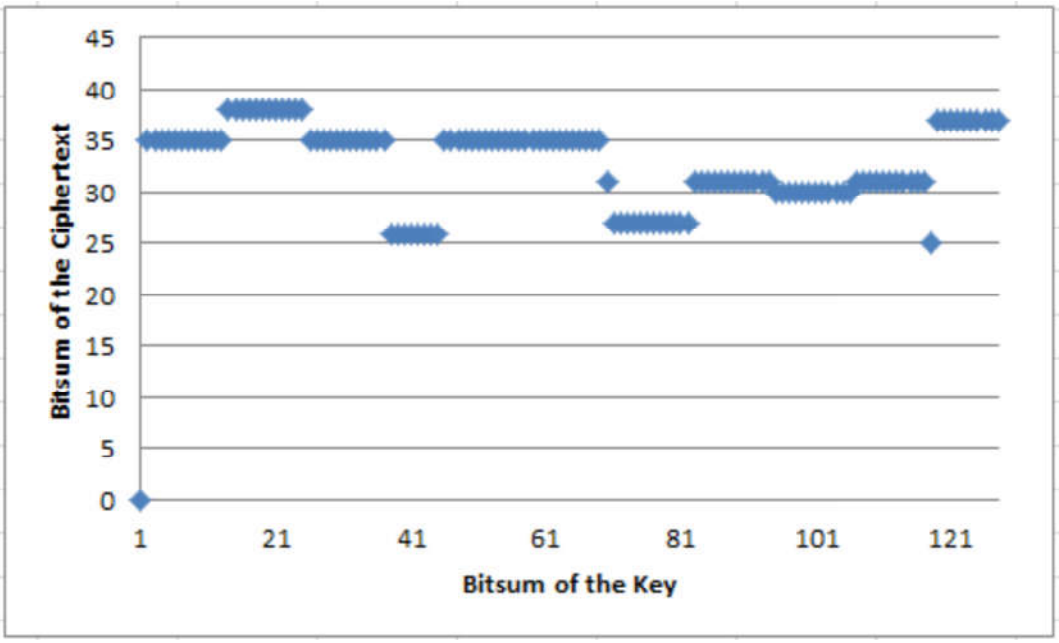

Figure 24: Relation between Bitsum of the Ciphertext with Bitsum of the Key Chosen at Random(TEA).

During analysis of this data, we have realized that there is a pattern of the key, which produce a constant Bitsum of the Ciphertext for every plaintext. Then to endorse that fact we produced more data with such keys while keeping the key constant and changed

the plaintext a number of times. It has been observed that result is factual for explicit patterns of the key.

A set of keys deduced in this research is as follows:

K = {0000...01, 000...011, 00..0111, 00..01111, 00..011111, and so on upto 011111..11,

100....00, 110....0000000, 1110....000000, 111100....000, and so on upto 11111....110,

and 00..010..00, 00..0110..00, 000..111..000, 00..1111..00, and so on upto 0111..1110}.

Results for TEA

Table 10: Value of Bitsum of Ciphertext for a Specific Pattern of the Key for TEA

Key (k) from the Set(K)	Bitsum of the Plain Text	Bitsum of the Key	Bitsum of the Ciphertext
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000011111111111111	14	16	37
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000011111111111111	16	16	37
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000011111111111111	19	16	37
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000011111111111111	23	16	37

When the key is selected from the above set for encrypting the plaintext, the Bitsum of the Ciphertext remains constant, no matter what the plaintext was.

For example, taking the key k Є K is 0000…..00000000000011111111111111. This key is used to encrypt the plaintext. Both TEA and XTEA are used for encryption. Results from both the ciphers are tabulated below.

Table 10 shows the values obtained after encryption. Bitsum of Plaintext, Key, and Ciphertext are taken for analysis. Whenever the value of the Bitsum of the ciphertext for any plaintext is found constant, the pattern of the key can be guessed. It may belong to the set of the deduced keys.

Results for XTEA

Table 11 shows the findings for the XTEA(extended tiny encryption algorithm).

Table 11: Value of Bitsum of Ciphertext for a Specific Pattern of the Key for XTEA

Key (k) from the Set(K)	Bitsum of the Plain Text	Bitsum of the Key	Bitsum of the Ciphertext
000000000000000000000000000000000000 000000000000000000000000000000000000 000000000000000000000000000000000000 00000011111111111111	14	16	31
000000000000000000000000000000000000 000000000000000000000000000000000000 000000000000000000000000000000000000 00000011111111111111	16	16	31
000000000000000000000000000000000000 000000000000000000000000000000000000 000000000000000000000000000000000000 00000011111111111111	19	16	31
000000000000000000000000000000000000 000000000000000000000000000000000000 000000000000000000000000000000000000 00000011111111111111	23	16	31

The table shows the values of Bitsum of plaintext, Bitsum of the key, and Bitsum of ciphertext. These Bitsum values are calculated after encrypting the random messages with the key of a specific pattern. These results are similar to those obtained from the experiment performed on TEA.

4.6.1 Results Obtained with the Specific Key Patterns

The obtained results are further analyzed to prove the fact.

Let us elucidate the cases using the following terminology:

M1, M2……Mn : Messages to be encrypted.

K1, K2…..Kn : Keys.

C1, C2,…Cn : Bitsum of the Ciphertext.

All the cases having specific pattern(K1 to Kn) are listed in this section.

Case 1: Key Pattern K1

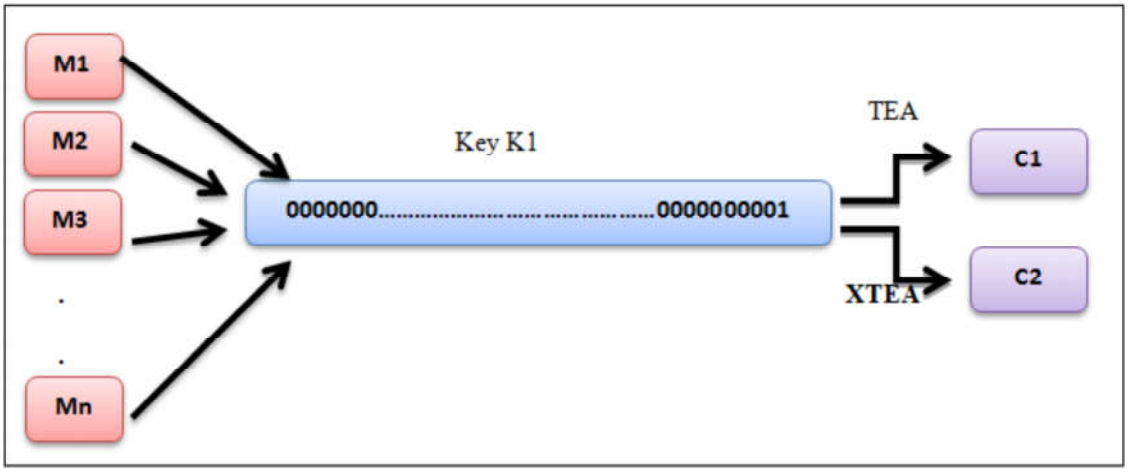

Figure 25: Encryption of Randomly Generated Messages using Key 'K1' (TEA)

TEA and XTEA encryption routines were used to encrypt a set of randomly generated messages. All the messages were encrypted using the key 'K1 – 000000000……..000000000001', a constant Bitsum of Ciphertext 'C1' (in case of

TEA) and ‘C2’ (in case of XTEA) were produced. These results of this experimentation are shown in Figure 25 .

Case 2: Key Pattern K2

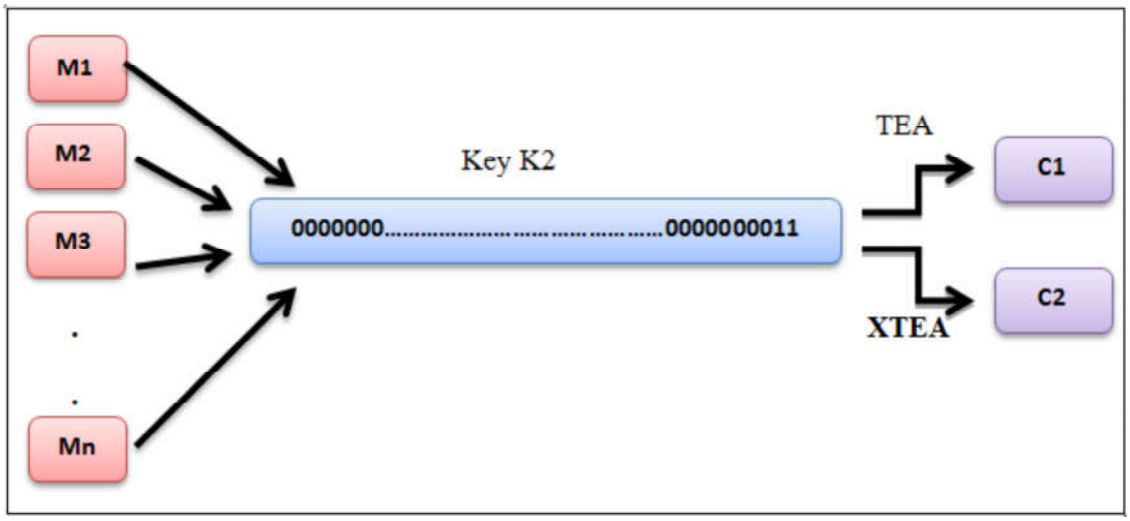

Figure 26: Encryption of Randomly Generated Messages using Key 'K2' (TEA and XTEA)

The results were further analyzed using another key of the same type. Figure 26 displays the results of this experimentation.

Again randomly generated messages were encrypted with ‘K2 – 0000…….000011’ (the next key pattern of the same type). Same kind of results have been produced i.e. the Bitsum of the Ciphertext (C1 and C2 for TEA and XTEA respectively) were constant.

All the key patterns of the same type have been analyzed. The results obtained prove the fact that if the random messages are encrypted with the key of this pattern, the Bitsum of the Ciphertext would remain constant.

And so on…

Case n: Key Pattern Kn

All such patterns were analyzed with same type the key pattern and got the results of same style. This experiment was continued until the last key pattern of the same type. Figure 27 shows the results for Key pattern ‘Kn – 0111111111……111111111111’.

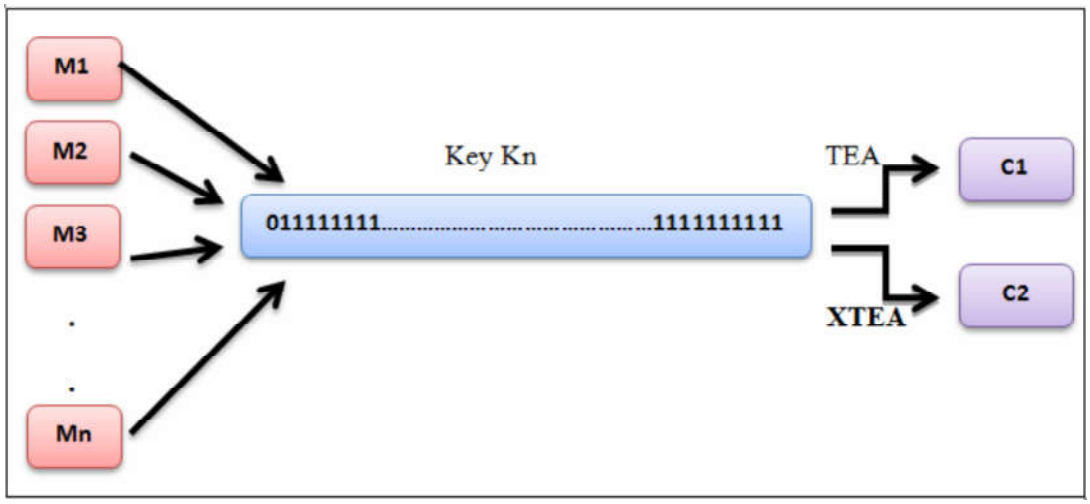

Figure 27: Encryption of Randomly Generated Messages using Key 'Kn' (TEA and XTEA)

A similar key pattern was tried in the reverse order.

Case 1:

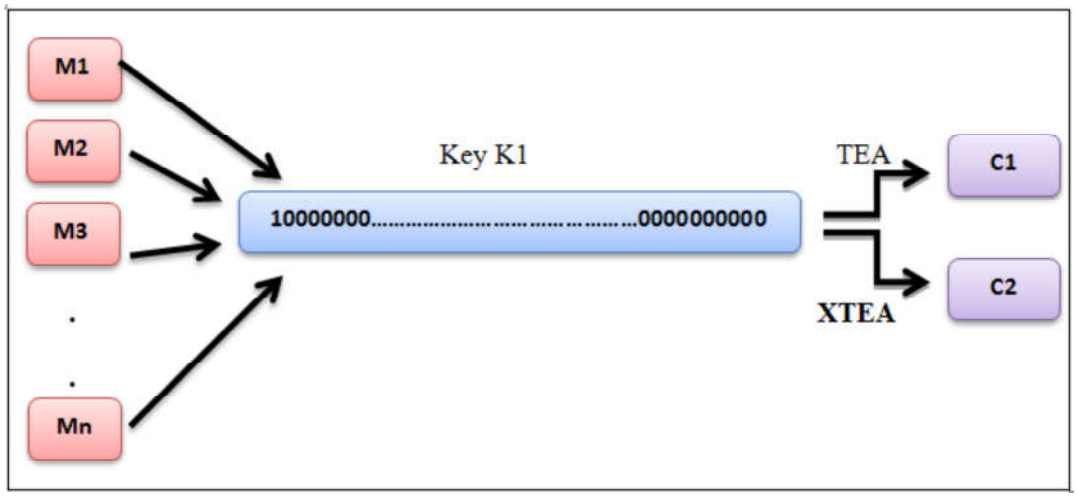

Figure 28: Encryption of Randomly Generated Messages using Reverse Order Key Pattern 'K1' (TEA and XTEA).

Same experimentation was done using randomly generated messages. Even when the order of key pattern was reversed, the results were of the same nature. Results will be shown pictorially in the following sections.

Once again the randomly generated messages were encrypted using reverse order Key pattern K1. Key K1 was 000000…..000000000001 and the reverse order of the Key K1 became 1000000…..000000000000. Figure 28 shows the results of the experimentation.

Case 2:

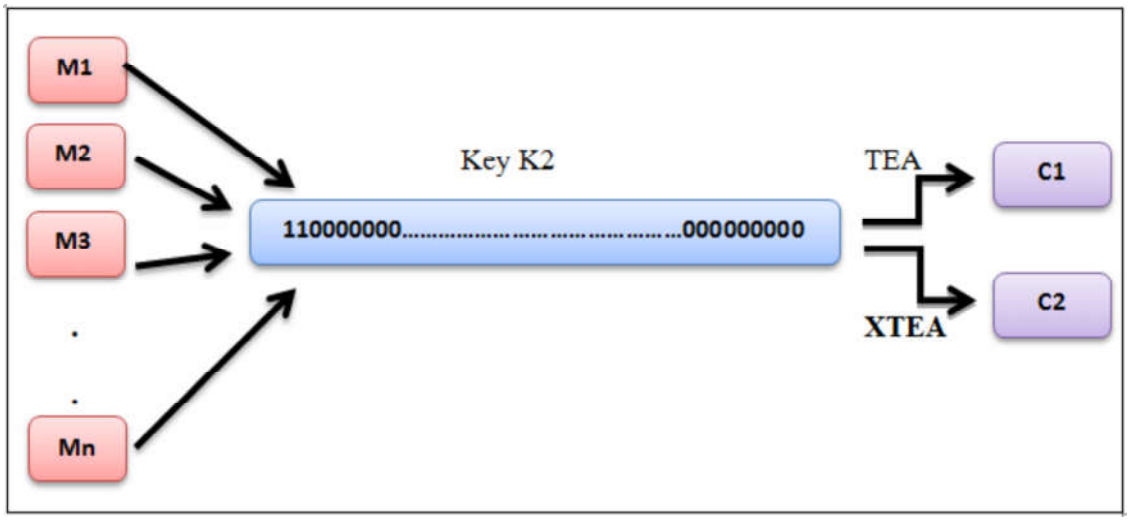

Figure 29:Encryption of Randomly Generated Messages using Reverse Order Key Pattern 'K2' (TEA and XTEA).

Following the pattern of Case 1, Key pattern K2 (000000000000000……00000000000000000000000000000000011) became 110000000000000……00000000000000000000000 in reverse order. This key was used to encrypt the randomly generated messages for TEA and XTEA. Figure 29 shows the result of the experimentation

And so on …

Case n:

This experiment was also performed for all such key patterns and the results found were comparable for all such keys.

At the end, the results obtained for Key pattern 'Kn – 1111111111……..11111111110' are shown in Figure 30.

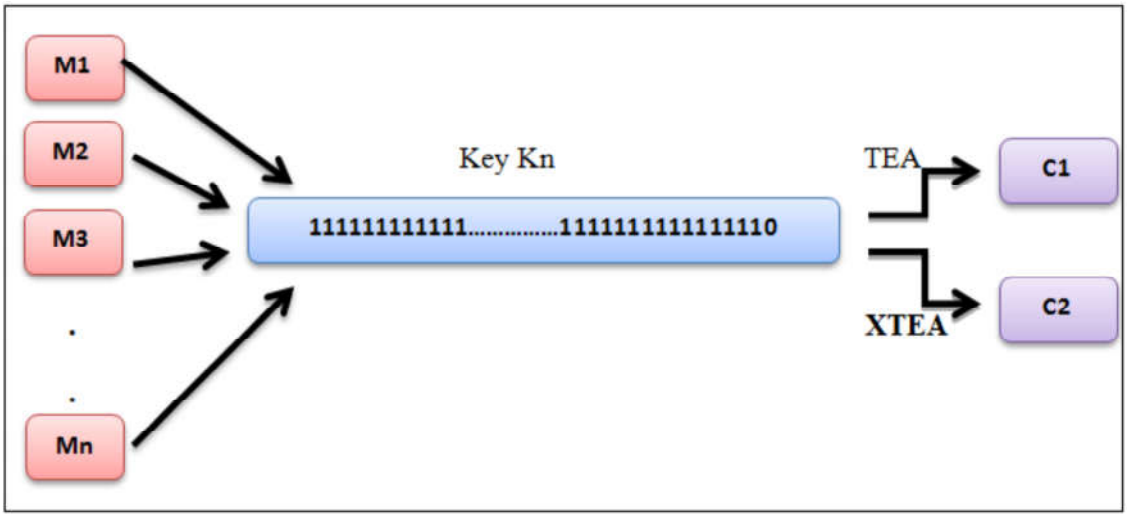

Figure 30: Encryption of Randomly Generated Messages using Reverse Order Key Pattern 'Kn' (TEA and XTEA).

This experiment was extended to test some other patterns also. A pattern of the keys where some number of 1s clustered together and rest of the bits are 0s, then also constant Bitsum of the Ciphertext is produced.

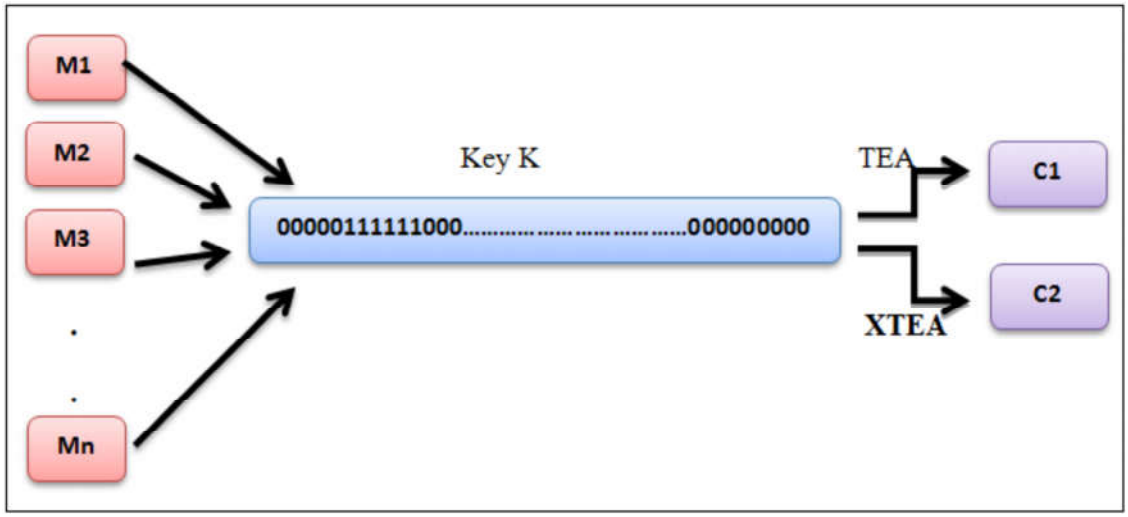

Figure 31: Encryption of the Randomly Generated Messages with a Specific Key Pattern 'K' (TEA and XTEA).

For illustration, it was assumed that M_1, M_2……M_n were the plaintexts to be encrypted and 'C' was a Bitsum of the ciphertext, which is a constant number.

Then we encrypted all the messages with the Key of such a pattern, the Bitsum of the Ciphertext remained same. Figure 31 shows the results of this illustration.

We have continued this experiment for other similar key patterns. Similar results were verified with different messages and different keys of this pattern.

4.6.2 The Key Pattern Theorem on TEA And XTEA

K = {0000...01, 0000...011, 0000...0111, 0000...01111, 0000...011111, and so on upto 01111....11,

10000...00, 11000...00, 11100...00, 111100...00, and so on upto 11111.....110 and

00..010..00, 00..0110..00, 00..111..00, 00..1111..00, and so on upto 01111..1110}

M= { m_1, m_2, m_3,……m_n } where n represents 2^{64}.

Considering k Є K and m_i Є M and computing,

$C = ALG_e(m_i, k)$

$m_i = ALG_d(C, k)$

$C_{BS}= ALG_e(m_i , k)$ is constant for constant Bitsum of Key for whatever value of m_i

Where ALG_e is encryption routine of TEA or XTEA

ALG_d is the decryption routine of TEA

C_{BS} is the Bitsum of the Ciphertext.

4.7 Results and Discussions

This chapter has focused on the experimentation done to check the implications of Bitsum algorithm on Reduced Tiny Encryption Algorithm, Tiny Encryption Algorithm

and XTEA. In case of Reduced TEA, the results has shown that the cipher is strong only if the Bitsum of the key lies between 14 and 51. If not, then the Bitsum of the ciphertext remains constant. In case of TEA the results of the experimentation has showed the absence of confusion and diffusion property in these ciphers. The results of correlation tests also proved that Bitsum of Ciphertext is related with Bitsum of Key as well as Bitsum of Plaintext.

The analysis of TEA and XTEA revealed a particular set of weak keys. Key pattern theorem is developed using this set of weak keys.

The experimentation would continue in next chapter. Chapter 5 will check the implications of Bitsum algorithm on FEAL, BLOWFISH and AES.

Chapter 5: Experiments on FEAL, BLOWFISH and AES

Observation is a passive science, experimentation is an active science.

-Claude Bernard

The first section of this chapter is devoted to the implementation of Bitsum Algorithm on FEAL(Fast Encryption Algorithm). The second section deals with the cryptanalysis of Blowfish with Bitsum Algorithm. The last section of this chapter delineates the implementation of Bitsum Algorithm on AES.

5.1 Implementation of Bitsum Algorithm on FEAL Algorithm

FEAL is a Feistel structured block cipher that was presented in Eurocrypt'87. Developed by Akihiro Shimizu and Shoji Miyaguchi [28], it is an N-round cipher having 64 bit block size and 64 bit key. FEAL cipher is easy to implement for both hardware and software which makes its acceptance promiscuously. The generic algorithm of FEAL has 4 rounds. This algorithm was implemented using Bitsum Algorithm.

5.1.1 Implementation

As FEAL was the cipher to be investigated, so around 10000 messages (plaintexts) were enciphered using FEAL. Random keys were chosen to encipher the messages. These messages were encrypted using those random keys. Around 40000-50000 records were produced to be analyzed. Bitsum of the plaintext, key, and ciphertext were taken. The idea was to derive a correlation between the Bitsum of the ciphertext and Bitsum of key and/or plaintext. During this analysis, it was found that there exists a correlation between

- Bitsum of Plaintext and Bitsum of Ciphertext.
- Bitsum of the Key and Bitsum of the Ciphertext

Case 1:

The correlation between Bitsum of the Plaintext and Bitsum of the Ciphertext was analyzed. Figure 32 depicts the scatter chart developed from the analysis. It is clearly visible in the scatter chart that value of the Bitsum of the ciphertext (BCT) lies in between 20 to 45 only irrespective of the value of Bitsum of the plaintext (BPT).

Though no weakness for this cipher could be detected but this cipher showed a pattern that Bitsum of the ciphertext always lies between 20 and 45.

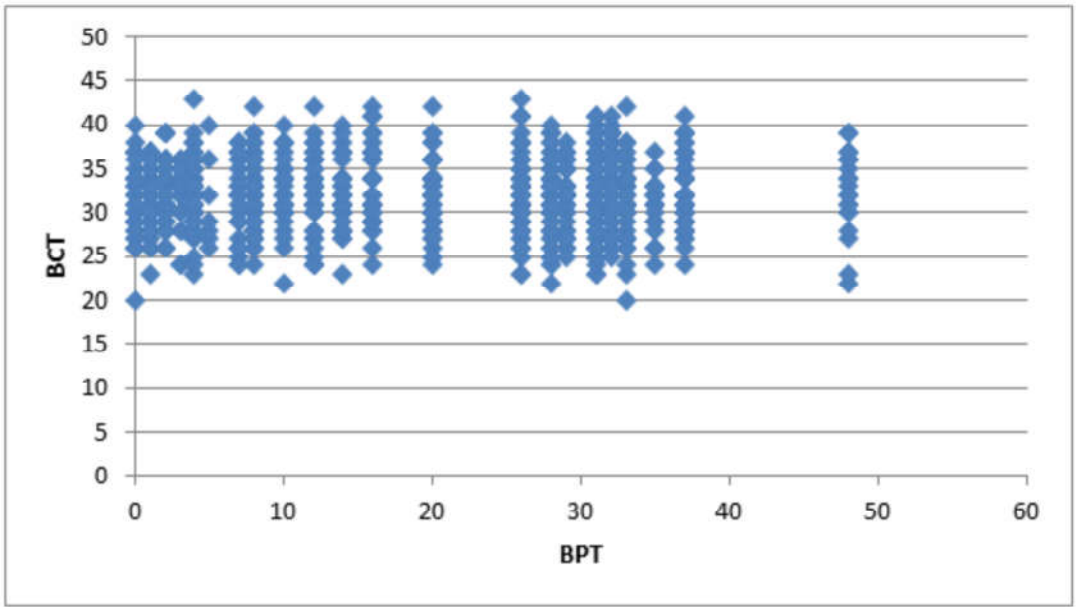

Figure 32: Correlation between Bitsum of Plaintext and Bitsum of Ciphertext (FEAL)

Case 2:

In the same way an analysis was attempted to find the correlation between Bitsum of the key and Bitsum of the ciphertext. Below is the scatter chart showing this correlation. It is again visible from the chart that the value of the Bitsum of Ciphertext lies between 20 and 45 only.

So the value of Bitsum of plaintext (BPT) and Bitsum of the key (BK) do not seem to make any difference on the range of values of Bitsum of the ciphertext(BCT).

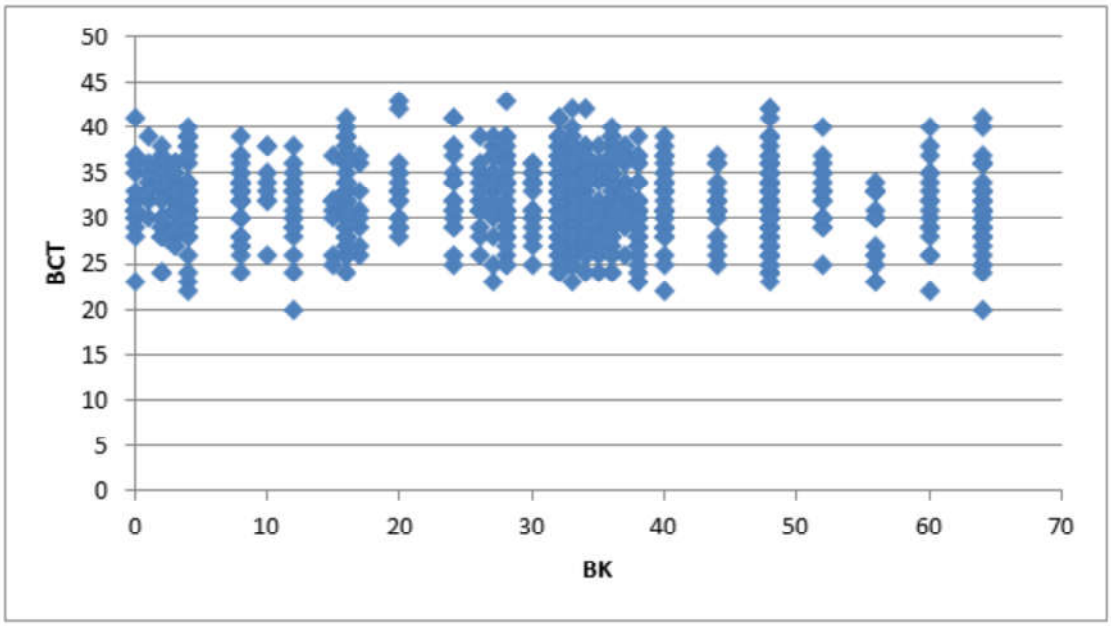

Figure 33: Correlation between Bitsum of Key and Bitsum of Ciphertext (FEAL)

5.1.2 Analysis of FEAL Algorithm using SPSS

Objective 1: To examine the relation between the Bitsums of key, plain texts and cipher texts.

H_0= Bitsums of cipher text and key are not correlated.

H_1= Bitsums of cipher text and key are correlated.

To test this hypothesis we have used correlation test in SPSS and the result found is shown in Table 12.

Table 12: Correlation statistics between Bitsums of plaintext and ciphertext for FEAL

		P_Bitsum	C_Bitsum
P_Bitsum	Pearson Correlation	1	-.008
	Sig. (2-tailed)		.701
	N	2184	2184
C_Bitsum	Pearson Correlation	-.008	1
	Sig. (2-tailed)	.701	
	N	2184	2184

As the significance value of the test is .701 which is > .05, it means that there is no correlation between the Bitsums of the keys and the cipher texts indicating that confusion property holds here. To strengthen this result we have also checked the correlation test results as between the Bitsums of cipher texts and the plaintexts to examine the diffusion property. We have found the result in SPSS as shown in Table 13 with the following hypothesis.

H_0= Bitsums of cipher text and key are not correlated.

H_1= Bitsums of cipher text and key are correlated.

Table 13:Correlation statistics between Bitsums of plaintext and ciphertext for FEAL

		C_Bitsum	K_Bitsum
C_Bitsum	Pearson Correlation	1	-.040
	Sig. (2-tailed)		.059
	N	2184	2184
K_Bitsum	Pearson Correlation	-.040	1
	Sig. (2-tailed)	.059	
	N	2184	2184

As the significance value of the test is .059 which is > .05, it means that there is no correlation between the Bitsums of the plaintexts and the cipher texts, which indicates that the diffusion property also holds here.

Table 14: Coefficients for regression equation for FEAL

	Unstandardized Coefficients		Standardized Coefficients		
Model	B	Std. Error	Beta	T	Sig.
1 (Constant)	22.497	2.585		8.704	.000
K_Bitsum	.069	.018	.081	3.770	.000
C_Bitsum	-.018	.077	-.005	-.233	.816
a. Dependent Variable: P_Bitsum					

We can estimate the overall effect of the variables under consideration as per Table 14. Table 14 provides us with a generalized regression equation which significantly provides this effect between Bitsums of plain text, cipher text and keys respectively. The equation is as follows.

$$P\_Bitsum = 22.497 + .069 (K\_Bitsum) - .018 (C\_Bitsum)$$

This means that one unit change in K_Bitsum will increase .069 unit in P-Bitsum and one unit change in C_Bitsum will decrease .018 unit of P_Bitsum where the other constant factors are 22.497. Another noticeable thing from the above table is that the significance value of C_Bitsum is .816 which is > .05. This means C_Bitsum alone cannot significantly predict the P_Bitsum, but when it is associated with K_Bitsum and the Constant (each significance value is 0 <.05), the overall prediction of P_Bitsum is significant by the regression model.

5.2 Implementation of Bitsum Algorithm on Blowfish

Blowfish [29] is a variable-length key, 64-bit block cipher. It is fast, compact, simple and secure cipher. The algorithm comprises of two parts: one part is data encryption and another part is key expansion. Key expansion transforms a key of at most 448 bits into various subkey arrays totalling 4168 bytes. Its encryption is also based on 16-round Feistel network. This cipher is suitable for the systems where key does not change frequently.

5.2.1 Experimentation

In this section, the cipher under investigation was Blowfish which is also a fiestel cipher. Around 10000 messages(plaintexts) were enciphered using Blowfish. Random keys were chosen to encipher the messages. These messages were encrypted using those random keys. Around 40000-50000 records were produced to be analyzed. Bitsum of the plaintext, key and ciphertext were taken. The idea was to derive a correlation between the Bitsum of ciphertext with Bitsum of key and/or plaintext. During this analysis, it was found that there exists a correlation between

- Bitsum of Plaintext and Bitsum of Ciphertext.
- Bitsum of the Key and Bitsum of the Ciphertext

Case 1:

The correlation between Bitsum of the Plaintext and Bitsum of the Ciphertext was analyzed. Figure 34 depicts the scatter chart developed from the analysis. It is clearly visible in the scatter chart that value of the Bitsum of the ciphertext (BCT) lies between 20 to 45 only irrespective of the value of Bitsum of the plaintext (BPT).

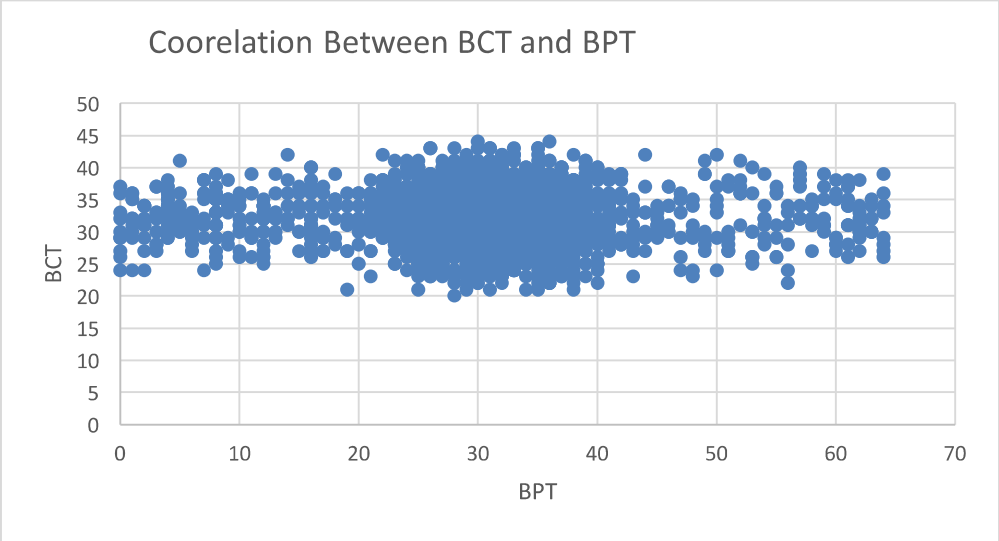

Figure 34: Correlation between Bitsum of Plaintext and Bitsum of Ciphertext (Blowfish)

This cipher did not yield any weak keys. But this cipher shows a pattern that Bitsum of the ciphertext always lies between 20 and 45.

Case 2:

In the same way, an analysis was attempted to find the correlation between Bitsum of the key and Bitsum of the ciphertext. Below is the scatter chart showing this correlation. It is again visible from the chart that the value of the Bitsum of Cipher text lies between 20 and 45 only.

The value of Bitsum of plaintext (BPT) and Bitsum of the key (BK) is not making any difference on the range of values of Bitsum of the ciphertext (BCT).

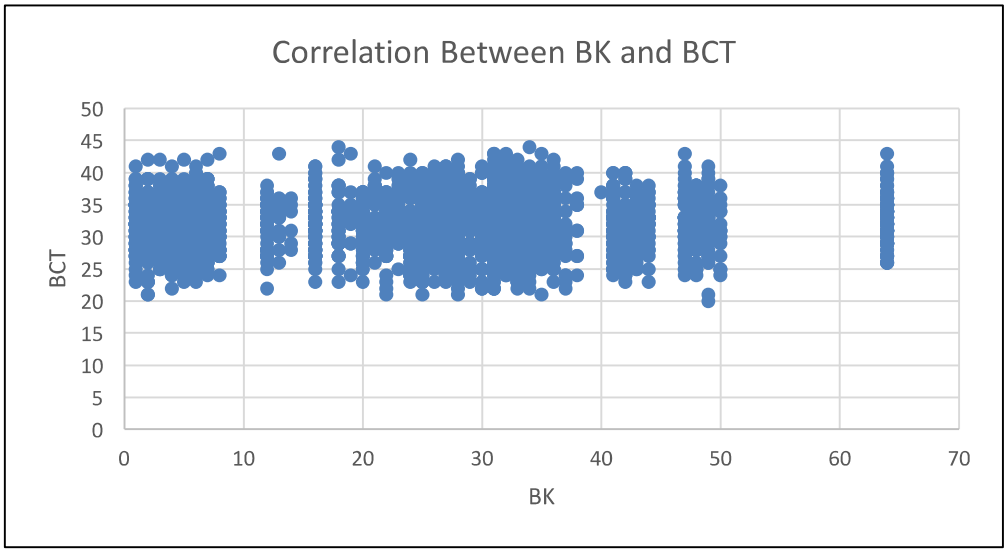

Figure 35:Correlation Between Bitsum of Key and Bitsum of Ciphertext(Blowfish)

5.2.2 Analysis of Blowfish results using SPSS

Objective: To examine the relation between the Bitsums of key, plain texts and cipher texts.

H_0= Bitsums of cipher text and key are not correlated.

H_1= Bitsums of cipher text and key are correlated.

To test this hypothesis, we have used correlation test in SPSS and the result obtained is shown in Table 15.

Table 15: Correlation between Bitsums of key and ciphertext for Blowfish

		K_Bitsum	C_Bitsum
K_Bitsum	Pearson Correlation	1	.024
	Sig. (2-tailed)		.181
	N	3042	3042
C_Bitsum	Pearson Correlation	.024	1
	Sig. (2-tailed)	.181	
	N	3042	3042

As the significance value of the test is .181, which is > .05, we are accepting the null hypothesis which suggests that there is no correlation between the Bitsums of the keys and the cipher texts. This indicates that confusion exists in Blowfish algorithm. For strengthening this result, we have also checked the correlation test results between the Bitsums of cipher texts and the plaintexts to examine the diffusion property with the hypothesis stated below. Table 16 summarizes the results.

H_0= Bitsums of cipher text and plain text are not correlated.

H_1= Bitsums of cipher text and plaintext are correlated.

Table 16: Correlation statistics between Bitsums of plaintext and ciphertext for Blowfish

		C_Bitsum	P_Bitsum
C_Bitsum	Pearson Correlation	1	-.001
	Sig. (2-tailed)		.956
	N	3042	3042
P_Bitsum	Pearson Correlation	-.001	1
	Sig. (2-tailed)	.956	
	N	3042	3042

As the significance value of the test is .956 which is > .05, we accept the null hypothesis which says that there is no correlation between the Bitsums of the plaintexts and the cipher texts. This establishes that Blowfish algorithm has diffusion property.

As an objective of our cryptanalysis, we wish to derive a generalized formula through which we can calculate the Bitsum of the plaintext directly from the Bitsums of the key and Bitsums of the cipher text. The data shown in Table 17 provides us with a generalized regression equation which significantly exhibits this effect between Bitsums of plain text, cipher text and keys.

The equation drawn from the values in the table is as follows.

P_Bitsum = 33.944 - .121 (K_Bitsum) + .008 (C_Bitsum)

This means that one unit change in K_Bitsum will decrease .121 unit in P-Bitsum and one unit change in C_Bitsum will increase .008 unit of P_Bitsum where the other constant factors are 33.944.

Table 17: Coefficients for regression equation for Blowfish

	Unstandardized Coefficients		Standardized Coefficients		
Model	B	Std. Error	Beta	T	Sig.
1 (Constant)	33.944	1.365		24.869	.000
K_Bitsum	-.121	.012	-.187	-10.485	.000
C_Bitsum	.008	.041	.004	.198	.843
a. Dependent Variable: P_Bitsum					

Another noticeable thing from the above table is that the significance value of C_Bitsum is .843 which is > .05. This means C_Bitsum alone cannot significantly predict the P_Bitsum, but when it is associated with K_Bitsum and the Constant (each significance value is 0 <.05), the overall prediction of P_Bitsum becomes significant by the regression model.

5.3 Implementation of Bitsum Algorithm on AES

This section is devoted to investigation of AES, a cipher with block size of 128 bit and key size that is variable, it can be 128, 192, or 256 bits. It is an iterative cipher and not a Feistel cipher. Around 500 messages (plaintexts) were enciphered using AES. Random keys were chosen to encipher the messages producing around 4000-5000 records for analysis. Bitsum of the plaintext, key and ciphertext were calculated and recorded. The idea was to find a correlation between the Bitsum of ciphertext with Bitsum of key and/or plaintext. By analysing this data we found some interesting patterns of the appearance of Bitsum of the Ciphertext. For this frequency analysis, sample of around 500 records of each variant of AES i.e. AES-128, AES- 192 and AES-256 were taken.

5.3.1 AES – 128

In this section, the algorithm under investigation was AES-128. The intention is again same, to find some correlation between Bitsum of Plaintext, Key, and Ciphertext. The

scatter chart in Figure 36 was drawn to find the relationship between Bitsum of the ciphertext and the key.

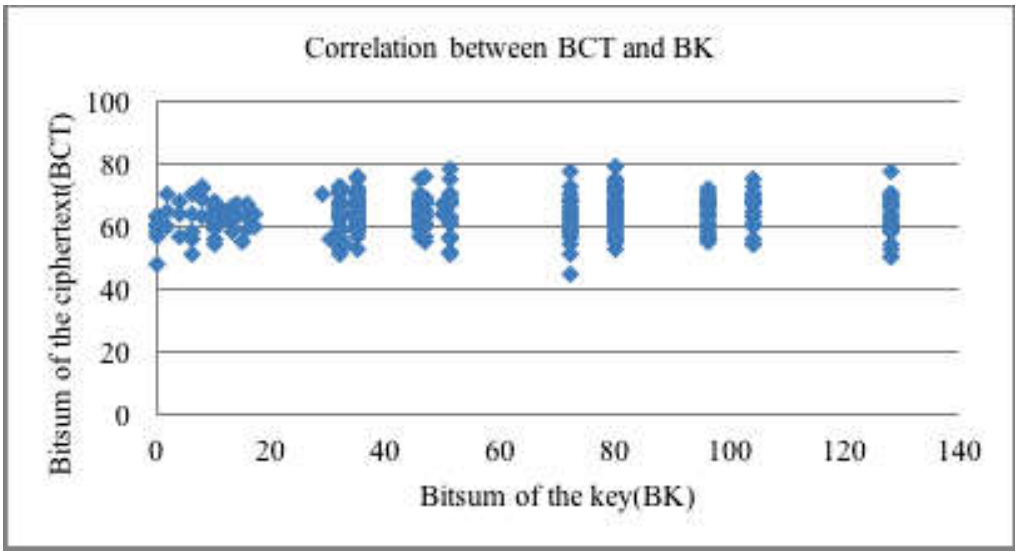

Figure 36: Correlation between Bitsum of Key and Bitsum of Ciphertext

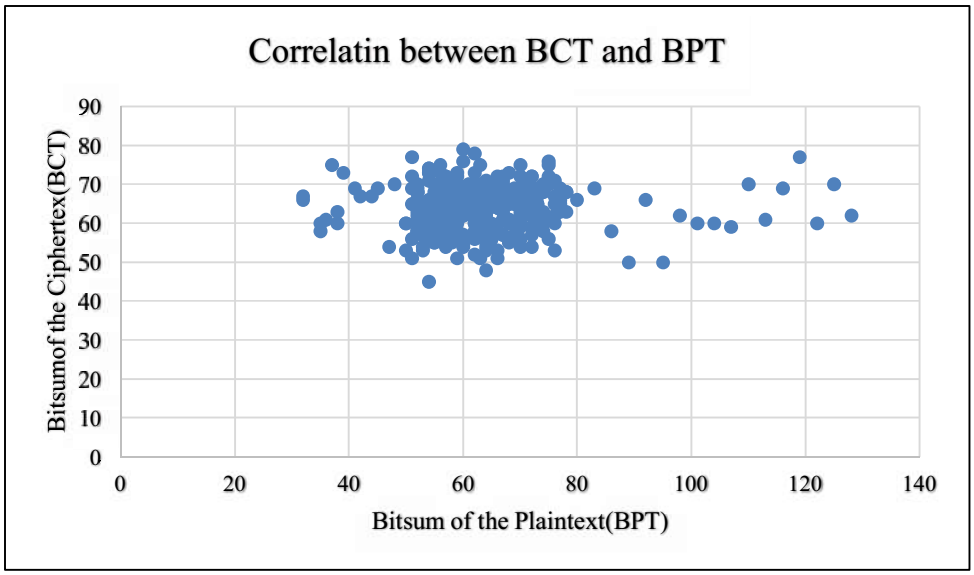

Figure 37: Correlation between Bitsum of Plaintext and Bitsum of Ciphertext

It is visible from the values in the scatter charts that Bitsum of the Ciphertext varies only from 45 to 82. Frequency analysis of Bitsum of Ciphertext is shown in Figure 38. The value of the maximum occurrence as shown in the graph is 63.

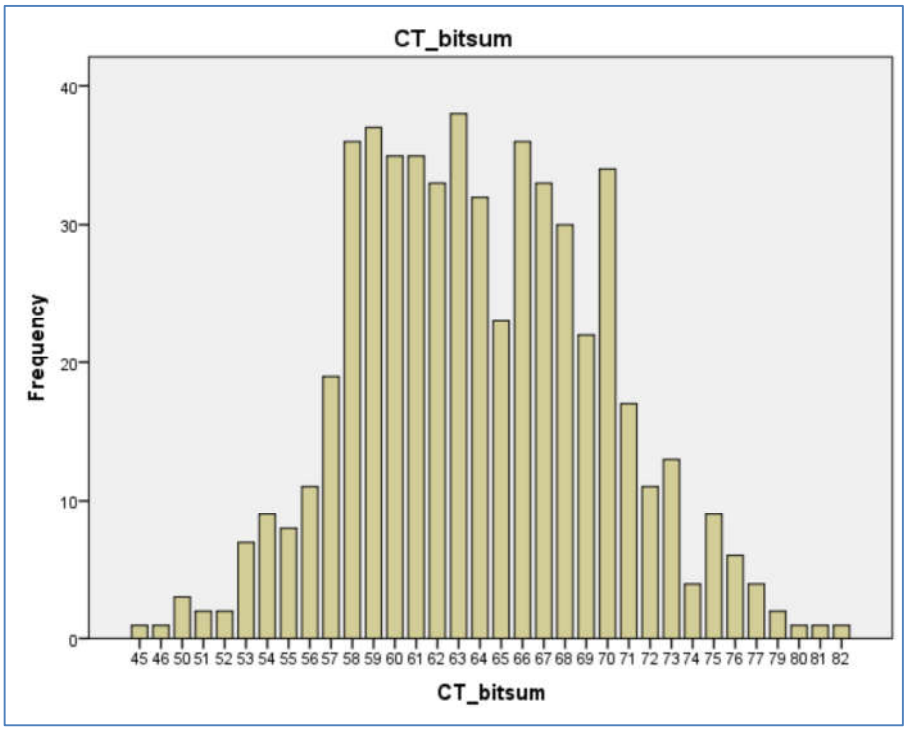

Figure 38: Frequency Analysis of Bitsum of Ciphertext for AES – 128

5.3.2 Analysis of 128 Bit AES using SPSS

Objective: To examine the diffusion property in AES-128. To test the correlation, null hypothesis was created.

H_0: No correlation was observed between the Bitsums of plaintext and ciphertext.

H_1: There is significant correlation between the Bitsums of plaintext and ciphertext

Table 18: Correlations Table of Bitsum of Plaintext and Ciphertext

		P_Bitsum	C_Bitsum
P_Bitsum	Pearson Correlation	1	.031
	Sig. (2-tailed)		.471
	N	556	556
C_Bitsum	Pearson Correlation	.031	1
	Sig. (2-tailed)	.471	
	N	556	556

In Table 18 the significance value is .471 which is > 0. It means the acceptance of the null hypothesis signifying that there is no correlation between the Bitsums of plaintext and ciphertext.

Objective: To examine the confusion property in AES-128. To test the correlation, null hypothesis was created.

H_0: There is no correlation between the Bitsums of key and ciphertext.

H_1: There is significant correlation between the Bitsums of key and ciphertext.

Table 19: Correlation Table of Bitsum of Ciphertext and Key

		CT_bitsum	K_bitsum
CT_bitsum	Pearson Correlation	1	.014
	Sig. (2-tailed)		.739
	N	556	556
K_bitsum	Pearson Correlation	.014	1
	Sig. (2-tailed)	.739	
	N	556	556

The significance value shown in Table 19 is .739 which is > 0. It means the null hypothesis is acceptable. It signifies that there is no correlation between the Bitsums of key and ciphertext. So, the overall correlation test on 128 bit key AES shows that this algorithm has both confusion and diffusion property.

For checking the overall effect of Bitsums of key and ciphertext, the observed data was put to regression test. The results obtained are tabulated in Table 20.

Table 20: Model Summary

Model	R	R Square	Adjusted Square	Std. Error of the Estimate
1	.251[a]	.063	.059	22.352

a. Predictors: (Constant), CT_Bitsum, K_Bitsum
b. Dependant Variable: PT_Bitsum

The model summary says that the regression model can predict the overall effect of ciphertext and key on the plaintext significantly by 63%.

Table 21: ANOVA Table

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	18517.706	2	9258.853	18.532	.000[b]
	Residual	276290.271	553	499.621		
	Total	294807.977	555			

a. Dependent Variable: PT_bitsum

b. Predictors: (Constant), CT_bitsum, K_bitsum

H_0: The regression model is not significant.

H_1: The regression model is significant.

The ANOVA table (Table 21) also emphasize the overall effect and the significance of the model. As the significance value of ANOVA test is 0 which is < .05, we reject the null hypothesis and accept the alternative hypothesis.
It means that, the regression model is significant for our purpose to predict the effect of ciphertext Bitsums and key Bitsums on the plaintext Bitsums.

Table 22 details about the coefficients for each of the parameter to deduce the regression equation. Therefore we can write the regression equation as below

PT_Bitsum = 49.101 – (.159 X K_Bitsum) + (.135 X CT_Bitsum)

The regression equation can be explained as, the one unit change in K_Bitsum can reduce .159 units of PT_Bitsum and similarly, one unit change in CT_Bitsum can increase .135 units of PT_Bitsum.

The regression line depicting the above equation is shown in Figure 39.

Table 22: Coefficients Table

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	49.101	10.532		4.662	.000
	K_bitsum	-.159	.026	-.249	-6.042	.000
	CT_bitsum	.135	.163	.034	.829	.407

a. Dependent Variable: PT_bitsum

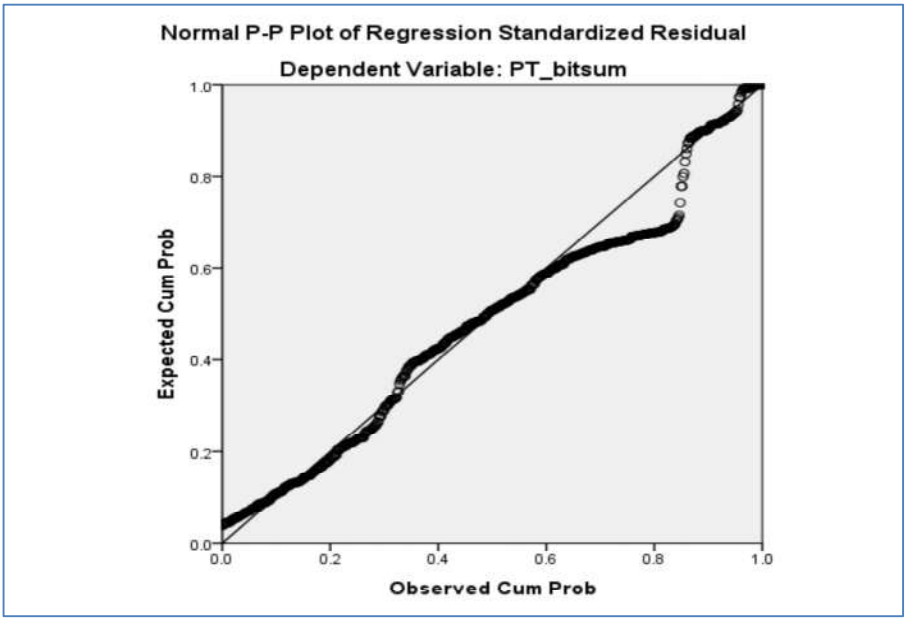

Figure 39: Regression Line

Another noticeable thing in Table 22 is that CT_Bitsum cannot predict alone about the regression model as the significance value is .407 which is > 0. But, when this parameter is attached with K-Bitsum and the constant, the overall regression model becomes significant.

5.3.3 AES – 192

Cipher under investigation in this section is AES – 192. The intention would be to find the correlation between Bitsum of Ciphertext, Key, and Plaintext.

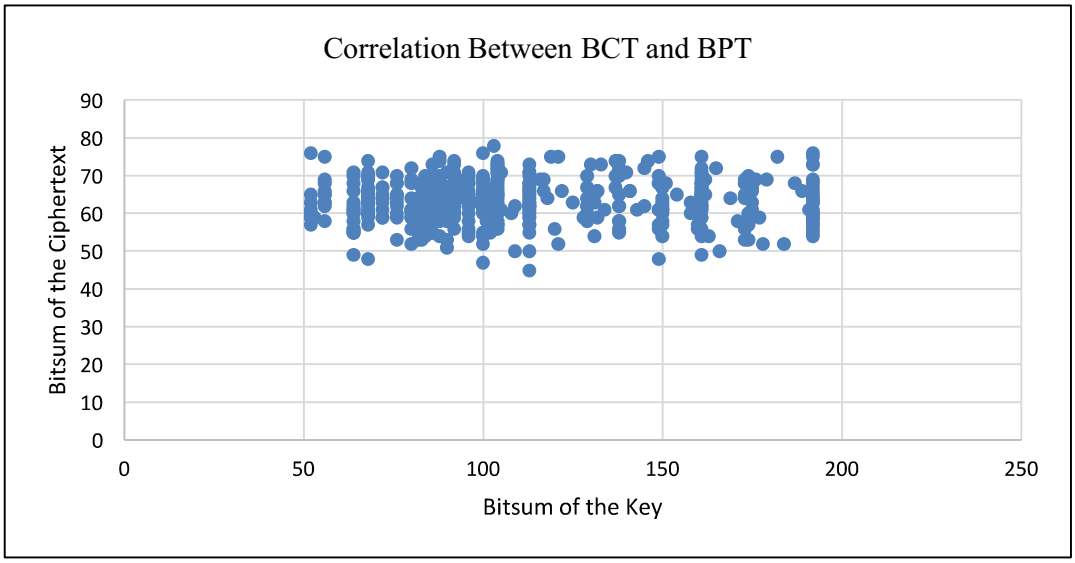

Figure 40: Scatter Chart for AES - 192 data

.

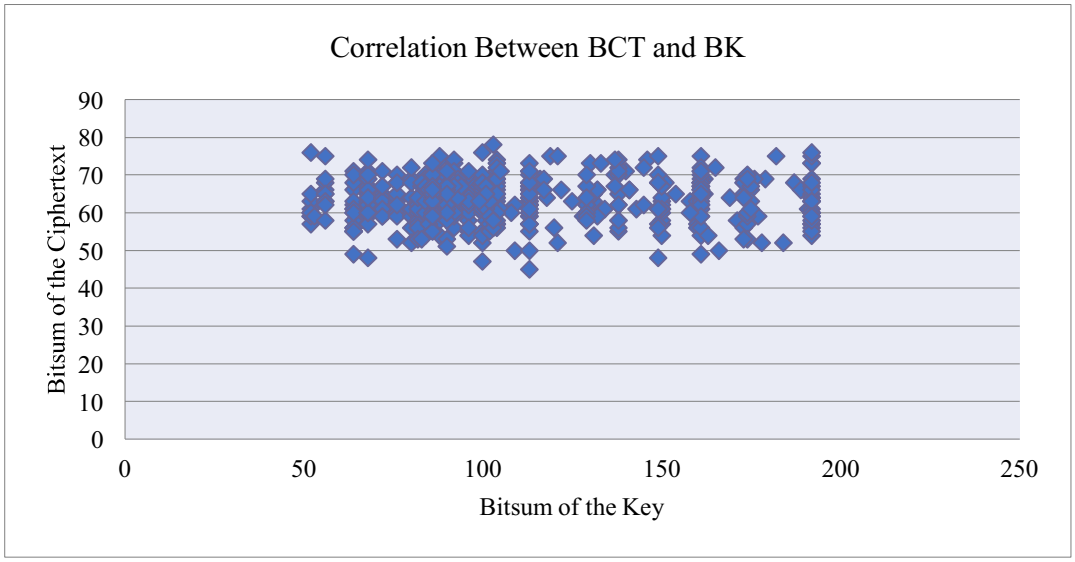

Figure 41: Correlation between Bitsum of key and Bitsum of Ciphertext

Figure 40 represents the correlation between Bitsum of Ciphertext and Bitsum of plaintext .

Figure 41 represents the correlation between Bitsum of Ciphertext and Bitsum of Key.

Frequency analysis of Bitsum of Ciphertext is shown in Figure 42. The values in both the charts lead to the inference that Bitsum of the Ciphertext varies only from 45 to 78.

Figure 42 shows the graph for frequency analysis of this data. The value of the maximum occurrence as shown in the graph is 65.

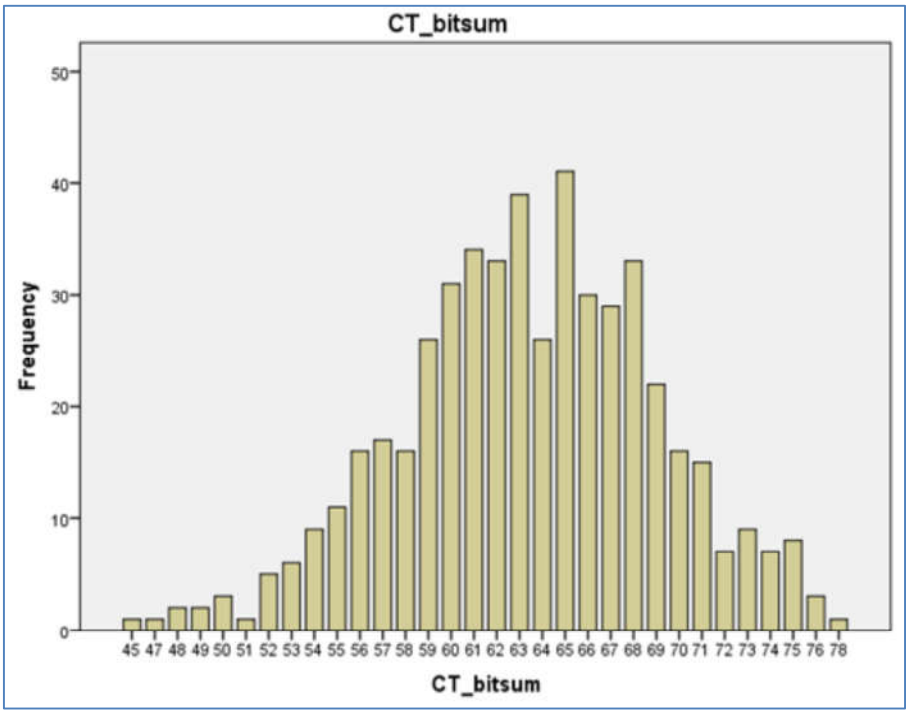

Figure 42: Frequency Analysis of Bitsum of Ciphertext for AES - 192

5.3.4 Analysis of 192 Bit AES using SPSS

Objective is to test the correlation between Bitsum of plaintext and Bitsum of the ciphertext. Null hypothesis is created for the purpose of testing .

H_0: There is no correlation between the Bitsums of plaintext and ciphertext.

H_1: There is significant correlation between the Bitsums of plaintext and ciphertext

As the data shown in Table 23 the significant value is .229 which is > .05. Therefore, we accept the null hypothesis which says that there is no correlation between Bitsums of ciphertext and plaintext. This means that confusion property exists in the 192 bit encryption of AES.

Table 23: Correlation Table for Bitsum of Plaintext and Ciphertext

Correlations

		PT_bitsum	CT_bitsum
PT_bitsum	Pearson Correlation	1	-.054
	Sig. (2-tailed)		.229
	N	500	500
CT_bitsum	Pearson Correlation	-.054	1
	Sig. (2-tailed)	.229	
	N	500	500

Another objective is to test the correlation between Bitsum of key and Bitsum of ciphertext. Again for this purpose, a null hypothesis is created.

H_0: There is no correlation between the Bitsums of key and ciphertext.

H_1: There is significant correlation between the Bitsums of key and ciphertext.

Table 24: Correlation Table for Bitsum of Ciphertext and Key

Correlations

		CT_bitsum	K_bitsum
CT_bitsum	Pearson Correlation	1	.034
	Sig. (2-tailed)		.446
	N	500	500
K_bitsum	Pearson Correlation	.034	1
	Sig. (2-tailed)	.446	
	N	500	500

The data shown in Table **24** articulates that, the significant value of correlation test is .446 which is > .05. Therefore, we accept the null hypothesis which signifies that there is no correlation between Bitsums of ciphertext and key. This means that diffusion property exists in the 192 bit encryption of AES.

If confusion and diffusion property exists here, then we have to check that what percentage of the ciphertext can be explained by Bitsum and we have to test the other constants affecting the process.

Table 25: Model Summary

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.467[a]	.218	.215	21.835

a. Predictors: (Constant), CT_bitsum, K_bitsum

b. Dependent Variable: PT_bitsum

The model summary in Table 25 details that only 21.8% can be explained by the ciphertext Bitsums and key Bitsums. It means there are also other factors remaining which effect the Bitsum calculation of the plaintext for deciphering. If we consider all those unknown factors as constant, we can check from Table 26 that these constant factor having a significant effect on the process.

Table 26: Coefficients Table

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	38.760	11.340		3.418	.001
	K_bitsum	.297	.025	.464	11.690	.000
	CT_bitsum	-.306	.174	-.070	-1.757	.080

a. Dependent Variable: PT_bitsum

5.3.5 AES – 256

The cipher under investigation in this section is AES – 256. The data was generated to analyze the correlation between Bitsum of Ciphertext, Key or Plaintext.

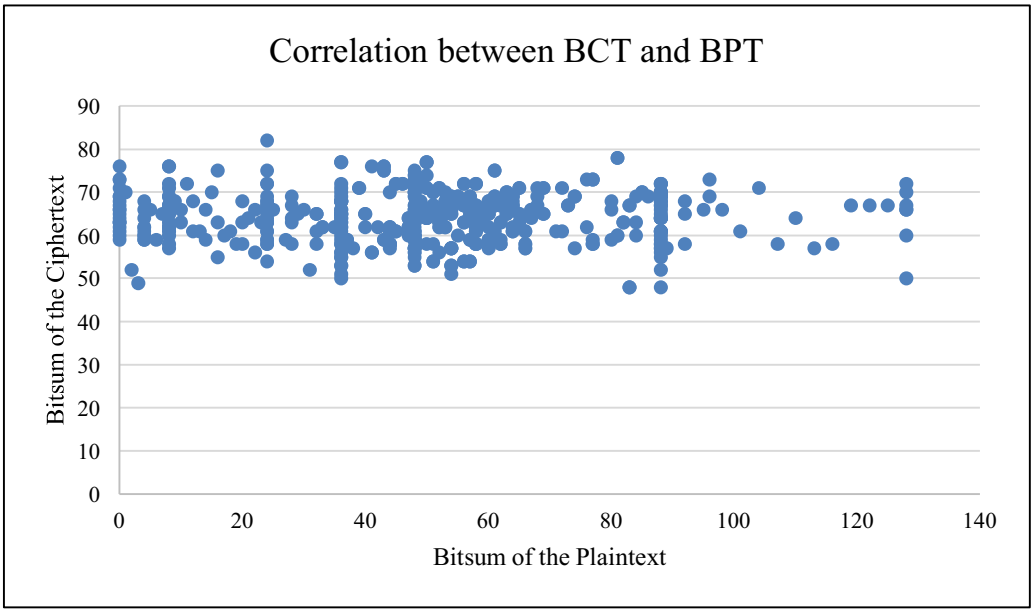

Figure 43: Scatter Chart for AES - 256 data

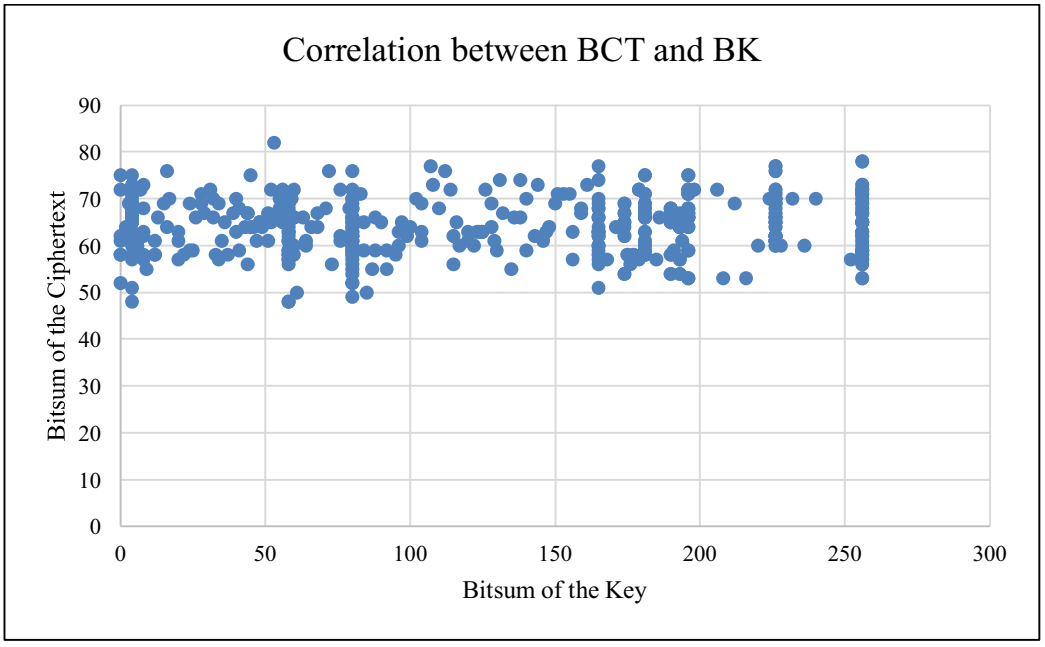

Figure 44: Correlation between Bitsum of Ciphertext and Bitsum of Key

Scatter charts in Figure 43 and Figure 44 are drawn for the the data. It is evident from the values in the charts that Bitsum of the Ciphertext varies only from 48 to 82.

Figure 45 depicts the frequency analysis of Bitsum of ciphertext. The frequency of their appearance is also visible in the chart. The maximum occurrence as shown in the graph is 66.

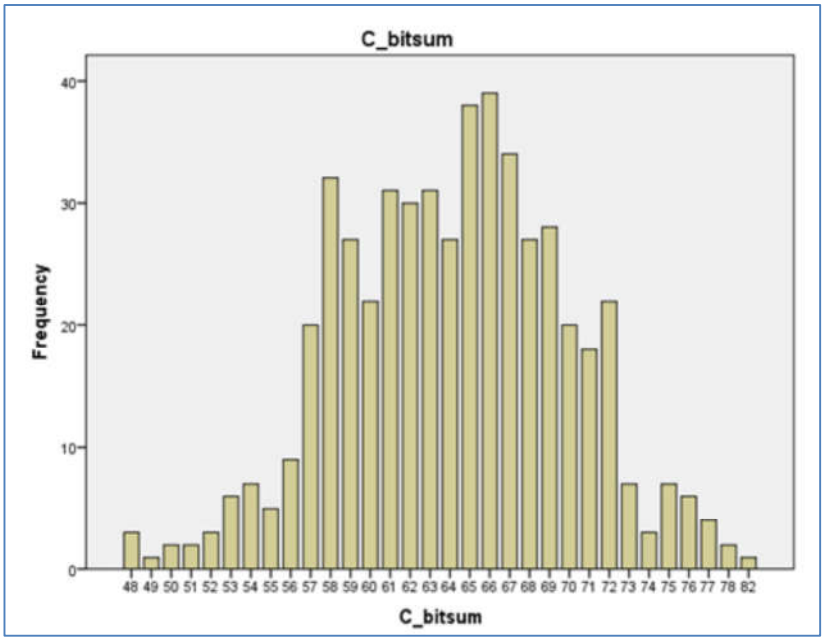

Figure 45: Frequency Analysis of Bitsum of Ciphertext for AES - 256

5.3.6 Analysis of 256 Bit AES using SPSS

The objective of this analysis is to test the confusion property of the cipher. This could be achieved by checking the correlation between between Bitsum of the plaintext and Bitsum of the ciphertext. A null hypothesis was created for this testing.

H_0: No correlation was observed between the Bitsums of plaintext and ciphertext.

H_1: Significant correlation was observed between the Bitsums of plaintext and ciphertext.

As the data shown in Table 27, the significant value is .929 which is > .05. Therefore, we accept the null hypothesis which says that there is no correlation between bitsums of ciphertext and plaintext. This means that confusion property exists in the 256 bit encryption of AES.

Table 27: Correlation Table depicting Bitsum of Plaintext and Bitsum of Ciphertext

		P_Bitsum	C_Bitsum
P_Bitsum	Pearson Correlation	1	-.004
	Sig. (2-tailed)		.929
	N	514	514
C_Bitsum	Pearson Correlation	.052	1
	Sig. (2-tailed)	.929	
	N	514	514

Another null hypothesis was created to check the correlation between Bitsum of the key and Bitsum of the ciphertext. This was done to determine the diffusion property of the cipher.

H_0: No correlation was detected between the Bitsums of key and ciphertext.

H_1: Significant correlation was observed between the Bitsums of key and ciphertext

As per the data in Table 28 the significant value of correlation test is .239 which is > .05. Therefore, the null hypothesis was accepted as it signified that there was no correlation between Bitsums of ciphertext and key. This leads to the inference of existence of diffusion property in the 256 bit encryption of AES.

Table 28: Correlation Table for Bitsum of Ciphertext and Bitsum of Key

		C_bitsum	K_bitsum
C_bitsum	Pearson Correlation	1	.052
	Sig. (2-tailed)		.239
	N	514	514
K_bitsum	Pearson Correlation	.052	1
	Sig. (2-tailed)	.239	
	N	514	514

The model summary says that only 0.1% can be explained by the ciphertext Bitsums and key Bitsums.

Table 29: Model Summary for AES - 256

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.037[a]	.001	-.003	26.785

a. Predictors: (Constant), C_bitsum, K_bitsum

b. Dependent Variable: PT_bitsum

This means that there remain other factors which effect the Bitsum calculation of the plaintext for deciphering. To check these unknown constants, coefficient table for AES- 256 is prepared.

These coefficients determine that 256 bit encryption of AES is not significantly affected by the calculations of the Bitsums of ciphertext, plaintext, and key. This also depend on the other factors tabulated in Table 30.

Table 30: Coefficients Table for AES- 256

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	49.968	13.384		3.733	.000
	K_bitsum	.012	.014	.037	.831	.406
	C_bitsum	-.027	.207	-.006	-.132	.895

a. Dependent Variable: PT_bitsum

5.4 Results and Discussions

There are various properties of the Block ciphers which contribute to their strength. These properties were also analyzed from the specific tests done in SPSS. The results from the analysis done in Chapter 4 and 5 are summarized in the following comparison table.

Table 31: Comparison of the Algorithms

	Blowfish	FEAL	TEA/XTEA	AES 128 bits	AES 192 bits	AES 256 bits
Confusion property holds	yes	yes	No	yes	yes	Yes
Diffusion property holds	yes	yes	No	yes	yes	Yes
Constant Factor Coefficient	33. 944	22.497	15.104	49.101	38.760	49.968
Effect of Bitsum of ciphertext alone on Bitsum of plaintext	No	No	Yes	No	No	No

The above analysis and the comparison (shown in Table 31) bring out the following important points.

TEA does not exhibits either diffusion or confusion property which are the prerequisites of a strong cryptographic algorithm.

Apart from the key, ciphertext, and plaintext, an efficient algorithm also depends upon some other factors such as number of rounds, substitution and transposition techniques, keysize and mode of operation. All these terms, considered to be constant here, are focused primarily for plaintext, ciphertext, and key. This study found that Blowfish and FEAL depend more on those constant factors [188] rather than the TEA. So it can be deduced that TEA is mainly dependent on the ciphertext and plaintext. Therefore TEA algorithm in comparison to the other algorithms is less secure.

The above comparison also shows that the Bitsums of ciphertexts alone in TEA algorithm have an effect on the Bitsums of the plaintext which is not present in Blowfish and FEAL.

In the comparison of the different key versions of AES, Table 31 shows that among all the versions of AES, the 256 bit key AES mostly depends on other factors (constant

factors) rather than on ciphertext or plaintext. The results previously shown also depicts that in the algorithm of 256 bit key AES, neither ciphertext Bitsums nor the plaintext Bitsums alone cannot interpret the Bitsums of plaintexts, whereas key Bitsums in the version of 128 bit key AES and 192 bit key AES are alone significant for the prediction of the plaintext Bitsums.

Chapter 6: Analysing the Applicability of Bitsum Algorithm on LSB Steganography Technique

The greatest value of a picture is when it forces us to notice what we never expected to see.

-John Tukey, American Mathematician

This chapter is mainly focused on the study done to analyze the effect of Bitsum Algorithm on the LSB steganographic technique. The first section of the chapter explains about the background of the LSB steganography. The second section concentrates on experimentation done to analyze the applicability of Bitsum Algorithm on LSB Steganography. This experimentation was done to check the correlation between Bitsum of the original image and stego- image, the correlation between Bitsum of the secret message and the value of R(Correlation Coefficient), and the correlation between type of image and the correlation coefficient.

6.1 Background

Steganography, a class of methods that deals with privacy of the data, is the process of hiding data into a covering medium making detection during process of communication almost impossible/very difficult. This technique involves hiding a text in other data, especially in image files, taking care to retain the quality as well as the size of the image. The simplest method in Steganography is the substitution of the Least Significant Bit (LSB) in an image that acts as a vehicle for the message or text. By using up to 4 least significant bits in each pixel, the hiding capacity can be increased to an extent that makes detection quite hard.

A number of approaches on this aspect of data hiding have been identified. Some of the processes have been reviewed in the paper [188] . A mathematical analysis of the LSB Steganography is undertaken in [189]. In paper [190], a Spatial Domain technique has been shown in which difference between consecutive pixels and the mean of median

values is determined to embed payload in 3bits of LSB and one bit of MSB in a chaotic manner.

Paper [191] makes use of a modified LSB method by combining cryptography and steganography. This approach used in this paper provides security at three levels: At the first level, the text file is compressed and zipped followed by encryption at the second level using the proposed algorithm; and at the final level, to ensure the protection of the hidden text against Stego-attacks a secret key is used. When this message is embedded into the image, the change in image resolution is negligible. Above all personal password helps to conceal and secure the image. The confidential data is thus secured against any damage by an unauthorized intruder.

The statistical analysis of the LSB method has been explained in paper [192]. The problem exist when a bit sequence is encoded to match the statistics of the covering random bit-sequence. The solution to this problem lies in concealing confidential information in LSBs of JPEG coefficients. In this method, the chi-square statistic of JPEG coefficients or their distribution is followed in which two-bit codes are used to encode the message bits. The results show that the solution is very effective on JPEG images depicting natural scenes.

In the paper [193] presents a novel steganography technique that combines two methods i.e. Discrete Cosine Transform (DCT) and LSB. This combination, with minimal modification to the cover image (at most k-bits per block), ensures optimization of the capacity and invisibility of the secret image which uses DCT to transform to frequency domain. To construct the optimum quantization for embedding the DCT coefficients in k-bits, an algorithm is executed. This facilitates the hiding of k-bits in the LSBs of the cover image.

An approach based on arithmetic progression with LSB has been found in the paper [194]. The algorithm of encoding a message is given as: Find the LSBs of each grey pixel in the cover image. This goes to each byte. Should the LSB be not the bit of the message position, flip it, else do nothing. Apply a progression scheme on height and width for getting the position. The results obtained through the proposed approach are better as compared to the classical LSB. The LSB based techniques are not that robust

against various attacks, but still these provide an easy way for embedding large amount of data, with high PSNR and perceptual quality. In addition, the proposed approach is not dependent on the different file formats. This approach is limited to grey scale images.

The application of LSB in different file formats, such as BMP, PNG and GIF, is shown in [195]. Variable LSBs were used for different image types and their effects were compared. The comparison shows that BMP uses loseless method to use LSB. But for that it needs a larger cover image to hide the message in. GIF, on the other hand, is efficient to hide a larger message when a proper cover image is selected.

An RGB based LSB steganography work has been executed in the paper [196]. This improved LSB technique is for colour images. In this method, embedding of the information is done in three planes of RGB image. In doing so, the quality of the image is enhanced as well as high embedding capacity is achieved. The proposed technique hides data 2-bits in 2 LSBs of Red component (the most significant Byte), 2-bits in 2 least significant bits of Green component and 4-bits in 4 LSBs of Blue component (the least significant Byte) of each selected pixel. This method, generated on the basis of sensitivity of human eye to different colour wavelengths, is a selective approach. It induces lower noise but ensures high security to transferring images.

Generally, the last bit of the carrier image bytes is modified to include the message bit. This does not yield to a high resistance capability of message concealing. To overcome this, the authors in the paper [197] have suggested a way to modify the last 4 bits in the LSB. They have implemented the technique on Bitmap and Wave file formats. LSB steganography, based on bit inversion, has been shown in the paper [198]. This technique improves the quality of stego-images in 24-bit colour image. The inversion is carried out on some of the pixels of the LSBs of the cover image on getting input of specific patterns of some bits related to the pixels. In doing so, lesser number of pixels are modified as compared with the standard LSB method.

Paper [199] uses a newer version of LSB steganography in which Extended Substitution Algorithm is used to encrypt data. The cipher text so obtained is concealed at two or

three LSB positions in the carrier image. This algorithm has been developed to encompass almost all types of symbols and alphabets. As the encrypted text in this method is hidden at variable positions in the LSBs, this becomes a stronger approach. The visible characteristics of the carrier image, after concealment, does not betray tampering. This method is able to retain the image quality similar to two LSB scheme.

Another recent approach [200] is use of polynomials in LSB steganography. In this approach, the original image as cover image and the text file that need to be embedded into original image are inputted. To generate the stream of bits, binary conversion is done by considering the conversion of ASCII value of the character into binary format. For holding the total number of bits of message, counter variable is taken. In case of cover image, bytes representing the pixels are taken in single array to generate byte stream. Message bits, taken sequentially, are then placed in LSB bit of image byte. The polynomial equation, given in the key, controls the index number of the image byte where replacement of LSB is to be done. The Stego-image could then be sent to the recipient through open systems environment, which in turn runs its programme to extract those randomly stored LSBs. This facilitates the covert communication of the secret message.

A combination of LSB steganography, LZ compression, and RSA algorithm is shown by the authors in [201]. They have shown that the embedding process of LSB steganography replaces the values of the LSB plane with messages, which alters the pixel values of the LSB plane of the stego-medium in comparison to those of the original medium.

V. Lokeswara Reddy et.al [202] have shown the application of a genetic algorithm based LSB steganography in JPEG images. This improved adaptive LSB steganography, can achieve high capacity while preserving the first order statistics. Also, the bits-order of the message is shuffled for minimizing visual degradation of the stego-image. The shuffling is based on chaos and the parameter for selection of the chaos is done by the genetic algorithm that finds the best mapping between the secret message and the cover image. Shuffling of bits order of the message improves the performance of steganography.

6.2 LSB Substitution Method and Methodology for Experimentation

Substitution of least significant bits of the pixel intensity values of the cover image with the secret data bits, is the most common technique for image steganography. For example, the use of an image with 8-bit pixel depth, owe can write one bit (the LSB) of each pixel by XORing it with 1 bit of the secret data bit. This would yield in strong 3 bits of secret data per pixel. If the size of the cover image is 100 x 100 pixels , it is possible to embed a total 10,000 bits of secret data in the stego image generated from the cover image.

As far as the quality of the stego image is concerned, there would almost be no perceptible difference between the colour quality of the cover image and stego image. In practical terms, 8 bits could represent 256 levels of intensity for a colour component. The maximum change in colour intensity will be 1/256 (i.e. 0.39%) per colour component and this can not be perceived by human eye even after keeping both the cover image and the stego image in front together.

6.2.1 Bitsum effect on LSB Steganography

The motivation for this experimentation is taken from the results obtained in Chapter 3. Bitsum attack poses a threat on XOR cipher. Since LSB substitution method uses XOR to hide the data bits, so this method must be inspected against Bitsum attack.

The data has been generated by hiding the messages into the pictures with LSB substitution method. Images with the 8 – pixel depth had been used for this experimentation. The image of m x n can be represented in the following equation:

$$\text{Img} = p(i, j) \text{ where } 0 \le i \le m \text{ and } 0 \le j \le n$$

Secret data of length l can be represented as:

$$D = \{ d(i) \mid 0 \le i \le l,\ d(i) \in \{0,1\}\}$$

The stego image of length m x n can again be represented in the following equation:

$$\text{Stego\_Img} = p(i, j) \text{ where } 0 \le i \le m \text{ and } 0 \le j \le n$$

6.2.2 Methodology

Different types of images were chosen to perform the experiment by hiding the secret messages into these images by using LSB substitution method. The value of the pixels is converted into binary form and their Bitsums were taken. Bitsums of pixel values (binary form) were again taken and stored. Bitsum values were added row-wise. Then these Bitsum values of the rows of original image and stego – image were put to a correlation test. The method is explained below:

1. Select a Secret Message
2. Choose a Cover Image to embed the secret message
3. Generate a Stego Image by using LSB method
4. Convert the pixel values into the binary form and calculate their Bitsum
5. Take Bitsum of all the rows of the Original and Stego- Image
6. Calculate value of correlation coefficient these values.
7. Repeat the process for different images as well as for different messages and keep a track of the values of the correlation coefficient.

6.2.3 Cases to study

Case 1: To check the correlation between Bitsum of the original image and stego- image
Case 2: To check the correlation between Bitsum of secret message and value of R(Correlation Coefficient)
Case 3: To check the correlation between type of image and the correlation coefficient.
The following section explains the results of the experimentation done to analyze these cases.

6.3 Results of Experiments

6.3.1 Results of the experimentation for case 1:

Different types of images were chosen to conduct this study. To explain the conduct of this experiment, the image in Figure 46(i.e. Penguins) is taken. This figure contains both the original as well as stego image.

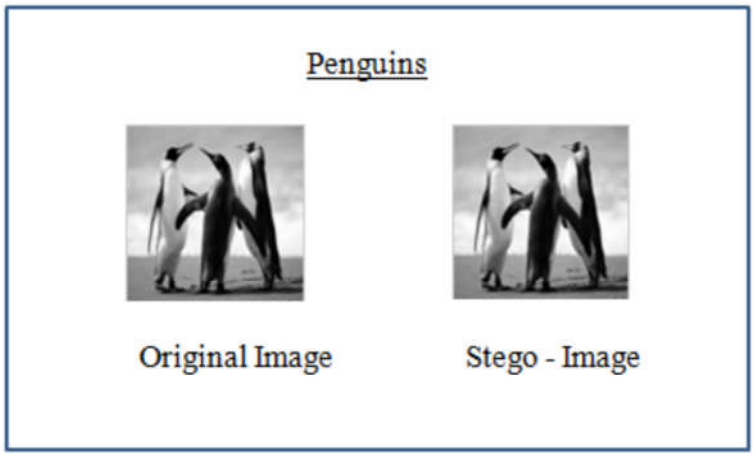

Figure 46: Data hidden in Penguins - image

Binary values of original and stego image are shown in Figure 47.

orgbits

orgbits <100x100 cell>

	1	2	3	4	5	6	7	8	9	10
1	10100010	10100011	10100100	10100100	10100101	10100111	10101000	10101000	10100111	10100111
2	10100011	10100011	10100010	10100100	10100110	10101000	10100111	10100111	10100110	10100110
3	10100010	10100001	10100001	10100100	10100110	10100111	10100111	10100111	10100110	10100111
4	10100010	10100000	10100010	10100101	10100111	10100111	10100111	10100111	10100111	10101000
5	10100000	10100010	10100011	10100110	10100111	10100111	10100111	10100111	10100111	10101000
6	10100010	10100011	10100011	10100101	10100110	10100111	10100111	10100111	10100111	10101000
7	10100010	10100010	10100011	10100100	10100110	10100110	10100111	10100111	10100111	10101001
8	10100010	10100011	10100011	10100100	10100101	10100110	10100111	10100111	10101000	10101010

finbits

finbits <100x100 cell>

	1	2	3	4	5	6	7	8	9	10
1	10100011	10100010	10100101	10100101	10100100	10100110	10101001	10101001	10100110	10100110
2	10100010	10100010	10100011	10100101	10100111	10101001	10100110	10100110	10100111	10100111
3	10100011	10100000	10100000	10100101	10100111	10100110	10100110	10100110	10100111	10100110
4	10100011	10100001	10100011	10100100	10100110	10100110	10100110	10100110	10100110	10101001
5	10100001	10100011	10100010	10100111	10100110	10100110	10100110	10100110	10100110	10101001
6	10100011	10100010	10100010	10100100	10100111	10100110	10100110	10100110	10100110	10101001
7	10100011	10100011	10100010	10100101	10100111	10100111	10100110	10100110	10100110	10101000
8	10100011	10100010	10100010	10100101	10100100	10100111	10100110	10100110	10101001	10101011

Figure 47: Binary Values of the Pixels of Original Image(orgbits) and Stego - Image(finbits)

Bitsum values of the original and stego images are shown in Figure 47. The binary values of all the 8 – bit pixels is shown in the cells. The number of 1's in each cell would be counted and will be written as Bitsum of the pixel.

Figure 48 shows the Bitsum value for the original image as well as for the stego image. To generate the Bitsum of each row, all the values in each row will be added.

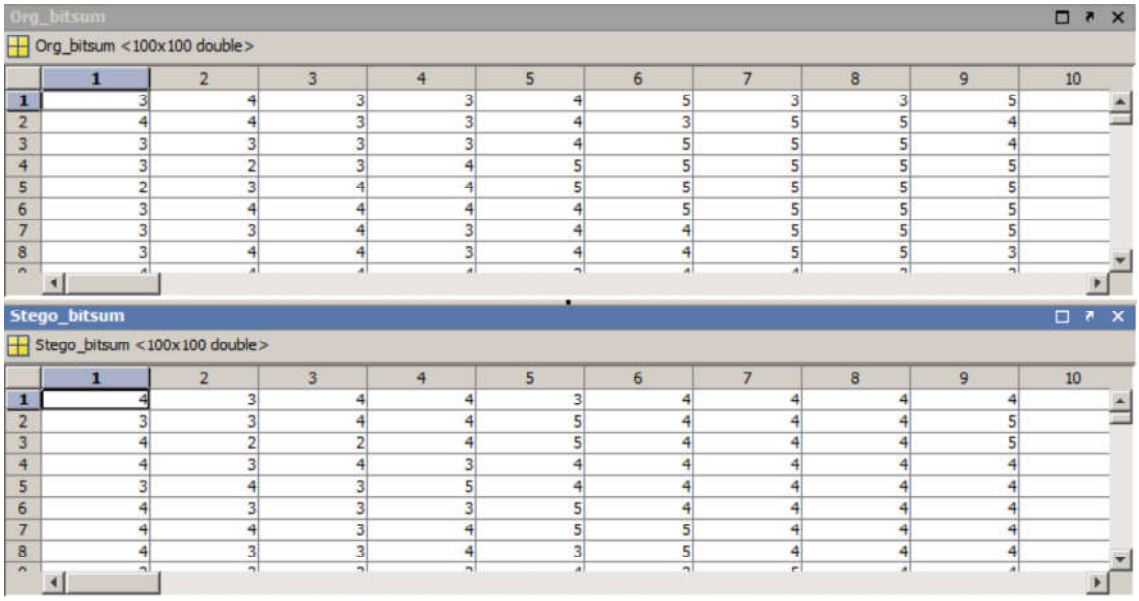

Figure 48: Bitsum values of the Pixels of Original Image (Org_Bitsum) and Stego – Image (Stego_Bitsum)

After adding these Bitsum values row – wise, correlation test was conducted.

The value of R i.e. correlation coefficient is 0.9979. This means that a strong positive correlation exists between the Bitsum value of the of original image and the stego - image. The correlation coefficients for different images for a specific message with Bitsum 72 is tabulated below:

Table 32: Correlation coefficients of different images

Image Name	Value of R
Chrysanthemum	0.9784
Penguins	0.9868
Desert	0.9868
Lighthouse	0.9784
Sunil-Gavaskar	0.9824
Red	0.9236
Green	0.7907

It is visible from Table 32 that correlation is there since all the values are showing positive value of R.

6.3.2 Results of the experimentation for case 2:

The correlation between Bitsums of the original image and the stego image for a particular message is persistent. The purpose of this study is to check correlation between the Bitsum of different data (secret messages) and the value of R. The messages with different Bitsum were taken to analyze this. Images of size 30 X 20 were taken i.e. these images would represent 600 pixels of 8 – bits each. Now the data would be hidden in some of these pixels. The number of pixels need to hide the data would directly depend on the number of bits in the secret message since we are using the single bit LSB steganographic technique. The values in Table 33 are the result of experiment done on Penguins image. Messages with different Bitsum were taken to analyze its effect on the Value of R.

Table 33: values of R for Penguins

Bitsum of the secret message	Value of R(Correlation Coefficient)
16	0.9949
60	0.9893
72	0.9868
300	0.9323
400	0.8942
600	0.8314

We can analyze from the values in the table that if we increase the Bitsum of the secret message, the value of R decreases. The same experiment was conducted on the other images listed in Table 32. It was tried that variable types of the images should be chosen and tested e.g. flowers, animals, human face, desert, buildings, and the single coloured plane images.

Eight images were analyzed on six messages with different Bitsum values. The resulting values for the correlation coefficients for these images are summarised in Table 34. These values strengthen the result obtained from Table 33.

Table 34: Value of R for Different Secret Messages with Different Images

Bitsum of the secret message	Value of R(Correlation Coefficient) for						
	Chrysanthemum	Penguins	Desert	Lighthouse	Sunil-Gavaskar	Red	Green
16	0.9962	0.9949	0.9949	0.9962	0.9964	0.9853	0.9698
60	0.9774	0.9893	0.9893	0.9774	0.9836	0.9483	0.8191
72	0.9784	0.9868	0.9868	0.9784	0.9824	0.9236	0.7907
300	0.9555	0.9323	0.9323	0.9555	0.9506	0.6105	0.6659
400	0.9513	0.8942	0.8942	0.9513	0.9433	0.4653	0.6056
600	0.8655	0.8314	0.8314	0.8655	0.8349	0.1809	0.156

6.3.3 Results of the experimentation for case 3:

The intention for this study was to check the value of the correlation coefficient for the different types of images. We took different types of images for this analysis e.g. flowers, buildings, human face, planes etc. The image and the table for the values for the correlation coefficient is given below.

The value of the correlation coefficients for different messages is tabulated in Table 34. Again it can be seen from the table values that when we increase the Bitsum of the secret message, value of R decreases. Similarly, this experiment was conducted on other images also. These images are shown below:

Figure 49: Desert

Figure 50: Lighthouse

Figure 51: Sunil Gavaskar

Figure 52: Chrysanthemum

Figure 53: Red Colour

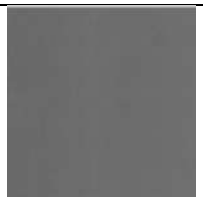

Figure 54: Green Colour

The variation in the values of the correlation coefficient R can be seen from Table 34.

Following bar chart would explain this variation.

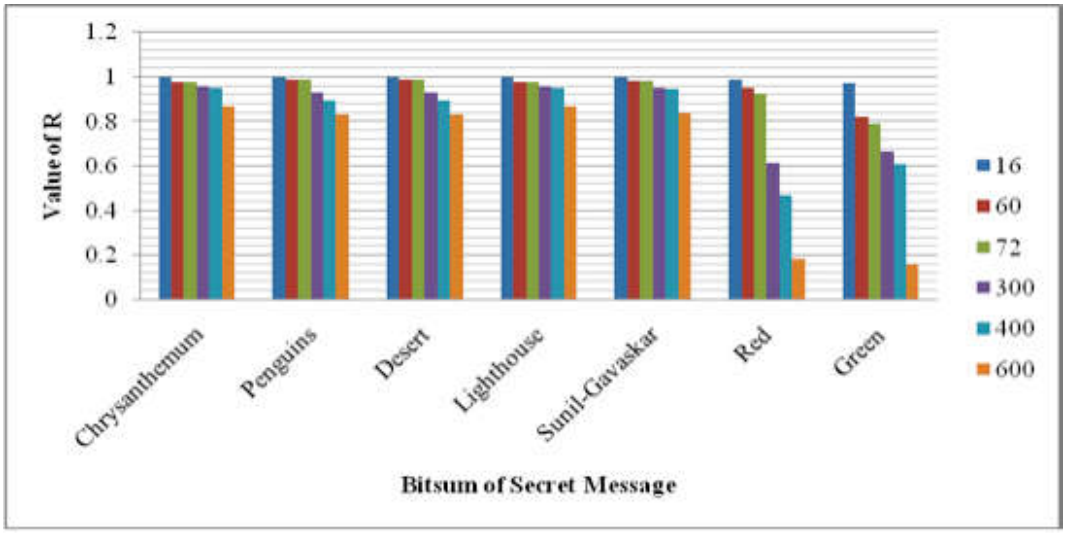

Figure 55: Bar Graph for the Values of R for Different images with Different Secret Messages

It can be easily analyzed from Figure 55 that the variation in the values of R is quite minimal in case of Flowers, buildings and human face. But when it comes on to the simple images having single colour, shows more variation in the value of R with respect to the Bitsum of the secret message.

6.4 Results and Discussions

This chapter was dedicated to LSB steganography and the impact of Bitsum attack on this technique. In particular three cases were under analysis. The observations for these three cases are :

a) During the experimentation, it was found that there is correlation between Bitsum of the original image and stego – image.

b) The correlation between the Bitsum of secret message and the value of correlation coefficient is also found. The results showed that whenever Bitsum of the secret message is increased, the value of R decreases.

c) Third observation was regarding the correlation coefficient and the types of images. The results has shown that the simple/plain images show more variation on changing the Bitsum of the secret message, whereas the variation is quite less in the case of complicated images.

Next chapter is based on conclusion which includes the discussion on results, and future work related to this thesis.

Chapter 7: Conclusion and Future Work

The true function of philosophy is to educate us in the principles of reasoning and not to put an end to further reasoning by the introduction of fixed conclusions.

-George Henry Lewes

7.1 Conclusion

In this thesis we have concentrated mainly on the block ciphers. An algorithm was designed and developed to check the strength of the block ciphers and was named "Bitsum Algorithm". Some block ciphers were chosen to conduct the analysis. This selection was based on the results obtained from the implementation of XOR cipher. TEA, XTEA, FEAL, Blowfish and AES were chosen since these ciphers also use XOR as their basic operational function. The chosen ciphers were implemented for the data generation. Generated data was analyzed to find the correlation and to check the robustness of these ciphers. The idea behind this experimentation was to find out the correlation between Bitsum of Ciphertext with Bitsum of Key or the Plaintext.

LSB Steganography is another area where XOR is used to hide the secret bits. It became important to test this technique against Bitsum algorithm. Experiments were conducted to check the correlation between the Bitsum of the original image and stego-image. Experimentation was done to evaluate the effect of Bitsum of secret message on the value of correlation coefficient. The variation in the value of Correlation coefficient was checked with respect the type of images.

We can conclude our experiments with the following findings.

7.1.1 Applicability of Bitsum Algorithm in caparison to various cryptanalytic algorithms on Chosen Block Ciphers

In security world, symmetric ciphers have been ruling since so many years. So we decided to implement this algorithm on some famous and strong symmetric ciphers to check its applicability. We decided to implement the algorithm on some of the popular algorithms such as XOR, TEA, XTEA, FEAL, Blowfish and AES. XOR cipher was the

first to be implemented and analyzed against Bitsum algorithm. Taking motivation from DES, security of reduced key Tiny Encryption Algorithm has been analyzed. TEA and XTEA, FEAL, BLOWFISH and AES are also analyzed through Bitsum algorithm.

The attacks which are implemented on these algorithms are studied in this section. In Table 35 we have summarized this analysis.

Table 35: Cryptanalytic attacks on the ciphers under study

Cryptanalytic Algorithms	TEA	XTEA	FEAL	BLOWFISH	AES
Exhaustive search	Yes	Yes	Yes	Yes	Yes
Zero correlation [203]	Yes	Yes	No	No	No
Related Key [204]	Yes	Yes	No	No	Yes
Slide Attack [205]	No	No	No	Yes	No
Reflection Attack [206]	No	No	No	Yes	No
Bit Sum [187]	Yes	Yes	No	No	No

Exhaustive search can be done on all the ciphers. Zero correlation attack is applicable on TEA and XTEA. Related key attack is pertinent on TEA, XTEA and AES. Slide attack is applicable on Blowfish. Reflection attack is applicable on Blowfish. Bitsum algorithm is able to cryptanalyze TEA and XTEA.

It can be observed from this table that for comparing the performance of Bitsum algorithm, four algorithms/attacks can be considered. These four techniques are Exhaustive search, Zero correlation, related key and Bitsum algorithm. The comparison of these algorithms is done on the basis of their complexity.

The choice of the algorithms in Table 36 for the comparison is due to he reason that all the three algorithms are used for the cryptanalysis of TEA and XTEA. Hence the comparison table is drawn for them only. Their performance on the basis of their complexities is compared in Table 36. From the comparison table we can conclude that performance of the Bitsum algorithm is better than the other algorithms. Although complexity of related key attack is lesser, but it comes with a condition of having 2^{23} chosen plaintexts.

Table 36: Comparison of complexities of various algorithms

Cryptanalytic Algorithms	**TEA**	**XTEA**	**Reduced Tiny Encryption Algorithm** [183]
Exhaustive search	2^{128}	2^{128}	2^{64}
Zero Correlation	$2^{119.64}$	$2^{120.71}$	-
Related Key attack	2^{32}(with a condition of having 2^{23} chosen plaintexts)	$2^{104.33}$	-
Bitsum	2^{64}	2^{64}	2^{32}

7.1.2 Performance of TEA, XTEA, FEAL, BLOWFISH and AES(128, 192 & 256) based on Correlation Test

This section is focusing on the comparative analysis of implementation of Bitsum Algorithm on selective block ciphers. These ciphers are TEA [207], XTEA [27], FEAL [208], BLOWFISH [29] and AES(128, 192 & 256) [209]. We have used SPSS to do this analysis.

Table 37: Analysis of Confusion and Diffusion Property of Selective Ciphers

Algorithm	**Significance Value of Bitsum of Plaintext – Ciphertext relation**	**Significance Value of Bitsum of Ciphertext – Key relation**
TEA/XTEA	.000	.000
FEAL	.701	.059
BLOWFISH	.956	.181
AES - 128	.471	.739
AES – 192	.229	.446
AES - 256	.929	.239

Table 37 is showing the significance values of the correlation test performed on the values of Bitsum of Plaintext, Bitsum of Ciphertext and Bitsum of Key. The second

column of the table gives the significance value of relation between Bitsum of Plaintext and Bitsum of the Ciphertext which indicates the diffusion property of the algorithms.
Significance values of the relation between Bitsum of Ciphertext and Bitsum of key are listed in the third column of the table. This column indicates the confusion property of the ciphers.
We have used the .05 level of significance. Where ever the significance value of the test(for Bitsum of Ciphertext and Bitsum of Key) is greater .05, this means that there is no correlation between the Bitsums of the keys and the cipher texts i.e. FEAL, BLOWFISH, AES -128, AES -192 and AES – 256 holds the confusion property.
In the similar way, where ever the significance value of the test (for Bitsum of Plaintext and Bitsum of Ciphertext) is greater .05, this means that there is no correlation between the Bitsums of the keys and the cipher texts i.e. FEAL, BLOWFISH, AES -128, AES -192 and AES – 256 holds the diffusion property.
The noticeable values are for TEA and XTEA, where the significance values are .000 in both the cases. That means these ciphers doesn't hold confusion or diffusion property. As a result, Bitsum algorithm poses a threat on TEA and XTEA.

7.1.3 Effect of Bitsum Algorithm on LSB steganography

One chapter of this thesis is devoted to the study of LSB steganography technique. The intention was to check the correlation between the Bitsum of Original Image, Stego – Image and Bitsum of the secret message. Even the variation of the correlation coefficient with respect to the Bitsum of the secret message was also evaluated. The variation in the value of correlation coefficient because of the types of the images was also checked.
The conclusion of this experimentation was quite encouraging. The value of the correlation coefficient was found quite high for some messages. But when we increase the Bitsum of the secret message, value of the correlation coefficient decreases.
The normal/complicated images shows a limited range in the values of R, even if the Bitsum of the secret messages varies a lot. But in case of plain images(Only one colour), the range of value of R is quite considerable.

7.2 Future Work

The thesis presented the analysis of Bitsum algorithm on chosen block ciphers. In our future work we wish to have the attack implementation on the stream ciphers as well.

In our literature review we have seen a number of algorithms, for which no cryptanalysis attack has been reported yet. We shall try to impose our introduced attack on those algorithms to check their strength.

We wish to generate hardware implementation procedure that will accept the ciphertext and can be directly converted to plaintext and along with the effects of Bitsums.

LSB steganography was analyzed only for one bit LSB. We would extend this experimentation to 4 least significant bits. The values of the correlation coefficients would be compared with the results of one bit LSB.

RGB based LSB steganography is another technique used for coloured images. This technique can be tested for one bit LSB, 2 – bit LSB or even 4 bit LSB steganography.

Analysis could also be done for Hybrid Steganographic techniques both in the spatial and spectral domains like Discrete Cosine Transformation and LSB, Discrete Wavelet Transformation and LSB, Discrete Curvelet Transformation and LSB steganography.

Bibliography

[1] H. V. Tilborg, *Fundamentals of Cryptology,* 2000.

[2] B. Schneier, Applied Cryptography, New York: John Wiley & Sons Inc., 1996.

[3] S. V. Kartalopoulos, "A primer on cryptography in communications," *IEEE Commun. Mag.,* vol. 44, no. 4, p. 146–151, April 2006.

[4] A. Arora, Priyanka and S. Pal, "A Survey of Cryptanalytic Attacks on Lightweight Block Ciphers," *International Journal of Computer Science and Information Technology & Security (IJCSITS),* vol. 2, no. 2, pp. 472-481, April 2012.

[5] G. C. Kessler, November 2006. [Online]. Available: http://www.garykessler.net/library/crypto.html.

[6] W. Diffie, "The First Ten Years of Public-Key Cryptography," *Proceedings of the IEEE,* vol. 76, no. 5, pp. 560-577, May 1988.

[7] J. S. Grah, *Hash functions in cryptography,* June 2008 .

[8] C. Woodbury, April 2007. [Online]. Available: http://www.mcpressonline.com/security/ibm-i-os400-i5os/your-guide-to-a-successful-encryption-project.html.

[9] E. Schaefer, "An introduction to Cryptography and Cryptanalysis," Santa Clara University, 1999.

[10] B. Schneier, "Self-study course in block cipher cryptanalysis," *CRYPTOLOGIA,* vol. 24, no. 1, pp. 18-34, 2000.

[11] A. Menezes, P. V. Oorschot and S. Vanstone, Handbook of Applied Cryptography, CRC Press Series on Discrete Mathematics and its Applications ed., CRC Press, 1997.

[12] X. Lai, "On the design and security of block ciphers," 1992.

[13] T. Yamanouchi. [Online]. Available: http://http.developer.nvidia.com/GPUGems3/gpugems3_ch36.html.

[14] T. Roeder. [Online]. Available:

http://www.cs.cornell.edu/courses/cs5430/2010sp/TL03.symmetric.html.

[15] P. Kaushal, R. Sobti and G. Geetha, "Random Key Chaining (RKC): AES Mode of Operation," *International Journal of Applied Information Systems,* vol. 1, no. 5, pp. 39-45, February 2012.

[16] C. Shannon, "Communication Theory of Secrecy Systems," *Bell System Technical Journal,* vol. 28, pp. 656-715, October 1949.

[17] C. Shannon, "Prediction and Entropy of printed English," *Bell System Technical Journal,* vol. 30, pp. 50-64, January 1951.

[18] M. Blaze, W. Diffie, R. Rivest, B. Schneier and T. Shimomura, "Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security," 1996.

[19] [Online]. Available: http://www.wikiwand.com/simple/Feistel_cipher.

[20] A. Sorkin, "Lucifer, A Cryptographic Algorithm," *Cryptologia,* vol. 8, no. 1, pp. 22-41, January 1984.

[21] F. P. 46-3, "Data Encryption Standard," 1999.

[22] D. Rudolf , "DEVELOPMENT AND ANALYSIS OF BLOCK CIPHERS AND THE DES SYSTEM," 2000. [Online]. Available: http://homepage.usask.ca/~dtr467/400/figure2-des_block.gif. [Accessed 9 January 2016].

[23] D. Wheeler and R. Needham, "TEA, a Tiny Encryption Algorithm," in *Proc. FSE*, 1994.

[24] S. Shepherd, "The Tiny Encryption Algorithm," in *Cryptologia*, 2007.

[25] R. Needham and D. Wheeler, "Two Feistel rounds (one cycle) of TEA," 1994. [Online]. Available: https://en.wikipedia.org/wiki/Tiny_Encryption_Algorithm. [Accessed 9 January 2016].

[26] D. Wheeler and R. Needham, "TEA extensions," 1997.

[27] R. Needham and D. Wheeler , "XTEA," [Online]. Available: https://en.wikipedia.org/wiki/XTEA. [Accessed 9 January 2016].

[28] S. Shimizu and Miyaguchi, "Fast Data Encipherment Algorithm FEAL," in *Advances in Cryptology - Eurocrypt '87*, 1987.

[29] B. Schneier, "Description of a New Variable- Length Key, 64-bit Block Cipher (Blowfish)," in *Springer-Verlag, Fast Software Encryption: Second International Workshop*, Leuven, Belgium, 1994.

[30] January 2015. [Online]. Available: http://www.iis.ee.ethz.ch/~kgf/acacia/fig/aes.png.

[31] D. W. Davies, "Some Regular Properties of the DES," in *Advances in Cryptology: A Report on CRYPTO 81*, 1981.

[32] M. Matsui, " Linear Cryptanalysis Method for DES Cipher," in *Advances in Cryptology -- EUROCRYPT '93*, 1993.

[33] M. Matsui, "The First Experimental Cryptanalysis of the Data Encryption," in *Advances in Cryptology -- CRYPTO '94, Springer-Verlag*, 1994.

[34] S. Langford and M. Hellman, "Differential–linear cryptanalysis," in *Advances in Cryptology - Crypto'94, Springer Verlag*, 1994.

[35] E. Biham, "New types of cryptanalytic attacks using related keys," *Journal of Cryptology,* vol. 7, no. 4, pp. 229-246, Fall 1994.

[36] L. R. Knudsen, "Truncated and higher order differentials," in *Fast Software Encryption*, B. Preneel, Ed., Springer Berlin Heidelberg, 1995, pp. 196-211.

[37] L. Knudsen, "Partial and higher order differentials and its application to the DES," 1995.

[38] M. Robshaw and K. L. , "Non-linear approximations in linear cryptanalysis," in *Advances in Cryptology — EUROCRYPT '96*, U. Maurer, Ed., Springer Berlin Heidelberg, 1996, pp. 224-236.

[39] J. Daemen, L. Knudsen and V. Rijman, "The block cipher Square," in *Fast Software Encryption*, E. Biham, Ed., Springer Berlin Heidelberg, 1997, pp. 149-165.

[40] D. Wagner, J. Kelsey and B. Schneier, "Related-Key Cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA," in *DES, RC2, and TEA, Proceedings of the 1997 International Conference on Information and Communications Security*, Springer-Verlag, 1997, pp. 233-246.

[41] E. Biham, A. Biryukov and A. Shamir, "Cryptanalysis of Skipjack Reduced to 31 Rounds using Impossible Differentials," in *Advances in Cryptology — EUROCRYPT '99*, J. Stern, Ed., Springer Berlin Heidelberg, 1999, pp. 12-23.

[42] H. Gilbert and M. Minier, "A collision attack on 7 rounds of Rijndael," in *The third Advanced Encryption Standard Candidate Conference*, 2000.

[43] E. Biham, O. Dunkelman and N. Keller, "The Rectangle Attack — Rectangling the Serpent," in *Advances in Cryptology — EUROCRYPT 2001*, P. B, Ed., Springer Berlin Heidelberg, 2001, pp. 340-357.

[44] N. Courtois and J. Pieprzyk, "Cryptanalysis of Block Ciphers with Overdefined Systems of Equations," in *Advances in Cryptology — ASIACRYPT 2002*, Y. Zheng, Ed., Springer Berlin Heidelberg, 2002, pp. 267-287.

[45] A. Shamir, J. Patarin, N. Courtois and A. Klimov, "Efficient Algorithms for solving Overdefined Systems of Multivariate Polynomial Equations," in *Eurocrypt'2000, LNCS 1807, Springer*, 2000.

[46] E. Filiol, "Plaintext-dependant Repetition Codes Cryptanalysis of Block Ciphers - The AES Case," 2003.

[47] D. Wagner, "Towards a Unifying View of Block Cipher Cryptanalysis," in *Fast Software Encryption*, M. Roy. B, Ed., Springer Berlin Heidelberg, 2004, pp. 16-33.

[48] P. Junod, *Statistical cryptanalysis of block ciphers,* 2005.

[49] H. Handschuh and D. Naccache, "SHACAL, preproceedings of NESSIE first workshop," 2000.

[50] J. Lu, J. Kim, N. Keller and O. Dunkelman, "Differential and rectangle attacks on reduced-round SHACAL-1," in *Progress in Cryptology - INDOCRYPT '06, Springer-Verlag*, 2006.

[51] O. Özen, K. Varıcı, C. Tezcan and C. Kocair, "Lightweight Block Ciphers Revisited: Cryptanalysis of Reduced Round PRESENT and HIGHT," in *Information Security and Privacy*, C. a. G. N. J. Boyd, Ed., Springer Berlin Heidelberg, 2009, pp. 19-107.

[52] B. Su, W. Wu and W. Zhang, "Differential cryptanalysis of SMS4 block cipher," 2010.

[53] J. Chen, M. Wang and B. Preneel, "Impossible differential cryptanalysis of the lightweight block ciphers TEA, XTEA and HIGHT," 2011.

[54] K. Jia, L. Li, C. Rechberger, J. Chen and X. Wang, "Improved Cryptanalysis of the Block Cipher KASUMI," in *Selected Areas in Cryptography*, L. a. W. H.

Knudsen, Ed., Springer Berlin Heidelberg, 2013, pp. 222-233.

[55] S. Ahmadi, Z. Ahmadian, J. Mohajeri and M. Aref, "Low-Data Complexity Biclique Cryptanalysis of Block Ciphers With Application to Piccolo and HIGHT," *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY,* vol. 9, no. 10, pp. 1641-1652, October 2014.

[56] J. Smith, "The design of Lucifer: a cryptographic device for data communications," N.Y., USA, 1971.

[57] E. Biham and A. Shamir, "Differential Cryptanalysis of Snefru, Khafre, REDOCII, LOKI and Lucifer," 1991.

[58] E. Biham and I. Ben-Aroya, "Differential cryptanalysis of Lucifer," *Journal of Cryptology,* vol. 9, no. 1, pp. 21-34, 1996.

[59] P. Rogaway and J. Kilan, "How to protect DES against exhaustive key search," *Journal of Cryptology,* vol. 14, no. 1, pp. 17-35, 2001.

[60] B. S. a. D. W. J. Kelsey, "Related-key cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA," in *First International Conference on Information and Communication Security ICICS'97*, London , 1997.

[61] M. Matsui and A. Yamagishi, "A New Method for Known Plaintext Attack of FEAL Cipher," in *EUROCRYPT 1992*, 1992.

[62] H. Gilbert and G. Chasse, "A Statistical Attack of the FEAL-8 Cryptosystem," in *10th Annual International Cryptology Conference on Advances in Cryptology*, 1990.

[63] R. Rivest, "A description of the RC2(r) encryption algorithm," 1998.

[64] R. Merkle, "Fast software encryption functions," in *Advances in Cryptology - Crypto'90*, A. M. a. S. Vanstone, Ed., Santa Barbara, California, Springer-Verlag, 1990, pp. 476-501.

[65] E. Biham, A. Biryukov and A. Shamir, "Miss in the Middle Attacks on IDEA and Khufu," in *FSE 1999*, L. R.Knudsen, Ed., Heidelberg, Springer, 1999, p. 124–138.

[66] S. Miyaguchi, "The FEAL cipher family," in *Advances in Cryptology - Crypto'90*, A. M. a. S. Vanstone, Ed., California, Springer-Verlag, 1990, pp. 627-638.

[67] H. Gilbert and A. Tardy-Corfdir, "A Known Plaintext Attack of FEAL-4 and FEAL-6," in *11th Annual International Cryptology Conference on Advances in Cryptology*, Santa Barbara, California, 1991.

[68] L. Brown, M. Kwan, J. Pieprzyk and J. Seberry, "Improving resistance to differential cryptanalysis and the redesign of LOKI," in *Advances in Cryptology - ASIACRYPT'91*, R. R. a. M. H. Imai, Ed., Fujiyoshida, Springer-Verlag, 1991, pp. 36-50.

[69] L. Knudsen, "Cryptanalysis of LOKI," in *Advances in Cryptology—ASIACRYPT' 91*, 1993.

[70] M. Wood and T. Cusick, "The RedocII cryptosystem," in *Advances in Cryptology - Crypto'90*, A. M. a. S. Vanstone, Ed., California, Springer-Verlag, 1990, pp. 545-563.

[71] X. Lai, J. Massey and S. Murphy, "Markov ciphers and differential cryptanalysis," in *Advances in Cryptology - Eurocrypt'91*, D. Davies, Ed., Brighton, Springer-Verlag, 1991, pp. 17-38.

[72] D. Khovratovich, G. Leurent and C. Rechberger, "Narrow-Bicliques: Cryptanalysis of Full IDEA," in *Advances in Cryptology – EUROCRYPT 2012*, D. P. a. T. Johansson, Ed., Heidelberg, Springer, 2012, pp. 392-410.

[73] V. Rijmen and J. Deamen, "The Block Cipher Rijndael," in *Smart Card Research and Applications*, B. S. Jean-Jacques Quisquater, Ed., Springer-Verlag, 2000, pp. 277-284.

[74] T. Gonzalez, " A Reflection attack on blowfish," *J Latex Files 6,* 2007.

[75] J. Massey, "SAFER-K: a byte-oriented block-ciphering algorithm," in *Fast Software Encryption*, R. Anderson, Ed., Cambridge, Springer-Verlag, 1994, pp. 1-17.

[76] J. Kelsey, B. Schneier and D. Wagner, "Key-schedule cryptanalysis of IDEA, G-DES, GOST, SAFER, and triple-DES," in *Advances in Cryptology—CRYPTO '96*, N. Koblitz, Ed., Springer-Verlag, 1996, pp. 237-251.

[77] W. Wlfovics and A. Di Porto, "VINO: a block cipher including variable permutations," in *Fast Software Encryption*, R. Anderson, Ed., Cambridge, Springer-Verlag, 1994, pp. 205-210.

[78] "GOST - Gosudarstvennyi Standard 28147-89," *Cryptographic protection for data processing systems," Government Committee of the USSR for Standards,*

1989.

[79] E. Fleischmann, M. Gorski, J. Huehne and S. Lucks, "Key Recovery Attack on full GOST Block Cipher with Negligible Time and Memory," in *Western European Workshop on Research in Cryptology (WEWoRC)*, 2009.

[80] B. Schneier and M. Blaze, "The MacGuffin cipher algorithm," in *ast Software Encryption: Second International Workshop*, B. Preneel, Ed., Leuven, Springer-Verlag, 1995, pp. 97-110.

[81] B. Preneel and V. Rijmen, "Cryptanalysis of MacGuffin," in *FSE 1994*, B. Preneel, Ed., Heidelberg, Springer, 1995, p. 353–358.

[82] R. Rivest, "The RC5 encryption algorithm," in *Fast Software Encryption: Second International Workshop*, B. Preneel, Ed., Leuven, Springer-Verlag, 1995, pp. 86-96.

[83] Y. Yin and B. Kaliski, "On Differential and Linear Cryptanalysis of the RC5 Encryption Algorithm," in *Advances in Cryptology — CRYPTO'95*, 1995.

[84] M. Matsui, "New block encryption algorithm MISTY," in *Fast Software Encryption: 4th International Workshop, FSE'97*, E. Biham, Ed., Haifa, Springer-Verlag, 1997, pp. 53-67.

[85] U. Kühn, "Cryptanalysis of Reduced-Round MISTY," in *Advances in Cryptology — EUROCRYPT 2001*, 2001.

[86] G. Alvarez, F. Montoya, A. Peinado and D. De La Guia, "Akelarre: a new block cipher algorithm," in *SAC'96, Workshop Record*, Queen's University, Kingston, Ontario, Canada, 1996.

[87] B. Schneier and N. Ferguson, "Cryptanalysis of Akelarre," in *SAC'97 Fourth Annual Workshop on Selected Areas in Cryptography*, Carleton University, 1997.

[88] E. Biham and R. Anderson, "Two practical and provably secure block ciphers: BEAR and LION," in *Fast Software Encryption*, D. Gollmann, Ed., Cambridge, Springer-Verlag, 1996, pp. 99-111.

[89] L. R. Knudsen, "On the security of Bear & Lion & ladder - DES," in *Fast Software Encryption Workshop*, Haifa, Israel, 1997.

[90] C. Adams, "Constructing symmetric ciphers using the CAST design procedure," *Designs, Codes and Cryptography,* vol. 12, no. 3, pp. 283-316, 1997.

[91] M. Wang, X. Wang, K. Chow and L. Hui, "New Differential Cryptanalytic Results for Reduced-Round CAST-128," *IEICE TRANSACTIONS on Fundamentals of Electronics,* Vols. E93-A, no. 12, pp. 2744-2754, 10 2010.

[92] S. Moriai, T. Shimoyama and T. Kaneko, "Higher order differential attack of a CAST cipher," *FSE 1998,* vol. 1372, p. 17–31, 1998.

[93] V. Rijmen, J. Daemen, B. Preneel, A. Bossalaers and E. Win, "The cipher Shark," in *Fast Software Encryption*, D. Gollman, Ed., Cambridge, Springer-Verlag, 1996, pp. 99-111.

[94] M. Kwan, "The design of the ICE encryption algorithm," in *Fast Software Encryption: 4th International Workshop,FSE'97*, E. Biham, Ed., Haifa, Springer-Verlag, 1997, pp. 69-82.

[95] L. Knudsen, V. Rijmen and B. Rompay, "Differential cryptanalysis of the ICE encryption algorithm," in *6th International Workshop Fast Software Encryption - FSE'98*, 1998.

[96] H. Mala, "Biclique-based cryptanalysis of the block cipher SQUARE," *Information Security, IET,* vol. 8, no. 3, pp. 207-212, May 2014.

[97] D. M'Raihi, D. Naccache, J. Stern and S. Vaudenay, "XMX: A firmware-oriented block cipher based on modular multiplications," in *Fast Software Encryption*, E. Biham, Ed., Haifa, Springer-Verlag, 1997, pp. 166- 171.

[98] N. Borisov, M. Chew, R. Johnson and D. Wagner, "Multiplicative differentials," in *Fast Software Encryption 2002*, Berlin, Springer, 2002, pp. 17 -33.

[99] J. Deamen and V. Rijmen, *The Design of Rijndael: AES - The Advanced Encryption Standard,* 2002.

[100] A. Bogdanov, D. Khovratovich and C. Rechberger, "Biclique cryptanalysis of the full AES," *ASIACRYPT 2011,* vol. 7073, no. Springer, Heidelberg (2011), pp. 344-371, 2011.

[101] H. Nover, "Algebraic Cryptanalysis of AES: An Overview," 2005.

[102] F. Abed, C. Foller, E. List, S. Lucks and J. Wenzel, "A Framework for Automated Independent-Biclique Cryptanalysis," in *Fast Software Encryption - 20th International Workshop, FSE 2013*, S. Moriai, Ed., Springer, 2013, pp. 561-581.

[103] H. Heys, C. Adams, S. Tavares and M. Wiener, "CAST256: a submission for the Advanced Encryption Standard," in *First AES Candidate Conference (AES1)*,

Ventura, California, USA, 1998.

[104] A. Pestunov, "Differential cryptanalysis of 24-round cast-256," in *IEEE Region 8 International Conference on Computational Technologies in Electrical and Electronics Engineering, SIBIRCON 2008*, Novosibirsk , 2008.

[105] S. Vaudenay and J. Stern, "CS-Cipher," in *Fast Software Encryption*, S. Vaudenay, Ed., Paris, Springer-Verlag, 1998, pp. 189-205.

[106] S. Vaudenay, "On the security of CS-cipher," in *Fast Software Encryption*, L. Knudsen, Ed., Springer Berlin Heidelberg, 1999, pp. 260-274.

[107] C. Lim, "Crypton: a new 128-bit block cipher," in *First AES Candidate Conference (AES1)*, Ventura, California, USA, 1998.

[108] J. Cheon, M. Kim, K. Kim, J. Lee and S. Kang, "Improved impossible differential cryptanalysis of rijndael and crypton," in *Information Security and Cryptology — ICISC 2001*, K. Kim, Ed., Springer Berlin Heidelberg, 2002, p. 39–49.

[109] L. Knudsen, "DEAL: a 128-bit block cipher," in *First AES Candidate Conference (AES1)*, Ventura, California, USA, 1998.

[110] B. Schneier and J. Kelsey, "Key-Schedule Cryptanalysis of DEAL," in *Selected Areas in Cryptography*, C. A. H. Heys, Ed., Springer Berlin Heidelberg, 1999, pp. 118-134.

[111] H. Gilbert, M. Girault, P. Hoogvorst, T. Pornin, G. Poupard, J. Stern and S. Vaudenay, "Decorrelated fast cipher: an AES candidate," in *First AES Candidate Conference (AES1)*, California, USA, 1998.

[112] L. Knudsen and V. Rijmen, "On the Decorrelated Fast Cipher (DFC) and Its Theory," in *Fast Software Encryption*, L. Knudsen, Ed., Springer Berlin Heidelberg, 2001, pp. 81-94.

[113] M. Kanda, S. Moriai, K. Aoki, H. Ueda, M. Ohkubo, Y. Takashima, K. Ohta and T. Matsumoto, "E2 - a candidate cipher for AES," in *First AES Candidate Conference (AES1)*, Ventura, California, USA, 1998.

[114] Y. Wei, X. Yang and C. Li, "Impossible differential cryptanalysis on cipher E2," *Concurrency and Computation: Practice and Experience,* vol. 26, no. 8, p. 1477–1489, June 2014.

[115] D. Georgoudis, D. Leroux and B. Chaves, "The "FROG" encryption algorithm,"

in *First AES Candidate Conference (AES1)*, Ventura, California, USA, 1998.

[116] D. Wagner, N. Ferguson and B. Schneier, "Cryptanalysis of FROG," in *Second AES Candiate Conference, The National Institute of Standards and Technlogy*, Gaithersburg, 1999.

[117] R. Schroeppel and H. Orman, "Overview of the hasty pudding cipher," in *First AES Candidate Conference (AES1)*, Ventura, California, USA, 1998.

[118] C. D'Halluin, G. Bijnens, B. Preneel and V. Rijman, "Equivalent Keys of HPC," in *Springer Berlin Heidelberg*, 1999.

[119] L. Brown, J. Pieprzyk and J. Seberry, "Introducing the new LOKI97 block cipher," in *First AES Candidate Conference (AES1)*, Ventura, California,USA, 1998.

[120] W. Wu, B. Li, D. Feng and S. Qing, "Cryptanalysis of Some AES Candidate Algorithms," in *Information and Communication Security, Second International Conference, ICICS'99*, Sydney, 1999.

[121] M. Jacobson Jr and K. Huber, "The Magenta block cipher algorithm," in *First AES Candidate Conference (AES1)*, Ventura, California, USA, 1998.

[122] E. Biham, A. Biryukov, N. Ferguson, L. Knudsen, B. Schneier and A. Shamir, "Cryptanalysis of MAGENTA," in *Second AES Conference*, 1999.

[123] D. Coppersmith, C. Burwick, E. D'Avignon, R. Gennaro, S. Halevi, C. Jutla, S. Matyas, L. O'Connor, M. Peyravian, D. Safford and N. Zunic, "Mars - a candidate cipher for AES," in *First AES Candidate Conference (AES1)*, Ventura, California, USA, 1998.

[124] B. Schneier and J. Kelsey, "MARS Attacks! Preliminary Cryptanalysis of Reduced-Round MARS Variants," in *The Third AES Candidate Conference*, National Institute of Standards and Technology, Gaithersburg, 169-185.

[125] R. Rivest, M. Robshaw, R. Sidney and Y. Yin, "The RC6 block cipher," in *First AES Candidate Conference (AES1)*, Ventura, California, USA, 1998.

[126] J. Borst, B. Preneel and J. Vandewalle, "Linear Cryptanalysis of RC5 and RC6," in *6th International Workshop, FSE '99*, 1999.

[127] N. Ferguson, J. Kelsey, S. Lucks, B. Schneier, M. Stay, D. Wagner and D. Whiting, "Improved cryptanalysis of Rijndael," in *Fast Software Encryption – FSE'2000*, Springer-Verlag, 2000, pp. 213-230.

[128] J. Massey, G. Khachatrian and M. Kuregian, "Nomination of SAFER+ as candidate algorithm for the Advanced Encryption Standard (AES)," in *First AES Candidate Conference (AES1)*, California, USA, 1998.

[129] J. Nakahara Jr, B. Preneel and J. Vandewalle, "Family, Linear Cryptanalysis of Reduced-Round Versions of the SAFER Block Cipher," in *Fast Software Encryption*, J. H. J. v. L. a. B. S. G. Goos, Ed., Springer Berlin Heidelberg, 2000, pp. 244-261.

[130] E. Biham, R. Anderson and L. Knudsen, "Serpent: a new block cipher proposal," in *Fast Software Encryption*, S. Vaudenay, Ed., Paris, Springer-Verlag, 1998, pp. 222-238.

[131] N. I. o. S. a. Technology, "Skipjack and KEA algorithm specifications," 1998.

[132] E. Biham, A. Biryukov and A. Shamir, "Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials," in *Advances in Cryptology — EUROCRYPT '99*, J. Stern, Ed., Springer Berlin Heidelberg, 1999, pp. 12- 23.

[133] B. Schneier, J. Kelsey, D. Whiting, D. Wagner and C. Hall, "Twofish: A 128-bit block cipher," in *First AES Candidate Conference (AES1)*, California, USA, 1998.

[134] N. Ferguson, J. Kelsey, B. Schneier and D. Whiting, "A Twofish Retreat: Related-Key Attacks Against Reduced-Round Twofish," 2000.

[135] W. Diffie and M. E. Hellman, "Exhaustive cryptanalysis of the NBS Data Encryption Standard," *Computer,* vol. 10, no. 6, pp. 74-84,, 1977.

[136] A. Hemme, "A Differential Fault Attack Against Early Rounds of (Triple-)DES," in *Cryptographic Hardware and Embedded Systems - CHES 2004*, Springer Berlin Heidelberg, 2004, pp. 254-267.

[137] S. Vaudenay and H. Handschuh, "A universal encryption standard," in *Selected Areas in Cryptography*, H. H. a. C. Adams, Ed., Ontario, Springer-Verlag, 2000, pp. 1-12.

[138] V. Rijmen and P. Barreto, "The Khazad legacy-level block cipher," in *First Open NESSIE Workshop*, Leuven, Belgium, 2000.

[139] F. Muller, "A new attack against Khazad," in *Advances in Cryptology - ASIACRYPT'03*, C. Laih, Ed., Springer-Verlag, 2003, pp. 347 - 358.

[140] V. Rijmen and P. Barreto, "The Anubis block cipher," Leuven, Belgium, 2000.

[141] A. Biryukov, "Analysis of Involutional Ciphers: Khazad and Anubis," *Springer,* vol. 2887, no. FSE 2003, pp. 45-53, 2003.

[142] J. Kang, K. Jeong, J. Sung, S. Hong and K. Lee, "Collision Attacks on AES-192/256, Crypton-192/256, mCrypton-96/128 and Anubis," *Journal of Applied Mathematics,* vol. 2013, pp. 1-10, August 2013.

[143] K. Aoki, T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakajima and T. Tokita, "Camellia: a 128-bit block cipher suitable for multiple platforms - design and analysis," in *Selected Areas in Cryptography*, D. S. a. S. Tavares, Ed., Ontario, Springer-Verlag, 2001, pp. 39-56.

[144] W. Wu, W. Zhang and D. Feng, "Impossible differential cryptanalysis of reduced-round ARIA and Camellia," *J. Comput. Sci. Technol,* vol. 22, no. 3, pp. 449-456, May 2007.

[145] L. Granboulan, P. Nguyen, F. Noilhan and S. Vaudenay, "DFCv2," in *Selected Areas in Cryptography*, D. S. a. S. Tavares, Ed., Ontario, Springer-Verlag, 2001, pp. 57-71.

[146] J. Borst, "The block cipher: GrandCru," in *First Open NESSIE Workshop*, Leuven, Belgium, 2000.

[147] Toshiba Corporation, "Specification on a block cipher: Hierocrypt-L1," in *First Open NESSIE Workshop*, Leuven, Belgium, 2000.

[148] B. Taga, S. Moriai and K. Aoki, "Differential and Impossible Differential Related-Key Attacks on Hierocrypt-L1," *Information Security and Privacy,* vol. 8544, pp. 17-33, 2014.

[149] Toshiba Corporation, "Specification of Hierocrypt-3," in *First Open NESSIE Workshop*, Leuven, Belgium, 2000.

[150] J. H. Cheon, M. Kim and K. Kim, "Impossible Differential Cryptanalysisof Hierocrypt-3 Reduced to 3 Rounds," 2002.

[151] ETSI/SAGE, "Kasumi Specification,Part of the Specification of the 3GPP Confidentiality and Integrity Algorithms," 1999.

[152] E. Biham, O. Dunkelman and N. Keller, "A Related-Key Rectangle Attack on the Full KASUMI," in *ASIACRYPT 2005*, B. Roy, Ed., Heidelberg, Springer, 2005, pp. 443-461.

[153] A. Machado, "The Nimbus cipher: a proposal for NESSIE," in *First Open*

NESSIE Workshop, Leuven, Belgium, 2000.

[154] V. Furman, "Differential cryptanalysis of Nimbus," in *Fast Software Encryption*, M. Matsui, Ed., Springer Berlin Heidelberg, 2001, pp. 187-195.

[155] J. Daemen, M. Peeters, G. Assche and V. Rijmen, "The Noekeon block cipher," in *First Open NESSIE Workshop*, Leuven, Belgium, 2000.

[156] S. F. Abdul-Latip, M. R. Reyhanitabar, W. Susilo and J. Seberry, "On the Security of NOEKEON against Side Channel Cube Attacks," in *5th Information Security Practice and Experience Conference - ISPEC 2010*, 2010.

[157] A. Volchkov and A. Labedev, "NUSH," in *First Open NESSIE Workshop*, Leuven, Belgium, 2000.

[158] W. Feng and D. Wu, "Linear cryptanalysis of NUSH block cipher," *Science in China Series F: Information Sciences,* vol. 45, no. 1, pp. 59-67, February 2002.

[159] L. McBride, "Q: a proposal for NESSIE," in *First Open NESSIE Workshop*, Leuven, Belgium, 2000.

[160] J. Massey, G. Khachatrian and M. Kuregian, "Nomination of SAFER++ as candidate algorithm for the New European Schemes for Signatures, Integrity, and Encryption (NESSIE)," in *First Open NESSIE Workshop*, Leuven, Belgium, 2000.

[161] A. Biryukov, C. Cannière and G. Dellkrantz, "Cryptanalysis of SAFER++," in *Advances in Cryptology - CRYPTO 2003*, D. Boneh, Ed., Springer, 2003, pp. 195-211.

[162] T. Shimoyama, H. Yanami, K. Yokoyama, M. Takenaka, K. Itoh, J. Yajima, N. Torii and H. Tanaka, "Specification and supporting document of the block cipher SC2000," in *First Open NESSIEWorkshop*, Leuven, Belgium, 2000.

[163] L. Ji-Qiang, "Differential attack on five rounds of the SC2000 block cipher," *JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY,* vol. 26, no. 4, pp. 722-731, July 2011.

[164] H. Handschuh, H. Helena and D. Naccache, "SHACAL," in *First Open NESSIEWorkshop*, Leuven, Belgium, 2000.

[165] J. Lu, J. Kim, N. Keller and O. Dunkelman, "Differential and Rectangle Attacks on Reduced-Round SHACAL-1," in *Progress in Cryptology - INDOCRYPT 2006*, T. L. R. Barua, Ed., Springer, 2006, pp. 17-31.

[166] L. Knudsen, G. Leander, C. Paar, A. Poschmann, M. Robshaw, Y. Seurin, C. Vikkelsoe and A. Bogdanov, "PRESENT: An Ultra-Lightweight Block Cipher," in *Cryptographic Hardware and Embedded Systems - CHES 2007*, Springer Berlin Heidelberg, 2007, pp. 450-466.

[167] C. De Canniere, O. Dunkelman and M. Knežević, "KATAN and KTANTAN—a family of small and efficient hardware-oriented block ciphers," in *Cryptographic Hardware and Embedded Systems-CHES 2009*, 2009.

[168] A. Bogdanov and C. Rechberger, "A 3-subset meet-in-the-middle attack: cryptanalysis of the lightweight block cipher KTANTAN," in *Selected Areas in Cryptography*, Springer Berlin Heidelberg, 2011, pp. 229-240.

[169] Z. Gong, S. Nikova and Y. Law, "KLEIN: a new family of lightweight block ciphers," *RFID Security and Privacy,* pp. 1-18, 2012.

[170] Z. Ahmadian, M. Salmasizadeh and M. Aref, "Biclique Cryptanalysis of the Full-Round KLEIN Block Cipher," *IACR Cryptology ePrint Archive,* vol. 2013, p. 97, 2013.

[171] J. Guo, T. Peyrin, A. Poschmann and M. Robshaw, "The LED block cipher," in *Cryptographic Hardware and Embedded Systems–CHES 2011*, 2011.

[172] F. Mendel, V. Rijmen, D. Toz and K. Varıcı, "Differential analysis of the LED block cipher," in *Advances in Cryptology–ASIACRYPT 2012*, Springer Berlin Heidelberg, 2012, pp. 190-207.

[173] D. Hong, J. Lee, D. Kim, D. Kwon, K. Ryu and D. Lee, "LEA: A 128-bit block cipher for fast encryption on common processors," in *Information Security Applications*, Springer, 2014, pp. 3-27.

[174] Y. Kim and H. Yoon, "First Experimental Result of Power Analysis Attacks on a FPGA Implementation of LEA".

[175] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks and L. Wingers, "The SIMON and SPECK lightweight block ciphers," in *52nd Annual Design Automation Conference (DAC '15). ACM*, New York, NY, USA, 2015.

[176] A. Biryukov, A. Roy and V. Velichkov, "Differential Analysis of Block Ciphers SIMON and SPECK," in *Fast Software Encryption*, Springer Berlin Heidelberg, 2015, pp. 546-570.

[177] M. Gomulkiewicz and M. Kutylowski, "Hamming Weight Attacks on Cryptographic Hardware—Breaking Masking Defense," in *Computer Security—*

ESORICS 2002, Springer, 2002, pp. 90-103.

[178] W. Hnath, "Differential power analysis side-channel attacks in cryptography," 2010.

[179] P. E. Black, "big-O notation," 31 August 2012. [Online]. Available: http://www.nist.gov/dads/HTML/bigOnotation.html. [Accessed 4 February 2016].

[180] A. Bagga and G. Geetha, "Implications of Bitsum Attack on XOR," in *Proc. 2nd National Conference on Emerging Trends in Computer Application*, Chennai, 2012.

[181] E. Berlekamp, "Factoring Polynomials Over Finite Fields," *Bell System Technical Journal,* vol. 46, no. 8, pp. 1853 - 1859, October 1967.

[182] "Linear Feedback Shift Registers," [Online]. Available: http://www.newwaveinstruments.com/resources/articles/m_sequence_linear_feedback_shift_register_lfsr.htm. [Accessed 17 April 2016].

[183] Amandeep and G. Geetha, "On the Security of Reduced Key Tiny Encryption Algorithm," in *International Conference on Computing Sciences (ICCS), 2012* , Phagwara, 2012.

[184] J. Guttman and M. Nadel, "What Needs Securing," in *Proc. CSFW*, 1988.

[185] L. Gordon and M. Loeb, "The economics of information security investment," *ACM Transactions on Information and System Security (TISSEC),* vol. 5, no. 4, pp. 438-457, November 2002.

[186] Y. Chan, "Biostatics 104: correlation analysis," *Singapore Med J,* vol. 44, no. 12, pp. 616-619, December 2003.

[187] Amandeep and G. Geetha, "Implications of bitsum attack on tiny encryption algorithm and XTEA," *Journal of Computer Science,* vol. 10, no. 6, pp. 1077-1083, 2014.

[188] G. Prashanti and K. Sandhyarani, "A New Approach for Data Hiding with LSB Steganography," in *Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India CSI Volume 2*, 2015.

[189] R. Chandramouli and N. Memon, "Analysis of LSB based image steganography techniques," in *Image Processing, 2001. Proceedings. 2001 International Conference on*, 2001.

[190] N. Sathisha, G. Madhusudan, S. Bharathesh, K. Suresh Babu, K. Raja and K. Venugopal, "Chaos Based Spatial Domain Steganography using MSB," in *5th International Conference on Industrial and Information Systems*, 2010.

[191] R. Boopathy, M. Ramakrishnan and S. Victor, "Modified LSB Method Using New Cryptographic Algorithm for Steganography," in *Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012), December 28-30, 2012*, 2014.

[192] Z. Duric, D. Richards and Y. Kim, "Minimizing the statistical impact of LSB steganography," in *Image Analysis and Recognition*, Springer, 2005, pp. 1175-1183.

[193] B. Mohd, S. Abed, B. Al-Naami and S. Alouneh, "Image steganography optimization technique," in *Signal Processing and Information Technology*, Springer, 2011, pp. 205-209.

[194] S. Goel, S. Gupta and N. Kaushik, "Image Steganography-Least Significant Bit with Multiple Progressions," in *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014*, Springer, 2015, pp. 105-112.

[195] D. Neeta, K. Snehal and D. Jacobs, "Implementation of LSB steganography and its evaluation for various bits," in *1st International Conference on Digital Information Management, 2006*, 2006.

[196] A. Singh and H. Singh, "An improved LSB based image steganography technique for RGB images," in *International Conference on Electrical, Computer and Communication Technologies (ICECCT), 2015*, 2015.

[197] J. Jasril, I. Marzuki and F. Rahmat, "Modification four bits of uncompressed steganography using least significant bit (LSB) method," in *International Conference on Advanced Computer Science and Information Systems (ICACSIS), 2012*, 2012.

[198] M. Majeed and R. Sulaiman, "An improved LSB image steganography technique using bit-inverse in 24 bit colour image," *Journal of Theoretical and Applied Information Technology,* vol. 80, no. 2, pp. 342-348, 2015.

[199] R. Gutte, Y. Chincholkar and P. Lahane, "Steganography for two and three LSBs using extended Substitution algorithm," *ICTACT Journal on communication technology,* vol. 4, pp. 685-690, 2013.

[200] A. Sivasankar, T. Jayachandra Prasad and M. Giriprasad, "LSB based image

steganography using polynomials and covert communications in open systems environment for DRM," in *International Conference & Workshop on Emerging Trends in Technology (ICWET '11)*, 2011.

[201] V. Lokeswara Reddy, A. Subramanyam and P. Chenna Reddy, "Implementation of Least Significant Bit Steganography and Statistical Steganalysis," in *Second International Conference on Computational Science, Engineering and Information Technology*, 2012.

[202] L. Yu, Y. Zhao, R. Ni and T. Li, "Improved adaptive LSB steganography based on chaos and genetic algorithm," *EURASIP Journal on Advances in Signal Processing,* p. 32, 2010.

[203] A. Bogdanov and M. Wang, "Zero Correlation Linear Cryptanalysis with Reduced Data Complexity," *Lecture Notes in Computer Science ,* vol. 7549, pp. 29-48, 2012.

[204] J. Kelsey, D. Wagner and B. Schneier, "Related-key cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA," in *First International Conference on Information and Communication Security ICICS'97*, London, 1997.

[205] D. Wagner and A. Biryukov, "Slide attacks," in *International Workshop on Fast Software Encryption, Springer*, Heidelberg, 1999.

[206] O. Kara and C. Manap, "A New Class of Weak Keys for Blowfish," in *14th International Workshop, FSE 2007, Luxembourg, March 26-28*, Luxembourg, 2007.

[207] D. Needham and R. Wheeler, "TEA, a Tiny Encryption Algorithm," in *Proc. FSE*, 1994.

[208] A. Shimuzu and S. Miyaguchi, "Fast Data Encipherment Algorithm FEAL," in *Advances in Cryptology - Eurocrypt '87*, 1987.

[209] V. Rijmen and J. Deamon, *The Design of Rijndael: AES - The Advanced Encryption Standard,* 2002.

List of Publications

I. Amandeep and G. Geetha, “Bit Sum Attack”, The Security Journal, vol.35, pp.21-22, Fall 2011.

II. G. Geetha and Amandeep, “Implication of Bit Sum Attack on XOR”, in Proc. 2nd National Conference on Emerging Trends In Computer Application, pp. 47-50, Feb-2012.

III. Amandeep and G. Geetha, “ On the Security of Reduced Key Tiny Encryption Algorithm” in Proc. International Conference on Computing Sciences, Punjab, 2012, pp. 323-326.

IV. Amandeep and G. Geetha, “Implications of Bitsum attack on tiny encryption algorithm and XTEA”, Journal of Computer Science, Vol. 10, pp. 1077-1083 , 2014.

V. Amandeep and G. Geetha, “Analysis of Bitsum Attack on Block Ciphers”, Discrete Journal of Mathematical Sciences and Cryptography. [Accepted for Publication].

VI. Amandeep and G. Geetha, “On the Complexity of Algorithms affecting the security of TEA and XTEA”, Far East Journal of Electronics and Communication. [Accepted for Publication].

VII. Amandeep and G. Geetha, “Analysis the applicability of Bitsum algorithm on LSB steganography technique”, [submitted for publication].