INTERPOLATION ANALYSIS: NEW CRITERIA FOR CRYPTANALYTIC ATTACK TOOLSET

By

Umar Malik

00000325138

A thesis submitted to the faculty of Information Security Department, Military College of Signals, National University of Sciences and Technology,Islamabad, Pakistan, in partial fulfillment of the requirements for the degree of MS in Information Security

SEPTEMBER 2021

Declaration

I certify that this research work titled “Interpolation Analysis: New Criteria for Cryptanalytic attack Toolset” is my own work. No portion of this work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere. The material that has been used from other sources has been properly acknowledged / referred.

Signature of Student

Umar Malik

00000325138

ABSTRACT

The importance of data has been increased manifolds in the recent past. Therefore, its protection from the unauthorized persons also demands utmost attention especially when this data is in transit because during this phase it is more vulnerable to attacks from the adversary. The only way to protect this data is with the help of cryptographic techniques. Many different schemes, algorithms and ciphers help users according to their needs. Block ciphers are a kind of symmetric ciphers where a block of text is encrypted with the help of a cryptographic key rather than encrypting one bit at a time as in the case of stream ciphers. The only non-linear and the most vital component in these ciphers is the s-box which provides enhanced security to these ciphers. Therefore, the designing of the s-box of any block cipher holds utmost importance. Block ciphers have many benefits and applications across many different fields in the world. With its growing demand and its usage, these ciphers are also vulnerable to different kinds of attacks. Some of the important known attacks are Linear Cryptanalysis and Differential Cryptanalysis. However, a more powerful attack against these ciphers is known as interpolation attack. Such attacks are effective against ciphers whose s-boxes are represented with simple algebraic functions. Researchers have been applying such attacks on various prototype ciphers like PURE and variants of block ciphers like SHARK and were successful. This definitely raises the concerns in the researcher community that the real world ciphers like DES and AES are also under threat from such attacks and one day when there won't be any issues of computational and memory resources, these ciphers can also be broken in polynomial time. Therefore, in this thesis we have analyzed s-boxes critically and tried to find what role these s-boxes can play in making ciphers resistant against these attacks. Finally, a design criteria is given for a stronger s-box against interpolation attacks.

COPYRIGHT STATEMENT

DEDICATION

This thesis is dedicated to

MY TEACHERS, PARENTS AND FAMILY

for their love, endless support and encouragement

ACKNOWLEDGEMENTS

I am grateful to Allah Almighty for giving me strength to keep going on with this thesis, irrespective of many challenges and troubles. All praises for HIM and HIM alone.

I am very grateful to my Project Supervisor Asst Prof. Dr. Fawad Khan, Co-Supervisor Asst Prof Dr Shahzaib Tahir and GEC members who supervised the thesis / research in a very encouraging and helpful manner. They always guided me with their profound and valuable support that have helped me in achieving my research aims.

I would also like to extend my feelings of gratitude towards Brig Humayun Ajmal Bhatti, Lt Col Khurram Shahzad and Maj Wajahat Sultan for their continuous guidance, motivation and endless support.

Finally, I would like to express my appreciation to all the people who have provided valuable support to my study and whose names I couldn't bring to memory.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

In the modern era of digital world, the importance of the data has been increased manifolds. Therefore, its protection from the unauthorized persons also demands utmost attention especially when this data is in transit because during this phase it is more vulnerable to attacks from the adversary. The only way to protect your any kind of data is with the help of cryptographic techniques [1]. Numerous such techniques and algorithms exist in the present era to protect your data from the adversary. If we want to categorize ciphers into two main types then they would be symmetric and asymmetric ciphers. In symmetric ciphers, a single key is used for the encryption and decryption. The only problem here is to distribute that key safely among the communicating parties. And for that purpose, asymmetric ciphers are used where a concept of public and private keys exist which means that the encryption and decryption is done by with the help of different keys.

In this thesis, our main focus is on block cipher which is a kind of symmetric cipher where a plaintext is divided into different blocks and then those blocks are encrypted with the help of a cryptographic key instead of one bit at a time as in the case of stream ciphers. The only non-linear and the most vital component in the block ciphers is the S-box which provides security to these ciphers. However, the main issue with the conventional s-boxes is their statistic behavior which means that they are used as a lookup table of fixed size [2]. Therefore, the construction of the S-box of any block cipher holds utmost importance. Block ciphers have many benefits and applications across many different fields in the world. With its growing demand and usage, these ciphers are also vulnerable to different types of attacks. S-boxes have been exploited from time to time in different attacks. Substitution boxes are designed based on some Boolean functions. These functions must be selected after carefully evaluating their cryptographic properties to make the cipher resistant against all types of attacks. Much work has already been done in the field of substitution boxes and analyzing its cryptographic properties and much more required to make them strong against specific attacks. Our focus will be primarily on the interpolation attacks which is one kind of an algebraic attack and try to analyze S-box with respect to this attack. Some of the important known attacks are Linear Cryptanalysis [4] and Differential Cryptanalysis [5]. These attacks exploit the statistical properties of ciphers. After these attacks, a new attack was presented against these ciphers which proved to be more powerful which is known as interpolation attack.

Interpolation attacks were initially introduced by Jakobsen and Knudsen in 1997 [6]. These attacks are effective against block ciphers which uses simple algebraic functions as S-box. Any cipher in which ciphertext can be represented as a polynomial of plaintext is vulnerable to interpolation attack. The attackers' aim is to examine the algebraic structure of a block cipher and tries to express the ciphertext in the polynomial form using the langrange interpolation formula with the help of chosen plaintexts [6]. The purpose of the interpolation attack varies from the attacker to attacker. The recovery of round keys is usually the main objective of many cryptanalysts. However, the other two methods are global deduction and instance deduction. Cryptanalysts have been applying such attacks on various prototype ciphers like PURE and variants of block ciphers like SHARK and were successful [3]. This definitely raises the concern that the real world ciphers like DES and AES are also under threat from such attacks and one day when there won't be any issues of computational and memory resources, these ciphers can also be broken in polynomial time.

1.1 Problem Statement

Whether it is Pakistan Army or any other important organization / institution of the country, they must deal with the data on the daily basis. And most of that data is highly sensitive, meaning by that if that data is lost to the adversary it can leave a devasting effect on that organization and country in general. This data also needs to be communicated over different channels in the real time for decision making. So, their protection needs utmost importance. For the protection of this data, cryptographic algorithms / ciphers are being used.

Among them, block ciphers are being employed mostly and in every corner of the world due to their effectiveness for the protection of critical data. However, certain issues and concerns of their security keep on arising from time to time. Among many other attacks to these ciphers, algebraic attacks holds key importance due to their strong nature. Some of the block ciphers which provably secure against linear and differential cryptanalysis were found vulnerable to these types of attacks. Substitution box being the most vital component of block ciphers provides enhanced security to these ciphers. Interpolation attack is one kind of the algebraic attack which exploits plaintext / ciphertexts relations in the form of polynomial. What role does any S-box play in thwarting or assisting this kind of attack will be the primary focus of this work and likewise study the properties of S-box in detail with respect to interpolation attack.

1.2 Motivation

With the growing demand and need of block ciphers in every field of work, continuous evaluation of these ciphers is mandatory. S-box being the only non-linear component, provides security to these ciphers. We know that difference distribution table (DDT) and linear approximation table (LAT) are exploited in the differential and linear cryptanalysis respectively. Likewise, what design criteria should a particular s-box in block cipher should possess to make it secure against interpolation attack. At present, the real-world ciphers like DES and AES are secure against interpolation attack. However, the main motivation for selection of this topic is to make the ciphers secure by carefully evaluating the design of S-boxes with respect to interpolation attack. The exploration of S-Box and finding its relation to interpolation attack is the main motive behind this study, which may provide certain level of mitigation against this attack.

1.3 Research Objectives

The main objectives of this thesis are

- Detailed literature review and exploring the existing mathematical structures
- Mathematical design of s-box and evaluation methods
- Define a method based upon interpolation analysis for the s-box evaluation
- Pseudocode and implementation for different structures

1.4 Contribution

Interpolation analysis: New criteria for cryptanalytic attack toolset will contribute in following ways:

- Making people aware of the interpolation attack by providing a detailed literature review
- Paving a way for future work in order to make ciphers more secure against this attack
- Providing a detailed analysis of S-boxes including its design criteria to thwart interpolation attacks
- Will help designers to cater for interpolation attack while designing S-boxes/ ciphers by providing them certain guidelines

1.5 Thesis Outline

The research work has been organized and distributed in following chapters:

- Chapter 1: A brief introduction is given, problem statement is highlighted, followed by motivation behind the research and research objectives are enumerated. Furthermore, the contributions made through this research are highlighted.

- Chapter 2: An overview of the attack is presented followed by birdseye view of existing / recent research already carried out in the field of interpolation analysis of block ciphers using various techniques.

- Chapter 3: A brief introduction of the cryptographic preliminaries used in our research work including Boolean functions, Lagrange interpolation formula and substitution boxes along with its important cryptographic properties being consider while designing of s-box are discussed.

- Chapter 4: This chapter includes the analyzing of s-boxes of certain ciphers using the sage math tool. A methodology adopted to carry out our research is also given in this chapter. Finally, a design criteria of s-box in order to resist interpolation attack is presented.

- Chapter 5: This chapter summarizes our research work followed by some of the recommendations and future work objectives.

Chapter 2

EXISTING RESEARCH / OVERVIEW OF INTERPOLATION ATTACKS ON BLOCK CIPHERS

2.1 Introduction

In this chapter, overview of the interpolation attack is given first and then the various research work already carried out on interpolation analysis of block ciphers at different times since its inception are discussed. Different techniques used / applied for carrying of this attack are highlighted. Some of the details (including the complexities) regarding this attack on different block ciphers are also mentioned in the form of table.

2.2 Overview of the Interpolation Attack

Interpolation attacks are one of the kind of algebraic attacks. They were introduced in 1997 when some of the variants of block ciphers or prototype ciphers which had provable security against linear and differential attacks were attacked using interpolation technique. This attack analyzes the algebraic structure of substitution boxes or round function as a whole and tries to express ciphertext in the form of polynomial of plaintext. The chances of success of the attack increases if a block cipher uses simple algebraic function while designing S-boxes. The attacker aim is to determine the polynomial expression of low degree representing that cipher. Because that polynomial would be key dependent, the possibility of extracting some round keys is there by using some chosen plaintexts and applying some useful techniques. However, on the other hand one can say that any cipher which cannot be represented by the simple polynomial of low degree, that cipher would be very difficult to break. This is the reason behind that until now all the globally known block ciphers like DES and AES are secure by now. The complexity of the attack relies mainly on the degree or the number of unknown terms (coefficients) of the interpolated polynomial.

2.3 Types of Interpolation attacks

There are basically three different types of interpolation attack depending upon the attackers aim. For explaining these attacks, let us consider a cipher with R rounds and block size 2m. If we represent plaintext x as the concatenation of n subblocks (number of S-boxes), size of S-box as s and ciphertext with y, then we know *2m=s x n*, Therefore,

$x = (x_n, x_{n-1}, \dots. x_1) \in GF(2^s)^n, \qquad x_i \in GF(2^s)$

$y = (y_n, y_{n-1}, \dots. y_1) \in GF(2^s)^n, \qquad y_j \in GF(2^s)$

2.3.1 Global Deduction

This is a kind of attack in which attacker's aim is to find the algorithm of any block cipher which can encrypt any plaintext into valid ciphertext and vice versa for a given key but without knowing that key. In other words, an attacker is able to represent a cipher's algorithm in the form of a polynomial. For better understanding of this, let us assume that a secret key k is fixed, then we can represent ciphertext subblock $y_j \in GF(2^m)$ in the polynomial of plaintext subblocks $\{x_n, x_{n-1}, \dots, x_1\}$ as follow:

$y_j = f_{jk}(x_n, x_{n-1}, \ldots x_2, x_1) \in GF(2^s) [x_n, x_{n-1}, \ldots x_2, x_1];$ where $GF(2^s) [X]$ denotes the polynomial ring of $X = \{ x_n, x_{n-1}, \ldots x_1\}$ over $GF(2^s)$. In this attack, if the number of unknown coefficients in $f_{jk}(x_n, x_{n-1}, \ldots x_1)$ is N, then $f_{jk}(x_n, x_{n-1}, \ldots x_2, x_1)$ can easily be computed from N pairs of unique plaintext / ciphertext. If we represent the degree of $f_{jk}(x_n, x_{n-1}, \ldots x_2, x_1)$ as $deg_{xi}\, f_{jk}$ with respect to xi, then relationship between degree and number of coefficients can be given as follows:

$$N \leq \prod_{1 \leq i \leq n} (deg_{xi}\ \ f_{jk} + 1)$$

After constructing this polynomial by an attacker successfully, (s)he can encrypt any plaintext into a valid ciphertext. Similarly, in order to obtain a decryption algorithm an attacker needs to swap plaintext and ciphertext and construct x_i as a function of ciphertext. By doing this, (s)he will be able to do decryption of any ciphertext into a valid plaintext for any fixed key and this attack is known as global deduction where any plaintext/ciphertext can be converted into corresponding ciphertext/plaintext without having any knowledge about secret key.

2.3.2 Instance Deduction

This is a kind of attack in which a cryptanalyst discovers an algorithm which can only encrypt a small chunk of plaintexts into the corresponding ciphertexts without the knowledge of symmetric key. (S)he does this by fixing values of some of the plaintext subblocks e.g. $(0,0,\ldots x_2,0)$, then the ciphertext subblock $y_j \in GF(2^s)$ can be represented by a polynomial as mentioned:

$y_j = f_{jk} (x_2) \in GF(2^s)[x_2].$

It is clearly visible that $f_{jk}(x_2)$ is a polynomial in one variable i.e x_2. Therefore, less number of chosen p/c pairs will be used to construct this polynomial. If the degree of the $f_{jk}(x_2)$ is d then the number of coefficients is estimated to be N ≤ d+1. After computing a polynomial expression $f_{jk}(x_2)$ from N different p/c pairs an attacker can only encrypt a small chunk of plaintexts for which the algorithm was designed e.g. $x = (0,0,\ldots x_2,0)$ into their valid ciphertexts for a given secret key. Similarly, by swapping the plaintexts and ciphertexts, an attacker can construct x_i as a function of ciphertext by fixing some of the ciphertexts sub blocks, then (s)he will be able decrypt a subset of ciphertexts. This kind of attack is known as instance deduction.

2.3.3 Key Recovery

The purpose of most of the interpolation attacks is to recover secret key. This attack works by using different techniques in order to recover some round keys. Basically, the output from the reduced cipher (say after r-1 rounds) is expressed as a polynomial of plaintext and same can also be expressed as a polynomial of ciphertext with the guessed last round key k^r.

2.4 Generalized steps of interpolation attack to recover round keys

Several different methods / techniques have been used for this attack but the generalized steps which are pertinent to recover round key are described. In this type of attack, the attackers main objective is to find some round keys as illustrated in figure 2.1 and for doing this (s)he gets the intermediate ciphertext from the reduced cipher (y^{r-1}) and expresses it as a polynomial $p(x) \in GF(2^s)[x]$ of plaintext rather than representing the final output. For understanding, we take some block cipher B having r number of rounds,

Step 1: The first and the foremost step is to find the upper bound degree of the polynomial expression after *(r-1)* rounds from the reduced cipher, we can call the output from reduced cipher as y^{r-1} and the degree of the polynomial expression as d. This *d* is related to the number of unknown coefficients which would be at most *d+1*. These coefficients are key dependent which helps in determining the key bits.

Step 2: After finding the upper bound d, an attacker guesses the last round key $K^{(r)}$ as seen in figure 2.1 and computes the value of y^{r-1} (output from the reduced cipher) as $y^{r1}=f(y,K^{(r)})$. This will be done according to the guess and determine strategy.

Step 3: We know that this attack is chosen plaintext, so the cryptanalyst will choose ***d+1*** distinct p/c pairs in order to construct polynomial representation of output from the reduced cipher using well known Lagrange interpolation formula which implies, If we assume F is a field, then for given 2m elements as $x_1, \ldots, x_m, y_1, \ldots, y_m \in F$, and all x_i*s* are unique. Then Lagrange interpolation formula implies

$$f(x) = \sum_{i=1}^{m} yi \prod_{1 \le j \le m, j \ne i} \frac{x - x_j}{x_i - x_j}$$

This equation gives us a polynomial over F having degree not more than m -1 such that $f(x_i) = y_i$ for i = 1,….,m.

The resultant polynomial representing the block cipher will be constructed by solving the system of linear equations as: $Y^{r-1}= f(X) \in GF(2^s)[X];$ where X constitutes the set of plaintext pairs and Y represents corresponding ciphertexts.

Step 4: The guessed key K^r can be determined with an additional p/c pair as seen in the figure below. Actually, what happens is that one decrypts the last round and find the value of Y^{r-1} as $y^{r-1}=f(y,K^{(r)})$. And the same value is also evaluated by the above-mentioned polynomial of the corresponding plaintext.

Step 5: If the decrypted value from the last round and the evaluated value from the polynomial of plaintext matches, then that would be the valid last round key. The process is repeated for the other values of keys in case match does not occurs and this process continues until one finds the valid key K^r.

The basic interpolation attack procedure for key recovery can also be understood easily with the help of figure 2.1.

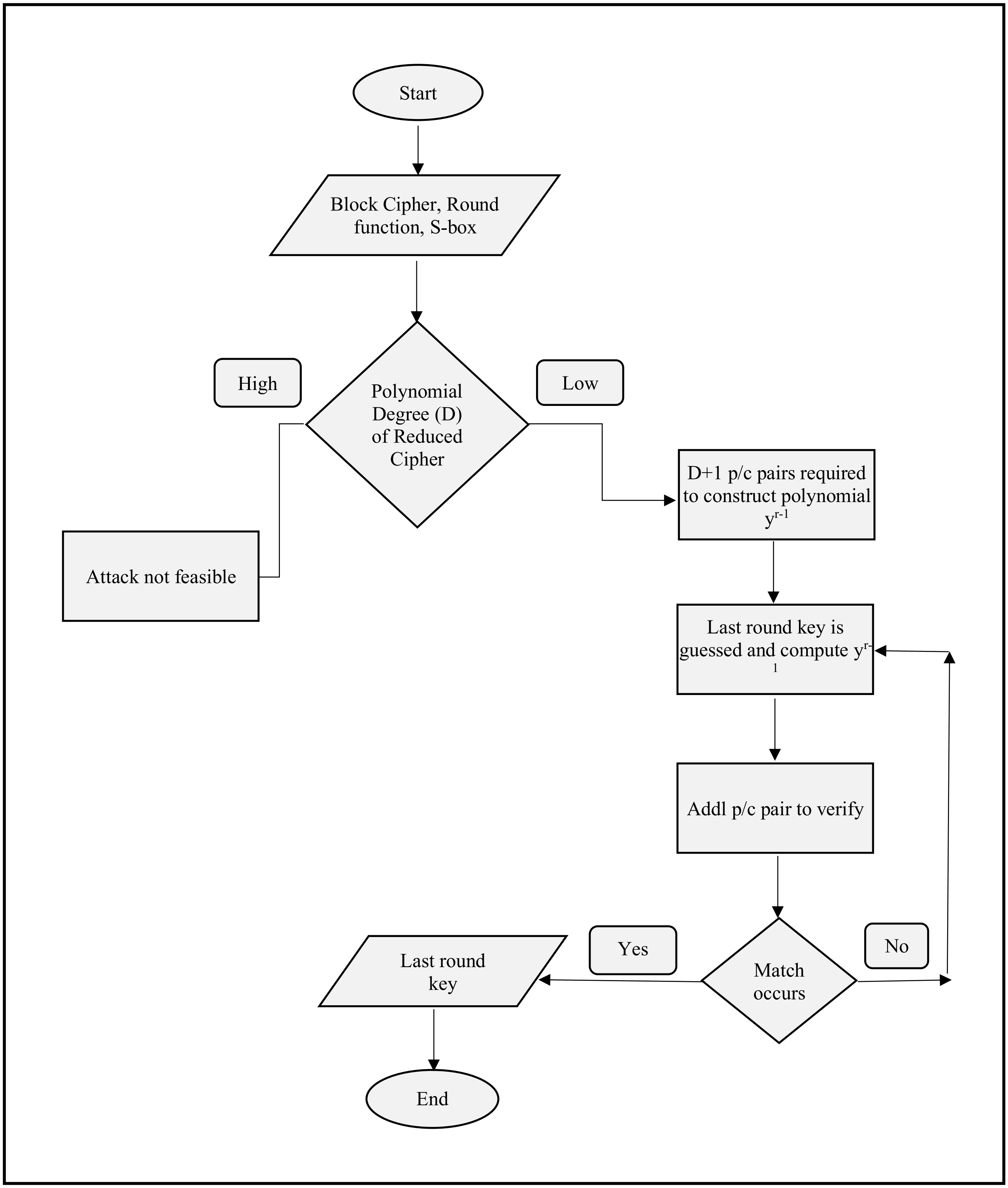

Start
Block Cipher, Round function, S-box
High
Polynomial Degree (D) of Reduced Cipher
Low
Attack not feasible
D+1 p/c pairs required to construct polynomial y^{r-1}
Last round key is guessed and compute y^{r-1}
Addl p/c pair to verify
Yes
No
Last round key
Match occurs
End

Figure 2.1: Basic Interpolation Attack Flow chart

2.5 Existing Research

Interpolation attack was initially presented by the authors Jakobsen and Knudsen in 1997 [6] and since then a lot of research work has been carried out in this domain by many different researchers. In [6], the authors basically presented that the ciphers which are considered to be secured and protected against the Linear and Differential cryptanalysis are actually vulnerable to higher order and interpolation attacks. They took the variant of SHARK cipher which was presented by *Rijmen et al*. [7] to cryptanalyze with the help of interpolation analysis. They have claimed that the ciphers constructed using this strategy can be broken in less than the claimed time. To prove their claim, they successfully cryptanalyze 5 rounds variant of SHARK. Furthermore, they also managed to cryptanalyze the new block cipher concept presented by the Kiefer [8] and a KN cipher by Knudsen and Nyberg [9] with the help of higher order differential attack. I refer [10, 11] for the understanding of higher order differentials.

Moriai et al. in [12] has applied the interpolation attack on the block cipher SNAKE. This cipher was presented by the Lee and Cha in 1997 [13]. This is a Fiestal cipher which has provable resistance against Linear and differential cyptanalysis as well as higher order differential cyptanalysis. Moriai took the actual version and not the variant of this cipher. He has contradicted with the Jakobsen and Knudsen findings regarding the number of plaintext / ciphertext (p/c) – pairs required for constituting the polynomial by saying that it is often overestimated by them specially when we talk about the multivariate polynomial or the rational expression. He has given solution to this problem by computing the rational expression having significantly less number of coefficients by choosing the plaintexts with the help of computer algebra system. By using computer algebra system, they also managed to reduce complexity of attack by finding the actual number of coefficients in less variables in the rational expression. They have mounted an interpolation attack by representing the cipher as rational expression and explained that if this cipher is represented by the polynomial, the attack is not possible after few rounds. They have demonstrated their claim by attacking both versions of SNAKE which are SNAKE(1) and SNAKE(2). They both are different with respect to their round functions. They were able to recover keys of all rounds for SNAKE(1) and for SNAKE(2) they only managed to recover round keys up to 11 rounds for the 64 bit block size and 8 bit S-box variant of the cipher. Furthermore, when they took block size of 128 bits and S-box as 16 bits, they recovered all round keys up to 15 rounds for SNAKE(1) and up to 16 rounds for SNAKE(2).

The authors in [14] have combined the partial sum technique and interpolation attack to reduce the attack complexity against the 6 round Rijndael-128 to 250 against the previous best results which is 272. Rijndael being one of the candidate of AES ciphers [15] holds much importance and its security evaluation has been carried out from time to time by many researchers. However, by now no known attack is heard against the full version of Rijndael algorithm. *Sun et al*. has presented an improved interpolation attack [16] in which he described that there is no requirement of storing the plaintexts and their resultant ciphertexts as in the case of original interpolation attack [6], thereby reducing the memory complexity to manifolds. Moreover, they have also demonstrated that key for the first round can be found by using only the plaintexts / ciphertexts pairs and solving the algebraic equations over finite field contrary to the original attack in which the guess and determine strategy was applied to determine the last round key. Therefore, again reducing the complexity of the attack drastically. They applied their attack on PURE cipher by determining the degree as well as the coefficients of special terms in the polynomial. The complexity of this attack mostly depends upon the degree or the number of unknown coefficients in the polynomial expression. The authors in [17] analyses the S-boxes (or round function as a whole) to study the input – output bits relationship of substitution box in order to see whether the resulting polynomial is having low degree or with less number of unknown coefficients. In the same article [17], the authors have also explained that the degree of the resultant polynomial is affected by the choice of the irreducible polynomial we use to construct the finite field. Therefore, affecting the complexity of the attack. Furthermore, they have also explained and presented the formula which relates lagrange interpolation and the Galois Field Fourier Transform. The significance of this relation in cryptography can be seen in [18] where the authors have modeled numerous block ciphers as Non Linear Feedback Shift Register. Jakobsen and Knudsen in [6] claimed that if the degree of the polynomial is n-1, then for deriving the last round key of a cipher, n+1 chosen plaintexts are required and exhaustive key search method was employed. However, *Kurosawa et al*. [19] described that for n+1 different plaintexts, not a single key but several equivalent keys are found. They also showed an upper bound for these last round keys by choosing these plaintexts. They have used Rabin's root finding algorithm for finding all these equivalent keys of the last round and termed their attack a root finding attack.

Interpolation attack has been mounted on several prototype ciphers like PURE and the modified versions of block ciphers like SHARK but it is hard to launch this attack on globally used ciphers like DES and AES. This difficulty is because of the fact that these ciphers cannot be represented by a low degree polynomial expression. Kazumaro Aoki in [3] introduced a linear sum attack which focuses on finding the effective basis for launching this attack. In linear sum attack, a cipher can be attacked the same way as the interpolation attack because $f_k(x)$ is represented by a sum of linearly independent polynomials. He termed this attack as a generalized form of interpolation attack because with the introduction of linear sum attack it is easier to study the security of ciphers against interpolation attacks. He has also given algorithm which tells us that whether a linear sum attack can be mounted on any block cipher or not and for those ciphers on whom this attack can be applied, his algorithm proficiently evaluates the security of a cipher against this attack. Furthermore, he also applied his algorithm on E2, CRYPTON and RIJNDAEL ciphers for security evaluation. In article [20], the authors have used the Moebius Transform method for carrying out an interpolation attack. Moebius transform is an algorithm which can efficiently convert the truth table of any Boolean function to its algebraic normal form (ANF). This approach actually helped them in curtailing the time complexity for getting a linear system of equations for specified intermediate state bits. They used their technique to apply attack on Elephant-Delerium and claimed that it was the first third-party cryptanalysis on this cipher. Elephant [21] belongs to a family of light weight ciphers employing the authentication schemes. Moreover, they also performed interpolation attack on the Kravatte [22] and Xoofff [23].

MiMC is a family of block ciphers having low multiplicative complexity which is designed to enhance the performance of applications like MPC, Zero-knowledge, SNARK and STARK etc. In article [24], the authors proposed low memory interpolation attack on the MiMC by minimizing the number of multiplications in large finite fields. They have demonstrated that the requirement of the memory is reduced substantially for interpolation attack. They were managed to break a round reduced MiMC but not the full round MiMC. LowMC are basically the family of block ciphers presented at Eurocrypt 2015 by Albrecht [25]. Their design is basically optimized for the cryptographic primitives like zero-knowledge proofs, fully homomorphic encryption (FHE) and multi-party computation (MPC) etc. The authors in [26] have launched an interpolation attack on LowMC-80 and LowMC-128 bits version and managed to reduce the attack complexity manifolds and refuted the security claims offered by the designers of these ciphers. The most valuable contribution by the authors is their variable transformation algorithm which efficiently reduces the number of variables and helps in mounting interpolation attacks. *Thomas Jakobsen et al*. in [27] has shown the possibility of breaking block ciphers operating on GF(q) where the ciphertext is probabilistically expressible as low degree polynomials over the plaintext quicker than key search strategy. This has open room for the new design criteria of block ciphers in order to thwart these attacks as only the complexity of Boolean round functions is not enough. He employed the Sudan's algorithm which is actually used for the decoding of Reed-Solomon codes. His work also shows that the properties seems to be good against differential and linear attacks does not provide enough security here. Moreover, the seemingly perfect Boolean non-linear functions should also be avoided if they are algebraically simple. They used their technique to break the KN cipher upto 10 rounds which was considered to be safe against linear and differential attack. Some of the details of interpolation attacks on different block ciphers are given in figure 2.3:

Cipher	**Number of rounds**	**Time complexity (XOR operations)**	**Data Complexity (bit/block)**	**Memory complexity (bits)**	**Purpose**	**Approach**
SNAKE(1) [12]	11	2^{47}	-	-	Key recovery	Global and instance deduction
SNAKE(2) [12]	11	2^{46}	-	-	Key recovery	Instance Deduction
Reduced-round PURE [16]	22	148 hours	3 x 2^{32} bits	neglectable	Round key	Solving algebraic equations
Rijndael-128 [14]	6	2^{50}	2^{32} bits	-	-	Interpolation attack and partial sum technique
Elephant – Delirium [20]	8/18	$2^{98.3}$	2^{70} blocks	2^{70}	Key recovery	Moebius Transform
Kravatte Achouffe [20]	4/6	$2^{106.2}$	$2^{78.3}$ blocks	2^{72}	Key recovery	Moebius Transform
Xoofff [20]	4/6	$2^{90.4}$	$2^{75.2}$ blocks	2^{69}	Key recovery	Moebius Transform
SHARK variant [28]	6	2^{225}	2^{75} bits	2^{150}	-	Meet-in-the-middle
LowMC-80 [26]	11	2^{57}	2^{39}	2^{39}	Key recovery	Deriving and solving linear system of equations
LowMC-128 [26]	12	2^{118}	2^{73}	2^{80}	Key recovery	Deriving and solving linear system of equations

Table 2.1: Comparison table of Interpolation Attacks on Block Ciphers

Chapter 3

CRYPTOGRAPHIC PRELIMINARIES FOR INTERPOLATION ANALYSIS ON S-BOXES

3.1 Introduction

This chapter gives a broad view of the cryptographic preliminaries regarding interpolation attacks on block ciphers. Boolean functions and their relation to the s-boxes and designing of s-boxes are discussed. Some important and relevant properties of s-box and important functions of sage math tool helpful for interpolation analysis of block ciphers are also highlighted.

3.2 Block Ciphers

Block ciphers are those ciphers which takes the input in the form of blocks of data with fixed length to transform them into the corresponding ciphertext. It is kind of a symmetric cipher which can be easily differentiated from a stream cipher, as a block cipher performs operations on a chuck of data contrary to a stream cipher which encrypts data bit wise at a time. As block ciphers operates on block of data instead of one bit at a time as seen from figure 3.1, therefore they are fast as compared to stream ciphers. The important example of block cipher is that of Advanced Encryption Standard (AES) which operates on 128 bits block of data. The major designing principles of block ciphers are:

- **Number of Rounds**. It is one of the important factors which determines the strength of any block cipher. Ciphers are designed according to the needs of different entities and resources available. Therefore, number of rounds are not fixed and vary in different ciphers. However, a cipher with more number of rounds provide more complexity break. Some ciphers are having with weak round function but their more rounds provide them strength against cryptanalysis.

- **Design of Round Function F**. The major strength that any block ciphers possesses is due to this round function in which a data is passes through different layers of substitution (confusion) and permutation (diffusion). The criterion that strengthens this function is its non-linearity, and this non-linearity is provided by the S-box being used. Therefore, the designing of S-box for any block cipher is of utmost importance as the whole security of the cipher depends on this. More this function F is non-linear, more it would be difficult to crack the cipher. This function must be so designed that it should have basic cryptographic properties like good avalanche effect and bit independence property criterion. (See section)
- **Key schedule algorithm**. This algorithm is the basis of any block cipher which is responsible for generating a unique key for every round from generates from the basic key bits. This algorithm should also confirm the bit independence criterion and strict avalanche effect.

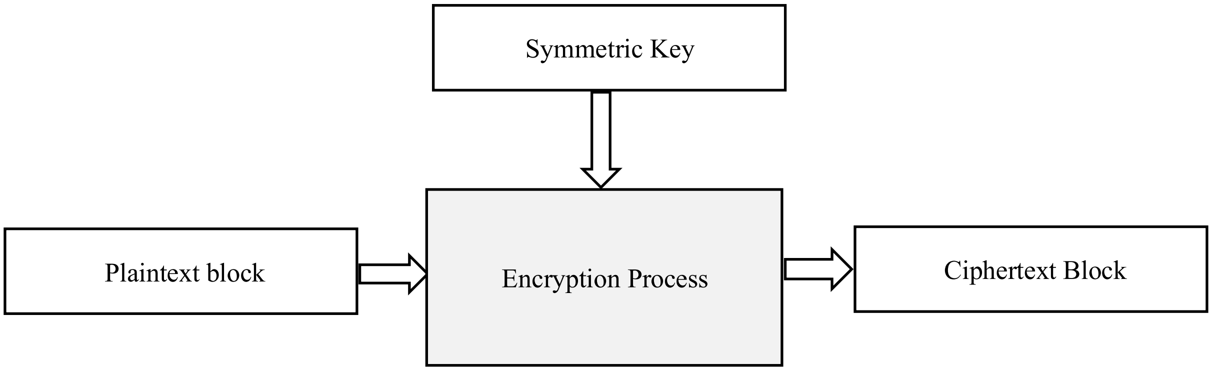


Figure 3.1: Generalized Block Cipher Diagram

3.3 Lagrange Interpolation Polynomial / Formula

The lagrange interpolation polynomial with n number of data points (f(x1) = (x1, y1)) (f(x2) = (x2, y2)), ……. (f(xn) = (xn, yn)) is a polynomial P(x) which passes through those n points having degree ≤ (n-1) and is represented by

$$P(x) = \sum_{j=1}^{n} P_j(x)$$

where

$$P_j\,(x) = y_j \prod_{\substack{k=1 \\ k\neq j}}^{n} \frac{x - x_k}{x_j - x_k}$$

The formula was published by Lagrange in 1795. With the help of this formula any polynomial can be constructed given some points. Once formed, the value of any unknown can also be determined with some certainty. The significance of this formula for constructing polynomial is much for cryptanalysts who wants to mount an interpolation attacks on block ciphers. As this attack is a known plaintext or chosen plaintext attack. Therefore, with the help of some fixed points and using this formula, a polynomial representing the cipher's algorithm can be easily constructed provided, that cipher is using simple algebraic function as their s-box.

3.4 Boolean Functions

A n-bit Boolean function is a mapping from $\{0,1\}^n \rightarrow \{0,1\}$. F: $GF(2)^n \rightarrow GF(2)$ This implies that both the arguments as well as function itself takes the values from set (0,1) only. The set {0,1} with the operations XOR and AND(⊕ , .) is denoted by GF(2). This is called the prime field of characteristic two.

$GF(2)^n = GF(2) \times GF(2) \times \ldots \times GF(2)$

$= \{(x_1, \ldots , x_n) : xi \; \varepsilon \; GF(2), \text{ for all } i \; \varepsilon \; \{0,1\}\}$

3.4.1 Representations of Boolean function

- **Truth Table**

The table which is used to represent the output values in bits (0 or 1) of a particular Boolean function for all the possible combinations of input variables is called the truth table. In other words, we can say that this is the mathematical representation of any logic gate function. Truth table of some function with three variables can been seen in figure 3.1.

y3	*y2*	*y1*	$F(y_3,y_2,y_1)$	*y3.y2+y1*
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	1	1
1	0	0	0	0
1	0	1	1	1
1	1	0	0.	0
1	1	1	0	0

Table 3.1: Truth Table of a Boolean function

If the ordering of the elements of $GF(2)^n$ is fixed as shown in table 3.1, then the truth table of function can be represented by the array (0,1,0,1,0,1,0,0).

- **Algebraic Normal Form**

A n-bit boolean function when represented in minimal sum (XOR, +) of products (AND, **.**) is said to to be in ANF. Mathematically,

$f(x_1, \ldots , x_n) = a_0 + a_1.x_1 + \ldots.. + a_n.x_n + a_{1,2}.x_1.x_2 + \ldots.\ a_{n-1,n}.x_{n-1}.x_n + \ldots\ldots\ a_{1,2\ldots.n}.x_1.x_2\ldots..x_n$.

This representation is called an ANF of a Boolean function which is unique. If the AND terms have all zero coefficients then we have an affine function. Furthermore, if the constant term is also zero, then we get a linear function. In the above equation, the " + " sign indicates the XOR operation. Therefore, all possible Boolean functions (2^{2n}) can be represented by above equation. Figure 2.1 shows that the ANF of Boolean function (0,1,0,1,0,1,0,0) in three variables is *y3.y2+y1*. There are some algorithms which can easily transform truth table representation of a Boolean function into its ANF. The importance of ANF form is that one can easily obtain the degree of a particular Boolean equation.

3.4.2 Hamming distance

Hamming distance between the two functions is defined as the number of input points at which their output differs. E.g. the distance between h and g is d(h,g) =4 as can be seen in the table 3.2.

y1	*y2*	*y3*	*h (y1, y2, y3)*	*g (y1, y2, y3)*
0	0	0	0	1
0	0	1	0	0
0	1	0	0	1
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	0

Table 3.2: Hamming Distance between two functions

3.4.3 Affine functions

Any Boolean function which can be denoted in the form $F(x_1, …. , x_n) = a_0 \oplus a_1x_1 \oplus …..\oplus a_nx_n$,where $a_0,a_1, … , a_n$ ε GF(2) is said to be an affine function. And the same function is said to be a linear function if $a_0 = 0$. The set containing all affine functions is defined by A_n and set of linear functions is denoted by L_n.

3.5 Substitution-Box

The Substitution Block is the most important cryptographic component in any block cipher which plays very crucial role in providing enhanced security due to its non-linear property. S-box is responsible to provide the basic cryptographic property also known as Shannon's property of confusion to these ciphers. The strength of a cipher mainly depends upon the quality of the s-box being used. It is usually implemented as *mxn* mapping, where m and n represents the number of input and output bits respectively which may be same or differ in size. It can also be denoted by a lookup table containing 2^m values of n bits. In other words, S-boxes are Boolean mappings from $\{0,1\}^m \rightarrow \{0,1\}^n$. This means that there are n coordinate functions, where each coordinate function is a Boolean function in m variables. Finding an optimal S-box is very hard and tricky because of the huge number of permutations mapping between input and output bits, even for small input size of s-box.. Therefore, checking all possible permutations along with their cryptographic properties to find an optimal s-box is a cumbersome task and is not practical for $m > 4$ [38].

3.5.1 Boolean Functions in S-box

An m x n s-box can be represented by a function from $GF(2)^m$ to $GF(2)^n$. This s-box can also be thought of as a sequence of n Boolean function from $GF(2)^m$ to GF(2).

Given any F: $GF(2)^m \rightarrow GF(2)^n$, then

$F(x) = (f_1(x), \ldots , f_n(x))$ *for all* $i \varepsilon \{1, \ldots , n\}$; f_i*'s* are said to coordinate functions of F. For a good S-box it is normally assumed that its coordinate functions should have good cryptographic properties but this is not enough. Besides, these coordinate functions, we also have to check component functions which are formed by taking all the linear combinations of coordinate functions. If we want to have a good s-box with strong cryptographic properties, then both coordinate functions as well as its component functions must possess good cryptographic properties.

3.5.2 Important Cryptographic properties of S-box

3.5.2.1 Balanceness

A boolean function is said to be balanced if its output yields equal number of zeros and ones for all possible input values. This property is essential to maintain the randomness of a particular Boolean function as the probability of having zero or one as output value is same i.e. 1/2. Mathematically, for a n bit Boolean function, there will be 2^{n-1} number of zeros and ones or in other words, its hamming weight $HW(f) = 2^{n-1}$. This property is is the hallmark of all component functions of the s-box to resist against different attacks especially linear cryptanalysis because more the function is imbalance, the more probability of the linear approximations obtained. Therefore, no s-box can be considered good unless all of its component functions are balanced.

3.5.2.2 Strict Avalanche criteria (SAC)

All cryptosystems must possess this property of SAC. This concept was initially introduced by Tavares and Websters for the designing of strong s-boxes [33]. It means that changing one input bit must changes 50% of the output bits. An avalanche of 50% is important to diminish any correlation between the input / output combination and resist against leakage of any information. This makes harder to analyze the ciphertext while making an effort to attack the cipher. Any cryptographic Boolean function which satisfies the above mentioned condition is said to fulfill the SAC.

3.5.2.3 Non-Linearity

The non-linearity of any Boolean function is defined as the minimum of all the distances between that function and the set of all affine functions. Mathematically,

$\mathrm{nl}(\mathrm{f}) = \min\limits_{\phi E A n} d(f, \phi)$ where A_n is the set of all affine functions over Σ^n.

y2	*y1*	$F(y1 \wedge y2)$	*0*	*y1*	*y2*	*y1+y2*	*1*	*1 + y1*	*1 + y2*	*1+y1+y2*
0	0	0	0	0	0	0	1	1	1	1
0	1	0	0	1	0	1	1	0	1	0
1	0	0	0	0	1	1	1	1	0	0
1	1	1	0	1	1	0	1	0	0	1
Distance of function f to all affine functions			1	1	1	3	3	3	3	1

Table 3.3: Calculation of non-linearity

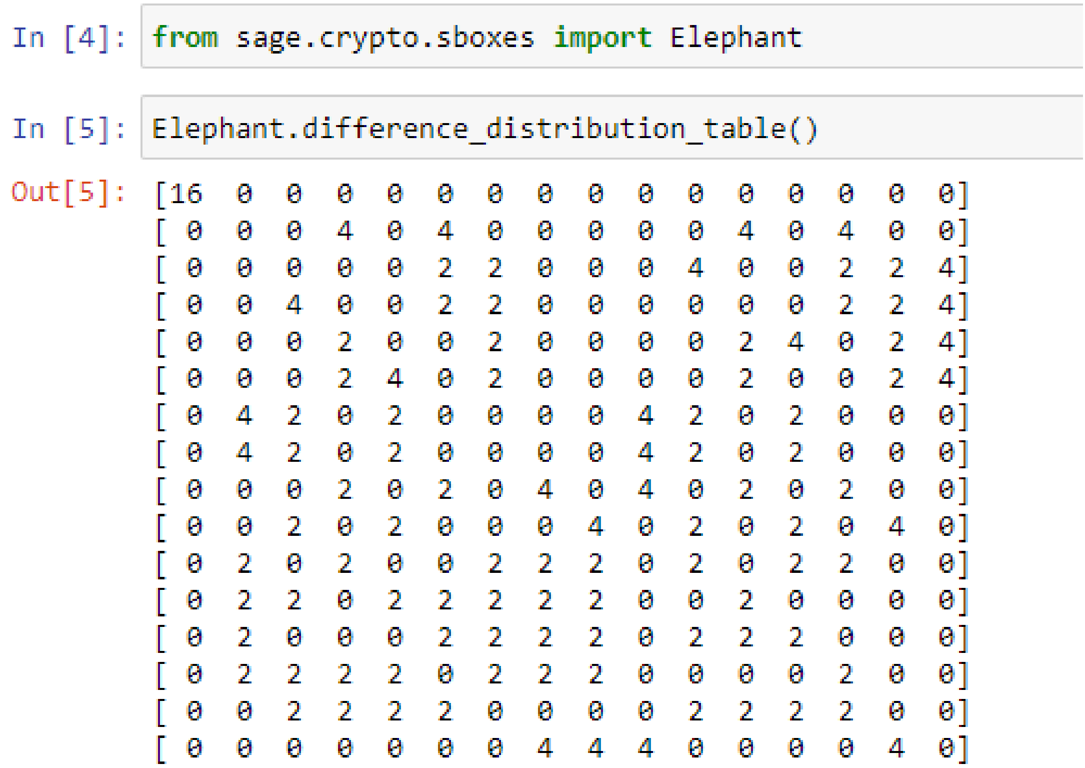
Figure 3.3 illustrates the example of a function $f: (y1 \wedge y2)$ in two variables. The hamming distance is calculated from this function to all possible affine functions. The least of all values of hamming distances is called the non-linearity of that particular Boolean function which is 1 in this case. The upper bound for calculating non-linearity of any function is given by 2^{n-1}, but in practical scenario this value cannot be achieved. However, efforts are made to keep it close to maximum value to attain maximum non-linearity. We can see that only the linear functions differ from each other with the hamming weight 2^{n-1}, so designing a non-linear function is very tricky. It must lie in between those linear functions attaining the maximum possible distance from all of them. The high value of non-linearity of s-box is required to resist against linear cryptanalysis [4]. The non-linearity of some of the ciphers with their upper bound is given in the table 3.4.

Cipher	S-box size	Non-Linearity	Upper bound
LowMC	3	2	3
SEA	3	2	3
ELEPHANT	4	4	6
PRESENT	4	4	6
GIFT	4	4	6
SHARK/ SNAKE	8	112	120
AES	8	112	120
Belt	8	102	120

Table 3.4: Non-linearities of few ciphers

3.5.2.4 Differential Uniformity

Differential uniformity is another effective property which shows the effectiveness or ineffectiveness of a differential attack against any cipher. The largest value in the difference distribution table (DDT) of any s-box is the differential uniformity. The smaller the value of differential uniformity, the more the cipher will be resistant to differential attack. The DDT of the Elephant cipher s-box is given in figure 3.2, where entries in the column and rows denotes the input and output difference respectively. It can been be seen in the figure that when input difference was set to 1 then the output difference of 3 appeared 4 times (second row & fourth column).



```
In [4]: from sage.crypto.sboxes import Elephant

In [5]: Elephant.difference_distribution_table()

Out[5]: [16  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
        [ 0  0  0  4  0  4  0  0  0  0  0  4  0  4  0  0]
        [ 0  0  0  0  0  2  2  0  0  0  4  0  0  2  2  4]
        [ 0  0  4  0  0  2  2  0  0  0  0  0  0  2  2  4]
        [ 0  0  0  2  0  0  2  0  0  0  0  2  4  0  2  4]
        [ 0  0  0  2  4  0  2  0  0  0  0  2  0  0  2  4]
        [ 0  4  2  0  2  0  0  0  0  4  2  0  2  0  0  0]
        [ 0  4  2  0  2  0  0  0  0  4  2  0  2  0  0  0]
        [ 0  0  0  2  0  2  0  4  0  4  0  2  0  2  0  0]
        [ 0  0  2  0  2  0  0  0  4  0  2  0  2  0  4  0]
        [ 0  2  0  2  0  0  2  2  2  0  2  0  2  2  0  0]
        [ 0  2  2  0  2  2  2  2  2  0  0  2  0  0  0  0]
        [ 0  2  0  0  0  2  2  2  2  0  2  2  2  0  0  0]
        [ 0  2  2  2  2  0  2  2  2  0  0  0  0  2  0  0]
        [ 0  0  2  2  2  2  0  0  0  0  2  2  2  2  0  0]
        [ 0  0  0  0  0  0  0  4  4  4  0  0  0  0  4  0]
```

Figure 3.2: DDT of Elephant S-box

The data in the figure 3.2 shows that the maximum value is 4 which is basically the differentially uniformity of elephant s-box. In order to avoid differential cryptanalysis, this value has to be as minimum as possible.

3.5.2.5 Bit Independence criteria

According to this criterion [36,37], changing one bit of input results in modifying the output bits without any interdependence. An S-box whose output bits behave independently having no dependency on each other is considered to fulfill this important property. Furthermore, if any s-box fulfills this bit independence criterion than that implies that all its coordinate functions have high non-linearity and also satisfy SAC. This is highly desirable property of any crypto system in order to make system complex. In other words, there will be no statistical dependencies between output bits and are acting independently. The authors in [35] have proposed a technique to test BIC.

3.5.2.6 Bijectiveness

For n x n S-box, bijectiveness means that each of the input vector of s-box must map one-to-one and onto to the output vectors. In other words, the LUP table of that s-box must have unique values ranging from 0 → 2^n – 1. For an s-box to be a bijective implies that all of its component functions must be balanced [38].

3.6 SageMath

Sagemath is a free and open-source software written mostly in python and Cython language and was initially released on 24 February 2005 [31][32]. It has many in built libraries mostly related to computer algebra system and provides a common interface by integrating different specialized packages. This tool is also very helpful for cryptographers as it has a specialized library of crypto which deals specifically with analyzing various ciphers and their s-boxes. A very user friendly and easy to learn tool being used by students as well as professionals.

3.6.1 Some Important Sage Math related properties / functions

3.6.1.1 Interpolated Polynomial

This function is very important and helps in analyzing ciphers with respect to algebraic and interpolation attacks. In sage math, the function interpolation_polynomial() helps in computing the algebraic expression of any s-box. This property reveals the weakness or strength of any cipher against algebraic and interpolation attacks. From the computed algebraic expression, one gets to know the maximum degree and the number of terms in that expression. The upper bound of number of terms in that expression for a n bit s-box are given by 2^n-1 and maximum degree by 2^n-2. Any cipher having s-box which uses simple algebraic function is vulnerable to these attacks. The interpolation polynomial of PRESENT cipher's s-box computed by sage math in figure 3.3.

3.6.1.2 Maximum degree

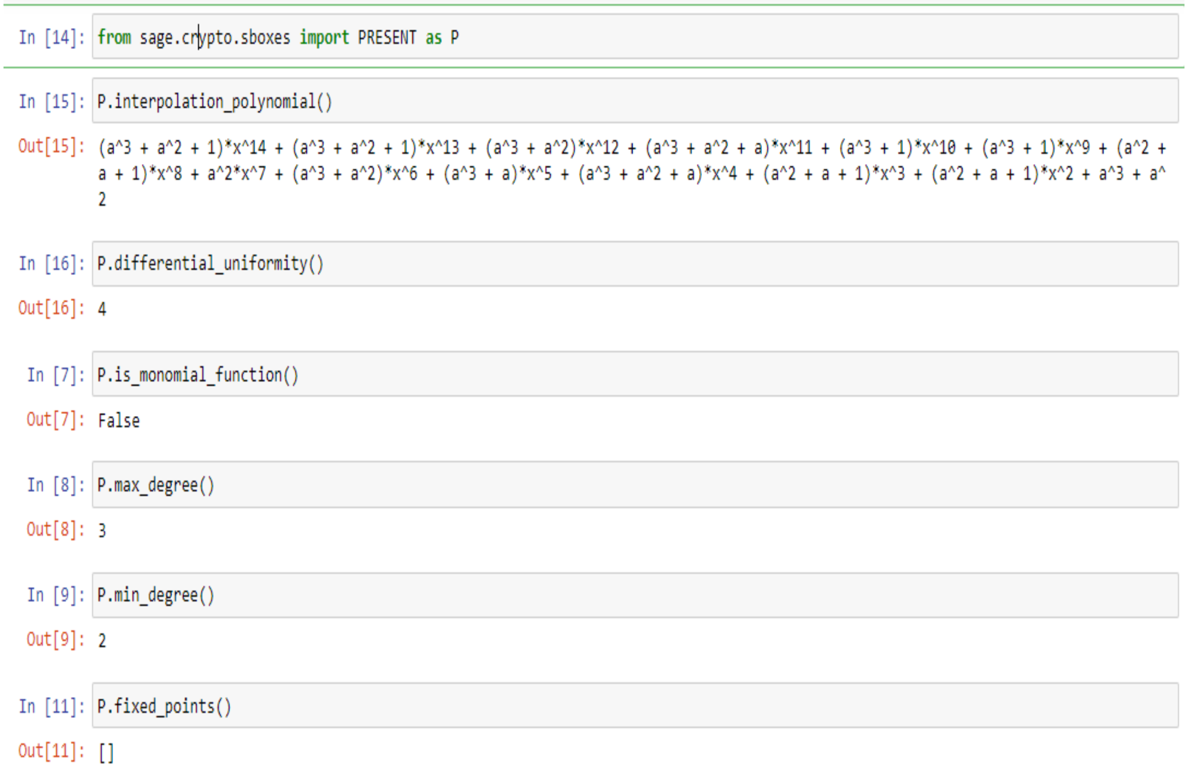
.This property helps in finding the maximum degree out of all component functions. The upper bound / desirable maximum degree for all of the component functions in a n bit s-box is given by n-1. The function max_degree() helps in computing the maximum degree of component functions in any s-box

3.6.1.3 Minimum degree

. This property helps in finding the minimum degree out of all component functions. The upper bound / desirable minimum degree for all of the component functions in a n bit s-box is given by n-1. But some ciphers use s-boxes which have low degree component functions. The function min_degree() helps in computing the minimum degree of component functions in any s-box.

3.6.1.4 Fixed points

Fixed point means that s-box input valus is equal to output value. There are several s-boxes in which some of the points are fixed. This is undesirable property and must be avoided or at least kept these points as minimum as possible to avoid statistical attacks. The function fixed_points() is used in sage math to determine these points for any particular s-box. Some of the important properties of s-box of cipher PRESENT can be seen in figure 3.3.



```
In [14]: from sage.crypto.sboxes import PRESENT as P

In [15]: P.interpolation_polynomial()

Out[15]: (a^3 + a^2 + 1)*x^14 + (a^3 + a^2 + 1)*x^13 + (a^3 + a^2)*x^12 + (a^3 + a^2 + a)*x^11 + (a^3 + 1)*x^10 + (a^3 + 1)*x^9 + (a^2 +
         a + 1)*x^8 + a^2*x^7 + (a^3 + a^2)*x^6 + (a^3 + a)*x^5 + (a^3 + a^2 + a)*x^4 + (a^2 + a + 1)*x^3 + (a^2 + a + 1)*x^2 + a^3 + a^
         2

In [16]: P.differential_uniformity()

Out[16]: 4

 In [7]: P.is_monomial_function()

Out[7]: False

 In [8]: P.max_degree()

Out[8]: 3

 In [9]: P.min_degree()

Out[9]: 2

In [11]: P.fixed_points()

Out[11]: []
```

Figure 3.3: Some important properties of PRESENT cipher s-box in sage math

3.7 Different Complexities of Attack

3.7.1 Time Complexity

The time complexity of an algorithm is an expression which tells us about the time requirement for running that algorithm. In cryptanalysis domain, it defined as the number of XOR operations required for breaking any cipher.

3.7.2 Memory Complexity

The memory complexity of an algorithm is an expression which tells us about the computer memory requirement for running that algorithm. In cryptanalysis domain, it defined as the amount of memory resources required for breaking any cipher.

3.7.3 Data Complexity

The data complexity in cryptanalysis domain means the number of plaintext/ciphertext (*p*/*t*) pairs required for breaking any cipher.

Chapter 4

METHODOLOGY ADOPTED TO GIVE A DESIGN CRITERIA

4.1 Introduction

In this chapter, a methodology adopted to conduct a detailed research on the topic is discussed. We analyzed ciphers and their s-boxes on whom this attack occurred. Most of the work carried out is with the help of sage math tool. The security claims given by different authors and parameters responsible to strengthen or weaken the particular s-boxes / cipher are critically analyzed. Moreover, a comparison between s-boxes of ciphers on whom interpolation attack occurred and the strong s-boxes being used widely was also carried out to find a design criteria for good s-box against interpolation attack.

4.2 Work Flow for defining design criteria of S-box

All parameters related to interpolation attack and their relation with the s-boxes are studied critically to find any weak properties of s-boxes against the attack in order to finally give some design criteria. The methodology adopted in our research can be easily represented/ understood with the help of figure 4.1 below.

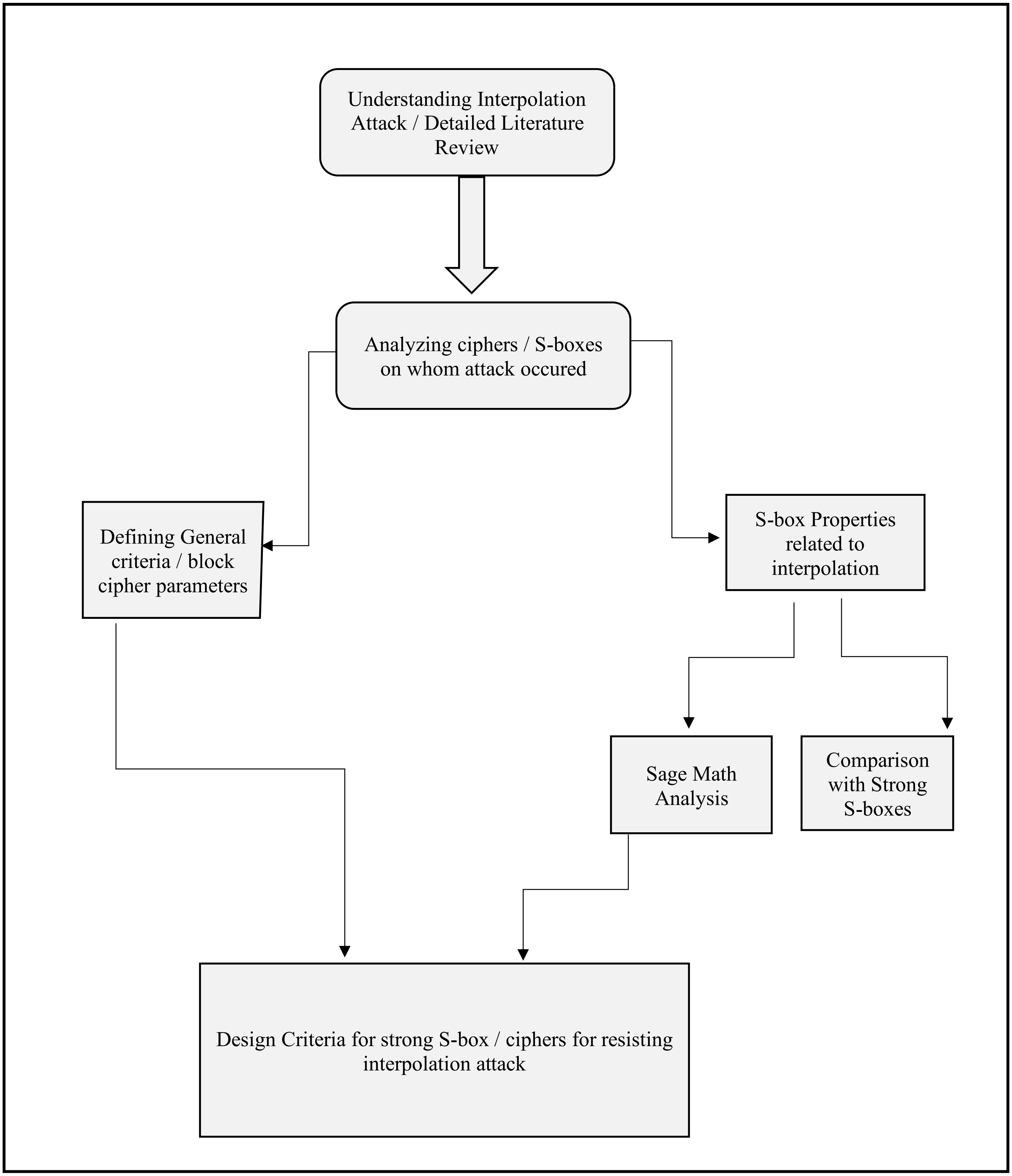

Understanding Interpolation Attack / Detailed Literature Review
Analyzing ciphers / S-boxes on whom attack occured
Defining General criteria / block cipher parameters
S-box Properties related to interpolation
Sage Math Analysis
Comparison with Strong S-boxes
Design Criteria for strong S-box / ciphers for resisting interpolation attack

Figure 4.1: Work Flow Diagram for S-Box Design Criteria against Interpolation Attack

4.3 Parameters Expediating Interpolation Attack

4.3.1 S-box with simple Algebraic function or (Degree of the Algebraic Function Representing the S-box)

Ciphers which use simple algebraic functions as their s-boxes are more vulnerable to interpolation attack [6][28]. By simple algebraic function we mean that the algebraic complexity of that s-box is minimal. In other words, the number of terms in the algebraic expression of that s-box are not touching their upper bound and/ or the degree is low. e.g. cubing function $f(x) = x^3$ or simple inverse function $f(x) = x^{-1}$ in GF (2^m) have only one non zero term in their algebraic expression. For an s-box of 8 bits, the upper bound on the number of terms is 255 and the maximum degree of that polynomial must be 254. The complexity of the such cryptanalytic attacks depends upon the degree and/ or the number of terms in the polynomial expression of the cipher being attacked [17]. The more number of terms and higher the degree of polynomial increases the complexity exponentially after every round of block cipher. For the same reason, no known ciphers have been attacked for their full rounds and only the toy ciphers and some variants of ciphers are attacked.

4.3.2 Number of S-boxes

Block ciphers used in cryptography have different designs considering the size of s-boxes, number of s-boxes in each layer, round function being used, key scheduling algorithms and different block sizes etc. One another aspect regarding the interpolation attack pointed out by many researchers is the role of the number of s-boxes in complexity of attack. Interpolation attack is independent of the size of s-box, however it depends upon the number of s-boxes being employed in ciphers' design [6][29]. The more the number of s-boxes, the more will be the data, memory and time complexity of the attack. Data complexity implies more number of p/c pairs requires to mount an attack whereas time and memory complexity means the requirement of more resources of time and memory. An interpolation attack was mounted on variant of SHARK cipher having s-box size 8 bits and it can be seen from the figure 4.2 that by keeping the number of rounds constant, the data, memory and time complexity of the attack increases proportionally with the increasing number of s-boxes [6][29]. The use of bigger and few s-boxes does not mean result in more secure ciphers [6].

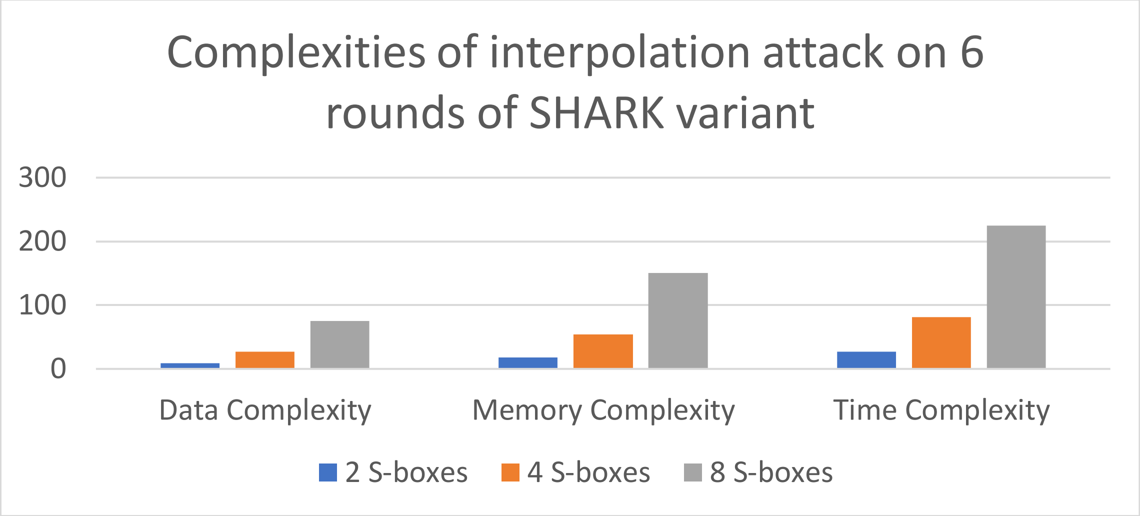


Figure 4.2: Complexities of Interpolation attack on SHARK variant

4.3.3 Size of S-box

Size of the s-boxes vary from cipher to cipher. Increasing size does not enhance the complexity of the attack and is independent [6]. However, to show the relation of the size of s-box with the complexity of the attack results from the cipher SNAKE can be seen in the figure 4.3 and figure 4.4. SNAKE uses inverse function S(x) = x^{-1} in GF(2^m) as s-box in round function. It is a Fiestal cipher having two variants SNAKE(1) and SNAKE(2) employing different round functions. The beauty of this cipher is that it is flexible as far as input of block size is concerned. It can encrypt both 64 and 128 bits of block data. The interpolation attack has been mounted on this cipher using rational expressions and computer algebra [12]. The graphs below show the complexities of different block sizes (only differs in size of s-box) along with the number of rounds attacked. In case of SNAKE(1), when the block size is 128 bits and size of s-box is 16 bits, all round keys are recovered upto 15 rounds and when the block size is 64 bits and size of s-box is 8 bits, only round keys upto 11 rounds are recovered. The complexity of the attack decreases with the increases size of the s-box as seen from the figure 4.3 below because the maximum number of available p/c pairs for the attacker increases when the block size is 128 bits compared to that of 64 bits block size [12]. The data in the figure 4.3 demonstrate that the complexity for breaking round 14 is 2^{30} in case of 16 bit s-box which is less than the complexity of breaking 10th round i.e. 2^{39} as in case of 8 bit s-box variant of SNAKE(1).

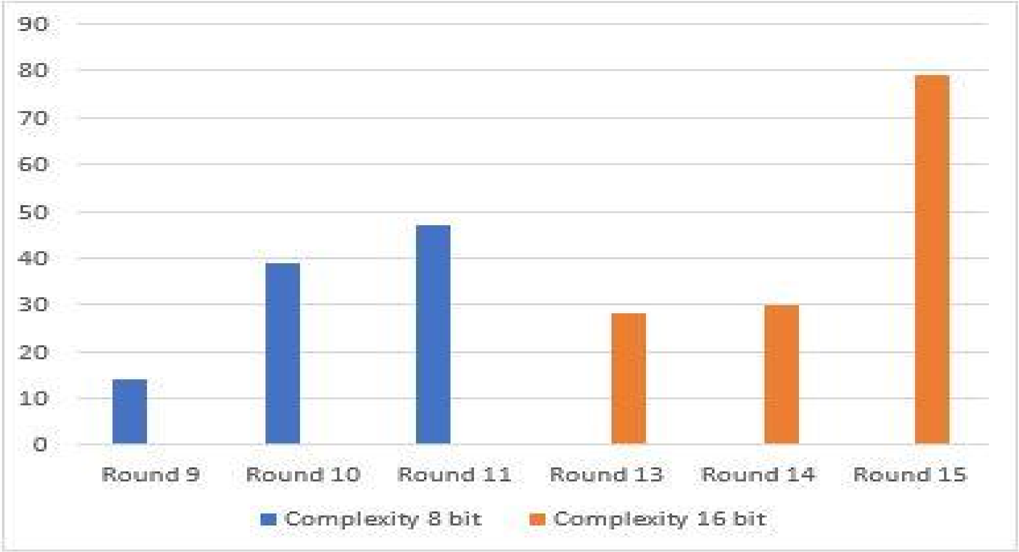


Figure 4.3: Complexities of Interpolation attack on 8 &16 bit S-box with varying rounds

4.3.4 Round Function / Number of rounds

One of the important parameters of block ciphers is its round function. In substitution-permutation network round function consists of two basic operations of substitution and permutation which are performed with the help of substitution and permutation boxes. These boxes must possess good cryptographic properties in order to thwart cryptanalytic attacks. Round function must be so designed that it should have high algebraic complexity. It means that when a plaintext in transformed into some ciphertext after passing through round function, the polynomial of that intermediate ciphertext must be complex having high algebraic degree and more number of terms. In [6], the authors were able to mount an interpolation attack (Global and Instance deduction) and also for key recovery on the cipher PURE which uses simple algebraic function as its round function i.e. F(x) = x3. This function is very weak algebraically, therefore weakens the cipher against interpolation attack. Furthermore, number of rounds play an equally important role in strengthening the cipher against interpolation attack like any other attack. The more the number of rounds, the more difficult to break the cipher to extract secret key. It is due to this fact that attackers are only able to break the ciphers upto few rounds and not the complete cipher. The demonstration of this fact can be seen from the interpolation attack carried out on modified version of cipher SHARK [6][28], which uses inverse function as s-box S(x) = x-1. From the data in the figure 4.5, it can be seen that more number of rounds enhance complexity of the attack while keeping the number of s-boxes same.

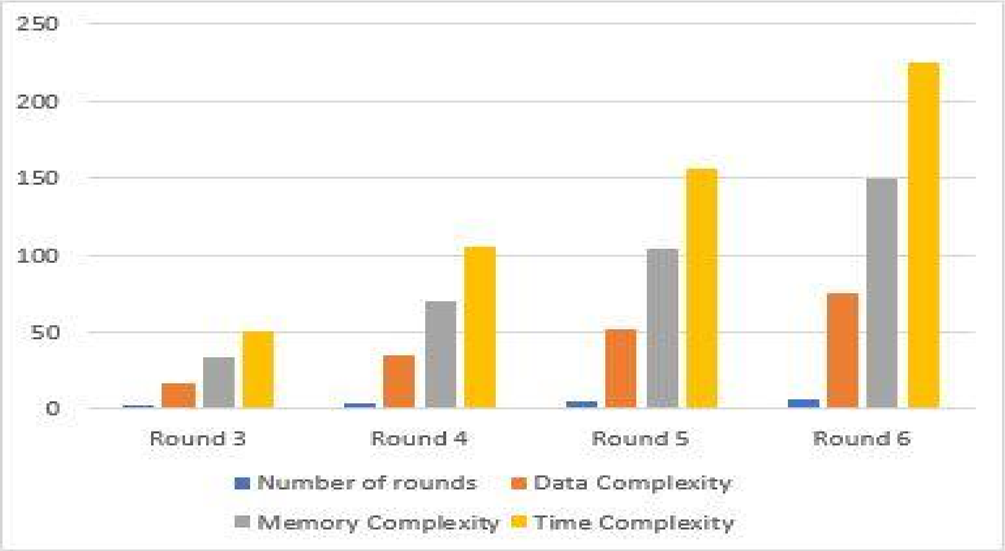


Figure 4.4: Complexities of Interpolation attack on SHARK variant

4.4 Sagemath Analysis

Sage math is a good cryptographic tool for analyzing ciphers and their s-boxes to check for their cryptographic properties [31]. It can be used to determine the strengths and weakness of any cipher/ s-boxes to check for any vulnerability or weak properties against different attacks. In this section, we have employed sagemath tool to analyse s-boxes of various ciphers against which interpolation attacks had met partial or complete success. Furthermore, a comparison is carried out between these s-boxes and s-boxes of widely used ciphers like AES.

For the study purposes we have taken ciphers SHARK [28], SNAKE [4], ELEPHANT [5] and LowMC-80/ 128[26]. The table 4.1 below demonstrate related properties of s-boxes of above mentioned ciphers. From the data in the table, it can be easily said that the most of these ciphers uses s-boxes which are represented by simple algebraic expression (less number of terms in their interpolated polynomial or simply with a monomial). Furthermore, the ciphers SHARK and SNAKE, though having high non-linearity were still easily attacked. Therefore, this property of non-linearity does not play any significant role in thwarting the interpolation attack.

Cipher	**S-box size(bits)**	**Non Linearity**	**DU**	**Degree Interpolated Polynomial**	**NTAE**	**Monomial**	**Is Almost Bent**	**Fixed Points**
Shark variant	8	112	4	254	1	Yes	No	-
Snake	8	112	4	254	1	Yes	No	0,1
LowMC-80/128	3	2	2	6	3	No	Yes	0,1
Elephant	4	4	4	14	15	No	No	-

Table 4.1: S-boxes Properties of attacked ciphers

NTAE: Number of terms in algebraic expression DU: Differential uniformity

4.5 Comparison of S-box Properties

In order to compare above mentioned s-boxes with others on whom no interpolation attack is carried out, we have divided our analysis into three categories according to the size of s-box for consistency purposes. e.g. LowMC-80/128 cipher uses 3 bit s-boxes, therefore we have compared it with the ciphers using same size s-boxes like cipher SEA and Pyjamask_3.

4.5.1 3 bit S-boxes

We have taken three different 3 bit s-boxes for analysis purposes. Among these 3, only LowMC-80/128 is a cipher where interpolation attack is mounted and remaining two are for comparison purposes. As seen from the table 4.2 below, we can deduce that all the properties like non-linearity, balanceness, differential uniformity etc are same for all. However, the only difference is that of number of terms in the algebraic expression of their s-boxes and fixed points. Furthermore, the data in the table also shows that the degree of the interpolated polynomial of these s-boxes are same which is 6 but the number of terms vary in their algebraic expression. LowMC has less number of terms than ciphers SEA and Pyjamask_3 which reduces the algebraic complexity of LowMC. The maximum number of terms for a 3 bit s-box algebraic expression should be 7. LowMC and SEA both have two fixed points each which can be an indicator for successful bit predictions in statistical analysis.

Cipher	Non-linearity	Max deg	Min deg	Balance	Degree of S-box Expression	Monomial	Fixed Points	DU	NTAE
LowMC	2	2	2	Yes	6	No	0,1	2	3
SEA	2	2	2	Yes	6	No	0,4	2	6
Pyjamask_3	2	2	2	Yes	6	No	-	2	6

Table 4.2: Properties of 3-bit S-boxes

NTAE: Number of terms in algebraic expression DU: Differential uniformity

4.5.2 4 bit S-boxes

Elephant-Delirium is a cipher on whom interpolation attack has been carried out and we have compared it with other 4 bit s-boxes of ciphers Gift, Panda and Present. The properties shown in the table 4.3 below does not give any clear indication of the possible reason for attack occurring on cipher Elephant, however we know that Elephant is a lightweight LFSR based authentication encryption scheme [21], therefore due to its dissimilar structure from other block ciphers some other parameters may have played a role in assisting interpolation attack and not the s-box. Furthermore, the results in table 4.3 also illustrates the weaknesses of the other ciphers' s-boxes as well. e.g. The s-box of the cipher Panda has two fixed points as well as less number of terms in the algebraic expression of its s-box weakening the cipher against interpolation attack, therefore we may believe that Panda is vulnerable to interpolation attack and is not yet analyzed by the cryptanalysts against this attack. One another parameter minimizing the algebraic complexity of ciphers Gift, Present and Elephant is that one or few of the component functions of their s-boxes are not attaining the maximum degree which is 3.

Cipher	Non-Lin	Max Deg	Min deg	Balance	Degree of S-box expression	Monomial	Fixed points	DU	NTAE
Gift	4	3	2	Yes	14	No	-	6	15
Present	4	3	2	Yes	14	No	-	4	14
Panda	4	3	3	Yes	14	No	0,1	4	12
Elephant	4	3	2	Yes	14	No	-	4	15

Table 4.3: Properties of 4 bit S-boxes

NTAE: Number of terms in algebraic expression DU: Differential uniformity

4.5.3 8 bit S-boxes

SHARK and SNAKE ciphers are known to be attacked with interpolation technique. Both of them use 8 bits s-boxes. On the other hand, AES is the worldwide used block cipher which employs 8 bit s-box and is considered very good due to its strong cryptographic properties. Therefore, we have compared SHARK / SNAKE ciphers with that of AES and BELT. Both SNAKE and SHARK variant uses inverse function as their s-boxes and their algebraic expression is simply a monomial (one term only) which makes them vulnerable to interpolation attack. On the other hand, AES and BELT s-box' algebraic expression is very complex and involves 255 in their polynomial expression as can be seen in the table 4.4, therefore making them strong against this attack. Initially, the algebraic expression of AES s-box used to have only 9 terms. Therefore, to enhance the algebraic complexity of AES s-box expression the authors in [30] proposed a new criteria for AES s-box by increasing the number of terms to 255 in its algebraic expression.

Cipher	Non-linearity	Max deg	Min deg	Balance	Deg of S-box expression	Monomial	Fixed Points	DU	NTAE
SHARK/ SNAKE	112	7	7	yes	254	yes	0,1	4	1
AES	112	7	7	yes	254	No	-	4	255
Belt	102	7	6	yes	254	No	-	8	255

Table 4.4: Properties of 8 bit S-boxes

NTAE: Number of terms in algebraic expression DU: Differential uniformity

4.6 S-box Design Criteria

S-boxes are not designed in isolation. It also depends upon the cipher. The s-box properties needs to be crafted together with the design of the block cipher to make sure that they work well together. However, there are some basic criteria along with other desirable properties of s-box which makes a cipher strong against interpolation attack. The figure 4.6 describes s-box design criteria which makes a cipher secure to some extent against interpolation attack.

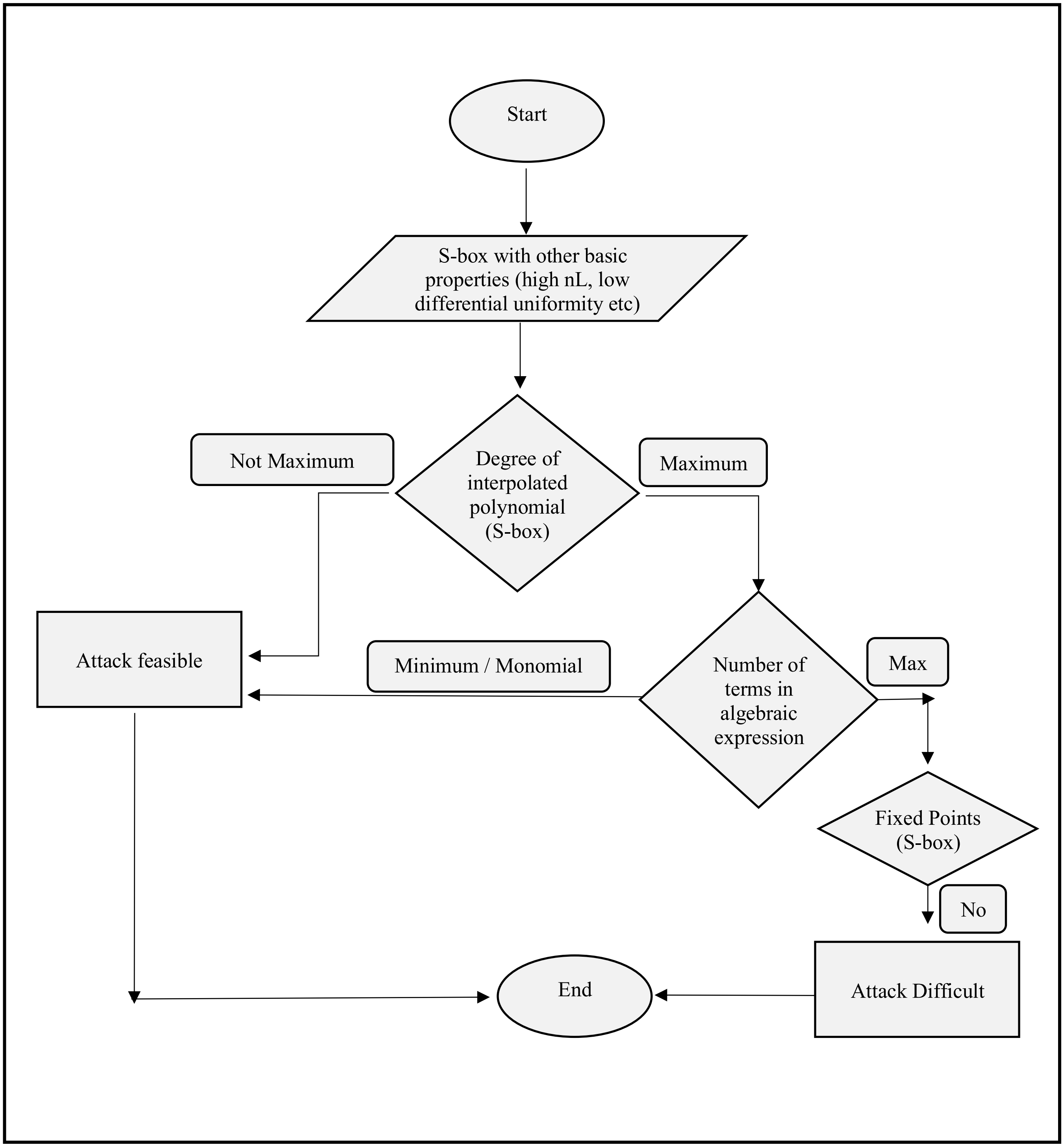

Start
S-box with other basic properties (high nL, low differential uniformity etc)
Degree of interpolated polynomial (S-box)
Not Maximum
Maximum
Attack feasible
Minimum / Monomial
Number of terms in algebraic expression
Max
Fixed Points (S-box)
No
End
Attack Difficult

Figure 4.6: Flow Chart S-box Design Criteria

Figure 4.6 explicitly highlighting the important properties that any s-box must possess in order to resist or enhance the complexity of interpolation attack. From the figure, it can be seen that the interpolated polynomial of any n-bit s-box must be complex, meaning by that it must have the maximum degree (2^n - 2) and the maximum number of terms (2^n - 1) in its polynomial expression. These two are the dominant properties as far as interpolation attack on block ciphers is concerned which must be catered for while designing a s-box. However, another property of fixed points must also be avoided to give them further strength against the attack.

Chapter 5

RECOMMENDATIONS, CONCLUSION AND FUTURE WORK

5.1 Recommendations

Keeping in view of the research findings, following are few recommendations for the designers and users.

- Interpolation attack depends upon the number of s-boxes in ciphers algorithm. The more the number of s-boxes, the more will be complexity of this attack. Therefore, it is suggested that number of s-boxes must be higher to enhance complexity of this attack.
- Employ a cipher with more number of rounds as a cipher with more rounds is comparatively more secure and is difficult to interpolate its polynomial expression.
- While designing s-boxes, it must be kept in mind that the algebraic expression of that s-box must be complex and is not represented with a simple algebraic function.
- S-box with the fixed points is easy to attack. Therefore, do not use such s-boxes in order to avoid interpolation attacks.
- Increasing the size of s-boxes does not guarantee extra security and attack is independent of size of s-box. Therefore, consider employing higher number of small size s-boxes over fewer large size s-boxes.

5.2 Conclusion

The importance of data has been increased manifolds in the recent years due to its growing demand, storage and processing at all levels. This data must also be protected from unauthorized entities for any kind of attacks. Block ciphers are used widely for the protection of critical data against any adversarial attacks. Every component of the block cipher plays its vital role in providing security to these ciphers. However, the major role is played by the s-box as being the only non-linear component of block ciphers. In this thesis we have analyzed interpolation attacks on block ciphers and gave some parameters which are necessary to enhance the complexity of these attacks. Some parameters are related to the ciphers design like number of rounds and number of s-boxes etc. However, the major focus of this work was to critically analyze s-boxes and its properties and find out what role s-box play in either facilitating or resisting these attacks. Finally, we have presented a s-box design criteria which is necessary to make this attack difficult or enhance the complexity more than brute force effort.

5.3 Future Work

S-box design criteria presented in this thesis enhance the complexity of interpolation attack. However, the possible future work objectives may be:

- To find the role / relation of key bits with the complexity of the attack.
- To find the complexity of attack for both static and dynamic s-boxes and finds out which one is better.

BIBLIOGRAPHY

[1] Gentry, Craig. (2010). Computing Arbitrary Functions of Encrypted Data. Commun. ACM. 53. 97-105. 10.1145/1666420.1666444.

[2] Rehman, Osama & Memon, Imran & Rizvi, Safdar. (2019). An efficient construction of key-dependent substitution box based on chaotic sine map. International Journal of Distributed Sensor Networks. 15. 155014771989595. 10.1177/1550147719895957.

[3] Aoki K. (2000) Efficient Evaluation of Security against Generalized Interpolation Attack. In: Heys H., Adams C. (eds) Selected Areas in Cryptography. SAC 1999. Lecture Notes in Computer Science, vol 1758. Springer, Berlin, Heidelberg.

[4] Matsui M. (1994) Linear Cryptanalysis Method for DES Cipher. In: Helleseth T. (eds) Advances in Cryptology — EUROCRYPT '93. EUROCRYPT 1993. Lecture Notes in Computer Science, vol 765. Springer, Berlin, Heidelberg.

[5] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems", Journal of Cryptology, vol. 4, no. 1, pp. 3-72, 1991.

[6] Jakobsen T., Knudsen L.R. (1997) The interpolation attack on block ciphers. In: Biham E. (eds) Fast Software Encryption. FSE 1997. Lecture Notes in Computer Science, vol 1267. Springer, Berlin, Heidelberg.

[7] V. Rijmen, 3. Daemen, B. Preneel, A. Bosselaers, and E. De Win. The cipher SHARK. In Gollmarm D., editor, *Fast Software Encryption, Third International Workshop, Cambridge, U.K., February 1996, LNCS 1039,* pages 99-112. Springer Verlag, 1996

[8] K. Kiefer. A New Design Concept for Building Secure Block Ciphers. In J. Pribyl, editor, *Proceedings of the 1st International Conference on the Theory and Applications of Cryptology, PRAGOCRYPT'96, Prague, Czech Republic,* pages 30-41.CTU Publishing House, 1996.

[9] K. Nyberg and L.R. Knudsen. Provable security against a differential attack. *The Journal of Cryptology,* 8(1):27-38, 1995

[10] L.R. Knudsen. Truncated and higher order differentials. In B. Preneel, editor, *Fast Software Encryption . Second International Workshop, Leuven, Belgium, LNCS 1008,* pages 196-211. Springer Verlag, 1995.

[11] X. Lai. Higher order derivatives and differential cryptanalysis. In *Proc. "Symposium on Communication, Coding and Cryptography", in honor of James L. Massey on the occasion of his 60'th birthday, Feb. 10-13, 1994, Monte- Verita, Ascona,Switzerland,* 1994.

[12] Moriai S., Shimoyama T., Kaneko T. (1999) Interpolation Attacks of the Block Cipher: SNAKE. In: Knudsen L. (eds) Fast Software Encryption. FSE 1999. Lecture Notes in Computer Science, vol 1636. Springer, Berlin, Heidelberg.

[13] C.Lee and Y.Cha, \The Block Cipher : SNAKE with Provable Resistance against DC and LC attacks," In Proceedings of 1997 Korea-Japan Joint Workshop on Information Security and Cryptology (JW-ISC'97), pp.3{17, 1997.

[14] J. Liu, S. Chen and L. Zhao, "Lagrange Interpolation Attack against 6 Rounds of Rijndael-128," 2013 5th International Conference on Intelligent Networking and Collaborative Systems, Xi'an, 2013, pp. 652-655.

[15] J. Daemen, V. Rijmen, "Proposal AES, Rijndael[C], "Proceedings from the First Advanced Encryption Standard Candidate Conference, National Institute of Standards and Technology (NIST). 1998.

[16] Sun B., Qu L., Li C. (2009) New Cryptanalysis of Block Ciphers with Low Algebraic Degree. In: Dunkelman O. (eds) Fast Software Encryption. FSE 2009. Lecture Notes in Computer Science, vol 5665. Springer, Berlin, Heidelberg.

[17] Youssef A.M., Gong G. (2001) On the Interpolation Attacks on Block Ciphers. In: Goos G., Hartmanis J., van Leeuwen J., Schneier B. (eds) Fast Software Encryption. FSE 2000. Lecture Notes in Computer Science, vol 1978. Springer, Berlin, Heidelberg.

[18] Guang Gong and S. W. Golomb, "Transform domain analysis of DES," in *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 2065-2073, Sept. 1999, doi: 10.1109/18.782138

[19] Kurosawa K., Iwata T., Duong Quang V. (2001) Root Finding Interpolation Attack. In: Stinson D.R., Tavares S. (eds) Selected Areas in Cryptography. SAC 2000. Lecture Notes in Computer Science, vol 2012. Springer, Berlin, Heidelberg.

[20] Zhou, Haibo, Rui Zong, Xiaoyang Dong, Keting Jia, and Willi Meier. "Interpolation Attacks on Round-Reduced Elephant, Kravatte and Xoofff." The Computer Journal (2020). https://doi.org/10.1093/comjnl/bxaa101

[21] Beyne, T., Chen, Y., Dobraunig, C. and Mennink, B. (2019) Elephant v1. In *NIST Lightweight Cryptography Project*, https://www.esat.kuleuven. be/cosic/elephant/

[22] Bertoni, G., Daemen, J., Hoffert, S., Peeters, M.,Assche, G. V. and Keer, R. V. (2018) Farfalle:Parallel Permutation-based Cryptography. In *FSE 2018*, Bruges, Belgium, March 5-7, pp. 1-38. IACR Transactions on Symmetric Cryptology

[23] Daemen, J., Hoffert, S., Assche, G. V. and Keer, R.V. (2019) The Design of Xoodoo And Xoofff. In *FSE 2019*, Paris, France, March 25-28, vol. 4 pp. 1-28. IACR Transactions on Symmetric Cryptology

[24] Li, Chaoyun, and Bart Preneel. "Improved interpolation attacks on cryptographic primitives of low algebraic degree." In International Conference on Selected Areas in Cryptography, pp. 171-193. Springer, Cham, 2019.

[25] M. R. Albrecht, C. Rechberger, T. Schneider, T. Tiessen, and M. Zohner. Ciphers for MPC and FHE. In E. Oswald and M. Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 430{454.Springer, 2015

[26] Dinur I., Liu Y., Meier W., Wang Q. (2015) Optimized Interpolation Attacks on LowMC. In: Iwata T., Cheon J. (eds) Advances in Cryptology – ASIACRYPT 2015. ASIACRYPT 2015. Lecture Notes in Computer Science, vol 9453. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-48800-3_22

[27] Jakobsen, Thomas. (1998). Cryptanalysis of Block Ciphers with Probabilistic Non-Linear Relations of Low Degree. Lecture Notes in Computer Science.

[28] Jakobsen, T., Knudsen, L. Attacks on Block Ciphers of Low Algebraic Degree. J. Cryptology 14, 197–210 (2001). 10.1007/BFb0055730.

[29] Rezaeipour, Davood & Md Said, Mohamad Rushdan. (2009). New Directions in Cryptanalysis of Block Ciphers. Journal of Computer Science. 5. 10.3844/jcssp.2009.1091.1094.

[30] L. Jinomeiq, W. Baoduui and W. Xinmei, "One AES S-box to increase complexity and its cryptanalysis," in Journal of Systems Engineering and Electronics, vol. 18, no. 2, pp. 427-433, June 2007, doi: 10.1016/S1004-4132(07)60108-X

[31] "Sagemath Mathematical Software System - Sage". 2021. Sagemath Mathematical Software System. https://www.sagemath.org/.

[32] https://github.com/sagemath

[33] Webster, A. F., & Tavares, S. E. (1986). On the design of S-boxes. Advances in Cryptology, Lecture Notes in Computer Science, 218, 523–534.

[34] Wen, Q., X. Niu and Y. Yang, Boolean Functions in Modern Cryptology, Science Press, Beijing, 2000.

[35] Adams, C., & Tavares, S. (1990). The structured design of cryptographically good S-boxes. Journal of Cryptology, 3(1), 27–41.

[36] Webster, A.F.; Tavares, S.E. On the Design of S-Boxes. In Proceedings of the Conference on Theory and Application of Cryptographic Techniques, Santa Barbara, CA, USA, 18–22 August 1986.

[37] 53. Adams, C.; Tavares, S. The Structured Design of Cryptographically Good S-Boxes. *J. Cryptol.* 1990, *3*, 27–31.

[38] Ahmad, M., Doja, M.N. & Beg, M.M.S. ABC Optimization Based Construction of Strong Substitution-Boxes. Wireless Pers Commun 101, 1715–1729 (2018).