Chapter 1: Overview

1.1 Security of Cryptography

In the year of 1949 *Claude Shannon* published a paper entitled "*Communication Theory of Secrecy Systems*" in the *Bell Systems Technical Journal*. This paper had a great influence on the scientific study of cryptography. Some of the various approaches for evaluating the security of cryptosystems are considered here.

Computational Security: This measure concerns the computational effort required to break a cryptosystem. A cryptosystem is conceptually secure if the best algorithm for breaking it requires at least ***N*** operations, where ***N*** is a specified and very large integer. The problem is that no known practical cryptosystem can be proved to be secure under this definition. In practice, people often study the computational security of a cryptosystem with respect to certain specific type of attacks like *exhaustive key search*. Security against one specific type of attack does not ensure security against some other type of attack.

Provable Security: This approach is to provide evidence of security by means of a reduction. It shows that if the cryptosystem can be *broken* in some specific way, then it would be possible to efficiently solve some well-studied problem that is thought to be difficult. It may be possible to prove a statement of the type "*a given cryptosystem is secure if a given integer n cannot be factored*". Cryptosystem of this type are sometimes termed as provably secure.

Unconditional Security: This measure concern the security of cryptosystems when there is no bound placed on the amount of computation. A cryptosystem is defined to be unconditionally secure if it cannot be broken, even with infinite computational resources.

A cryptosystem has perfect secrecy if $P_r[X|Y] = P_r[X]$. For all $X \in P,\ Y \in C$.

That is, posteriori probability that the plaintext is X, given that the ciphertext Y is observed is identical to a priori: probability that the plaintext is X.

1.2 Protocol of Cryptography

The whole point of cryptography is to solve problems. Cryptography solves problems that involve secrecy, authentication, integrity and dishonest people. The characteristics of protocol of cryptography are **[1]**:

- Everyone involved in the protocol must know the protocol and all the steps to follow.

- Everyone involved in the protocol must agree to follow it.
- The protocol must be unambiguous, each step must be well defined and there must be no chance of misunderstanding.
- The protocol must be complete; there must be a specified action for every possible situation.

A cryptographic protocol is a protocol that uses cryptography. It involves some cryptographic algorithms, but generally the goal of the protocol is something beyond simple secrecy. The whole point of using cryptography is a protocol is to prevent or detect eaves dropping and cheating.

Arbitrated Protocols: An *arbitrator* is a disinterested third party trusted to complete a protocol. Arbitrators can help to complete protocols between two mutually distrustful parties.

Adjucated Protocols: Because of the high cost of hiring arbitrators, arbitrated protocols can be sub-divided into two lower level sub-protocols. One is non-arbitrated protocol, and other is an arbitrated sub-protocol, executed only in exceptional circumstances – where there is a dispute. This special type of arbitrator is called an adjucator. An adjucator is also a disinterested and trusted third party. Unlike an arbitrator, it is not directly involved in every protocol.

Self-enforcing Protocols: A self-enforcing protocol is the best type of protocol. No arbitrator is required to complete the protocol. No adjucator is required to resolve the dispute. The protocol is constructed in a way so that there cannot be any dispute. If one of the parties try to cheat, the other party immediately detects the cheating and the protocol stops functioning.

1.3 Cryptographic Techniques

The cryptographic techniques are broadly discussed as followings **[22]**

1.3.1 Key Length: Key length of any cryptosystem may be measured in two ways:

- **Symmetric Key Length:** The security of symmetric cryptosystem is a combined function of two things: the strength of the algorithm and the length of the key. Assuming that the strength of the algorithm is perfect, there is no better way to break the cryptosystem other than trying every possible key in a brute-force attack.
  Calculating the complexity of a brute-force attack is easy. If the key is 8 bit long, there are 2^8 or 256 possible keys, and then it will take 256 attempts to find the correct key with a 50% chance of finding the key after half of the attempt. The security of cryptosystem should rest in the key, not in the details of the algorithm.
- **Public-key Length:** Public-key cryptography uses the idea to make a trap-door-one way function. Actually that is a lie, factoring is conjectured to be a

hard problem. Today's dominant public-key encryption algorithms are based on the difficulty of factoring large numbers that are the product of two large primes. These algorithms are also susceptible to a brute-force attack, but of a different type. Breaking these algorithms does not involve trying every possible key, breaking these algorithms involve trying to factor a large number. If the number is too small then there is no security, if the number is large enough, it has security against all the computing power.

- **How long should a key be?** There is no single answer to this question. To determine how much security you need, you must ask yourself some question. The key length must be such that there is a probability of not more than 1 in 2^{32}.
- **Key Generation:** Every security of an algorithm depends upon the key. If we use cryptographically weak process to generate key, the whole system will become weak. The generation of key may be handled by following means.
    a. **Reduced Key space.**
    b. **Random Keys.**
    c. **Pass Phrases.**
    d. **X9.17 Key Generation.**
    e. **DoD Key Generation.**

1.3.2 Key Exchanges: Although the asymmetric encryption algorithms are more secure than the symmetric types, they are also much slower and it is not feasible to use them to secure large quantities of data, as the consequent increase in transmission times would be excessive. Similarly, although chained mode of symmetric algorithms can process large quantities of plaintext at speed, they do not offer the requisite level of security because the key is a shared secret, that must be exchanged over the insecure medium prior to the transmission of the cipher-text. This paradox may be resolved as follows. A random secret, known as the *Session Key*, is generated and an asymmetric cipher secures this small piece of data for exchange over the Internet. A fast symmetric cipher then uses the *Session Key*, known only to the two security peers, to encrypt their exchanges of bulk data.

1.4 Objective of Work

The prime objective of this research work is to identify the demerits of the existing algorithms for testing the vulnerabilities of the encryption algorithms, especially S-boxes, and suggest improvement(s) or propose totally new test(s) for the same. To summarize, following are the main areas of investigation:

i. To study the already existing testing algorithms and analyze their relative strengths and weaknesses, and also to identify areas where there is scope of improvement.
ii. To suggest modifications, if possible, wherever the scope for improvement have been identified, so as to increase the strength of the underlying test.
iii. To propose new test wherever there is a scope.

1.4.1 Studying Block Ciphers: The Significance

Some public key algorithms such as RSA are capable of encrypting a block of as many bits as the size of the modulus - commonly 1024, 2048, or 4096 bits, dependent on the key. The problem is that most public key algorithms require a very large amount of CPU cycles to encrypt one block of data. In the case of RSA, the larger the modulus (the block size), the greater CPU cycles are required. RSA can take thousands or millions of times as many CPU cycles as a block cipher to encrypt the same amount of data. The slowdown is so significant that public key cryptography is often posed as the limiting factor of a system such as a web server. If encrypting every block of data requires that amount of CPU, the computer requirements for encrypting a stream of data would be prohibitive.

In comparison, symmetric key block ciphers are much more efficient in terms of speed. Given the amount of CPU it takes to encrypt 512 bytes of data with RSA, a symmetric block cipher such as AES would encrypt megabytes of data. The problem with symmetric algorithms is that of storing the keys securely, and the difficulty of exchanging keys with other people without the risk of interception. The public key algorithm is used only one time to encrypt a symmetric algorithm's key and the symmetric algorithm is then used to encrypt the data. The performance problem of public key cryptography suffers only once to exchange the keys, and the volumes of data are efficiently encrypted with a symmetric block cipher. Reasons for studying block ciphers may be summarized as:

- One Time Pads are believed to be the most secure cipher till date as long as the pad is used to encrypt to a single message. One Time Pads which are special purpose Stream Cipher are practically implemented less often as because the length of the key is as long as the message, but still it is considered to the most secure cipher. Block Ciphers may be implemented in a way to realize the powers of OTP.
- Though Asymmetric Ciphers are believed to be much stronger as compared to the Block Ciphers, they are mathematically very intense, which is the reason behind them being inherently slower as compared to the Block Ciphers, so much so that Asymmetric Ciphers are primarily used for key exchanges and not for actual data encryption [1].
- Kerckhoffs stated that for a secure cipher, "*it's key must be communicable and retainable without the help of written notes, and changeable and modifiable at the will of the correspondents"* [1]. But in case of Public Key Cryptography, the keys are of much greater length as compared to Block Ciphers. A comparison between the length of the keys required to achieve same level of security is listed in Table 1.1:

Security Level (in bits)	Asymmetric Ciphers (RSA, Elgamal)	Block Ciphers (RSA, 3DES)
80	1024	80
128	3072	128
192	7680	192
256	15380	256

Table 1.1: Security Levels of Various Ciphers

From the above discussion, it is justified why Block Ciphers are considered as the most fundamental building blocks for any modern cryptosystem, and, that is specifically the reason why the security of block ciphers is of great interest.

When a block cipher is used in a given mode of operation, the resulting algorithm should ideally be about as secure as the block cipher itself. ECB (*discussed in Sec. 1.4.4*) emphatically lacks this property: regardless of how secure the underlying block cipher is, ECB mode can easily be attacked. On the other hand, CBC mode can be proven to be secure under the assumption that the underlying block cipher is likewise secure. However, making statements like this require formal mathematical definitions for what it means for an encryption algorithm or a block cipher to "be secure". This section describes two common notions for what properties a block cipher should have. Each corresponds to a mathematical model that can be used to prove properties of higher level algorithms, such as CBC.

This general approach to cryptography---proving higher-level algorithms (such as CBC) are secure under explicitly stated assumptions regarding their components (such as a block cipher) --- is known as *provable security*.

1.4.2 Stream Cipher vs. Block Cipher

While both are symmetric ciphers, stream ciphers are based on generating an "infinite" cryptographic key stream, and using that to encrypt one bit or byte at a time (similar to the one-time pad), whereas block ciphers work on larger chunks of data (i.e. blocks) at a time, often combining blocks for additional security (e.g. AES in CBC mode).

- Stream ciphers are more difficult to implement correctly, and prone to weaknesses based on usage - since the principles are similar to one-time pad, the key stream has very strict requirements. On the other hand, that is usually the tricky part, they can be offloaded to external box.
- Because block ciphers encrypt a whole block at a time (and furthermore have "feedback" modes which are most recommended), they are more susceptible to noise in transmission, that is, if someone messes up one part, the rest of the data that helps to protect files and data in any web server, is provably unrecoverable. Whereas in stream ciphers there are bytes that are individually encrypted with no connection to other chunks of data (in most ciphers/modes), and often have support for interruptions on the line.

- Stream ciphers do not provide integrity protection or authentication also, but some block ciphers (depending on mode) can provide integrity protection, in addition to confidentiality.
- Because of all the above, stream ciphers are usually best for cases where the amount of data is either unknown, or continuous - such as network streams. Block ciphers, on the other hand are more useful when the amount of data is pre-known - such as a file, data fields, or request/response protocols, such as HTTP where the length of the total message is already known from the beginning and also used in most of the hand hold devices.

1.4.3 Some Standard Cryptographic Algorithms

In this section a few popular algorithms are discussed from different perspectives, clearly indicating the evolution in minimizing the chance of breaking ciphers. After all, these algorithms laid the foundation for efficient ciphers. Some of these algorithms have been even used before the advent of the computers.

Data Encryption Standard (DES): Up until recently, the main standard for encrypting data was a symmetric algorithm known as the *Data Encryption Standard (DES)* **[1]**. However, this has now been replaced by a new standard known as the *Advanced Encryption Standard (AES)*.

DES is a 64 bit block cipher which means that it encrypts data 64 bit at a time. This contrasted to a stream cipher in which only one bit at a time is encrypted. DES was the result of research project set up by IBM Corporation in the year of late 1960's which resulted in a cipher known as *LUCIFER*.

DES of course is not only symmetric cipher. There are many others, each with varying levels of complexity. Such ciphers include: IDEA, RC4, RC5, RC6 etc.

DES is based on a cipher known as Feistal block cipher. It consists of a number of round where each round contains bit shuffling, non-linear substitution (*S-box*) and *exclusive OR* operation. DES accepts two inputs – the plaintext to be encrypted and the secret key.

Advanced Encryption Standard (AES): All of the cryptographic algorithms discussed earlier have some problem. The earlier cipher can be broken with ease on modern computation system [1]. The DES algorithm was broken in 1998 using a system that cost about $250,000. Triple DES has three times, as many rounds as DES and is correspondingly slower. Moreover the 64 bit block size of triple DES & DES is not very efficient and is questionable when it comes to security. Then the requirement for any brand new algorithm was the resistant to all known attacks. The National Institute of Standards and Technology, United States (NIST) wanted to help in the creation of a new standard.

Like DES, AES is a symmetric block cipher. This means that it uses the same key for both encryption and decryption. The block and key can be chosen independently from

128, 160, 192, 224, 224, 256 bits. The AES standard states that the algorithm can only accept a block size of 128 bit and any choice of three keys – 128, 192, 256 bits. As well as these differences AES differs from DES is that it is not feistal structure.

1.4.4 Modes of Operation

In cryptography, a mode of operation is an algorithm that uses a block cipher to provide an information service such as confidentiality or authenticity. A block cipher by itself is only suitable for the secure cryptographic transformation one fixed length group of bits called *block*. A mode of operation describes how to repeatedly apply a cipher's single block operation to securely transform data that are larger than a block. Most modes require a unique binary sequence, often called *Initialization Vector (IV)* of each encryption operation **[21]**.

Electronic Codebook (ECB): The simplest of the encryption modes is the *electronic codebook (ECB)* mode. The message is divided into blocks and each block is encrypted separately.

The disadvantage of this method is that identical plaintext blocks are encrypted into identical Ciphertext blocks; thus it does not hide the data pattern well. It does not provide serious message confidentiality and it is not at all recommended for use in cryptographic protocol.

Cipher-block Chaining (CBC): IBM invented the *cipher-block chaining (CBC)* mode of operation in 1976. In CBC mode each block of plaintext is *XOR-ed* with the previous Ciphertext block before being encrypted. This way each Ciphertext block depends on all plaintext blocks processes up to the point. To make each message unique, an initialization vector must be used in the first block. If the first block has index 1, the mathematical formula for CBC encryption is:

$$C_i = E_k(P_i \oplus C_{i-1}), C_0 = IV$$

While the mathematical formula for CBC decryption is:

$$P_i = D_k(C_i) \oplus C_{i-1}, C_0 = IV$$

CBC has been the most commonly used mode of operation.

Propagating Cipher-block Chaining (PCBC): The propagating or plaintext cipher-block chaining mode was designed to cause small changes in the Ciphertext to propagate indefinitely when decrypting, as well as when encrypting. Formulas for encryption and decryption algorithms are as follows:

$$C_i = E_k(P_i \oplus P_{i-1} \oplus C_{i-1}), P_0 \oplus C_0 = IV$$
$$P_i = D_k(C_i \oplus P_{i-1} \oplus C_{i-1}), P_0 \oplus C_0 = IV$$

Cipher Feedback (CFB): The cipher feedback (CFB) mode, a close relative of CBC, is a mode block cipher. Operation is very similar; in particular, CFB decryption is almost identical to CBC encryption performed in reverse.

$$C_i = E_k(C_{i-1}) \oplus P_i$$
$$P_i = E_k(C_{i-1}) \oplus C_i$$
$$C_0 = IV$$

Output Feedback (OFB): The output feedback (OFB) mode makes a block cipher into a synchronous stream cipher. It generates key stream blocks, which are then *XORed* with the plaintext blocks to get the Ciphertext. Just as with other stream ciphers, flipping a bit in the Ciphertext produce flipped bit in the plaintext at the same location. This property allows many error correcting codes to function normally even when applied before encryption.

$$C_j = P_j \oplus O_j$$
$$P_j = C_j \oplus O_j$$
$$O_j = E_k(I_j)$$
$$I_j = O_{j-1}, I_0 = IV$$

Counter (CTR): Like OFB counter mode turns a block cipher into a stream cipher. It generates the next key stream block by encrypting successive values of a counter. CTR mode is widely accepted, and the problems resulting out of the input function are recognized as weaknesses of the underlying block cipher. CTR mode has similar characteristics to CFB, but also allows a random access property during decryption.

1.5 Review of Existing Works

Going with one of the prime objectives of this research work, i.e., findings the merits and demerits of existing and well accepted cryptanalytic models, a narrative review has been performed. During the review work of related scholarly paper, two key feature of literature review has been followed, that are, review article and systematic review. The thorough review of scholarly papers including the subject of this research topic, helped to find the theoretical and methodological knowledge and contribution to this research topic.

Cristof Paar et. al. has defined **[1]** that two very important security principles for block ciphers are diffusion and confusion. *Diffusion* is an encryption operation where the influence of one plaintext symbol is spread over many ciphertext symbols with the goal of hiding statistical properties of the plaintext. A simple diffusion element is the bit permutation, which is used frequently within DES. On the other hand, *confusion* is an encryption operation where the relationship between key and ciphertext is obscured. Today, a common element for achieving confusion is substitution which found is in both DES and AES. Ciphers which only perform confusion or diffusion are not secure. By using the both operation, a strong cipher can be built and such ciphers are known as product ciphers.

In the research paper titled "On the statistical testing of Block Cipher" **[2]**, it was shown that how a cryptanalyst can use algorithms of a certain kind to attack block cipher and it has been established when a cryptanalyst cannot break the given block cipher. There are two basic problems that a cryptanalyst could attempt to solve and if cryptanalyst cannot solve at least one of these for a given block cipher, they cannot break this block cipher. These two basic problems are:

a. To find an algorithm that is distinguishing for given block cipher.
b. To find an algorithm that is key-subset distinguishing for a given block cipher and for a given decomposition of the key space.

In statistical hypothesis testing, probabilities will be unknown is almost a universal assumption.

In Kerckhoffs's principle, it is assumed that the cryptanalyst knows the entire mechanism of encipherment, except for the value of the secret key. During the course of attack, some questions may arise in front of cryptanalyst that for chosen plaintext block what will be the corresponding ciphertext and vice versa. For all question, the cryptanalyst may try to solve in the following way:

a. *Ciphertext block chosen uniformly at random for decryption.*
b. *Plaintext block chosen uniformly at random for encryption.*
c. *Finding of additional entry in function table.*
d. *Find the secret key*

An invertible function f should be analyzed to solve the above mentioned 4 problems with any black box device that can compute an invertible function f and its inverse f^{-1}. The deterministic algorithms help to entry in the function table of f and additionally deterministic algorithm has access to random table that provides all the 'randomness' in the probabilistic algorithm. A random table can be loaded with a random string R chosen according to a specified probability distribution P_R. A probabilistic algorithm for analyzing an invertible function can be applied to a randomly chosen encryption function of a block cipher e where random variable F and random string R are the inputs and random variable A is the output. The encryption and decryption time do not change against the probabilistic algorithm for analyzing an invertible function. So, as concluded, the encryption and decryption time may be neglected and slow block cipher may have no advantage over a fast block cipher.

A cryptanalyst cannot break the block cipher e if no computationally feasible probabilistic algorithm for analyzing an invertible function is known which solves for the block cipher e the problem of:

- decrypting a ciphertext block chosen uniformly at random without asking the black box to encrypt it,
- finding an additional entry in the function table of the encryption function,
- finding the secret key.

The cryptanalyst can break the block cipher e if he knows a computationally feasible probabilistic algorithm for analyzing an invertible function that solves the block cipher e at least for one of these three problems.

Statistical testing of block ciphers is intended to provide tests that are capable of analyzing any practical block cipher, no matter what the internal structure of the block cipher may be. Therefore such tests should analyze a block cipher based only on the input-output-behavior for its bivariate function *e*.

A cryptanalyst can use statistical testing of a block cipher as a first step towards breaking a block cipher. After using several tests on block cipher, if some of these tests show a non-ideal behavior of the block cipher to see what caused the non-ideal behavior. This might give him ideas how he could break the block cipher. Statistical testing helps the cryptographer to ensure that the designed block cipher that does not have any weaknesses.

Several models have been proposed including the "*model for a probabilistic algorithm for extracting a feature from a sequence of invertible functions*", "*a model for independent execution of a probabilistic algorithm for extracting a feature from a sequence of invertible function and analysis of extracted features for randomly chosen encryption functions of a block cipher e*", "*a model for statistical testing of a block cipher e*".

The block cipher e being tested with block length N and key space Z_e, and the block cipher $e^{\perp}$ with block length N and key space $Z_{e^{\perp}} = Z_e$, are duals if the encryption function $e_{z^{\perp}}$ of the block cipher $e^{\perp}$ is identical to the decryption $e_{z^{\perp}}$ of the block cipher e for every secret key Z in the key space Z_e.

The derivation of block cipher can help to analyze the models that have been proposed. Those models show that instead of looking directly for a probabilistic algorithm for analyzing an invertible function that is key-subset distinguishing for the given block cipher *e*, it is preferred to look for a probabilistic algorithm for analyzing an invertible function that is distinguishing for some of the reduced-key-space versions of the given block cipher *e*.

Knudsen and Mathiassen **[3]** have considered iterated ciphers and their resistance against linear and differential cryptanalysis. It is shown by experiments that cipher with complex key schedules resists both the attack better than ciphers with more straightforward key schedules. It is presented in experiment to illustrate that some iterated ciphers with very simple key schedules will never reach this uniform distribution. It is also shown that cipher with well-designed, complex key schedules reach the uniform distribution faster using fewer rounds than ciphers with poorly designed key schedules.

The author believes that there exists cipher for which the differential of the highest probability for one fixed key is also the differential of the highest probability for any other key and it shown as a side result. The experimental results showed that the

uniform distribution is reached faster for the 10-bit and 12-bit block ciphers than for the 8-bit block ciphers. A good and complex key schedule therefore help to make a cipher more resistant to differential and linear attack.

According to W.S. Forsyth and R. Safavi-Naini **[4]**, in a ciphertext attack it is always possible to test every possible key. This is called exhaustive key search. For any alphabet of size *N* there are $N! > N^{n/2}$ possible substitutions and hence the size of the substitution space increases exponentially in proportion to the size of plaintext alphabet. An algorithm for finding the affine mapping from plaintext to the ciphertext would include:

- Finding the *N*-gram relative frequencies of the sample ciphertext.
- Matching the frequencies against those of the plaintext language and suggesting a key.
- Verifying the decryption obtained by using the suggested key.

Testing likely keys is a non-trivial test by itself and has traditionally required dictionary search, pattern matching and human assistance to except/reject a cryptogram decrypted under a suggested key.

Forsyth and Safavi-Naini formulate the cryptanalysis of the substitution cipher as a combinatorial optimization problem and use simulated analysis to find the optimal solution which corresponds to the affine mapping used encrypt the plaintext alphabet. This approach appealing as it completely eliminates human intervention and does not require any sophisticated pattern matching technique. It also provides an elegant way of solving substitution cipher which is also promising for block cipher algorithm.

G. Piret and F.-X. Standaert **[5]** showed their concern with the security of block cipher against the linear cryptanalysis and discussed the distance between the so-called practical security approach and the actual theoretical security provided by a given cipher. The comparison has been performed between the linear probability of the best linear characteristic and the actual best linear probability. A test is also done for the key equivalence hypothesis. An experiment to evaluate the relevance of the *use of characteristics for arguing the security of a construction* as defined by Knudsen et.al.**[3]** These experiment highlight another aspect of the practical security approach: if the best linear approximation of a given cipher is key-dependent, it can hardly be exploited by an actual adversary. All these have been discussed based on the (im)possibility to derive practical *design criteria* for block cipher.

Their experiments only considered Substitution Permutation Network (SPN), but similar investigation can be considered on the Feistal ciphers.

In order to evaluate the extent to which the practical security approach is meaningful for actual block ciphers, they first computed following quantities:

$$max_{char} := max_{\Omega} ELCP(\Omega)$$

$$max_{hull} := E_{Ě} max_{a,b} LP(a, b; Ě),$$

for various SPNs and as result the following facts are observed:

- After sufficient number of rounds, the average best approximation of a given cipher only depends on its block size *n* and suggested that the average best linear probability of 16-bit and 12-bit ciphers are 6.30×10^{-4} and 7.44×10^{-3}, respectively.
- The probability of the best characteristic goes on decreasing with the number of rounds *R*.
- The value of $E_{Ě}$ max LP is faster achieved with 8-bit S-box than 4-bit ones.
- Linear probability increases when one more round is added.

A new statistical test for randomness, the *strict avalanche criterion* (SAC) test, is there, together with its result over some well known generators in the literature is given and analyzed by J.C.H. Castro et.al **[6]**. The avalanche effect was originally proposed for s-boxes by Webster and Tavares in 1986 **[7]**.

The SAC was further generalized by R. Forre **[8]**. The SAC act as a generalization of the avalanche effect, but not formulated in concrete terms, in early works in the field of cryptography. The avalanche effect tries to reflect, to some extent, the intuitive idea of high-nonlinearity: a very small difference in the input producing a high change in the output, thus an avalanche of changes. Mathematically:

$$\forall x, y \mid H(x, y) = 1, average\ \left(H\big(F(x),\ F(y)\big)\right) = {}^{n}/_{2}$$

So, if *F* is to have the avalanche effect, the Hamming distance between the outputs of a random input vector and are generated by randomly flipping one of its bits should be, on average ${}^{n}/_{2}$. Forre presented a result obtained with the SAC test over a number of well-known pseudo-random number generators using different lengths, from 8 to 128 bits, and marked the results that have corresponding *p*-values less than 0.01 and thus proved a failure for the generator to pass the test.

Applying tests of randomness to block ciphers the cipher will be viewed as a black box such that the actual algorithm used is unknown, the only information being the block size for both the message and key **[9]**. Plaintext that appears random will generally produce ciphertext which also random. Therefore, the application of randomness measures need to indicate that there is no relationship between the plaintext and ciphertext block, i.e. the plaintext is independent of the ciphertext. In order to test this hypothesis, a large number of blocks of length *n* were examined for randomness. Two different ways to generate such set of blocks are:

- Non-random (patterned) plaintext as input and then corresponding ciphertext are tested for randomness.

- Purely random plaintext blocks are combined with the corresponding ciphertext blocks under bitwise modulo-two addition and then tested for randomness.

The Hamming weight, runs, linear complexity, sequence complexity are derived during randomness measures on whole block.

For randomness measures on block differences, for each ciphertext bit position, the strength of the Boolean function may be investigated by measuring the change in the ciphertext bit when subsets of input bits are complemented. This may be expressed as $F(P,K) \oplus F(P \oplus H_i, K)$where P is randomly chosen from the set of *n*-tuples, K is the constant key and H_i is an n-bit vector having hamming weight of *i*, i.e. $W(H_i) = i$.

The property of SAC may be applied to the complementation of plaintext bits as the *strict plaintext avalanche criterion (SPAC)*. For a fixed key, each bit of the ciphertext block changes with the probability of one half whenever a single bit of plaintext block is complemented. This property is applied to key changes where a block cipher satisfies the *strict key avalanche criterion (SKAC)*.

For the independent measures on sub-blocks, a block cipher algorithm aims to combine the elements of the plaintext and key using confusion and diffusion techniques. The subset of plaintext bit will be concatenated with the subset of ciphertext bits to give a combined subset of $l = p + c$ bits for testing. When l is small the classical test of uniformity is applied and for larger l a new test, involving the *Poisson* approximation to the classical occupancy problem distribution and following tests are introduced:

- Small sub-blocks and uniformity test.
- Large sub-blocks and repetition test.

In the paper titled '*Statistical Analysis of Block Cipher*' **[10]** it is stated that diffusion and confusion are the two important principle of security for block cipher. For diffusion, a little change in plaintext or key should result in massive change to the ciphertext i.e. each ciphertext bit depends on each bit of plaintext. Completeness and avalanche criterion are the measures of diffusion and both have been combined to define strict avalanche criterion (SAC). In SAC a change in a single bit of plaintext result in the change of each output with probability $1/2$ over all possible key and plaintext combination. Moreover, satisfying SAC property for encryption does not imply that it is satisfied for decryption.

The paper also enlists few tests under Distinguishing Properties Tests, all of which use fixed key. The Distinguishing Properties tests include:

1) Frequency test that examines the effect of weight of plaintext blocks in the weight of cipher test blocks or vice versa. In this test it is tested whether input with low or high density affect the output weight. The Chi-square test is applied to analyze the result.

2) Run test, here as input large/small numbers of runs are picked and its effect on the number of run in the output are analyzed again using chi-square technique.

3) Alphabetic character test, where, in the frequency of the alphabets (both upper and lower cases) are observed in the ciphertext generated from the plaintext. For this purpose, the ciphertext is broken into group of 8 bits in both overlapping and non-overlapping fashion.

The test detail of SAC states that it is impossible to test the security of the cipher for each key value but it is possible, with a very low probability, to identify weak key classes. The SAC test can be summarized that the hypothesis is plaintext and ciphertext blocks are not correlated when different types of keys in terms of their weight are used. The test for SAC is enough to test the diffusion principle, and can be concluded if a block cipher satisfies SAC, this means that it is also satisfies the completeness and the avalanche criteria.

In the thesis by Sreenivasulu Nagireddy **[11],** it was stated that for the cryptanalysis of DES than any other block cipher, the most practical attack is still a brute-force approach. There are three attacks known that can break the full sixteen rounds of DES with less complexity than a brute-force attack: differential cryptanalysis (DC), linear cryptanalysis (LC) and Devies' attack. However, these attacks are theoretical and are not feasible to mount in practice.

The best attack known on 3key TDES requires around 2^{32} known plaintext, 2^{113} steps, 2^{90} single DES encryption and 2^{88} bit memory **[12]**. This is not practically feasible at present. TDES is slowly disappearing from use, largely replaced by the Advance Encryption Standard (AES). TDES suffers from slow performance in software and AES tends to be around 6 times faster than the earlier.

The AES, also known as Rijndeal is a substitution-permutation network, not feistal network and fast in both software and hardware. So far, the only attack against AES implementations have been side channel attacks. By this attack, it is not possible to attack the underlying cipher, but attack implementations of the cipher on systems which inadvertently leak data.

According to Xuejia Lai and James L. Massey **[13]**, they considered the encryption of pair of distinct plaintext by an r-round iterated cipher, where the round function $Y = f(X, Z)$ is such that, for every round sub key Z, $f(., Z)$ establishes a one-to-one correspondence between the round input X and round output Y. They assumed the difference $\triangle X$ between two plaintext (or two ciphertext) X and X^* is defined as $\triangle X = X \otimes X^{*-1}$, where $\otimes$ denotes a specified group operation on the set of plaintext and X^{*-1} denotes the inverse of the element X^* in the group. The round function $Y = f(X, Z)$ is said to be cryptographically weak if, given a few triples$(\Delta X, Y, Y^*)$, it is feasible to determine the sub key Z.

It has been summarized the basic procedure of differential cryptanalysis attack on an r-round iterated cipher as:

1) Find an (r-1)-round differential (α, β) such that $P(\Delta Y(r-1)) = \beta \mid \Delta X = \alpha$ has maximum or nearly maximum probability.

2) Choose a plaintext X uniformly at random and compute X^* so that the difference $\triangle X$ between X andX^* isα. Submit X and X^* for encryption under the actual key Z. From the resultant ciphertexts $Y(r)$ and $Y^*(r)$, find every possible value of the sub key $Z^{(r)}$ of the last round corresponding to the anticipated difference is $\Delta Y(r-1) = \beta$. Add one to the count of the number of appearances of each such value of the sub key is $Z^{(r)}$.

3) Repeat (2) until one or more values of the sub key $Z^{(r)}$is counted significantly more often than the others. Take this more-often-counted sub key as the cryptanalyst's decision for the actual sub key is $Z^{(r)}$.

It is noted that, in a differential cryptanalysis attack, all the sub keys are fixed and only the plaintext can be randomly chosen.

The terminology "Markov Cipher" is been explained by the theorem as:

If an r-round iterated cipher is a Markov cipher and the r-round keys are independent and uniformly random, then the sequence of difference $\Delta \boldsymbol{X} = \Delta \boldsymbol{Y}(\mathbf{0})$, $\Delta \boldsymbol{Y}(\mathbf{1}), \ldots., \Delta \boldsymbol{Y}(\boldsymbol{r})$ *is a homogeneous Markov chain. Moreover, this Markov chain is stationary and* $\Delta \boldsymbol{X}$ *is uniformly distributed over the non- neutral elements of the group.*

In the research article "Cryptographic Randomness Testing of Block Cipher and Hash Function" **[14]** another version of SAC test was proposed along with three other tests namely 'linear span test', 'correlation test' and 'coverage test'. In this version of SAC test, a SAC matrix is prepared which is similar to the matrix prepared in **[10]** but, in this case unlike **[10]** 2^{20} randomly chosen plaintext are used instead of an arbitrary m number of plaintext. Since 2^{20} numbers of randomly chosen plaintext are used, each of the entries in SAC matrix is expected to have a value close to 2^{19} in order to have a probability of ½. After preparing the matrix, chi-square goodness of fit test is applied to evaluate the distribution of the values of the entire matrix, if the matrix produces a $p - value$ less than 0.01, then it is considered as non-random. Next to catch any correlation between a particular input bit and a particular out bit entry outside a particular range $2^{20} - 5009, 2^{20} + 5009$ are flagged and the test is applied once again. If the flagged entries deviate from the expected value once more significantly, it indicates that a specific input bit and a specific bit are correlated, which in term indicate a weakness in the underlying cipher.

In the linear span test, non-linearity of the underlying block cipher is tested by producing an input set of size $m = 2^t$ obtained from t independent plaintext. After obtaining the input set, a $m \times m$ matrix is obtained from the corresponding ciphertext

and the rank of matrix is calculated and compared with the rank of a random binary matrix. After rank is being obtained, the corresponding binary value is incremented by one. This process is repeated as many times as possible and the resulting binary values are then put through the chi-square goodness of fit test to produce $p - value$. If the $p - value$ is less than 0.01, it indicates a non-random mapping.

A block cipher should produce random looking outputs, the outputs randomness is generally evaluated using a pseudo-random number generator (PRNG). One of the most commonly used randomness test suites for PRNG is the Diehard test suite **[15]** which was considered to be the best test suite for PRNG until the authors came up with the concept in "Some difficult-to-pass tests for randomness" **[16]**. On the other hand, another statistical test suite was designed and used by NIST **[17]** in order to evaluate the randomness criteria of the AES finalist. Although the NIST test suite was designed to test the randomness characteristics of the block cipher, in fact it is a general purpose test suite for evaluating the randomness of a binary string which may come from any source, the NIST test suite describes 15 statistical tests and it also describes the implementation details for each of these tests. Generally there are two issues which are needed to be addressed while evaluating a block cipher. The first issue which needs to be addressed is that block ciphers are not PRNGs by itself and they do not generate arbitrary long binary strings. In this regard, NIST test suite is not well suited for evaluating block ciphers, it would be better if some statistical test directly evaluate the randomness of the block cipher mappings, without aiming it to turn the block cipher to behave as a PRNG.

The second issue with most statistical tests is that they take a frequentist approach to statistical testing. In other words, it can be said that while analyzing a binary string, each test computes a statistic and a $p - value$. Now even if the binary string being analyzed is actually random, if the $p - value$ falls below a threshold, it is assumed that the string is not random. It is assumed if the underlying PRNG is random, $p - value$ should be uniformly distributed between 0 to 1, so, the NIST test suite applies a *second level* statistical test to the $p - value$ for each *first level* test to determine whether the test $p - values$ are randomly distributed. From the $p - values$, the analyst has to decide whether the binary string and the underlying block cipher that produced the string holds good randomness property or not. The frequentist approach does not specify a procedure for combining multiple $p - values$ into a single number that yields an overall random/non random decision.

1.6 Justification of Proposed Research Work

After a thorough study of existing works of various cryptanalysis, it is found that there is very limited work of cryptanalysis has been done towards the strength analysis of S-boxes of different cryptographic algorithms.

In this proposed research work, S-boxes are analyzed with some novel approaches to measure the strength against the attacks. Different type of statistical methods has been used with the proposed algorithms to establish the conclusion.

The major aim of the proposed research work is to establish a testing suite for the existing S-boxes with the help of novel algorithms.

1.7 Cryptographic Standards

There are number of standards related to cryptography are available. Standard algorithms and protocols provide a lot of help for research study and attract to large amount of cryptanalysis. Some of the well known standards are being reviewed during the preparation of this research work.

1.6.1 Encryption Standards

1.6.1.1 Data Encryption Standard

Data Encryption Standard (DES) is a symmetric algorithm for encrypting data of recent time. But DES now has been replacing by another well known new standard algorithm called Advance Encryption Standard (AES) **[43]**. DES is the result of changes, made by IBM, requested by NBS (now NIST), on popular cipher LUCIFER. The notable change in DES on LUCIFER is the key size, which is reduced from 128-bits to 56-bits. DES actually accepts 64-bits key as input and remaining 8-bits for parity checking. Biham and Shamir **[44]** publicly discovered the concept of S-boxes which is appeared secure against an attack called Differential Cryptanalysis.

DES is based on Feistal Block Cipher, Developed by IBM researcher Horst Feistal. It consists of a number of rounds where each round contains bit-shuffling, non-linear substitutions (S-boxes) and XOR operations. As DES is 64-bits block cipher, if the number of bits in the message is not evenly divided by 64 then the last block will be padded. To increase the difficulty of cryptanalysis, multiple permutations and substitutions are there in DES. The sequence of events that occur during an encryption operation using DES is shown in Figure 1.1.

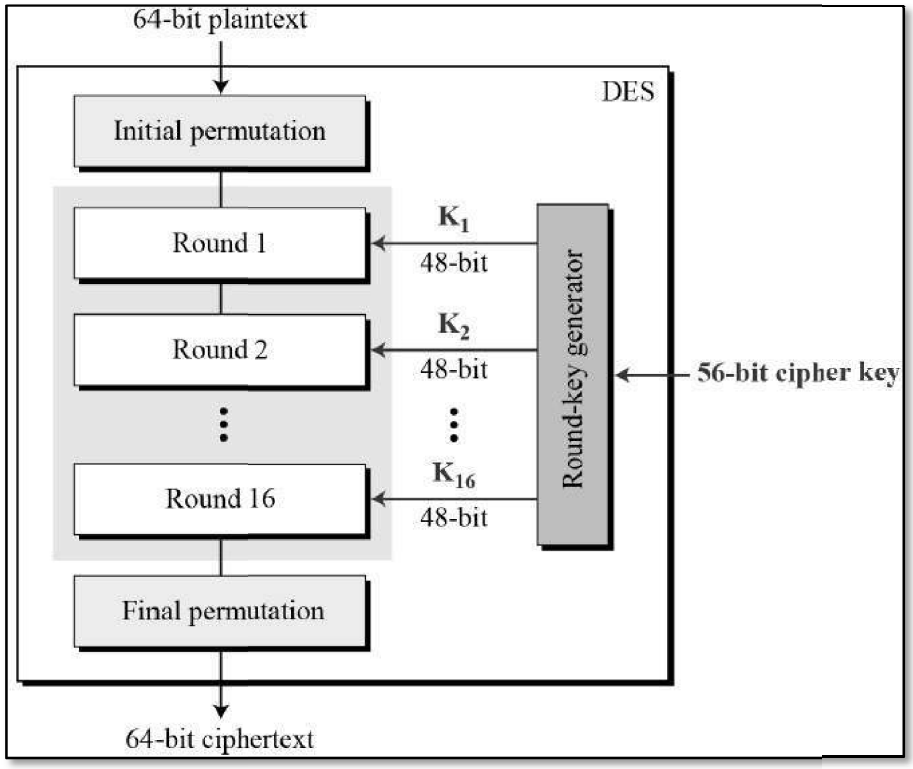

Figure 1.1: Flow Diagram of DES for Encryption Data

The Figure 1.2 shows the initial and final permutations (P-boxes) of DES. Each permutation takes 64-bit input and permutes them according to predefined rule. The permutation rules of these P-boxes are shown in Table 1.2.

The S-boxes of DES does the real confusion. DES uses 8 S-boxes, each with 6-bit input and a 4-bit output.

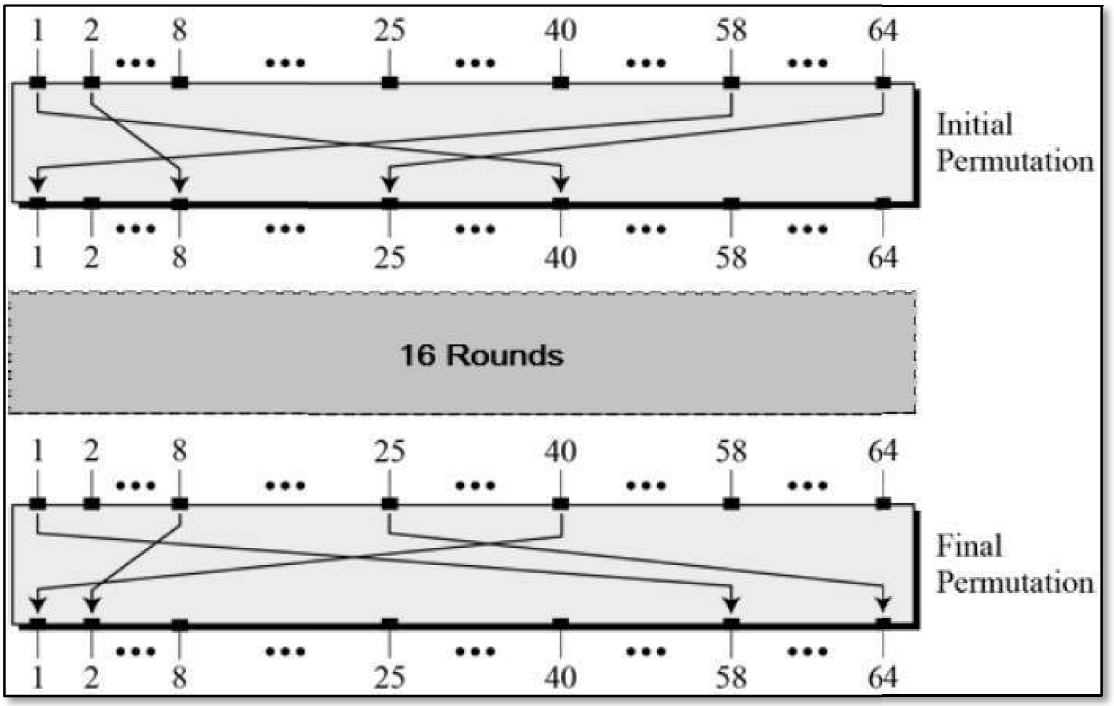

Figure 1.2: Initial and final permutation steps in DES

Initial Permutation								Final Permutation							
58	50	42	34	26	18	10	02	40	08	48	16	56	24	64	32
60	52	44	36	28	20	12	04	39	07	47	15	55	23	63	31
62	54	46	38	30	22	14	06	38	06	46	14	54	22	62	30
64	56	48	40	32	24	16	08	37	05	45	13	53	21	61	29
57	49	41	33	25	17	09	01	36	04	44	12	52	20	60	28
59	51	43	35	27	19	11	03	35	03	43	11	51	19	59	27
61	53	45	37	29	21	13	05	34	02	42	10	50	18	58	26
63	55	47	39	31	23	15	07	33	01	41	09	49	17	57	25

Table 1.2. Initial and Final Permutation Table

The decryption of DES is same as encryption process; it uses the ciphertext as input to DES algorithm and use the key K_i in reverse order.

1.6.1.2 Triple DES

The modified scheme Triple DES came in practice as a result of discomfort of users against the exhaustive key searching in DES. There are two variants of Triple DES known as 3-key Triple DES (3TDES) and 2-key DES (2TDES).

1. 3-Key Triple DES (3TDES): The key (K) of 3TDES is consists of three different DES keys K_1, K_2, K_3. The key length of K is $3 \times 56 = 168$ bits.
2. 2-Key Triple DES (2TDES): The 2TDES scheme is mostly same as 3TDES except that K_3 is replaced by K_1, therefore the key length for 2TDES is $2 \times 56 = 112$ bits.

1.6.1.3 Advanced Encryption Standard

The 64-bit DES and triple DES is not very efficient and questionable when it comes to security. NIST chose the algorithm known as Rijndeal [44] which is then named as AES as standard block cipher cryptographic model. The salient features of AES are as follows:

- AES is a block cipher with a block length of 128 bits.
- AES allows 3 different key lengths: 128, 192 and 256 bits.
- Encryption consists of 10 rounds for 128 bits key, 12 rounds for 192 bits key and 14 rounds for 256 bits key.
- Except the last round in each case, all other rounds are identical.
- Each round of processing includes one single-byte based substitution step, a row-wise permutation step, a column-wise mixing step, and the addition of the round key. The order in which these four steps are executed is different for encryption and decryption.
- Like DES, AES is an iterated block cipher in which plaintext is subject to multiple rounds of processing, with each round applying the same overall transformation function to the incoming block.
- Unlike DES, AES is an example of key-alternating block ciphers. In such ciphers, each round first applies a diffusion-achieving transformation operation — which may be a combination of linear and nonlinear steps — to the entire incoming block, which is then followed by the application of the round key to the entire block. As you'll recall, DES is based on the Feistel structure in which, for each round, one-half of the block passes through un-changed and the other half goes through a transformation that depends on the S-boxes and the round key. Key alternating ciphers lend themselves well to theoretical analysis of the security of the ciphers.

1.8 Action Plan of the Research Work

Analysis of block ciphers has been done efficiently, encrypted with well-known encryption methods like DES and AES, till date, and reviewed minutely in *1.5*. There are lots of excellent research works on the:

a) Statistical testing on block ciphers,

b) On the role of key schedules in attack on iterated ciphers,

c) Automated cryptanalysis on substitution cipher,

d) SAC randomness test including SPAC and SKAC,

e) A pattern recognition approach to block cipher identification, and

f) Many distinct ways of randomness tests.

All statistical approach has concentrated on computing $p - value$ and its randomness. This approach has been identified as one of the scopes of this research work.

The research work looks forward to make it feasible to cover both linear and differential cryptanalysis approaches. The bit level block cipher diffusion and confusion analysis are the key areas to be covered in this research work. The core intension of this research work is to establish a standard algorithmic test suite on block ciphers to test most of the internationally recognized encryption methods. Both statistical and randomness tests are taken into consideration to develop the suite.