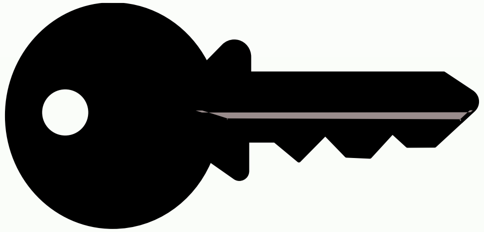
FEISTEL CIPHER (FC)

AND

SUBSTITUTION-PERMUTATION NETWORK (SPN)

Introduction to Feistel Cipher (1/2)



- *Feistel Cipher* (also known as Luby–Rackoff block cipher), developed by Horst Feistel in the early 1970s, is a symmetric structure used in the construction of block ciphers. It is widely used in many cryptographic application such as electronic payment systems, secure communication protocols, and data storage.

- This technique uses a combination of substitution and permutation operations to encrypt plaintext into cipher text. The plaintext is divided into two halves, which are then subjected to a series of rounds of encryption and de-encryption using a secret key.

Introduction to Feistel Cipher (2/2)

Feistel proposed the use of a cipher that alternates substitutions and permutations.

This is a practical application of producing a cipher text that alternates between *confusion* and *diffusion.*

Substitution: Each plaintext element or group of elements is uniquely replaced by a corresponding ciphertext element or group of elements.

Permutation: A sequence of plaintext elements is replaced by a permutation of that sequence. That is, no elements are added or deleted or replaced in the sequence, rather the order in which the elements appear in the sequence is changed.

Diffusion & Confusion (1/2)

These terms were introduced by **Claude Shannon**.

He was concerned that based on the normal distribution or other statistical analysis on the ciphertext, an attacker might be able to find out the patterns in them and correlate it to the frequency distribution of the words/phrases used in the plaintext.

This way, the encryption key, or a part of the key might be decoded which can lead to the finding of the original key. This will give the attacker access of the plaintext.

According to him, a *strongly ideal cipher* is one in which the statistical analysis on the ciphertext does not correlate to the original key used.

For achieving this, he suggested two methods for frustrating statistical cryptanalysis: **diffusion** and **confusion**.

Diffusion & Confusion (2/2)

Diffusion

Here, the statistical structure of the plaintext is dissipated into long-range statistics of the ciphertext.

Essentially, it makes each plaintext bit have effect on as many ciphertext bits as possible.

Thus, 1 bit change in the plaintext, makes a significant change on the ciphertext.

The mechanism of diffusion seeks to make the statistical relationship between the plaintext and ciphertext as complex as possible in order to thwart attempts to deduce the key.

E.g. –Transposition/Permutation

Confusion

Confusion seeks to make the relationship between the statistics of the ciphertext and the encryption key as complex as possible, again to thwart attempts to discover the key.

Here the relation between ciphertext and plaintext is obscured.

Thus, even if the attacker can get some handle on the statistics of the ciphertext, there will be no information about the plaintext, encryption key, encryption algorithm.

This is because, the way in which the key is used to produce that ciphertext is so complex as to make it difficult to deduce the key.

E.g. –Substitution (Complex substitution algorithm)

The Feistel Cipher Structure (1/2)

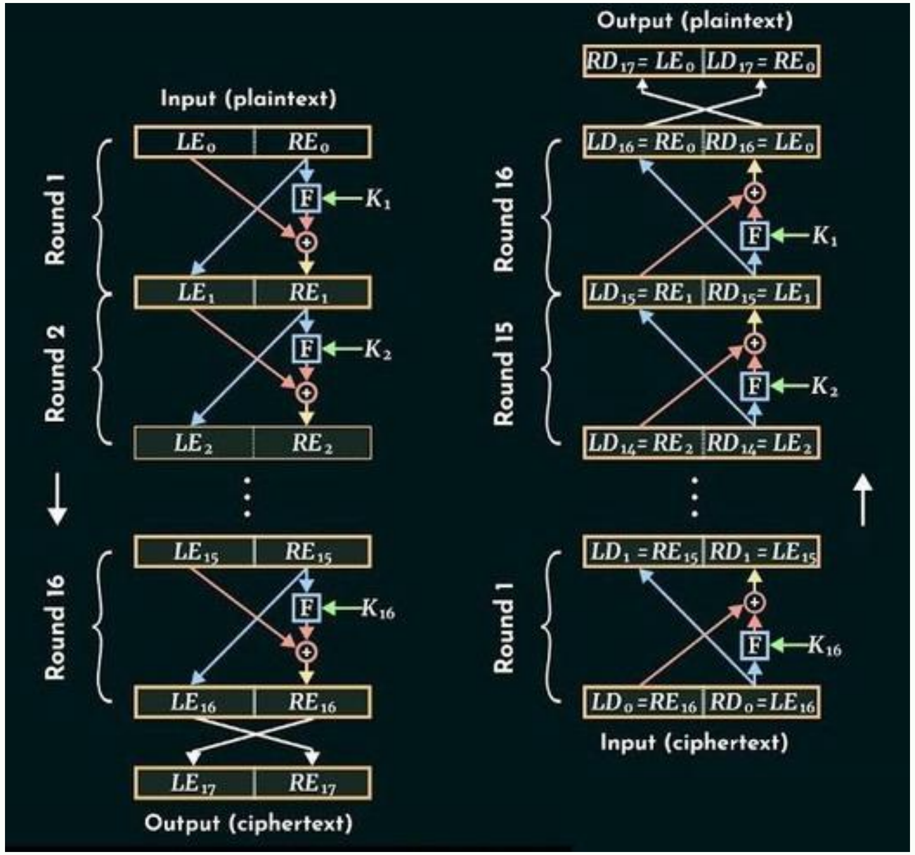


Fig.1 : Feistel Encryption and Decryption (16 rounds)

- The Feistel cipher structure is a *block cipher*.
- It consists of multiple rounds of encryption and decryption. Each round involves the following steps: dividing the input block into two halves, applying a round function to one half using a subkey, and swapping the two halves.
- The round function typically involves a combination of substitution and permutation operations, such as S-boxes and P-boxes. The subkey is derived from the main key using a key schedule algorithm, which generates a set of round keys for each round of encryption and decryption.

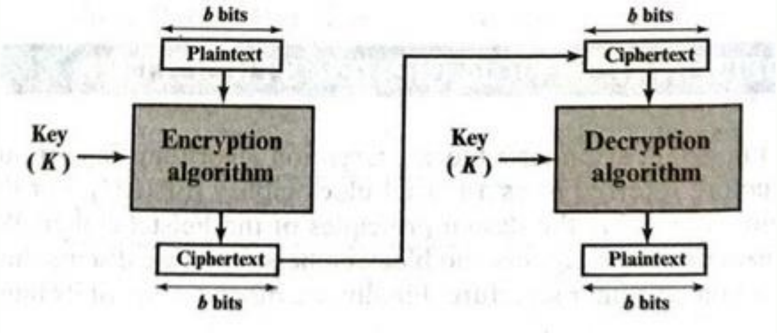
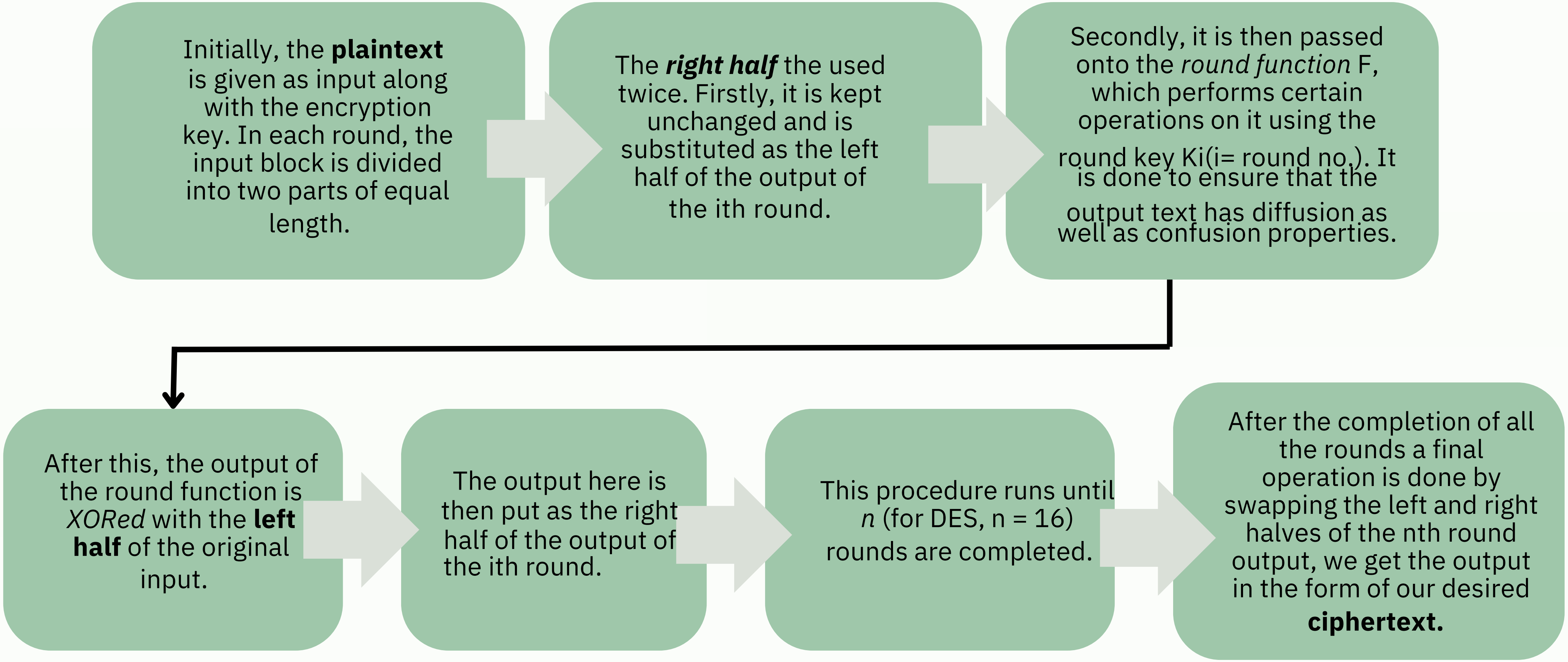


Fig.2 : Block Cipher

Note : A ***block cipher*** *is one in which a block of plaintext is treated as a whole and used to produce a ciphertext block of same length. Generally, a blocksizeof64-bitsor128-bitsisused.*

The Feistel Cipher Structure (2/2)



Initially, the **plaintext** is given as input along with the encryption key. In each round, the input block is divided into two parts of equal length.

The ***right half*** the used twice. Firstly, it is kept unchanged and is substituted as the left half of the output of the ith round.

Secondly, it is then passed onto the *round function* F, which performs certain operations on it using the round key Ki(i= round no.). It is done to ensure that the output text has diffusion as well as confusion properties.

After this, the output of the round function is *XORed* with the **left half** of the original input.

The output here is then put as the right half of the output of the ith round.

This procedure runs until *n* (for DES, n = 16) rounds are completed.

After the completion of all the rounds a final operation is done by swapping the left and right halves of the nth round output, we get the output in the form of our desired **ciphertext.**

Feistel Structure Design Features (1/2)

Block Size : As the Feistel cipher is a *block cipher*, thus group of bits are given as input.
Larger block size means greater security. However, this reduces the encryption/decryption speed for a given algorithm.
For the application in DES, a block size of 64-bits is used.

Key Size : Greater key size means greater security but it may decrease the encryption/decryption speed for a given algorithm.
If the key size is smaller, it will make the data vulnerable to brute-force attacks. It will also reduce the confusion.
As of now, key size of 64-bits or more is the standard with 128 bits being used in most applications.

Number of rounds : Only one round of encryption doesn’t offer adequate security. Thus, we must perform multiple rounds.
The standard number of rounds used is 16.

Feistel Structure Design Features (2/2)

Subkey Generation Algorithm : As, one generated key is used to make round keys for the subsequent rounds, greater complexity in this algorithm will lead to greater difficulty of cryptanalysis.

Round function F : Complex design of the round function also leads to greater difficulty of cryptanalysis.

Fast Software Encryption/Decryption : As, this cipher structure is mostly used inside of certain applications or implemented directly into the hardware, thus we need the encryption/decryption process to be very fast.
This becomes very important as in case of applications we need to transfer the data after performing these operations.
All the improvements/complexities in the algorithm should be implemented keeping in mind the execution time of the algorithm.

Ease of Analysis : If the algorithm is implemented with a clear and concise explanation, it would be easier to analyze it for any vulnerabilities. This in turn would make the algorithm stronger.

Feistel Decryption Algorithm

- The process of decryption with a fiestal cypher is essentially the same as the encryption process.

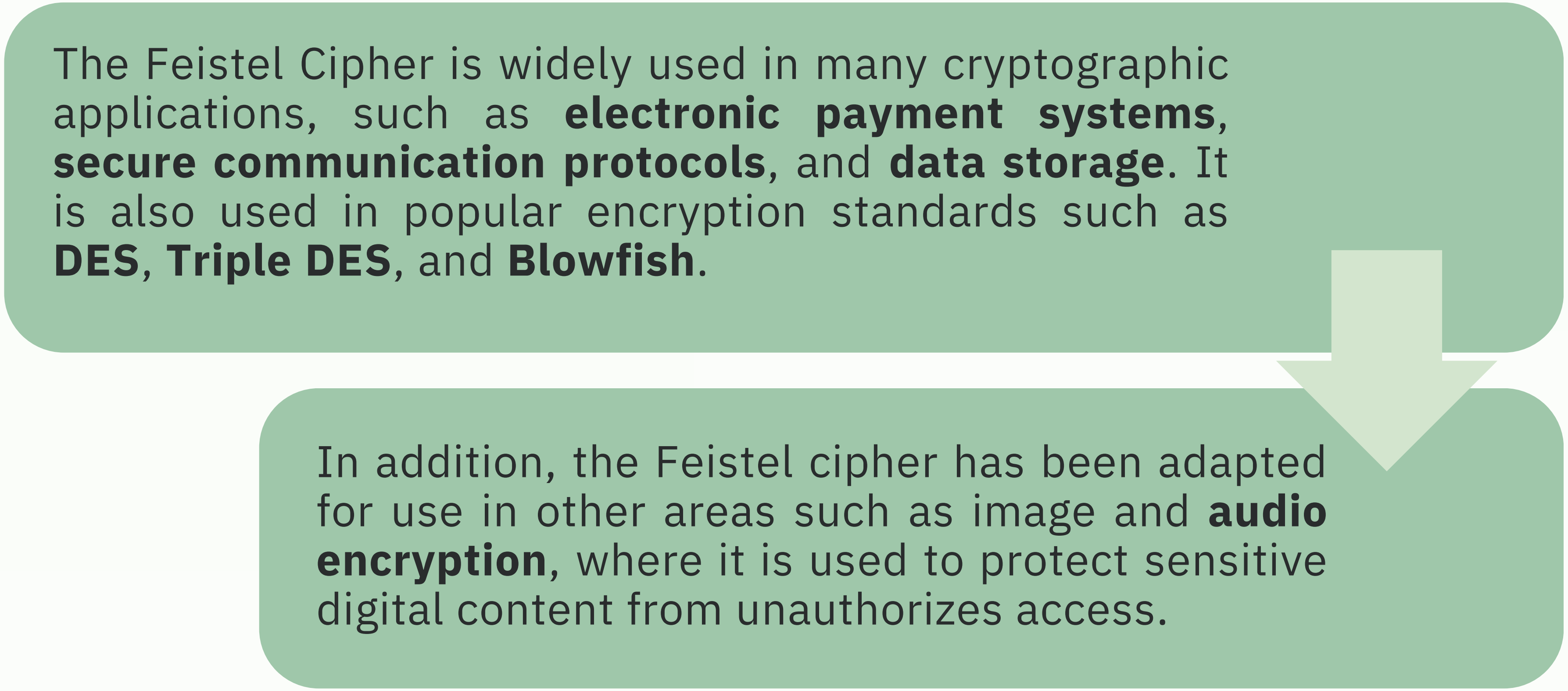
The process of decryption with a Feistel cipher is essentially the same as the encryption process.

- The rule is as follows:

The rule is as follows:

- ❑ Use the ciphertext as input to the algorithm but use the subkeys Ki in reverse order.
- ❑ That is, use Kn in the first round, Kn-1 in the second round,and so on, until K1, is used in the last round.
- ❑ At the end, the left and right halves are swapped giving us the plaintext as output.
- ❑ This is a nice feature, because it means we need not implement two different algorithms; one for encryption and one for decryption.
- ❑ The process can be seen in action in Fig. 1.

Application of Feistel Cipher



The Feistel Cipher is widely used in many cryptographic applications, such as **electronic payment systems**, **secure communication protocols**, and **data storage**. It is also used in popular encryption standards such as **DES**, **Triple DES**, and **Blowfish**.

In addition, the Feistel cipher has been adapted for use in other areas such as image and **audio encryption**, where it is used to protect sensitive digital content from unauthorizes access.

ADVANTAGES

One of the main advantages of Feistel Cipher is its **simplicity** and **efficiency**. The encryption and decryption operations are easy to implement and require only basic arithmetic and logical operations.

It is resistant to differential and linear cryptanalysis attacks.

Feistel ciphers can be *reversed* to decrypt data, even if the round function is not invertible.

It does not rely on substitution boxes, which can cause timing side-channels in software.

Feistel ciphers are a *versatile framework* that can be used to create many encryption algorithms.

DISADVANTAGES

Although the Feistel Cipher is generally considered to be a strong encryption algorithm, it does have some weaknesses. One weakness is its vulnerability to **brute force attacks**, where an attacker tries all possible keys until the correct one is found.

Another weakness is its susceptibility to **side-channel attacks**, where an attacker can gain information about the key or plaintext by observing the physical characteristics of the system, such as power consumption or electromagnetic radiation.

IMPROVEMENTS

- We can use the Feistel structure consisting of multiple rounds of processing of the plaintext.

- In this process we perform XOR operation and Function for both left half and right half.

- This new approach is shown in Fig. 3.

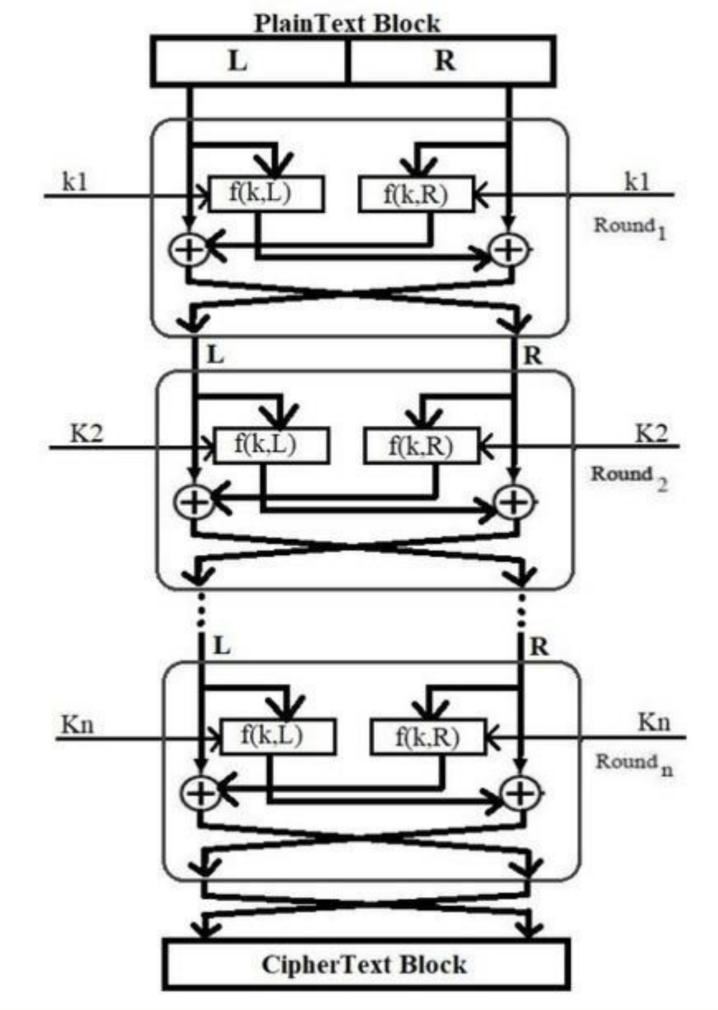


Fig. 3 : Improved Feistel Cipher

CRYPTOGRAPHY

SUBSTITUTION-PERMUTATION NETWORK AND COMPARISON

An in-depth exploration of cryptographic techniques, focusing on the Substitution-Permutation Network and its contrast with the Feistel Cipher.

INTRODUCTION TO SUBSTITUTION-PERMUTATION NETWORK (SPN)

Exploring the significance and functionality of SPNs in cryptography

1 DEFINITION OF SPN

SPN, or Substitution-Permutation Network, is a specific type of block cipher utilized in the field of cryptography. It consists of a series of interconnected mathematical operations that include both substitution (replacing elements) and permutation (rearranging elements). This structured approach enhances security by complicating the relationship between the plaintext and ciphertext.

2 PURPOSE OF SPN

The primary purpose of SPN is to achieve confusion and diffusion within the encryption process. Confusion refers to making the relationship between the key and the ciphertext as complex as possible, while diffusion ensures that a small change in the plaintext results in a significant change in the ciphertext. This dual focus enhances the overall security of the encryption mechanism.

3 EXAMPLE OF SPN

A prominent example of a Substitution-Permutation Network is the Advanced Encryption Standard (AES). AES is widely used for securing data in various applications, including financial transactions and secure communications. Its robust structure and efficiency have made it a standard in modern cryptography.

COMPONENTS OF SPN

Understanding the essential layers of the Substitution-Permutation Network

SUBSTITUTION LAYER

The Substitution Layer is a crucial non-linear component of the SPN structure. It functions by taking input bits or blocks and replacing them with output bits or blocks based on a predefined substitution box (S-box). This non-linearity helps to obscure the relationship between the plaintext and ciphertext, enhancing security.

PERMUTATION LAYER

Following the substitution process, the Permutation Layer plays a significant role in the SPN architecture. It is a linear layer that rearranges the bits, effectively shuffling them to disperse the data throughout the block. This spread of information is vital for increasing the diffusion, making it more difficult for attackers to analyze the data patterns.

KEY MIXING

Key Mixing is an integral process that combines the plaintext with a round key, typically through an XOR operation. This step ensures that the influence of the key is integrated into the data transformation process. By modifying the plaintext with the round key, this layer enhances the overall security and variability of the encryption process.

OPERATION OF SPN

Understanding the Sequential Steps in SPN Encryption

1 INITIAL ROUND KEY ADDITION

The SPN process begins with the addition of the initial round key to the plaintext. This step is critical as it sets the foundation for the encryption operation, ensuring that the data is transformed right from the start. The round key is derived from a master key, which is expanded into multiple round keys for use in subsequent steps.

2 SUBSTITUTION STEP

Following the initial round key addition, the SPN undergoes a series of substitution operations. This involves replacing bits of the input data with other bits according to a predefined substitution table, also known as an S-box. This process introduces non-linearity into the encryption, making it harder for attackers to predict the output.

3 PERMUTATION STEP

After the substitution step, a permutation operation rearranges the bits of the output from the substitution. This diffusion process spreads the influence of individual bits across the output, enhancing security by ensuring that small changes in input yield significantly different outputs.

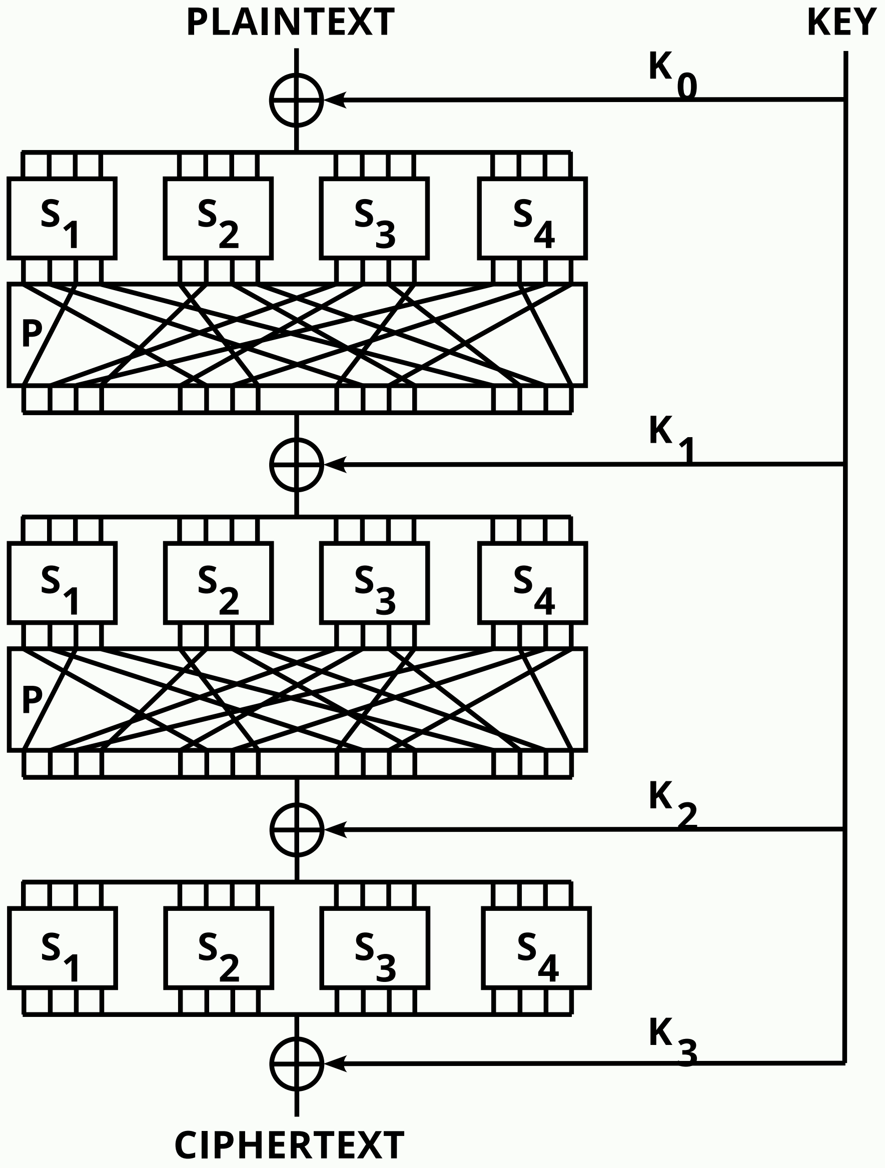
4 KEY ADDITION IN ROUNDS

The process of substitution and permutation is repeated for a fixed number of rounds, each time adding a new round key. This iterative approach strengthens the encryption, as each round adds complexity and obscures the relationship between the plaintext and the ciphertext.

5 FINAL ROUND KEY ADDITION

The encryption process concludes with a final addition of the round key. This step is crucial as it ensures that the last transformation of the data includes the final round key, resulting in the ciphertext. The completeness of this step ensures the security of the encryption scheme.

SKETCH OF A SUBSTITUTION-PERMUTATION NETWORK



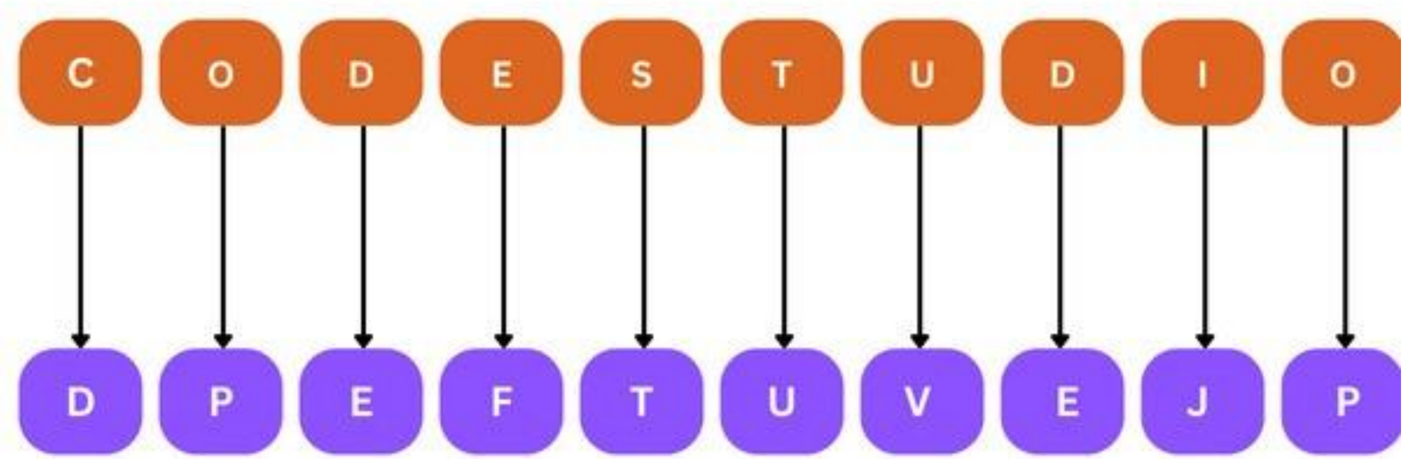
A sketch of a substitution– permutation network with 3 rounds, encrypting a plaintext block of 16 bits into a ciphertext block of 16 bits. The S-boxes are the Si, the P-boxes are the same P, and the round keys are the Ki.

SUBSTITUTION LAYER IN SPN

The substitution layer is a key component of the SPN structure. It operates using S-boxes (substitution boxes) to replace input data with predetermined output data. Each S-box is a lookup table that substitutes a block of input bits (e.g., 4 bits) with a corresponding block of output bits.

SUBSTITUTION

Plain text

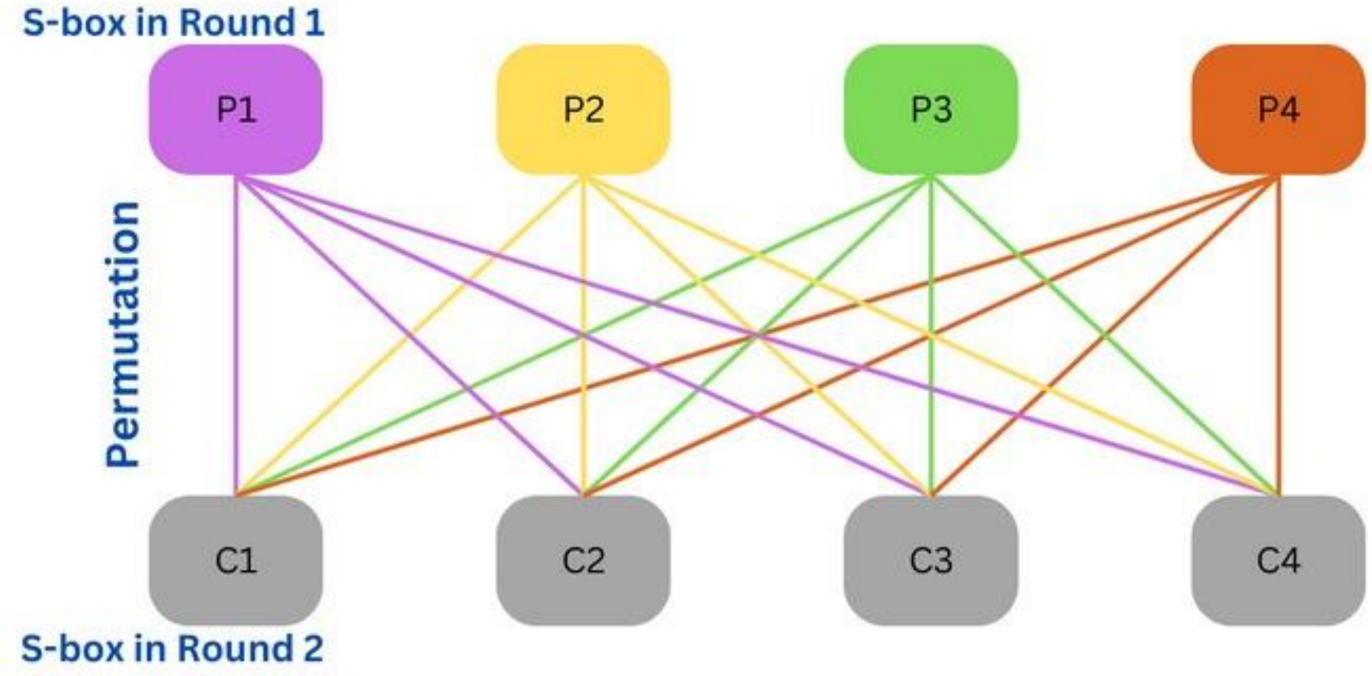


After substitution

PERMUTATION LAYER IN SPN

The permutation layer in SPN reorders (or permutes) the bits of the data to ensure thorough diffusion, spreading the influence of each bit of the input data across the ciphertext.

PERMUTATION



DEFINITION OF KEY SCHEDULING

A method used in block ciphers to derive a series of sub-keys (round keys) from the main secret key. These sub-keys are then used in each encryption or decryption round to introduce complexity and increase security. This ensures that every block of data is encrypted with a unique key, making it difficult for attackers to decrypt without knowing the specific sub-keys.

UNDERSTANDING SPN STRUCTURE AND OVERFLOW

Substitution-Permutation Networks (SPN) are a class of block ciphers widely used in cryptography for secure data encryption. The structure is typically composed of multiple rounds, with each round consisting of substitution and permutation stages.

COMPARISON OF SPN AND FEISTEL CIPHER

A Comprehensive Analysis of Two Cipher Structures

DESIGN STRUCTURE

- SPN: Substitution and permutation operations.
- Feistel: Splitting and processing halves.

ENCRYPTION PROCESS

- SPN: All rounds process data through substitution and permutation.
- Feistel: Only half of the data is processed in each round.

ROUND FUNCTION

- SPN: Uses a combination of S-boxes and P-boxes.
- Feistel: Uses a function that can be any reversible function.

KEY SCHEDULE

- SPN: Requires a single key to generate round keys.
- Feistel: Can use multiple keys throughout the process.

DECRYPTION

- SPN: Requires a separate decryption algorithm.
- Feistel: Decryption is the same as encryption but with reversed subkeys.

SECURITY

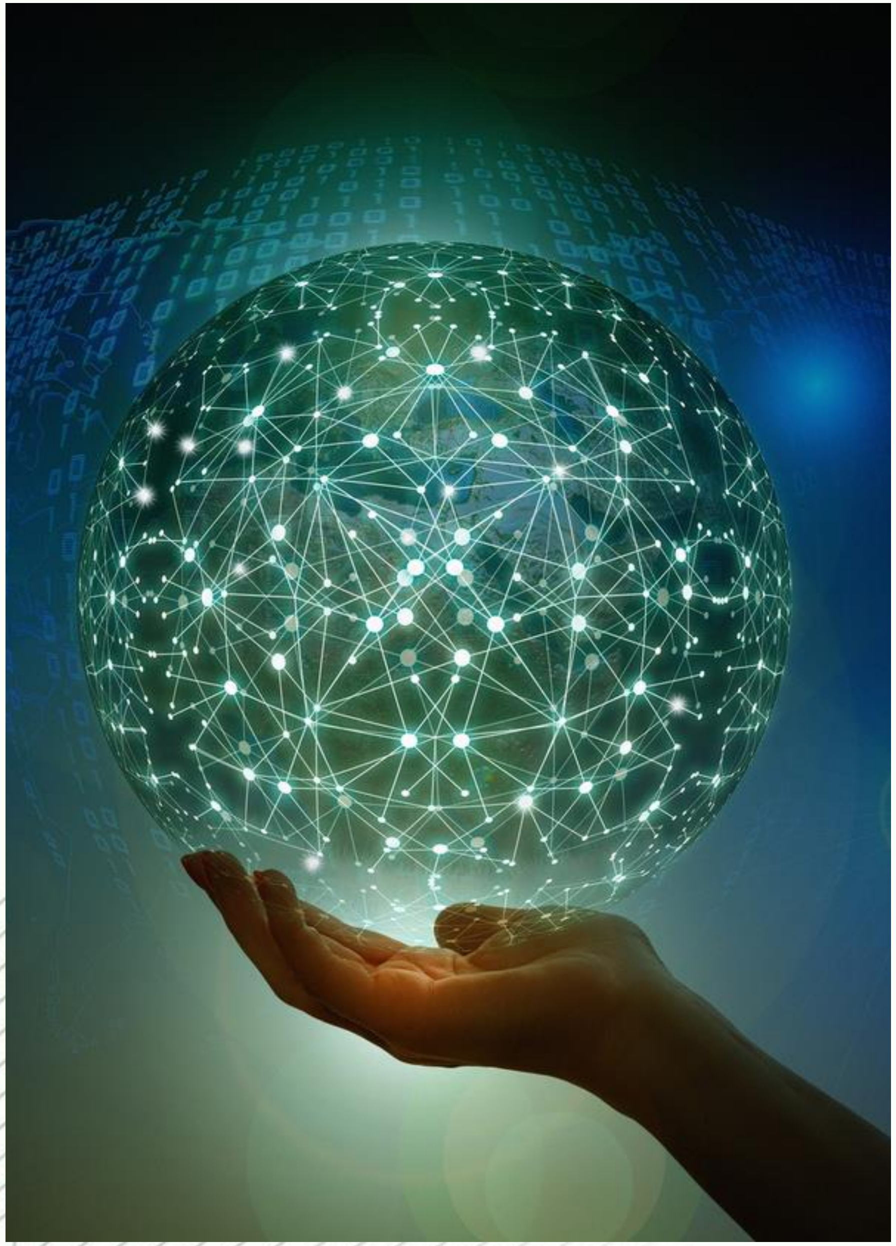
- SPN: Security is dependent on the strength of S-boxes.
- Feistel: Security is based on the complexity of the round function.

PERFORMANCE

- SPN: Generally faster due to parallel processing of rounds.
- Feistel: Performance can vary based on the round function used.

COMPLEXITY

- SPN: Complexity increases with the number of rounds.
- Feistel: Complexity can be adjusted by changing the round function.



EFFICIENCY

Analyzing the Efficiency of Modern Cipher Techniques

1 SPN (SUBSTITUTION-PERMUTATION NETWORK)

SPN is recognized for its speed and efficiency, making it the preferred choice in modern encryption standards like AES (Advanced Encryption Standard). Its structure allows for rapid processing of data, which is essential in applications requiring high throughput and low latency.

2 FEISTEL CIPHER STRUCTURE

Feistel ciphers are characterized by their unique structure that allows for a simplified key management process. They typically require fewer rounds of encryption compared to SPNs, which can make them less resource-intensive while still providing robust security. This efficiency is particularly advantageous in environments with limited computational resources.

SPN: ENHANCED SECURITY

SPNs enhance security by combining substitution and permutation layers to achieve confusion and diffusion. The substitution layer uses S-boxes for non-linearity, while the permutation layer shuffles bits to spread the effect of each input bit. Multiple rounds and a key scheduling algorithm amplify these effects, making SPNs resistant to various cryptanalytic attacks.

Karan

CONCLUSION

Both SPN (Substitution-Permutation Network) and Feistel Ciphers are foundational cryptographic techniques. SPN is characterized by a series of substitution and permutation operations, while Feistel Ciphers use a structure that allows the encryption function to be applied multiple times, enhancing security. Each method has its unique strengths and use cases, making them essential components in the field of cryptography.

REFERENCES

1. Cryptography and Network Security : Principles and Practice, 7e, by William Stallings
2. https://www.youtube.com/watch?v=8l9xAvuGJFo
3. https://www.youtube.com/watch?v=kbROQdcxdTQ
4. https://xiphcyber.com/articles/importance-of-encryption-in-the-workplace
5. An Improved in Feistel Cipher Structure of Encryption by Rajesh Kumar, https://www.ijsr.net/archive/v9i2/SR20223161316.pdf

Thank You