CS406 Project
Decrypting SPN Cipher using Linear Cryptanalysis

Midathana Pranay - 210050096
Magham Dipen Anjan - 210050092

May 2024

S-Box & P-Box

In our cipher, we divide the 16-bit data block into four 4-bit sub-blocks. Each sub-block forms an input to a 4×4 S-box (a substitution with 4 input and 4 output bits), which can be easily implemented with a table lookup of sixteen 4-bit values, indexed by the integer represented by the 4 input bits. The most fundamental property of an S-box is that it is a nonlinear mapping, i.e., the output bits cannot be represented as a linear operation on the input bits.

The permutation portion of a round is simply the tranposition of the output bits from the S-Box of the current round to give input bits of the next round. Each sub-blocks produced by the output of S-Box are glued to be taken as input of P-Box which produces a 16-bit output.It is to be noted that this output is not the input to next S-Box , we XOR this with next round's Key to get the input to next S-Box.

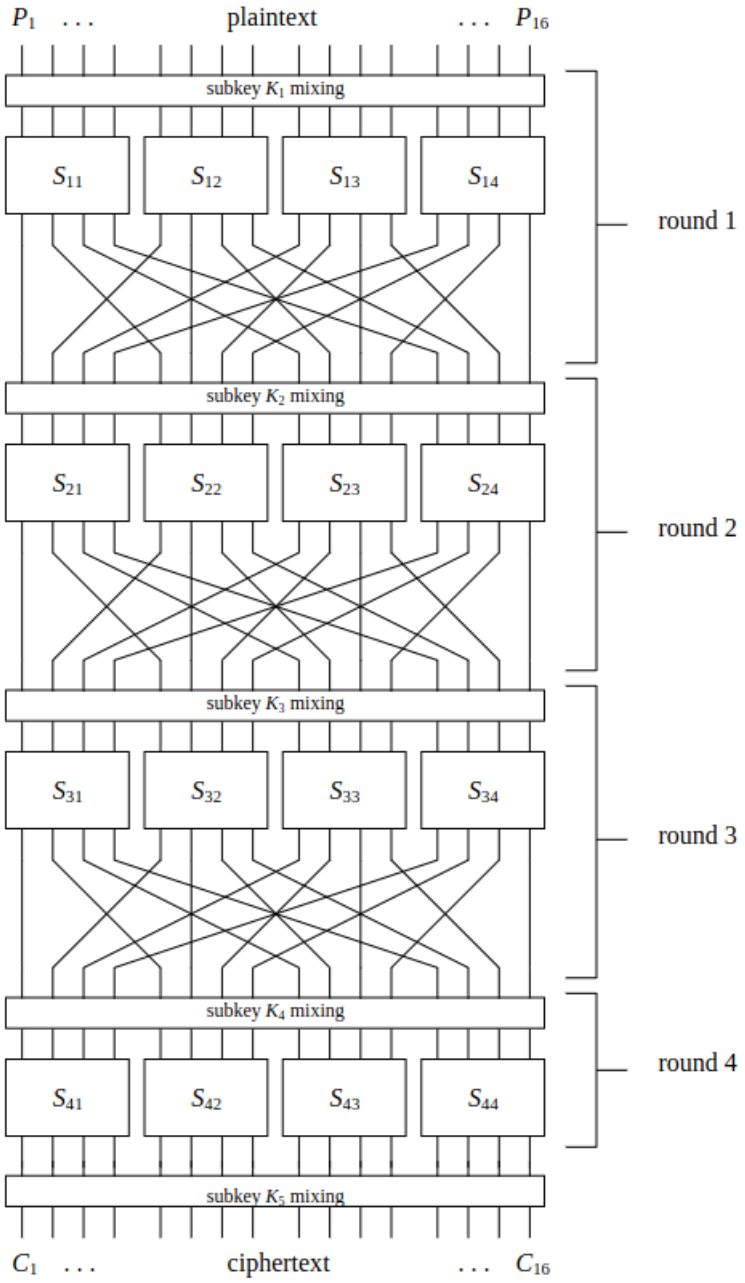
Encryption & Decryption

Encryption in an SPN cipher involves several rounds as follows:

1. **Key Expansion**: The encryption key is expanded into round keys.
2. **Initial Permutation**: Rearrange the plaintext bits.
3. **Round Function**: Substitute and permute the plaintext.
4. **Key Mixing**: Combine the round output with the round key.
5. **Final Permutation**: Rearrange the output bits.
6. **Output**: The final output is the ciphertext.

In an SPN cipher, decryption is the reverse of encryption:

1. **Key Expansion (Reverse)**: Expand the encryption key into round keys in reverse order.
2. **Initial Permutation (IP)**: Rearrange the ciphertext bits.
3. **Round Function (Reverse)**: Perform the inverse of substitution and permutation operations.
4. **Key Mixing (Reverse)**: Combine the round output with round keys in reverse order.
5. **Final Permutation (FP)**: Rearrange the output bits.
6. **Output**: The final output is the plaintext.



Linear Cryptanalysis of SPN Cipher

1. **Construct Linear Approximation for individual S-Boxes**:
   - Each entry in the Linear Approximation table represents the number of matches between the linear equation represented in hexadecimal as ”Input Sum” and the sum of the output bits represented in hexadecimal as ”Output Sum” minus 8 for that S-Box
   - Find the output which has more bias for given input of the S-Box to get an linear equation relating input and output bits with computed bias
2. **Query Data**:
   - Collect a sufficient number of plaintext-ciphertext pairs for cryptanalysis of the given cipher.
   - Store initial inputs for every S-box to initiate the attack on the cipher.

3. **Compute Paths & Linear Equations**:

    - Iterate through the following procedure until reaching the last round:
        - For a given input of a round, select the output with the maximum bias for that input. The input to the next round is the output after passing the maximum bias output through a P-box. Also, record the bias obtained in this round.
        - In the last round, calculate the final linear equation for the given input using the piling lemma for the biases stored and the output of the last round.

4. **Attack to Obtain Last Sub Key**:

    - Choose the linear equation with the maximum bias in the above step
    - Utilize this linear equation for a brute force attack on key bits for all set bits in the output of the last rounds as follows:
        - For each key, iterate through all possible plaintext-ciphertext pairs. Invert the XOR of the ciphertext and the key, and call this inverted_output. Calculate the bias with which this key satisfies the previously obtained linear equation.
        - From all biases obtained, select the sub key whose bias is closest to the computed bias of the linear equation
    - Iterate through the linear equations in decreasing order of the bias until we obtain the complete subkey

5. **Extending it to obtain complete key**:

    - Repeat the above procedure by replacing the ciphertext with s_inv[XOR(last_subkey,ciphertext)]
    - Make sure to use P-Box on the input obtained to the last round.

References

1. Howard M. Heys. *A Tutorial on Linear and Differential Cryptanalysis*
`https://ioactive.com/wp-content/uploads/2015/07/ldc_tutorial.pdf`

2. Christopher Kruegel. *CS177: Project 6 - Linear Cryptanalysis*.
`https://sites.cs.ucsb.edu/~chris/teaching/cs177/projects/proj6.html`

3. D. Mukhopadhyay *Linear CryptAnalysis*. nptelhrd
`https://www.youtube.com/watch?v=93PD9CqcZTg`