Substitution-Permutation Networks

Math 4175

§4.2. Substitution-Permutation Networks

There are two common ways of building round function in an iterated cipher:

1. Substitution-Permutation Networks or SPN
2. Feistel Cipher

In this section, we will discuss SPN. Any SPN consists of a mix between many Substitution Ciphers (denoted by π_S) and Permutation Ciphers (denoted by π_P) as discussed in the example of the previous slide. The Substitutions, also called S-boxes, are taken over a small bits, typically $\ell = 4, 6$, or 8 bits input. The S-boxes of this size can be efficiently inverted, a requirement for efficient decryption. Permutation π_P is performed over ℓm characters for some fixed positive integer m.

§4.2. Substitution-Permutation Networks

Notation: $\{0,1\}^\ell = \{(x_1, \cdots, x_\ell) : x_i \in \{0,1\} \text{ for all } i\}$ = set of all binary bits of length ℓ.

For given positive integers ℓ and m, both plaintext and ciphertext will be binary bits of length ℓm in SPN (i.e., ℓm is the block length of the cipher).

So an SPN is built from two components, which are denoted by π_S and π_P.

- $\pi_S : \{0,1\}^\ell \to \{0,1\}^\ell$ is a permutation called an S-box (the letter S denotes "substitution"). It is used to replace a set of ℓ bits with a different set of ℓ bits.
- $\pi_P : \{1, \ldots, \ell m\} \to \{1, \ldots, \ell m\}$ is a permutation used to shuffle ℓm bits, sometimes called index permutation.

§4.2. Substitution-Permutation Networks

Given an ℓm-bit string, say $x = (x_1, \cdots, x_{\ell m})$, we can regard x as the concatenation of m blocks of ℓ-bit substrings, which we denote by $x_{\langle 1 \rangle}, \ldots, x_{\langle m \rangle}$. More precisely,

$$x = x_{\langle 1 \rangle} || \cdots || x_{\langle m \rangle}$$

where

$$x_{\langle i \rangle} = (x_{(i-1)\ell+1}, \cdots, x_{i\ell})$$

The SPN will consist on N rounds. In each round (except for the last round, which is slightly different), we will perform m substitutions using π_S (called, S-box) , followed by a permutation using π_P.

Though the above operations are key independent, prior to each substitution operation, we will incorporate a round key bits via a simple XOR operation.

§4.2. Substitution-Permutation Networks

Let us illustrate with a specific example for SPN:

Example: Suppose that $\ell = m = N = 4$. Let

$$\mathbf{K} = 0011\ 1010\ 1001\ 0100\ 1101\ 0110\ 0011\ 1111$$

- Let $\pi_S : \{0,1\}^4 \to \{0,1\}^4$ and $\pi_P : \{1, \cdots, 16\} \to \{1, \cdots, 16\}$ be the permutations as defined on the next slide.
- To complete the description of SPN, we need to specify a key scheduling algorithm. Here is a simple possibility (though it is not a very secured one): suppose that we begin with a 32-bit key $\mathbf{K} = (k_1, \cdots, k_{32})$ as given above. For $1 \le r \le 5$, we define that $\mathbf{K}^r$ consists of 16 consecutive bits of $\mathbf{K}$, beginning with k_{4r-3}.

§4.2. Substitution-Permutation Networks

Let π_S be defined as follows in hexadecimal notation:

z	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\pi_S(z)$	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

Let π_P be defined as follows in regular decimal notation:

z	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\pi_P(z)$	1	5	9	13	2	6	10	14	3	7	11	15	4	8	12	16

S-box

input	output
0000 (0)	1110 (E)
0001 (1)	0100 (4)
0010 (2)	1101 (D)
0011 (3)	0001 (1)
0100 (4)	0010 (2)
0101 (5)	1111 (F)
0110 (6)	1011 (B)
0111 (7)	1000 (8)
1000 (8)	0011 (3)
1001 (9)	1010 (A)
1010 (A)	0110 (6)
1011 (B)	1100 (C)
1100 (C)	0101 (5)
1101 (D)	1001 (9)
1110 (E)	0000 (0)
1111 (F)	1110 (7)

Permutation

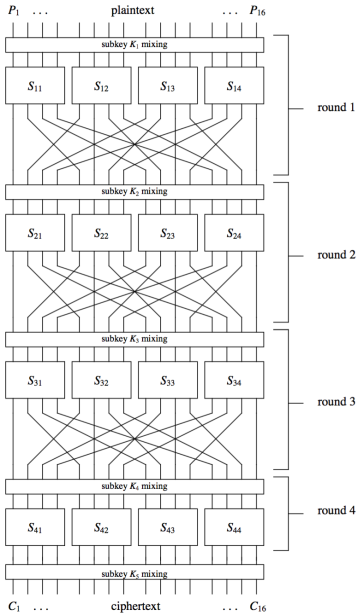
1 $\rightarrow$1
2 $\rightarrow$5
3 $\rightarrow$9
4 $\rightarrow$13
5 $\rightarrow$2
6 $\rightarrow$6
7 $\rightarrow$10
8 $\rightarrow$14
9 $\rightarrow$3
10$\rightarrow$7
11$\rightarrow$11
12$\rightarrow$15
13$\rightarrow$4
14$\rightarrow$8
15$\rightarrow$12
16$\rightarrow$16

Keys

$K_1 =$ 0011 1010 1001 0100
$K_2 =$ 1010 1001 0100 1101
$K_3 =$ 1001 0100 1101 0110
$K_4 =$ 0100 1101 0110 0011
$K_5 =$ 1101 0110 0011 1111

Suppose that the plaintext is $x = 26B7$. Determine the cipher text.

Correction: Last row should be 0111 (7)


P_1 . . . plaintext . . . P_{16}
subkey K_1 mixing
S_{11}
S_{12}
S_{13}
S_{14}
round 1
subkey K_2 mixing
S_{21}
S_{22}
S_{23}
S_{24}
round 2
subkey K_3 mixing
S_{31}
S_{32}
S_{33}
S_{34}
round 3
subkey K_4 mixing
S_{41}
S_{42}
S_{43}
S_{44}
round 4
subkey K_5 mixing
C_1 . . . ciphertext . . . C_{16}

$$
\begin{aligned}
w^0 &= 0010\ 0110\ 1011\ 0111 \\
K^1 &= 0011\ 1010\ 1001\ 0100 \\
u^1 &= 0001\ 1100\ 0010\ 0011 \\
v^1 &= 0100\ 0101\ 1101\ 0001 \\
w^1 &= 0010\ 1110\ 0000\ 0111 \\
K^2 &= 1010\ 1001\ 0100\ 1101 \\
u^2 &= 1000\ 0111\ 0100\ 1010 \\
v^2 &= 0011\ 1000\ 0010\ 0110 \\
w^2 &= 0100\ 0001\ 1011\ 1000 \\
K^3 &= 1001\ 0100\ 1101\ 0110 \\
u^3 &= 1101\ 0101\ 0110\ 1110 \\
v^3 &= 1001\ 1111\ 1011\ 0000 \\
w^3 &= 1110\ 0100\ 0110\ 1110 \\
K^4 &= 0100\ 1101\ 0110\ 0011 \\
u^4 &= 1010\ 1001\ 0000\ 1101 \\
v^4 &= 0110\ 1010\ 1110\ 1001 \\
K^5 &= 1101\ 0110\ 0011\ 1111, \quad \text{and} \\
y &= 1011\ 1100\ 1101\ 0110
\end{aligned}
$$

§4.2. Substitution-Permutation Networks

So the resulting cipher text is: BCD6.

Remark: In the above SPN Algorithm, u^j is the input to the S-boxes in round j, and v^j is the output of the S-boxes in round j. w^j is obtained from v^j by applying the permutation π_P and then u^{j+1} is constructed from w^j by x-or-ing with round $\mathbf{K}^{j+1}$. In the last round, the permutation π_P is not applied. As a consequence, the encryption algorithm can also be used for decryption.

Now we will give more formal definition of SPN cryptosystem.

§4.2. Substitution-Permutation Networks

Substitution-Permutation Network Cryptosystem (SNP): Let ℓ, m and N be positive integers, let $\pi_S : \{0,1\}^\ell \to \{0,1\}^\ell$ be a permutation, and let $\pi_P : \{1,\ldots,\ell m\} \to \{1,\ldots,\ell m\}$ be a permutation. Let $\mathcal{P} = \mathcal{C} = \{0,1\}^{\ell m}$ and let $\mathcal{K} \subseteq (\{0,1\}^{\ell m})^{N+1}$ consist of all possible key schedules that could be derived from an initial key K using the key scheduling algorithm. For a key schedule $\{K^1, \cdots, K^{N+1}\}$, we encrypt the plaintext x using the following algorithm.

§4.2. Substitution-Permutation Networks

SPN Algorithm: $\mathrm{SPN}(x, \pi_S, \pi_P, (\mathbf{K}^1, \mathbf{K}^2, \dots, \mathbf{K}^{N+1}))$.

- Initialize w^0 to be the plaintext x.
- For $1 \le j \le N-1$ do:
  - $u^j = w^{j-1} \oplus \mathbf{K}^j$.
  - $v^j_{\langle i \rangle} = \pi_S\big(u^j_{\langle i \rangle}\big)$, $1 \le i \le m$.
  - $w^j = \left(v^j_{\pi_P(1)}, v^j_{\pi_P(2)}, \cdots, v^j_{\pi_P(\ell m)}\right)$
- In round N
  - $u^N = w^{N-1} \oplus \mathbf{K}^N$.
  - $v^N_{\langle i \rangle} = \pi_S\big(u^N_{\langle i \rangle}\big)$, $1 \le i \le m$.
  - $y = v^N \oplus \mathbf{K}^{N+1}$
- Output y.

§4.2. Substitution-Permutation Networks

Remarks:

- Notice that the very first and last operations performed in this SPN are x-ors with subkeys. This is called whitening, and is regarded as a useful way to prevent an attacker from even beginning to carry out an encryption or decryption operation if the key is not known.
- Observe that the memory requirement of the S-box $\pi_S : \{0,1\}^\ell \to \{0,1\}^\ell$ is $\ell 2^\ell$, since we have to store 2^ℓ values, each of which needs ℓ bits to store.
- In the preceding example, we used four identical S-boxes in each round. The memory requirement of the S-box is 2^6 bits. If we instead used one S-box which mapped 16 bits to 16 bits, the memory requirement would be increased to 2^{20} bits, which would be prohibitively high for some applications.

§4.2. Substitution-Permutation Networks

Modified SPNs:

- The SPN in our example is not secure, because the key length is small and so exhaustive key search is possible.
- Larger SPNs with larger key size, larger S-boxes, and larger rounds will be more secure. An example of SPN called Rijndael has minimum key size of 128 bits, a minimum of 10 rounds and its S-box maps eight bits to eight bits.
- Unlike in our example, we could use four different S-boxes in each round, instead of the same S-box. For example, Data Encryption Standard uses eight different S-boxes in each round.
- Another strategy is to make permutation operation harder by including an invertible linear transformation, either in addition to, or instead of, permutation operation. This is done in the Advanced Encryption Standard.