Substitution-Permutation Networks

Substitution-Permutation Networks

- A one-bit change in the input should "affect" every bit of the output.
- The confusion-diffusion paradigm : construct a random looking permutation F with a large block length from many smaller random (or random-looking) permutations $\{f_i\}$ with small block length

We can define F as follows: the key k for F will specify 16 permutations $f_1, \ldots, f_{16}$ that each have an 8-bit (1-byte) block length.[3] Given an input $x \in \{0,1\}^{128}$, we parse it as 16 bytes $x_1 \cdots x_{16}$ and then set

$$F_k(x) = f_1(x_1) \| \cdots \| f_{16}(x_{16}). \tag{7.1}$$

These *round functions* $\{f_i\}$ are said to introduce *confusion* into F.

Confusion – diffusion

- For a truly random permutation changing the first bit of the input would be expected to affect all bytes of the output.
- A diffusion step is introduced whereby the bits of the output are permuted, or “mixed,” using a mixing permutation. This has the effect of spreading a local change
- The confusion/diffusion steps—together called a round—are repeated multiple times

Confusion – diffusion

As an example, a two-round block cipher following this approach would operate as follows. First, confusion is introduced by computing the intermediate result $f_1(x_1)\|\cdots\|f_{16}(x_{16})$ as in Equation (7.1), where we stress again that the $\{f_i\}$ depend on the key. The bits of the result are then "shuffled," or re-ordered, using a mixing permutation to give $x' = x'_1 \cdots x'_{16}$. Then $f'_1(x'_1)\|\cdots\|f'_{16}(x'_{16})$ is computed, using possibly different functions $\{f'_i\}$ that again depend on the key, and the bits of the result are again permuted using a mixing permutation to give output x''.

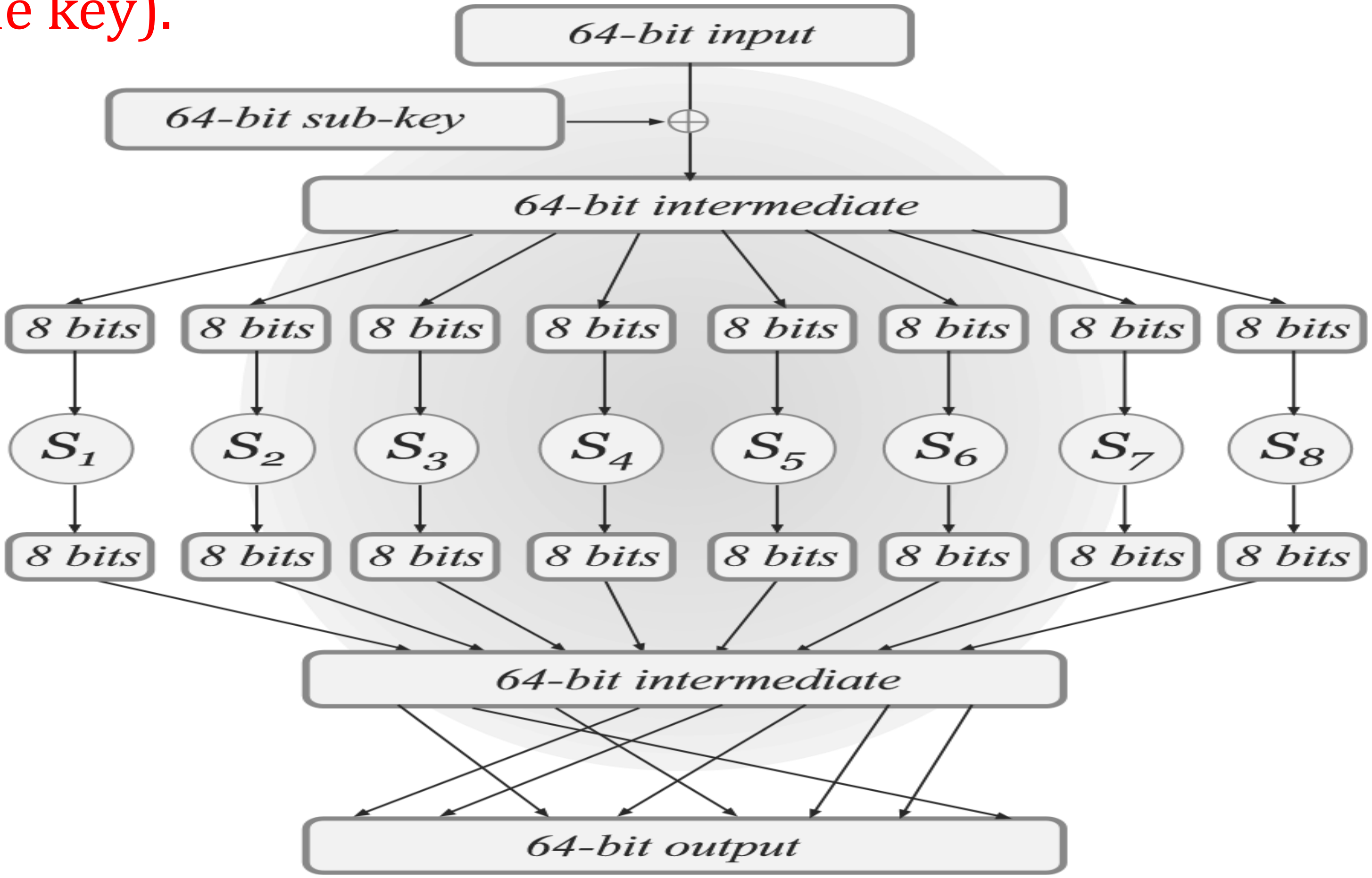
Substitution-permutation networks

- Consider an SPN with a 64-bit block length based on a collection of 8-bit (1-byte) s-boxes $S_1,\ldots,S_8$.
- Evaluating the cipher proceeds in a series of rounds, where in each round we apply the following sequence of operations to the 64-bit input x of that round

1. Key mixing: set $x := x \oplus k$, where k is the current-round sub-key;
2. Substitution: set $x := s_1(x_1) || \cdots || s_8(x_8)$, where xi is the i^{th} byte of x;
3. Permutation: permute the bits of x to obtain the output of the round

Substitution-Permutation Networks

Any SPN is invertible (given the key).



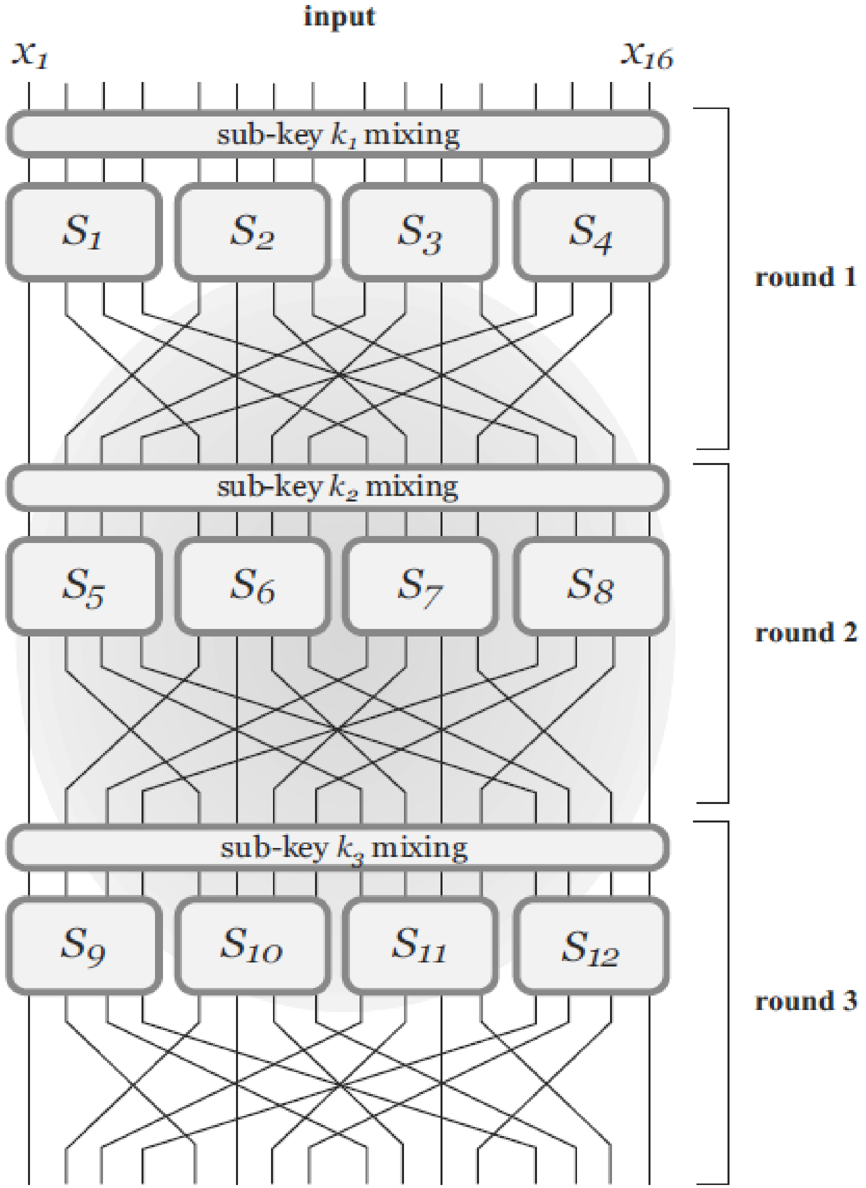
Substitution-permutation networks

- Different sub-keys (or round keys) are used in each round.
- The actual key of the block cipher is sometimes called the master key.
- The round keys are derived from the master key according to a key schedule.
- The key schedule is to use different subsets of the bits of the master key as the various sub-keys.
- An r-round SPN has
  - ***r rounds of key mixing,***
  - ***S-box substitution, and***
  - ***application of a mixing permutation,***
  - ***followed by a final key-mixing step***. (This means that an r-round SPN uses r + 1 sub-keys.)

Three rounds of a substitution-permutation network

- The number of rounds,
- Along with the exact choices of the *s*-boxes,
- Mixing permutations, and
- Key schedule,

Are what ultimately determine whether a given block cipher is trivially breakable or highly secure



Avalanche effect

- In any block cipher is that a small change in the input must "affect" every bit of the output.
- To induce the avalanche effect in a substitution-permutation network is to ensure that the following two properties hold (and sufficiently many rounds are used):

1. The s-boxes are designed so that changing a single bit of the input to an s-box changes at least two bits in the output of the s-box.
2. The mixing permutations are designed so that the bits output by any given s-box affect the input to multiple s-boxes in the next round.

For example, in previous figure the output from S1 affects the input to S5,S6,S7, and S8.

Feistel Networks

- An advantage of feistel networks over substitution-permutation networks is that
- The underlying functions used in a Feistel network—in contrast to the s-boxes used in SPNs—need not be invertible.
- A Feistel network operates in a series of rounds.
- In each round, a keyed round function is applied.
- Round functions need not be invertible.
- They will typically be constructed from components like S-boxes and mixing permutations

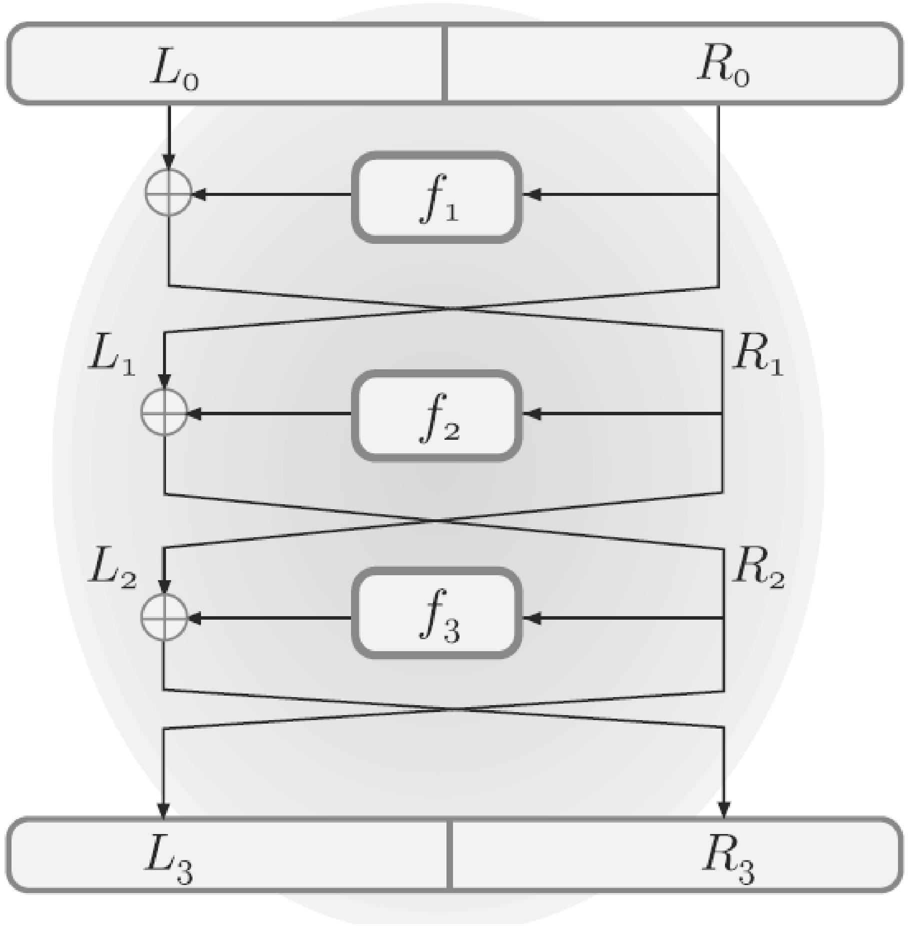
Feistel Networks

In a (balanced) Feistel network with ℓ-bit block length, the ith round function $\hat{f}_i$ takes as input a sub-key k_i and an $\ell/2$-bit string and generates an $\ell/2$-bit output. As in the case of SPNs, a master key k is used to derive sub-keys for each round. When some master key is chosen, thereby determining each sub-key k_i, we define $f_i : \{0,1\}^{\ell/2} \to \{0,1\}^{\ell/2}$ via $f_i(R) \stackrel{\text{def}}{=} \hat{f}_i(k_i, R)$. Note that the round functions $\hat{f}_i$ are fixed and publicly known, but the f_i depend on the master key and so are not known to the attacker.

The ith round of a Feistel network operates as follows. The ℓ-bit input to the round is divided into two halves denoted L_{i-1} and R_{i-1} (the "left" and "right" halves, respectively). The output (L_i, R_i) of the round is

$$L_i := R_{i-1} \quad \text{and} \quad R_i := L_{i-1} \oplus f_i(R_{i-1}). \tag{7.2}$$

A three-round Feistel network



Feistel Cipher Design Elements

- block size
- key size
- number of rounds
- subkey generation algorithm
- round function
- fast software en/decryption
- ease of analysis

Summary

Discussed

- Confusion – diffusion properties of Block cipher
- Substitution and Permutation Networks
- Feistel structure.