

# Introduction to Cryptology

## Lecture 16

# Announcements

- HW6 posted on course webpage, due Tuesday 4/11
- EC opportunity
  - Due on Tuesday, May 9

# Agenda

- Last time
  - Sponge Construction
  - Practical constructions of Stream Ciphers (K/L 6.1)
- This time
  - Practical constructions of Block Ciphers (K/L 6.2)

## Block Ciphers

Recall: A block cipher is an efficient, keyed permutation  $F: \{0,1\}^n \rightarrow \{0,1\}^\ell$ . This means the function  $F_k(x) := F(k, x)$  is a bijection, and moreover  $F_k$  and its inverse  $F_k^{-1}$  are efficiently computable given  $k$ .

- $n$  is the key length
- $\ell$  is the block length

## Block Cipher Security

Call for proposals for AES competition: 1997-2000

“The security provided by an algorithm is the most important factor. . . Algorithms will be judged on the following factors. . . The extent to which the algorithm output is indistinguishable from a random permutation. . .”

get to query  $F_k, F_k^{-1}$   
 $f, f^{-1}$  } strong pseudorandom permutation

## First Idea

- Random permutations over small domains are “efficient.”
  - What does this mean?
- First attempt to define  $F_k$ :
  - The key  $k$  for  $F$  will specify 16 permutations  $f_1, \dots, f_{16}$  that each have an 8-bit block length.
  - Given an input  $x \in \{0,1\}^{128}$ , parse it as 16 bytes  $x_1, \dots, x_{16}$  and then set
 
$$F_k(x) = f_1(x_1) \parallel \dots \parallel f_{16}(x_{16})$$
- Is this a permutation?
- Is this indistinguishable from a random permutation?

## Shannon's Confusion-Diffusion Paradigm

Above step is called the “confusion” step. Is combined with a “diffusion” step: the bits of the output are permuted or “mixed,” using a mixing permutation.

- Confusion/Diffusion steps taken together are called a round.
- Multiple rounds required for a secure block cipher.

Example: First compute intermediate value  $y = f_1(x_1) \parallel \dots \parallel f_{16}(x_{16})$ . Then permute the bits of  $y$ .

## Substitution-Permutation Network (SPN)

In practice, round-functions are not random permutations, since it would be difficult to implement this in practice.

- Why?

Instead, round functions have a specific form:

- Rather than having a portion of the key  $k$  specify an arbitrary permutation  $f$ , we instead fix a public “substitution function” (i.e. permutation)  $S$ , called an  $S$ -box.
- Let  $k$  define the function  $f$  given by  $f(x) = S(k \oplus x)$ .

## Informal Description of SPN

1. Key mixing: Set  $x := x \oplus k$ , where  $k$  is the current-round sub-key.
  2. Substitution: Set  $x := S_1(x_1) || \dots || S_8(x_8)$ , where  $x_i$  is the  $i$ -th byte of  $x$ .
  3. Permutation: Permute the bits of  $x$  to obtain the output of the round.
  4. Final mixing step: After the last round there is a final key-mixing step. The result is the output of the cipher.
    - Why is this needed?
- Different sub-keys (round keys) are used in each round.
    - Master key is used to derive round sub-keys according to a key schedule.

## Formal description of SPN

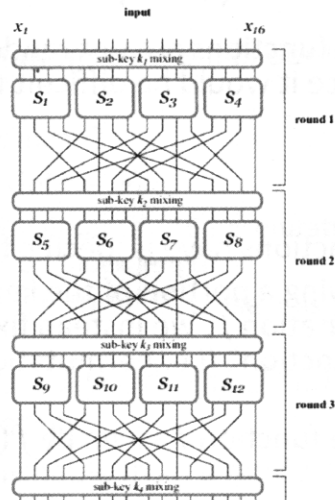


FIGURE 6.2: A substitution-permutation network.

## SPN is a permutation

Proposition: Let  $F$  be a keyed function defined by an SPN in which the  $S$ -boxes are all permutations. Then regardless of the key schedule and the number of rounds,  $F_k$  is a permutation for any  $k$ .

## How many rounds needed for security?

The avalanche effect.

Random permutation: When a single input bit is changed to go from  $x$  to  $x'$ , each bit of  $f(x)$  should be flipped with probability  $\frac{1}{2}$ .

- $S$ -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.
- The mixing permutations are designed so that the output bits of any given  $S$ -box are used as input to multiple  $S$ -boxes in the next round.

## The Avalanche Effect

$f(x)$  vs.  $f(x')$  where  $x, x'$  differ in one bit:

1. After the first round the intermediate values differ in exactly two bit-positions. Why?
2. The mixing permutation spreads these two bit positions into two different  $S$ -boxes in the second round.
  - At the end of the second round, intermediate values differ in 4 bits.
3. Continuing the same argument, we expect 8 bits of the intermediate value to be affected after the 3<sup>rd</sup> round, 16 after the 4<sup>th</sup> round, and all 128 bits of the output to be affected at the end of the 7<sup>th</sup> round.



## Practical SPN

- Usually use many more than 7 rounds.
- S-boxes are NOT random permutations.

## Attacking Reduced-Round SPN

Trivial case: Attacking one round SPN with no final key-mixing step.