Katz, Lindell
Introduction to Modern Cryptrography

Slides Chapter 6

Markus Bläser, Saarland University

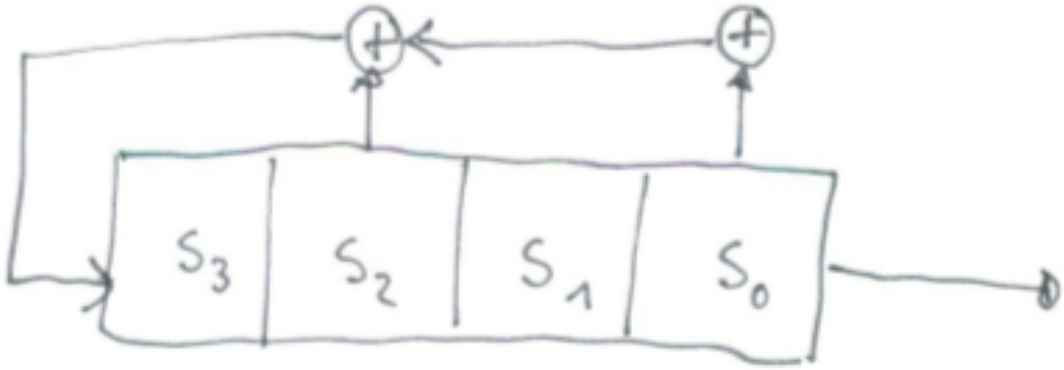
What is cryptography? (Only a partial answer . . .)

Science	▶ Construction of provably secure schemes (modulo assumptions) ▶ Mathematically investigations of the underlying assumptions
Engineering	▶ Building efficient candidate constructions ▶ Efficient implementations (soft/hardware) of cryptographic schemes ▶ Thinking about unmodelled threats (side channels)
Art, magic, stupidity	▶ Everything else

We proudly present

The cryptographic construction of the day

Nov 18, 2016: Linear feedback shift registers



$$s_i^{(t+1)} := s_{i+1}^{(t)}, \qquad i = 0, \ldots, n-2$$

$$s_{n-1}^{(t+1)} := \bigoplus_{i=0}^{n-1} c_i s_i^{(t)}$$

$$y_i := s_{i-1}^{(0)}, \qquad i = 1, \ldots, n$$

$$y_i := \bigoplus_{j=0}^{n-1} c_j y_{i-n+j}, \qquad i > n$$

Extensions of LFSR

LFSR have good statistical properties, but are **not** suited for cryptography.

Extensions:

- Nonlinear feedback: $s_{n-1}^{(t+1)} = g(s_0^{(t)}, \ldots, s_{n-1}^{(t)}$
- Nonlinear output: $y_i = g(s_0^{(i-1)}, \ldots, s_{n-1}^{(i-1)}$
  g needs to be balanced

Examples:

- Trivium
- RC4 (can be attacked)

Nov 30, 2016: Substitution-permutation networks (SPN)

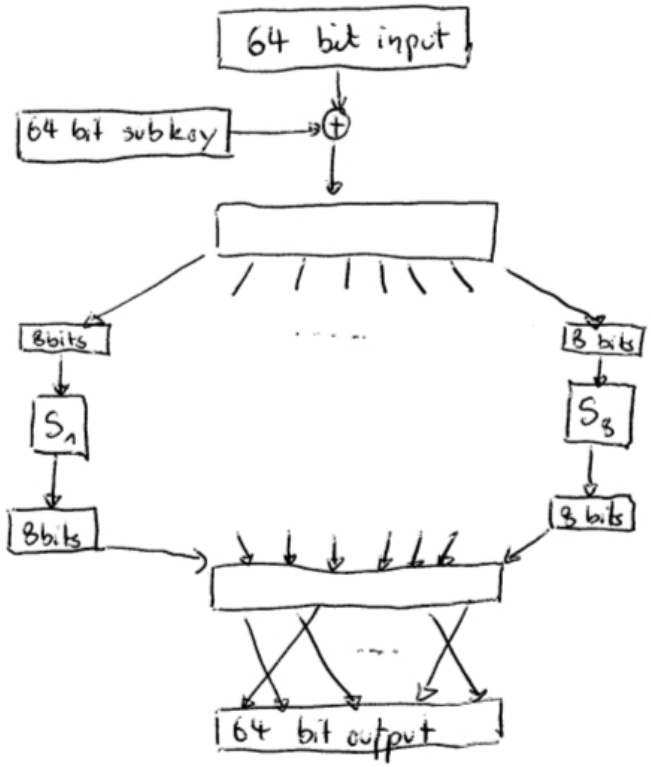
- Choose a public "substitution functions" (that is, permutations) S_i.
- S_i operates on few bits, say 8.

One round of an SPN consists of (here with 64 bits):

1. *Key mixing:* Set $x := x \oplus k$, where k is the current-round subkey
   (subkey is computed from key k for each round, e.g. by selecting some bits of k)
2. *Substitution:* Set $x := S_1(x_1) || \ldots || S_8(x_8)$, where x_i is the i byte of x.
3. *Permutation:* Permute the bits of x to obtain the output of the round.
   Permutation is fixed, but accross the outputs of all boxes.

"Confusion-Diffusion principle"

SPN (2)



Proposition (6.3)

The function computed by an SPN is always are permutation, regardless of the key schedule and the number of rounds.

The avalanche effect

To be pseudorandom, a small change in the input must "affect" all bits.

1. S-boxes are designed in such a way that changing one bit of the input changes at least *two* bits in the output.
2. The mixing permutations feed the output bits of any S-box into multiple S-boxes.

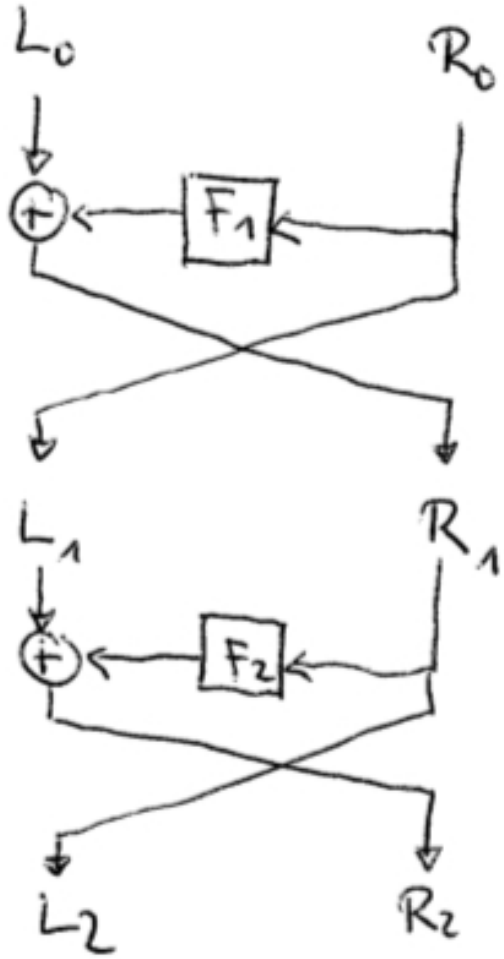
Dec 2, 2016: Feistel networks

- Out of a master k, we generate subkeys k_i, one for each round.
- In each round, the input is divided into two halves L_{i-1} and R_{i-1} of $\ell/2$ bits each.
- $\hat{f}_i$ is a keyed function taking a key k_i and and $\ell/2$-bit string and outputting an $\ell/2$ bit string.
- $f_i(.) := \hat{f}_i(k_i, .)$
- In each round, compute $L_i := R_{i-1}$ and $R_i := L_{i-1} \oplus f_i(R_i)$

Proposition (6.4)

Any Feistel network computes a permutation. If the key is known, then it is effectively invertible.

Feistel networks (2)



Application: DES

- 16 round Feistel network
- block length 64 bits, key length 54 bits
- $\hat{f}_i$ is almost an SPN:
  - 32 bit input is expanded to 48 bits by duplication
  - S-Boxes map 6 bits to 4 bits.

- DES is still "secure" in the sense that the best attacks essentially do an exhaustive search through the key space.
- However, the key length is too short (now)