Sketch of Lecture 20

Wed, 3/1/2017

4 Block ciphers (and DES in particular)

We now introduce block ciphers at the example of **DES** (short for data encryption standard).

This sketch only provides an overview but does not include all details. See Chapter 4 in our book for these internals and detailed diagrams.

DES was the first public cryptosystem. While a public standard, the design decisions have been kept secret.

1974: proposed by IBM (lead by Horst Feistel; Lucifer) with input from NSA (key size reduced from 128 to 56 bits)

1976–2000: US national standard (broken by exhaustive search in 1997)

2000: replaced with AES (Rijndael) by NIST; however, **3DES** still considered secure (more later)

Why was the design secret? For many years, a particular mystery about DES was the choice of the S-boxes. Much later, in 1990, Biham and Shamir discovered **differential cryptanalysis**, a general method for breaking block ciphers. Surprisingly, it turned out that the particular choice of S-boxes made DES rather resistant against that attack. Indeed, as confirmed later, the IBM researchers had already discovered and anticipated that attack in 1990, but were asked by the NSA to keep it secret (it was a powerful weapon against other cryptosystems).

https://en.wikipedia.org/wiki/Data_Encryption_Standard

Comment. As our discussion will show, DES was designed to be implemented in hardware.

4.1 General principles of block cipher design

A block cipher takes a plaintext block of, say, B bits and encrypts it into a ciphertext block of B bits.

For instance, for DES, $B = 64$: 64 bit blocks are encrypted to 64 bit blocks.

For now, we will just focus on encrypting a single block.

However, we will need to talk about how to use a block cipher to encrypt longer plaintexts that need to be broken into many blocks (it is generally a bad idea to individually and independently encrypt each block).

The design of a block cipher is almost an art, but there are two guiding principles due to Claude Shannon, the father of information theory:

- confusion

refers to making the relationship between the ciphertext and the key as complex and involved as possible (changing one bit of the key should change the ciphertext completely)

For instance. In DES, confusion is increased by the S-box substitutions.

- diffusion

refers to dissipating the statistical structure of plaintext over the bulk of ciphertext

(changing one bit of the plaintext should change the ciphertext completely; likewise, changing one bit of the ciphertext should change the plaintext completely)

For instance. In DES, diffusion is increased by the E-box and P-box permutations.

Example 118. The classical substitution cipher provides only confusion.

Diffusion is completely missing. Changing bits of the plaintext only changes corresponding parts of the ciphertext. That's why frequency analysis can break these ciphers so easily.

Typical block ciphers are built by iteration, and consist of several **rounds**. Each round should have steps to increase both confusion and diffusion.

- **(key expansion)** First, we need to expand key k into several **round keys** $k_1, k_2, \ldots, k_n$ (n rounds).

  For instance. For DES, each round key k_i has 48 bits, which are drawn from the 56 bit DES key k in such a way that each bit of k shows about in about 14 of the 16 rounds.

- **(round functions)** Then, the message m is encrypted successively with R_{k_1}, R_{k_2}, ..., R_{k_n} to obtain c in the end.

  $$m \rightarrow \boxed{R_{k_1}} \rightarrow \boxed{R_{k_2}} \rightarrow \ldots \rightarrow \boxed{R_{k_n}} \rightarrow c$$

  Each R_k is called a **round function**.

  For instance. For DES, there are $n = 16$ rounds; for AES-128, there are $n = 10$ rounds

A specific block cipher now needs specific algorithms for key expansion and round functions.

For DES, 16 rounds are used, which are identical in functionality but use different round keys k_i.

There is one additional, cryptographically irrelevant, step for DES: namely, there is a (fixed) **initial permutation** IP, which shuffles the bits of m before being sent to R_{k_1}. Similarly, the output of $R_{k_{16}}$ is shuffled with IP^{-1}, the inverse permutation, to produce c. (For the exact permutation see Chapter 4.4 in our book.)

Why? When implemented in hardware, this permutation does not cost any work, since it is just a wiring of the bits. In fact, the permutation somehow simplified the electrical engineering in the chips of the 70s.

4.2 A block cipher design: Feistel ciphers

Many ciphers, including DES (but not AES) are Feistel ciphers. This means that the encryption functions R_{k_i} are of a special format. The crucial ingredient is a **round function** $f_{k_i}(x)$.

This round function can be **any** function, such that x and $f_{k_i}(x)$ have the same size in bits (though only good choices will provide security). Also, several different round functions can be used for the different rounds.

To encrypt m using R_{k_i} (for DES, m is 64 bits and the round key k_i is 48 bits):

- Split the plaintext m into two halves (L_0, R_0) (for DES, each half is 32 bits).
- $L_1 = R_0$
  $R_1 = L_0 \oplus f_{k_i}(R_0)$
- Then, $R_{k_i}(m)$ is (L_1, R_1).

Example 119. How to decrypt one round? That is, how to obtain (L_0, R_0) from (L_1, R_1)?

Solution. First, $R_0 = L_1$. Then, $L_0 = R_1 \oplus f_{k_i}(R_0)$.

Important comment. In particular, we can take any round function f in the sense that we obtain some cipher, which can actually be decrypted (however, most choices for f will be insecure; see example below).

Comment. In hardware, the circuit for decryption is the same as for encryption, just reversed.

Example 120. What happens if we choose $f_{k_i}(R) = 0$ as the round function?

Solution. In that case, we are just swapping left and right half. No security whatsoever.

To finish the description of DES, we need to specify $f_{k_i}(R)$, where R is 32 bits and k_i is 48 bits.

We did that in class, but do not reproduce the description and diagrams here. See Chapter 4.4 of our book. The crucial ingredients are an E-box (expansion), 8 S-boxes (substitution) and a P-box (permutation).