Computer and Network Security

Computer & Information Sciences & Engineering
University Of Florida
Gainesville, Florida 32611-6120
nemo@cise.ufl.edu

Data Encryption Algorithm
and
Conventional Modern Cryptosystems
(Pfleeger Ch. 3,4; KPS Ch. 3)

1 Enigma

1.1 History

- Germans
- Between WWI and WWII
- automated symmetric encryption - an electromechanical device.
- Variants used throughout WWII for communications
  - diplomatic
  - military
  - transatlantic communications cable
- Polish obtained a commercial version
- classified in the 1930's
- Polish mathematicians and machine flown to England
- decisive impact on WWII

1.2 Enigma Structure

overall similar a teletype.

1.2.1 General Operation

1. electrical connection made to send power
2. through a number of permutation components
3. finally lighting up an output light
4. also caused some wheels to turn, changing permutations

1.2.2 rotors

1. a disk with electrical contacts on each side
2. contacts connected via an internal permutation of wires
3. each contact had a symbol on rim of the disk, used to set key
4. rotors moved like an odometer when keys pressed
5. overall permutation would not repeat until they all returned to their original position (26 symbols $\Rightarrow 26^r$ where $r =$ number of rotors).
6. interchangeable - units were sent sets of rotors in a box

1.2.3 math

- Rotors

  If a given rotor implements a permutation P, and R_i is the ith rotation permutation

  $$R_i(n) = n + i \ modulo \ N$$

  then the rotors implemented the concatenation of r permutations, each a rotation-permutation-inverse-rotation sequence, in which the rotation and its corresponding inverse rotation changed for one or more rotors for every plaintext character entered

- exchange permutation reflector
  - special disk that just connected pairs of contacts on the same side and did not move.
  - (Note: an exhange permutation is one that is its own inverse, i.e., if $p(x) = y$, then $p(y) = x$).
  - Signal was sent back out a different contact and then traversed the rotors in the reverse direction.

1.2.4 ”steckerbord,”

1. plug board similar to those used by old timey telephone operators,
2. allowed an arbitrary permutation to be entered before the signal traversed the rotors and reflector.

1.2.5 Keys

key was

1. set of rotors to be used
2. order in which the rotors were to be placed in the machine
3. reflector to use,
4. initial positions of each of the rotors and reflector
5. plugboard settings

1.2.6 Comments

Enigma is of importance in a number of ways:

1. cracked by Polish, British and American codebreakers
2. probably decided the outcome of the war
3. it shows how carefully one must take any claims of "unbreakability" of a cryptosystem
4. electronic computers were developed in secret during WWII expressly to crack intercepted Enigma transmissions (Colossus machines)

2 Feistel Structure

First reported by Horst Feistel of IBM in 1973

2.1 Overview

- based on some number of rounds of processing
- input plaintext is the input to the first round
- ciphertext is the last round's output
- encryption key K is used to generate a round key k_i for each round i
- each round computes a different function as long as the k_i's are different,
- any number of rounds may be employed
- same hardware for encryption and decryption

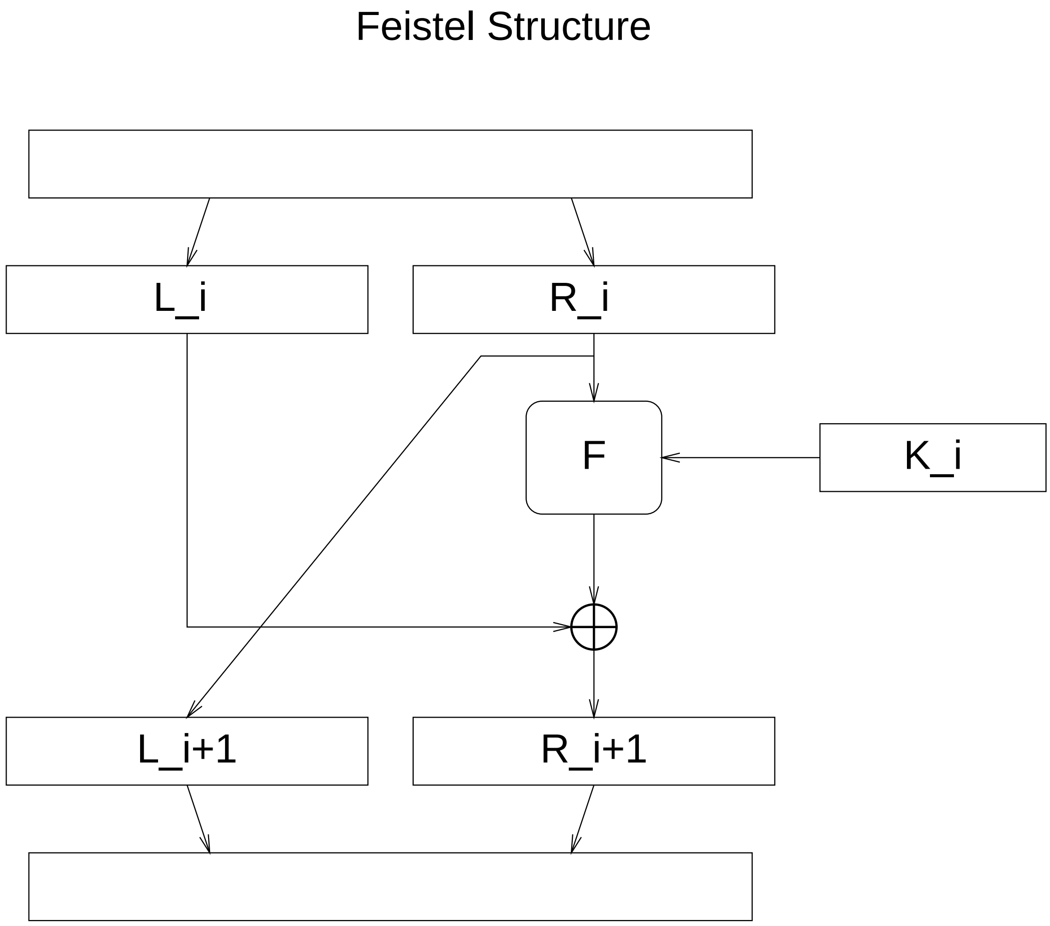

Figure 1: Feistel Structure

2.2 Round Structure

On each round,

1. input to the round is split into 2 halves of equal length, L_i and R_i
2. output $L_{i+1} = R_i$
3. output $R_{i+1} = F(R_i, k_i) \oplus L_i$, where $F()$ is some one-way round function

2.3 Decryption

The beauty of this structure is that, although F is difficult to invert, it is not necessary to do so.

To decrypt, given k_i, R_{i+1} and L_{i+1}:

1. R_i is easily obtained:
$$R_i = L_{i+1}.$$
2. L_{i+1} is run through F using k_i, result is XORed with R_{i+1} to obtain L_i
$$L_i = F(R_i, k_i) + R_{i+1} = F(L_{i+1}, k_i) + R_{i+1},$$

The structure lends itself to use of the same hardware or software for encryption as decryption due to its nature, running it with the reverse order of round keys.

Decryption uses the same keys in reverse order, and with the left and right halves of the 64 bit block reversed.

3 DES

3.1 Overview

1. 56-bit keys
2. 64-bit plaintext blocks
3. 16-round Feistel structure
4. (useless) initial and final permutations of the bits

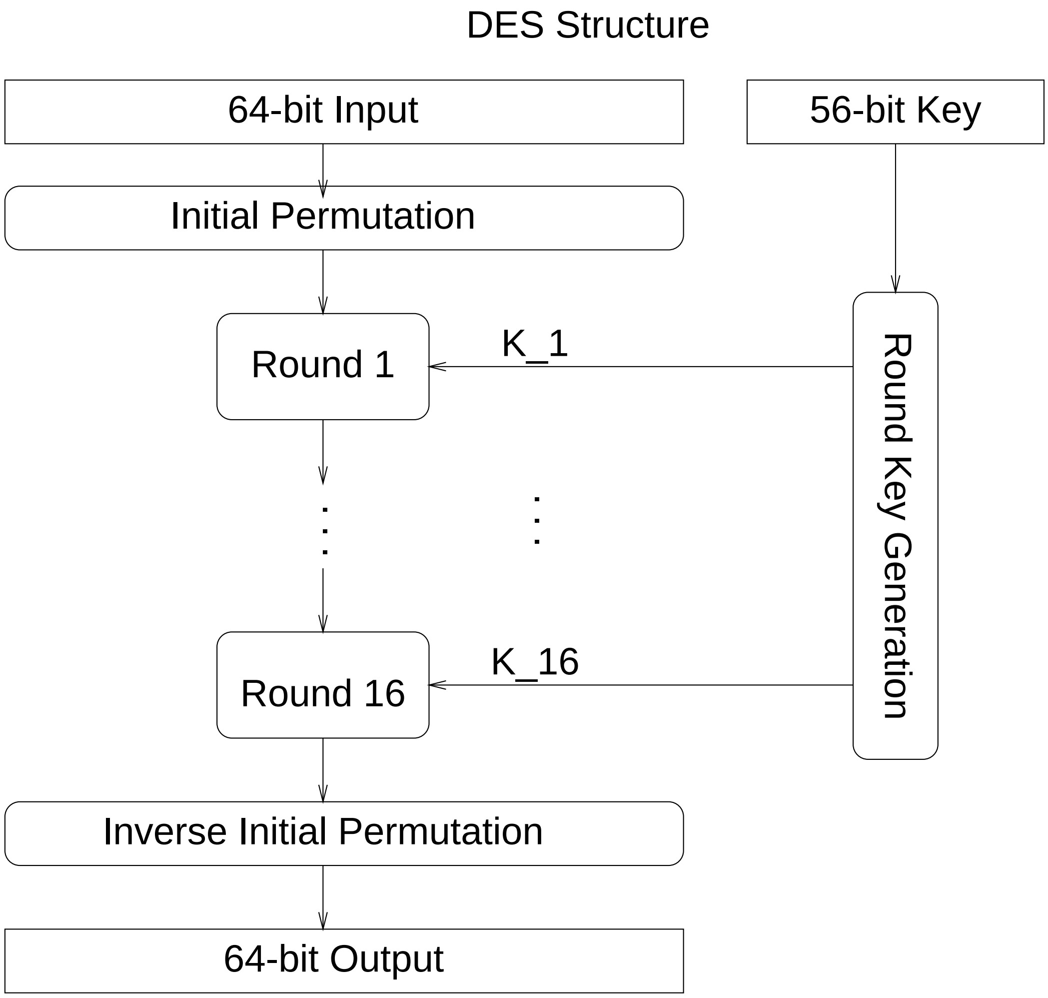

Figure 2: Overall DES Structure

3.2 Round Key Generation

round keys are generated from the 56-bit master key by

1. splitting the master key in half
2. rotating each half either one or two bits left (logical shift)
3. then selecting 24 bits from each half

3.3 Round Details

In each round,

3.3.1 Expansion permutation

1. right half is first expanded from 32 to 48 bits by considering it as 8 nybbles (4-bit quantities),
2. copying the 8 nybbles to the center 4 bits of 8 6-bit quantities.
3. leftmost and rightmost bits of the 6-bit quantities are copied from the bits that flanked the nybble in the original 32-bit input (wrapped around).
4. E.g., first 6-bit quantity consists of bits 31,0,1,2,3, and 4, while the second consists of bits 3,4,5,6,7 and 8.

3.3.2 Round key effect

- 48-bit quantity is XORed with the round key

3.3.3 S-Boxes

XORed 48-bit quantity is put through 8 S-boxes. Each S-box takes one of the 6-bit quantities,

1. uses the two end bits as a two-bit selector,
2. applies one of four substitutions to the 4 central bits
3. A 32-bit quantity results,

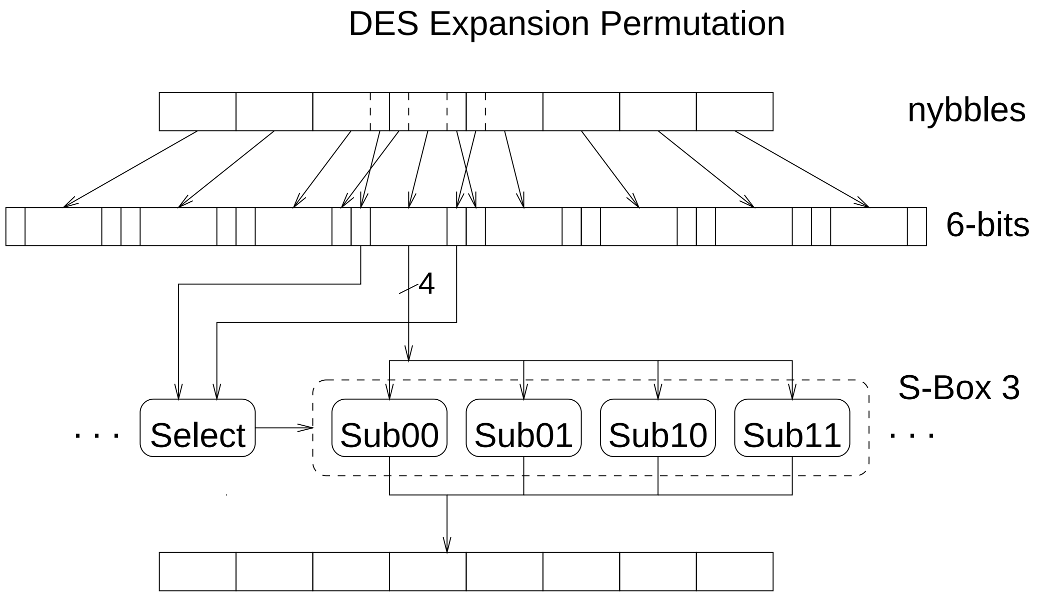

Figure 3: DES Expansion Permutation and S-Boxes

3.3.4 P-Boxes - diffusion

- 32-bit quantity is then put through a permutation (the P-boxes)

3.3.5 More diffusion

- permuted 32-bit quantity XORed with the left half in the Feistel structure.

4 DES Modes

4.1 ECB - Electronic Code Book

1. Simply break up M into 64-bit blocks and encrypt each one independently with the key, K.
2. Subject to cut&paste attack.

4.2 CBC - Cipher Block Chaining

1. M is again broken up into 64-bit blocks, each plaintext block is XORed with the preceding ciphertext block before encrypting with K
2. first plaintext block is XORed with initialization vector (IV).
3. Decryption requires that each ciphertext block be
   (a) decrypted with K,
   (b) then XORed with the preceding ciphertext block or IV.
4. Transmission error will affect two blocks.
5. Specific bits in one plaintext block may be changed at the expense of garbling the preceding block by an attacker.

4.3 OFB - Output Feedback

1. DES used as a PRNG
2. Converts DES to a stream cipher (specifically, a Vernam cipher).
3. IV is encrypted using K
4. leftmost k bits of the output are used as a stream key
5. These k bits shifted into the right end of the previous DES input (tossing out its k leftmost bits)
6. new quantity is encrypted to produce the next stream key
7. stream keys XORed plaintext symbols to produce ciphertext.
8. Decryption simply
   (a) generates the same stream keys
   (b) XORs them with the ciphertext.

9. Key stream may be generated ahead of time for more speedy processing.
10. Q: Why not use the rightmost k bits, so that an attacker will not have the input to DES even if some plaintext and ciphertext are known?

4.4 CFB - Cipher Feedback

1. Like OFB
2. except ciphertext symbol is shifted into the previous DES input to produce the next stream key
3. keys cannot be computed ahead of time
4. like CBC, improves diffusion greatly