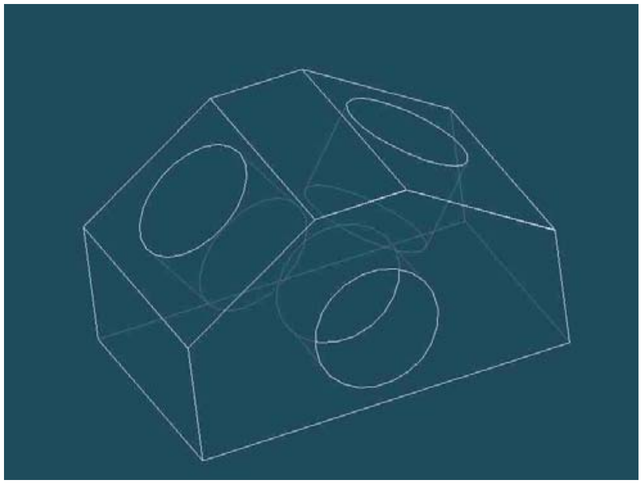
Symmetric Key Cryptosystems

Debdeep Mukhopadhyay
IIT Kharagpur

Definition

- Alice and Bob has the same key to encrypt as well as to decrypt
- The key is shared via a “secured channel”
- Symmetric Ciphers are of two types:
  - Block : The plaintext is encrypted in blocks
  - Stream: The block length is 1
- Symmetric Ciphers are used for bulk encryption, as they have better performance than their asymmetric counter-part.

Block Ciphers



What we have learnt from history?

- **Observation:** If we have a cipher C_1=(P,P,K1,e1,d1) and a cipher C_2 (P,P,K2,e2,d2).
- We define the product cipher as C_1xC_2 by the process of first applying C_1 and then C_2
- Thus C_1xC_2=(P,P,K1xK2,e,d)
- Any key is of the form: (k1,k2)

and $e=e_2(e_1(x,k1),k2)$. Likewise d is defined.

Note that the product rule is always associative

Question:

- Thus if we compute product of ciphers, does the cipher become stronger?
  - The key space become larger
  - 2^{nd} Thought: Does it really become larger.
- Let us consider the product of a
  1. multiplicative cipher (M): y=ax, where a is co-prime to 26 //Plain Texts are characters
  2. shift cipher (S) : y=x + k

Is MxS=SxM?

- MxS: y=ax+k : key=(a,k). This is an affine cipher, as total size of key space is 312.
- SxM: y=a(x+k)=ax+ak
  - Now, since gcd(a,26)=1, this is also an affine cipher.
  - key = (a,ak)
  - As gcd(a,26)=1, a^{-1} exists. There is a one-one relation between ak and k. Thus the total size of the key space in SxM is still 312. Thus this is also the affine cipher
- Thus S and M are commutative.

Note that:

- M is a permutation cipher.
- S is a substitution cipher.
- Composed cipher has a larger key space than each of them.
- If we had computed MxM or SxS, would that have lead to the increase of key space? No.
  - This is because SxS=S and MxM=M
  - These are called idempotent ciphers

Inference

- Thus there is no point of obtaining products of idempotent functions.
- Rather we would get "product ciphers" from non-idempotent ciphers
  - That is by iterating them (rounds)
- How to make non-idempotent functions?
  - Compose two small different cryptosystems which do not commute

Why?

- If there are two cryptosystems which are idempotent and also commute then their product is also idempotent.
- $(S_1 x S_2) \text{ x } (S_1 x S_2) = S_1 x (S_2 \text{ x } S_1) x S_2$

  $= S_1 x (S_1 x S_2) x S_2$

  $= (S_1 x S_1) \text{ x } (S_2 x S_2)$

  $= S_1 x S_2$

  Thus, MxS is also idempotent. Why?

Thus, composing MxS does not help.

Concept of Rounds

- Consider : S=f(x) and P=x+k
- What is SxP? f(x)+k
- What is (SxP)x(SxP)? f(f(x)+k)+k
  - For this multiplication to increase the key length, thus SxP should not be idempotent.
  - that is $f(f(x)+k)+k \neq f^2(x)+k'$
  - This happens if f is non-linear wrt. +
  - **Hence we compose linear and non-linear functions to increase the security of a cipher**

Data Encryption Standard (DES)

(Iterated) Block Cipher

- Plaintext and ciphertext consists of fixed sized blocks
- Ciphertext obtained from plaintext by iterating a **round function**
- Input to round function consists of key and the output of previous round
- These functions are obtained by the repeated application of Substitution and Permutation.
- Thus they are called Substitution Permutation Networks (SPN)

Feistel Cipher

- **Feistel cipher** refers to a type of block cipher design, not a specific cipher
- Split plaintext block into left and right halves: Plaintext = (L_0,R_0)
- For each round i=1,2,...,n, compute

  $L_i= R_{i-1}$

  $R_i= L_{i-1} \oplus F(R_{i-1},K_i)$

  where f is **round function** and K_i is **subkey**
- Ciphertext = (L_n,R_n)

Feistel Permutation

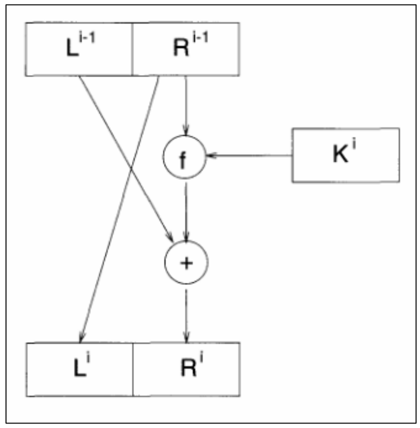
- Decryption: Ciphertext = (L_n,R_n)
- For each round i=n,n-1,…,1, compute

  $\mathbf{R_{i-1} = L_i}$

  $\mathbf{L_{i-1} = R_i \oplus F(R_{i-1},K_i)}$

  where f is round function and K_i is subkey
- Plaintext = (L_0,R_0)
- Formula “works” for any function F
- But only secure for certain functions F

Encryption



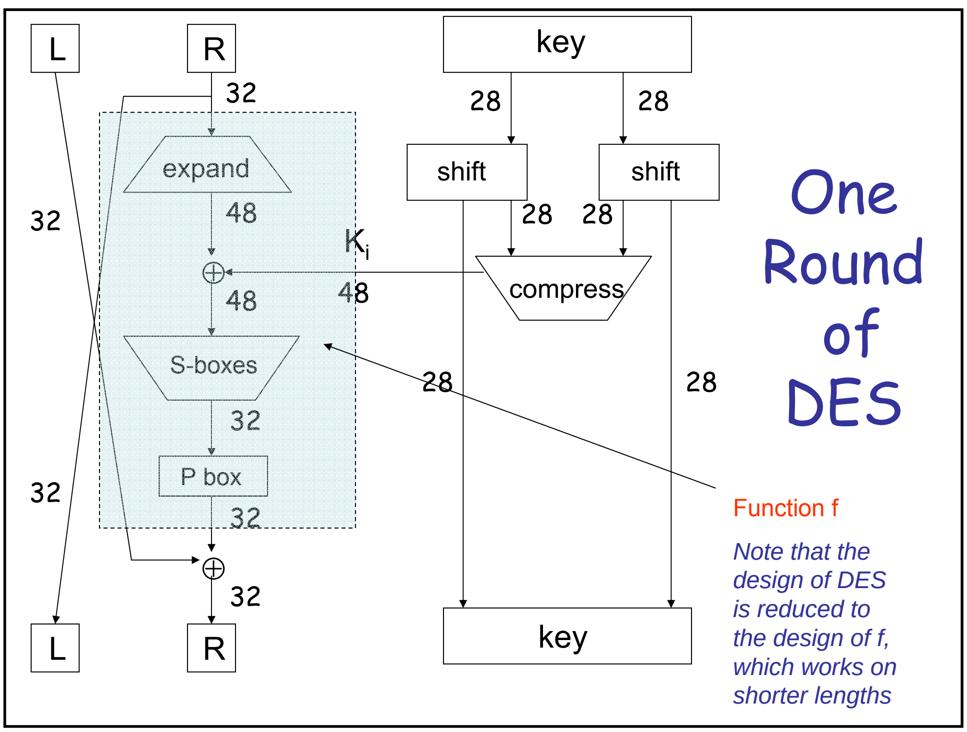
Repeating/ Iterating this transformation we obtain the Feistel Cipher

Data Encryption Standard

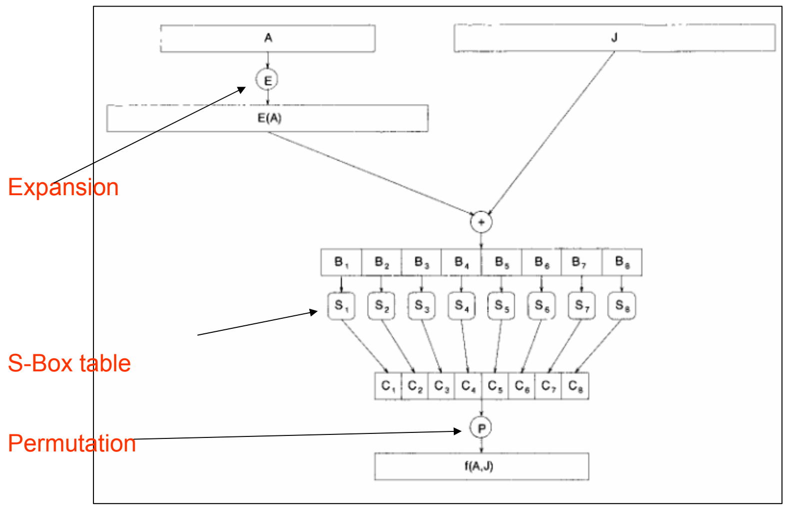
- DES developed in 1970's
- Based on IBM Lucifer cipher
- U.S. government standard
- DES development was controversial
  - NSA was secretly involved
  - Design process not open
  - Key length was reduced
  - Subtle changes to Lucifer algorithm

DES Numerology

- DES is a Feistel cipher
- 64 bit block length
- 56 bit key length
- 16 rounds
- 48 bits of key used each round (subkey)
- Each round is simple (for a block cipher)
- Security depends primarily on “S-boxes”
- Each S-boxes maps 6 bits to 4 bits



The function f



DES Expansion

- Input 32 bits

 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

- Output 48 bits

31 0 1 2 3 4 3 4 5 6 7 8
 7 8 9 10 11 12 11 12 13 14 15 16
15 16 17 18 19 20 19 20 21 22 23 24
23 24 25 26 27 28 27 28 29 30 31 0

DES S-box (Substitution Box)

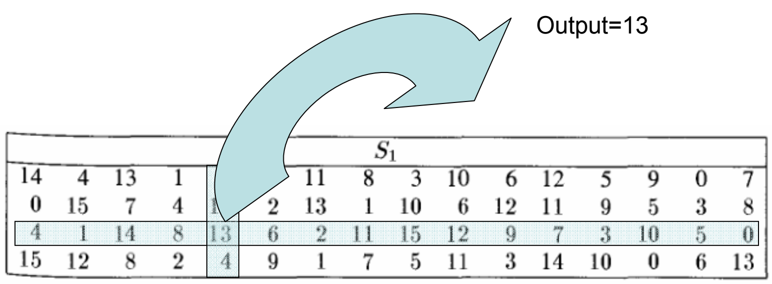
- 8 “substitution boxes” or S-boxes
- Each S-box maps 6 bits to 4 bits
- S-box number 1

input bits (0,5)
↓ input bits (1,2,3,4)

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1110	0100	1101	0001	0010	1111	1011	1000	0011	1010	0110	1100	0101	1001	0000	0111
01	0000	1111	0111	0100	1110	0010	1101	0001	1010	0110	1100	1011	1001	0101	0011	1000
10	0100	0001	1110	1000	1101	0110	0010	1011	1111	1100	1001	0111	0011	1010	0101	0000
11	1111	1100	1000	0010	0100	1001	0001	0111	0101	1011	0011	1110	1010	0000	0110	1101

For other tables refer to Stinson’s Book

S-Box with Table entries in decimal



S_1

14	4	13	1	[illegible]	[illegible]	11	8	3	10	6	12	5	9	0	7
0	15	7	4	[illegible]	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

What is the output if input is 101000?

Row=10=2

Column=0100=4

Properties of the S-Box

- There are several properties
- We highlight some:
  - The rows are permutations
  - The inputs are a non-linear combination of the inputs
  - Change one bit of the input, and half of the output bits change **(Avalanche Effect)**
  - Each output bit is dependent on all the input bits

DES P-box (Permutation Box)

- Input 32 bits

  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
  16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

- Output 32 bits

  15 6 19 20 28 11 27 16 0 14 22 25 4 17 30 9
  1 7 23 13 31 26 2 8 18 12 29 5 21 10 3 24

Principle of Confusion and Diffusion

- The design principles of Block Cipher depends on these properties
- The S-Box is used to provide confusion, as it is dependent on the unknown key
- The P-Box is fixed, and there is no confusion due to it
- But it provides diffusion
- Properly combining these is necessary.

DES Subkey

- 56 bit DES key, 0,1,2,…,55
- Left half key bits, LK

```
49 42 35 28 21 14  7
 0 50 43 36 29 22 15
 8  1 51 44 37 30 23
16  9  2 52 45 38 31
```

- Right half key bits, RK

```
55 48 41 34 27 20 13
 6 54 47 40 33 26 19
12  5 53 46 39 32 25
18 11  4 24 17 10  3
```

DES Subkey

- For rounds `i=1,2, … ,n`
  - Let LK = (LK circular shift left by r_i)
  - Let RK = (RK circular shift left by r_i)
  - Left half of subkey K_i is of LK bits

    ```
    13 16 10 23  0  4  2 27 14  5 20  9
    22 18 11  3 25  7 15  6 26 19 12  1
    ```
  - Right half of subkey K_i is RK bits

    ```
    12 23  2  8 18 26  1 11 22 16  4 19
    15 20 10 27  5 24 17 13 21  7  0  3
    ```

DES Subkey

- For rounds 1, 2, 9 and 16 the shift r_i is 1, and in all other rounds r_i is 2
- Bits 8,17,21,24 of LK omitted each round
- Bits 6,9,14,25 of RK omitted each round
- **Compression permutation** yields 48 bit subkey K_i from 56 bits of LK and RK
- **Key schedule** generates subkey

DES Some Points to Ponder

- An initial perm P before round 1
- Halves are swapped after last round
- A final permutation (inverse of P) is applied to (R_{16},L_{16}) to yield ciphertext
- None of these serve any security purpose

Security of DES

- Security of DES depends a lot on S-boxes
  - Everything else in DES is linear
- Thirty years of intense analysis has revealed no “back door”
- Attacks today use exhaustive key search
- In Crypto 93, a DES key search engine was shown
  - A cluster of 5760 chips were put.
  - Each chip could test $5x10^7$ chips per second.
  - Cost of each equal to around $10
  - DES could be broken in about 1.5 days

Complementation Property of DES

- DES also has some other weaknesses:
  - weak keys exist
    - there are some keys like 010101….01 for which all the round keys are 0….0
    - there are some partial weak keys also, where instead of 16 different keys, only two distinct round keys are generated.
  - Complementation Property exists…
    - Find out