Modern Block Cipher Standards (DES)

Debdeep Mukhopadhyay

Assistant Professor
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
INDIA -721302

Data Encryption Standard

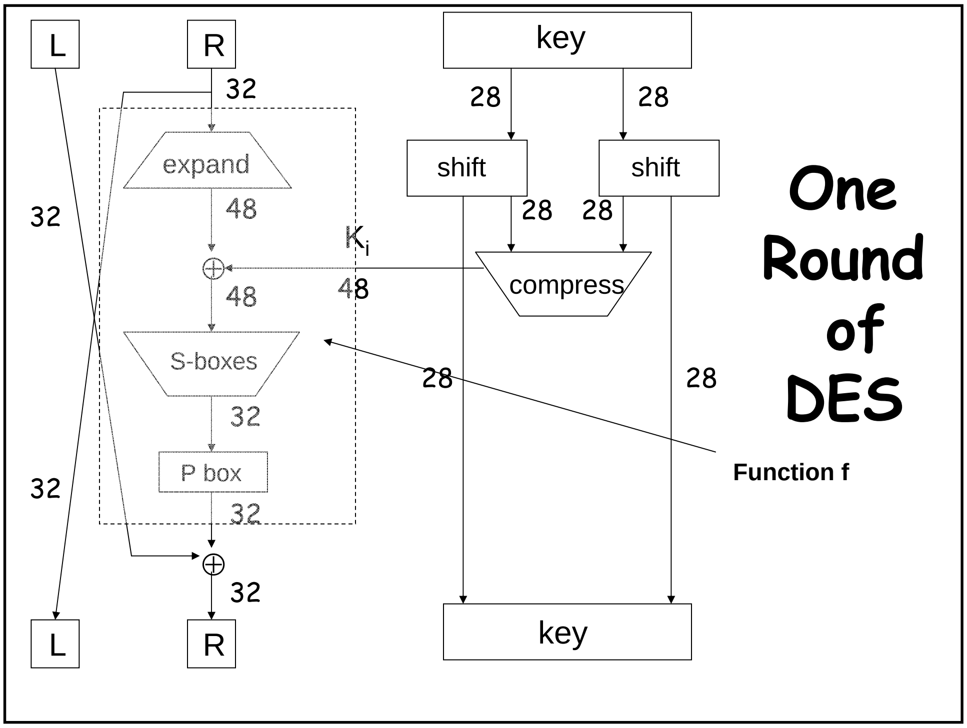
- DES developed in 1970's
- Based on IBM Lucifer cipher
- Federal Information Processing Standard (FIPS)
- DES development was controversial:
  - Design process was not open, people feared hidden trapdoors that would have allowed NSA to decrypt messages without the keys.
  - Key length was small (56 bits)

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

Initial Permutations

- DES has an initial permutation and a final permutation after 16 rounds.
- These permutations are inverses of each other and operate on 64 bits.
- They have no cryptographic significance.
- The designers did not disclose their purpose.



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

Output=13

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

DES Subkey

- Input key size: 64 bits, of which 8 are parity bits.
- 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 24 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 24 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, and its inverse at the end.
- Halves are swapped after last round.
- A final permutation (inverse of P) is applied to (R_{16},L_{16}) to yield ciphertext.

Exercise

Prove that decryption in DES can be done by applying the encryption algorithm to the ciphertext, with the key schedule reversed.

Weak keys

- A weak key is the one which after parity drop operation, consists either of all 0's, all 1's or half 0's and half 1's.
- Four out of the 2^{56} keys are weak keys.

Examples of weak keys

Keys before parity drop (64 bits)	**Actual key (56 bits)**
0101 0101 0101 0101	0000000 0000000
1F1F 1F1F 0E0E 0E0E	0000000 FFFFFFF
E0E0 E0E0 F1F1 F1F1	FFFFFFF 0000000
FEFE FEFE FEFE FEFE	FFFFFFF FFFFFFF

Consequence of weak keys

- The round keys created from any of these weak keys are the same.
  - For example, for the first weak key, all the round keys are 0.
  - The second key leads to half 0s, and half 1s.
- If we encrypt a block with a weak key and subsequently encrypt the result with the same weak key, we get the original block.

Semi Weak Keys

- A semi weak key creates only two different round keys and each of them is repeated eight times.
- There are six key pairs that are called semi-weak keys.
- The round keys created from each pair are the same in different order.

Semi weak keys

First key in the pair	**Second key in the pair**
01FE 01FE 01FE 01FE	FE01 FE01 FE01 FE01
1FE0 1FE0 0EF1 0EF1	E01F E01F F10E F10E
01E0 01E0 01F1 01F1	E001 E001 F101 F101
1FFE 1FFE 0EFE 0EFE	FE1F FE1F FE0E FE0E
011F 011F 010E 010E	1F01 1F01 0E01 0E01
E0FE E0FE F1FE F1FE	FEE0 FEE0 FEF1 FEF1

A Sample round key generation

1	9153E54319BD	6EAC1ABCE642
2	6EAC1ABCE642	9153E54319BD
3	6EAC1ABCE642	9153E54319BD
4	6EAC1ABCE642	9153E54319BD
5	6EAC1ABCE642	9153E54319BD
6	6EAC1ABCE642	9153E54319BD
7	6EAC1ABCE642	9153E54319BD
8	6EAC1ABCE642	9153E54319BD
9	9153E54319BD	6EAC1ABCE642
10	9153E54319BD	6EAC1ABCE642
11	9153E54319BD	6EAC1ABCE642
12	9153E54319BD	6EAC1ABCE642
13	9153E54319BD	6EAC1ABCE642
14	9153E54319BD	6EAC1ABCE642
15	9153E54319BD	6EAC1ABCE642
16	6EAC1ABCE642	9153E54319BD

There are 8 equal round keys in each semi-weak keys.

Also, the round key in the first set is the same as the 16th key in the second set.

This means that the keys are inverses of each other.

Thus, $E_{k2}(E_{k1}(P))=P$

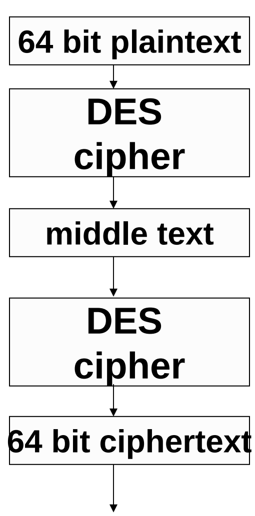
Multiple DES

- The major criticism against DES is the key length.
- So, we may try cascading several DES applications.
- Luckily, DES does not form a group under the composition operation. Thus, it is highly improbable that we can obtain k3 st.
  - $E_{k2}(E_{k1}(P))=E_{k3}(P)$

2DES

- Uses two applications of the DES cipher.
- The total key size is 56x2=112 bits.
- However 2DES is vulnerable to a known plaintext attack.

Meet in the middle attack



$M=E_{K2}(P)$ and $M=D_{k2}(C)$

Attacker performs a known plaintext attack. He collects (P,C) pairs.

Using 1st relation, he encrypts P using all possible 2^{56} keys, and records all values for M.

Using 2nd relation, he decrypts C using all possible 2^{56} keys, and records all values for M.

Security of 2 DES

- Then the attacker checks for a match in the table in the value of M. He notes the key pair (K_1,K_2)
- If there are more than one keys, he takes another (P,C) pair.
- The attacker continues until there is only key left.
- Thus attack complexity is around 2^{57}.
- What does this say about the security of 2DES?

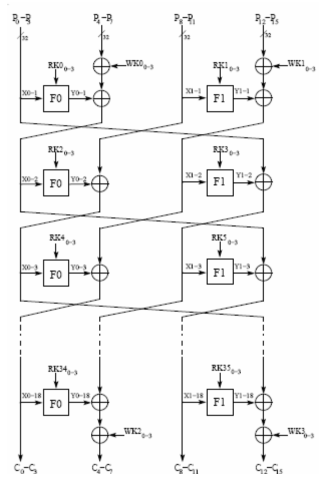
Triple DES

- Since 2DES was a bad design, people consider 3 applications of DES.
- The first and third stages use K_1 as key.
- The second stage use K_2 as the key.
- Also, the middle stage uses decryption.
- Thus, setting $K_1=K_2$ we have simple DES.

Generalization of the Feistel Cipher

Clefia:

128 bit block cipher designed by Sony Corporation



Further Reading

- B. A Forouzan, *Cryptography & Network Security, Tata Mc Graw Hills, Chapter 5*
- Douglas Stinson, *Cryptography Theory and Practice, 2nd Edition*, Chapman & Hall/CRC

Exercises

DES (Data Encryption Standard) although an elegantly designed cipher has become old. Its $n = 56$ bit key is being challenged by the present day computation power. As an alternative, it was thought of applying DES twice, i.e in creating a product cipher $DES' = DES \times DES$. If the key space of DES was $K = \{0,1\}^n$, the key size of the product cipher is expected to be $K_1 \times K_2 = (K_1, K_2)$, where $K_1, K_2 \in K$. The plaintext of the cipher is denoted by $P = \{0,1\}^m$ and the cipher is endomorphic (the plaintext and the ciphertext are the same set).

In regard to this composed cipher answer the following questions:

1. What is the property in the DES construction which helps to increase the key length by performing such composition? (Another way of asking the question is: why is DES not idempotent?)

Exercises

2. Using the DES cipher an attacker obtains l pairs of plaintexts and ciphertexts: $(p_1, c_1), \ldots, (p_l, c_l)$. The key is say (K_1, K_2) but unknown to the attacker (obviously, else why will he/she be an attacker).
   Prove that for all $1 \leq i \leq l$, $DES_{K_1}(p_i) = DES^{-1}_{K_2}(c_i)\ \forall i$, where $1 \leq i \leq l$.

3. Prove that of all the possible keys (K_1, K_2), the expected number of keys for which $DES_{K_1}(p_i) = DES^{-1}_{K_2}(c_i)\ \forall i$, where $1 \leq i \leq l$, is about 2^{2n-lm}.

4. Suppose $l \geq 2n/m$, what can you say to the attacker to help him in developing an attack against the composed cipher DES'?

Exercises

5. The attacker starts building up two lists: L_1 and L_2. Each entry in the list L_1 and L_2 has l tuples of elements of P followed by an element from K. The lists are filled with all possible keys.
   The lists are now sorted in a lexicographic manner on the l tuples. The attacker now does a linear search to find out the common l tuples in the lists.
   Explain how does the attacker maintain the list and how does this approach help him to find out the correct key? Show that the amount of memory required by the attacker is $2^{n+1}(ml+n)$ bits and number of encryptions and/or decryptions required to identify the key is $l2^{n+1}$.

   (Hint: Use the distinguisher: for the correct key $DES_{K_1}(p_i) = DES^{-1}_{K_2}(c_i)\ \forall i$)

6. Into what class does the above kind of attack fall?

Next Day's Topic

- Advanced Encryption Standard