Introduction to Cryptography
Lecture 8

Monika K. Polak

February 16, 2021

Content of this Lecture

- Once again about time and space
- Security of DES
- Double Encryption and Meet-in-the-Middle Attack
- Triple Encryption - 3DES
- Key Whitening - DESX

Key space and key length

- Key space – set of all possible keys for a cryptographic algorithm
- Key space size – `number` of possible keys
- Key size – amount of bits that we need to store the `number`

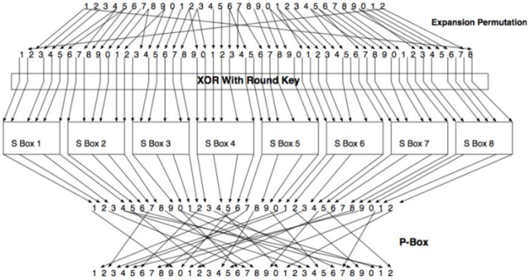
Chapter 3 – The Data Encryption Standard (DES)

DES: Week keys

Pierwotny ciąg słabego klucza	Faktyczny ciąg klucza
0101 0101 0101 0101	0000000 0000000
FEFE FEFE FEFE FEFE	FFFFFFF FFFFFFF
1F1F 1F1F 1F1F 1F1F	0000000 FFFFFFF
E0E0 E0E0 F1F1 F1F1	FFFFFFF 0000000

Computer Science @ RIT

DES: The f-Function



- DIFFUSION hides the relationship between the ciphertext and the plaintext
- CONFUSION hides the relationship between the ciphertext and the key

DES: Avalanche effect (change in plaintext)

Round		δ
	02468aceeca86420 12468aceeca86420	1
1	3cf03c0fbad22845 3cf03c0fbad32845	1
2	bad2284599e9b723 bad3284539a9b7a3	5
3	99e9b7230bae3b9e 39a9b7a3171cb8b3	18
4	0bae3b9e42415649 171cb8b3ccaca55e	34
5	4241564918b3fa41 ccaca55ed16c3653	37
6	18b3fa419616fe23 d16c3653cf402c68	33
7	9616fe2367117cf2 cf402c682b2cefbc	32
8	67117cf2c11bfc09 2b2cefbc99f91153	33

Round		δ
9	c11bfc09887fbc6c 99f911532eed7d94	32
10	887fbc6c600f7e8b 2eed7d94d0f23094	34
11	600f7e8bf596506e d0f23094455da9c4	37
12	f596506e738538b8 455da9c47f6e3cf3	31
13	738538b8c6a62c4e 7f6e3cf34bc1a8d9	29
14	c6a62c4e56b0bd75 4bc1a8d91e07d409	33
15	56b0bd7575e8fd8f 1e07d4091ce2e6dc	31
16	75e8fd8f25896490 1ce2e6dc365e5f59	32
IP^{-1}	da02ce3a89ecac3b 057cde97d7683f2a	32

DES: Avalanche effect (change in key)

Round		δ
	02468aceeca86420 02468aceeca86420	0
1	3cf03c0fbad22845 3cf03c0f9ad628c5	3
2	bad2284599e9b723 9ad628c59939136b	11
3	99e9b7230bae3b9e 9939136b768067b7	25
4	0bae3b9e42415649 768067b75a8807c5	29
5	4241564918b3fa41 5a8807c5488dbe94	26
6	18b3fa419616fe23 488dbe94aba7fe53	26
7	9616fe2367117cf2 aba7fe53177d21e4	27
8	67117cf2c11bfc09 177d21e4548f1de4	32

Round		δ
9	c11bfc09887fbc6c 548f1de471f64dfd	34
10	887fbc6c600f7e8b 71f64dfd4279876c	36
11	600f7e8bf596506e 4279876c399fdc0d	32
12	f596506e738538b8 399fdc0d6d208dbb	28
13	738538b8c6a62c4e 6d208dbbb9bdeeaa	33
14	c6a62c4e56b0bd75 b9bdeeaad2c3a56f	30
15	56b0bd7575e8fd8f d2c3a56f2765c1fb	33
16	75e8fd8f25896490 2765c1fb01263dc4	30
IP–1	da02ce3a89ecac3b ee92b50606b62b0b	30

DES: S-boxes

S1

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7
1	0	F	7	4	E	2	D	1	A	6	C	B	9	5	3	8
2	4	1	E	8	D	6	2	B	F	C	9	7	3	A	5	0
3	F	C	8	2	4	9	1	7	5	B	3	E	A	0	6	D

S2

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	F	1	8	E	6	B	3	4	9	7	2	D	C	0	5	A
1	3	D	4	7	F	2	8	E	C	0	1	A	6	9	B	5
2	0	E	7	B	A	4	D	1	5	8	C	6	9	3	2	F
3	D	8	A	1	3	F	4	2	B	6	7	C	0	5	E	9

S3

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	A	0	9	E	6	3	F	5	1	D	C	7	B	4	2	8
1	D	7	0	9	3	4	6	A	2	8	5	E	C	B	F	1
2	D	6	4	9	8	F	3	0	B	1	2	C	5	A	E	7
3	1	A	D	0	6	9	8	7	4	F	E	3	B	5	2	C

S4

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	7	D	E	3	0	6	9	A	1	2	8	5	B	C	4	F
1	D	8	B	5	6	F	0	3	4	7	2	C	1	A	E	9
2	A	6	9	0	C	B	7	D	F	1	3	E	5	2	8	4
3	3	F	0	6	A	1	D	8	9	4	5	B	C	7	2	E

S5

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	2	C	4	1	7	A	B	6	8	5	3	F	D	0	E	9
1	E	B	2	C	4	7	D	1	5	0	F	A	3	9	8	6
2	4	2	1	B	A	D	7	8	F	9	C	5	6	3	0	E
3	B	8	C	7	1	E	2	D	6	F	0	9	A	4	5	3

S6

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	C	1	A	F	9	2	6	8	0	D	3	4	E	7	5	B
1	A	F	4	2	7	C	9	5	6	1	D	E	0	B	3	8
2	9	E	F	5	2	8	C	3	7	0	4	A	1	D	B	6
3	4	3	2	C	9	5	F	A	B	E	1	7	6	0	8	D

S7

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	4	B	2	E	F	0	8	D	3	C	9	7	5	A	6	1
1	D	0	B	7	4	9	1	A	E	3	5	C	2	F	8	6
2	1	4	B	D	C	3	7	E	A	F	6	8	0	5	9	2
3	6	B	D	8	1	4	A	7	9	5	0	F	E	2	3	C

S8

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	D	2	8	4	6	F	B	1	A	9	3	E	5	0	C	7
1	1	F	D	8	A	3	7	4	C	5	6	B	0	E	9	2
2	7	B	4	1	9	C	E	2	0	6	A	D	F	3	5	8
3	2	1	E	7	4	A	8	D	F	C	9	0	3	5	6	B

Problem 1: S-boxes

One important property which makes DES secure is that the S-boxes are nonlinear. In this problem we verify this property by computing the output of S_1 for several pairs of inputs. Show that $S_1(x_1) \oplus S_1(x_2) \neq S_1(x_1 \oplus x_2)$, where "$\oplus$" denotes bitwise XOR, for:

a) $x_1 = 000000$, $x_2 = 000001$

b) $x_1 = 111111$, $x_2 = 100000$

c) $x_1 = 101010$, $x_2 = 010101$

Exhaustive Key Search Revisited

- A simple exhaustive search for a DES key knowing one pair (x_1, y_1):

$$DES_{k_i}(x_1) \stackrel{?}{=} y_1,\ \ i = 0, 1, ..., 2^{56} - 1$$

- However, for most other block ciphers a key search is somewhat more complicated
- A brute-force attack can produce false positive results
  - keys k_i that are found are not the one used for the encryption
  - The likelihood of this is related to the relative size of the key space and the plaintext space
  - A brute-force attack is still possible, but several pairs of plaintext–ciphertext are needed

An Exhaustive Key Search Example

- Assume a cipher with a block width of 64 bit and a key size of 80 bit
- If we encrypt x1 under all possible 2^{80} keys, we obtain 2^{80} ciphertexts
  However, there exist `only` 2^{64} different ones
- If we run through all keys for a given plaintext–ciphertext pair, we find on average $2^{80}/2^{64} = 2^{16}$ keys that perform the mapping $e_k(x_1) = y_1$

Expected number of false keys

Given a block cipher with a key length of k bits and block size of n bits, as well as t plaintext-ciphertext pairs $(x_1, y_1), \ldots, (x_t, y_t)$, the expected number of false keys which encrypt all plaintexts to the corresponding ciphertexts is:

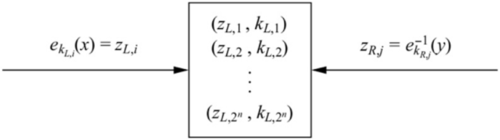
$$2^{k-tn}.$$

Increasing the Security of Block Ciphers

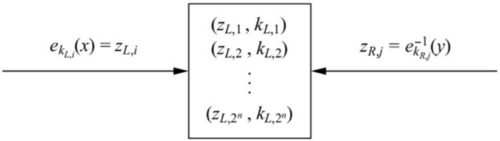
- Double Encryption and Meet-in-the-Middle Attack
- Triple Encryption
- Key Whitening

Double Encryption and Meet-in-the-Middle Attack

- A plaintext x is first encrypted with a key k_L, and the resulting ciphertext is encrypted again using a second key k_R
- Assuming a key length of k bits, an exhaustive key search would require $2^k \cdot 2^k = 2^{2k}$ encryptions or decryptions
- A Meet-in-the-Middle attack requires $2^k + 2^k = 2^{k+1}$ operations!



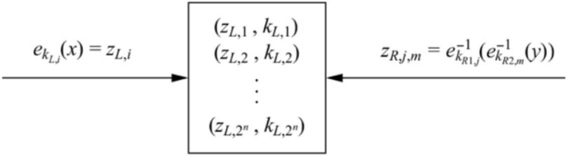
Double Encryption and Meet-in-the-Middle Attack



- `Phase I`: for the given (x_1, y_1) the left encryption is brute-forced for all $k_{L,i}$, $i = 1, 2, \ldots, 2^k$ and a `lookup table with` 2^k `entry` (each $n + k$ bits wide) is computed (the lookup table should be ordered by the result of the encryption (z_L, i))
- `Phase II`: the right encryption is brute-forced (using decryption) and for each (z_R, i) it is checked whether (z_R, i) is equal to any (z_L, i) value in the table of the first phase
- `Double encryption is not much more secure then single encryption!`

Triple Encryption

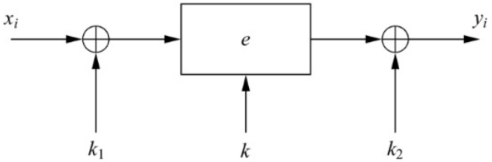
- The encryption of a block three times $y = e_{k_3}(e_{k_2}(e_{k_1}(x)))$
- In practice a variant scheme is often used EDE (encryption-decryption-encryption) $y = e_{k_3}(d_{k_2}(e_{k_1}(x)))$
- Still we can perform a meet-in-the middle attack, and it reduces the effective key length of triple encryption from $3K$ to $2K$!



- `Triple encryption effectively doubles the key length`

Key Whitening

- Makes block ciphers such as DES much more resistant against brute-force attacks (it does not strengthen block ciphers against most analytical attacks)
- In addition to the regular cipher key k, two whitening keys k_1 and k_2 are used to XOR-mask the plaintext and ciphertext



- It is not a "cure" for inherently weak ciphers
- The additional computational load is negligible
- Its main application is ciphers that are relatively strong against analytical attacks but possess too short a key space especially DES (a variant of DES which uses key whitening is called DESX)

Thanks for Your attention.