ICT Course: Introduction to Cryptography

Nguyen Minh Huong

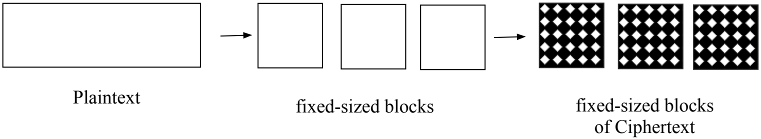
ICT Department, USTH

January 3, 2023

Session 4: Symmetric Cryptography - Block Cipher and Operation Modes

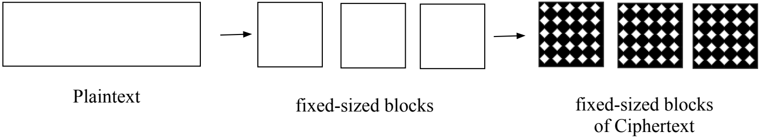
Block ciphers

- General idea:



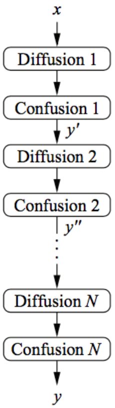
Block ciphers

- General idea:



- Claude Shannon: There are two primitive operations with which strong encryption algorithms can be built:
  - **Confusion**: An encryption operation where the relationship between key and ciphertext is **obscured** (e.g., substitution).
  - **Diffusion**: An encryption operation where the **influence of one plaintext symbol is spread over many ciphertext symbols** with the goal of hiding statistical properties of the plaintext (e.g., permutation).

- Combining the two primitives to build a so called **product ciphers**
- Most of today's ciphers are product ciphers as they consist of rounds which are applied repeatedly to the data.



Common block ciphers

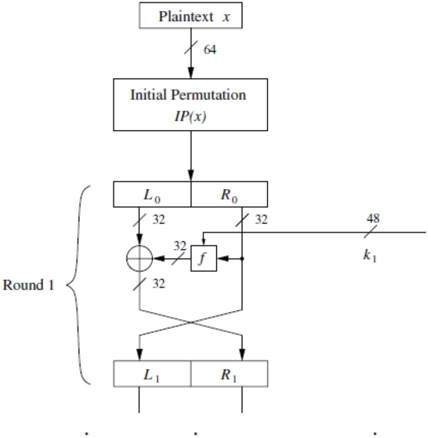
- Feistel Cipher
- DES
- AES

Feistel Cipher

???

Feistel network structure

Feistel network structure in DES

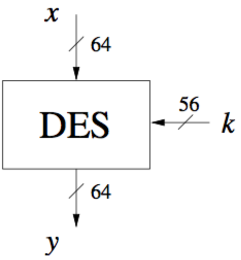


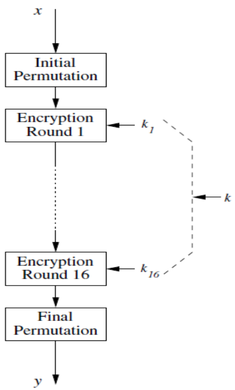
Data Encryption Standard

DES overview:

- Developed by IBM, based on Lucifer cipher
- Standardized in 1977 by the National Bureau of Standards (NBS) today called National Institute of Standards and Technology (NIST)
- Features:
  - Key length: ?
  - Block length?
  - how many rounds?
  - Subkey length?

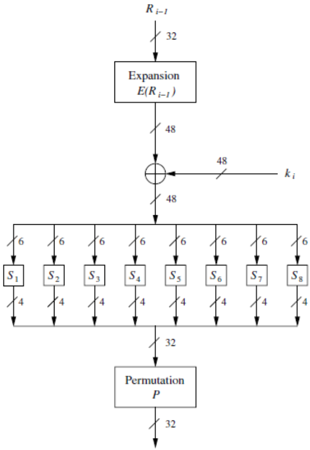
DES Algorithm





DES Algorithm - Encryption round

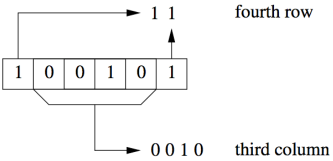
- Feistel encryption
- F-funtion



S-box substitution

- Eight substitution tables
- 6 bits of input, 4 bits of output
- Non-linear, crucial element for DES security

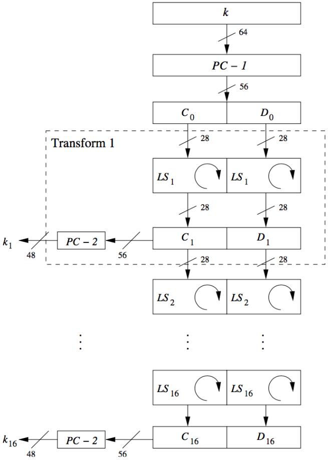
$$S(x_1) \oplus S(x_2) \neq S(x_1 \oplus x_2)$$



S_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	01	10	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

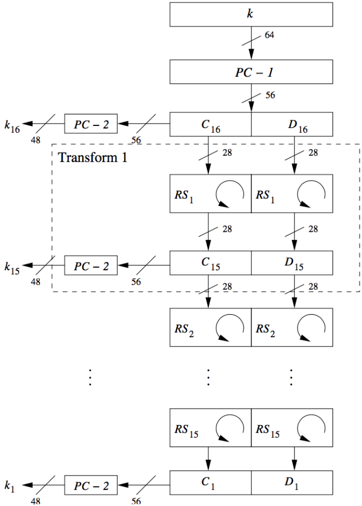
Key schedule

- PC-1: reduce 64 bits to 56 bits by ignoring every 8 bits and permute
- Round 1, 2, 9, 16: rotate left by 1 bits, others by 2 bits
- Rotation is only taken place within either left or right half



Decryption

Key schedule is reversed in decryption



Summary of DES

From above description of DES algorithm and key schedule, what we can summarize?

- DES Features:
  - Key length: 56 bits
  - Block lenghth: 64 bits
  - Subkey length: 48 bits
  - Number of rounds: 16
- 56-bit key is susceptible to an exhaustive key search

Advanced Encryption Standard

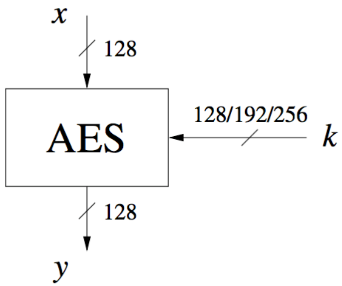
AES overview:

- The most widely used symmetric cipher today
- Approved by US NIST in 2001 after years of
- byte-oriented cipher

AES Encryption Algorithm

Features:

- block size: 128 bits
- 3 key lengths: 128/192/ 256 bits (subkey length =?)
- Number of rounds: 10/12/14
- Each round: 3 layers

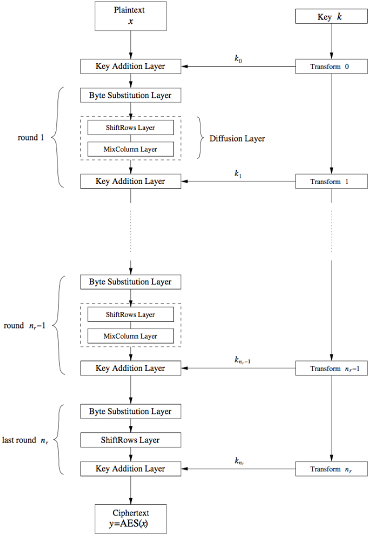


AES Encryption Algorithm

3 layers:

- Key addition layer
- Byte substitution layer
- Diffusion layer

Data path (state): 128 bits = 16 bytes



AES Internal structure

Plaintext block is arranged in matrix of 16 bytes

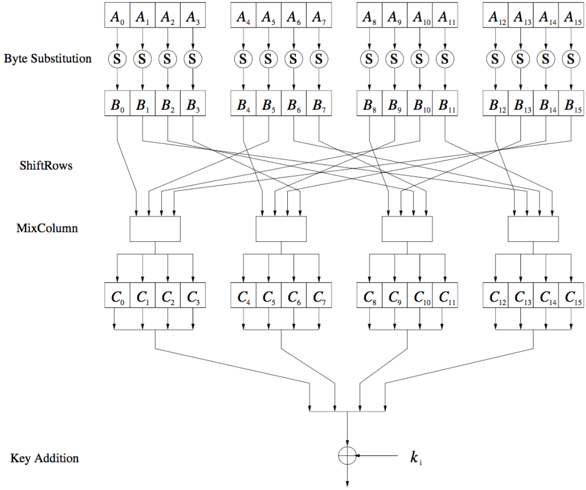
A_0	A_4	A_8	A_{12}
A_1	A_5	A_9	A_{13}
A_2	A_6	A_{10}	A_{14}
A_3	A_7	A_{11}	A_{15}

Key bytes are arranged in matrix of 4 rows x n columns

k_0	k_4	k_8	k_{12}	k_{16}	k_{20}
k_1	k_5	k_9	k_{13}	k_{17}	k_{21}
k_2	k_6	k_{10}	k_{14}	k_{18}	k_{22}
k_3	k_7	k_{11}	k_{15}	k_{19}	k_{23}

AES Internal structure - Round function

Round function for round 1,2,...,n_{r-1}



AES Round function - Byte substitution layer

- Consist of 16 S-boxes
- S-boxes are:
  - identical
  - non-linear
  - one-to-one mapping of input and output
- In implementation, S-boxes are realized as a look up table (mapping)

AES Round function- Diffusion layers

- provides diffusion
- consist of two sublayers:
  - ShiftRows sublayer: permutation
  - MixColumn sublayer: mixes block of 4 bytes
- perform linear operation on state matrices A and B (input matrices)

$$Diff(A) + Diff(B) = Diff(A + B)$$

ShiftRows sublayer

Rows of state matrix are shifted cyclically:

B_0	B_4	B_8	B_{12}
B_1	B_5	B_9	B_{13}
B_2	B_6	B_{10}	B_{14}
B_3	B_7	B_{11}	B_{15}

B_0	B_4	B_8	B_{12}	no shift
B_5	B_9	B_{13}	B_1	$\longleftarrow$ one position left shift
B_{10}	B_{14}	B_2	B_6	$\longleftarrow$ two positions left shift
B_{15}	B_3	B_7	B_{11}	$\longleftarrow$ three positions left shift

MixColumn sublayer

- Mixes each column of the state matrix

$$MixColumn(B) = C$$

- Each 4-byte column is multiplied by a fixed 4x4 matrix
- All arithmetic is done in Galois Feild GF(2^8)

Key Addition layer

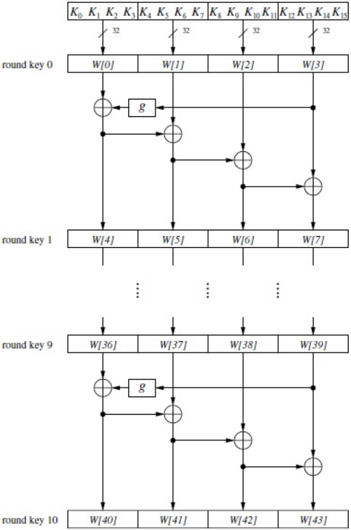
- Input: 16-byte state matrix C and subkey k_j
- Output: $C \oplus k_j$

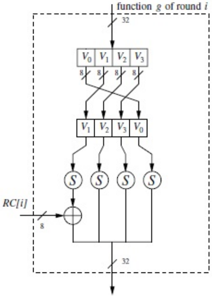
AES Key schedule

- 1 word = 32 bits
- In each round i, subkey k_i = 4 words $W[4i + j]$, $j = 0..3$
- Different key schedule for different key size

AES Key schedule - 128-bit key

- The leftmost word: $W[4i] = W[4(i-1)] \oplus g(W[4i-1])$
- The remain words: $W[4i+j] = W[4i+j-1] \oplus W[4(i-1)+j]$


K_0 K_1 K_2 K_3 K_4 K_5 K_6 K_7 K_8 K_9 K_{10} K_{11} K_{12} K_{13} K_{14} K_{15}
32
round key 0
W[0] W[1] W[2] W[3]
g
round key 1
W[4] W[5] W[6] W[7]
round key 9
W[36] W[37] W[38] W[39]
g
round key 10
W[40] W[41] W[42] W[43]


function g of round i
32
V_0 V_1 V_2 V_3
8
V_1 V_2 V_3 V_0
S
RC[i]
8
32

- g function:
    - S-box substitution: adds nonlinearity to the schedule
- RC[i]: a round coefficient, 8-bit value, vary from round to round

AES Decryption

All layers must be inverted for decryption:

- Inv MixColumn layer: inverse of the 4x4 matrix
- Inv ShiftRows layer: all the state of the state matrix B are shifted in the opposite direction
- Inv Byte substitution: inverse S-Box

Key schedule for decryption:

- Subkeys are needed in reversed order

Implementation in Software

- One requirement of AES was the possibility of an efficient software implementation
- Straightforward implementation is well suited for 8-bit processors (e.g., smart cards), but inefficient on 32-bit or 64-bit processors
- A more sophisticated approach: Merge all round functions (except the key addition) into one table look-up
  - This results in four tables with 256 entries, where each entry is 32 bits wide
  - One round can be computed with 16 table look-ups
- Typical SW speeds are more than 1.6 Gbit/s on modern 64-bit processors

AES Security

- Brute-force attack: ?
- Analytical attack: There is currently no analytical attack against AES known which has a complexity less than a brute-force attack

Mode of Operation

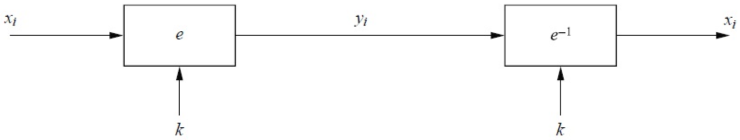
How to encrypt more than one single block of plaintext?

- Electronic Code Book Mode (ECB)
- Cipher Block Chaining Mode (CBC)
- Cipher Feedback Mode (CFB)
- Output Feedback Mode (OFB)
- Counter mode (CTR)
- others

They provides: confidentiality, integrity and authenticity

Electronic Code Book Mode

- The length of the plaintext must be an exact multiple of the block size of the cipher, if not it must be padded
- Each block is encrypted separately
- b-bit plaintext block x_i has b bits, b-bit ciphertext block y_i
- Message exceeding b-bit must be partitioned into b-bit blocks



ECB Mode

Let e() be a block cipher of block size b, and let x_i and y_i be bit strings of length b.

- Encryption: $y_i = e_k(x_i), i \geq 1$
- Decryption: $x_i = e_k^{-1}(e_k(x_i)), i \geq 1$

ECB security

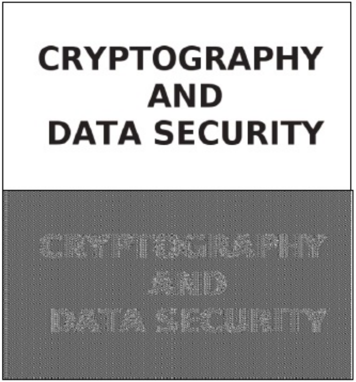
Substitution Attack:

- Example: electronic bank transfer: Block #1.Sending Bank A #2.Sending Account #3.Receiving Bank B #4.Receiving Account #5. Amount
- Attacker: transfers repeatedly from his account in bank A to his account in bank B
  - He obtains ciphertext for block 1,3,4
  - replaces block 4 of other transfers with his block 4 then all transfers are redirected to his account

ECB security

Encrypting bipmap in ECB mode:

- Statistical properties in the plaintext are preserved in the ciphertext



ECB Advantages and Disadvantages

Advantages:

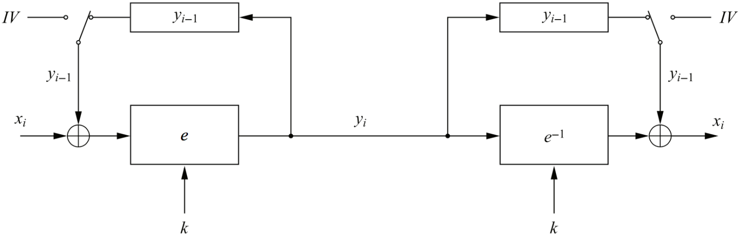
- no block synchronization between sender and receiver is required
- bit errors caused by noisy channels only affect the corresponding block but not succeeding blocks
- Block cipher operating can be parallelized, then advantage for high-speed implementations

Disadvantages: ECB encrypts highly **deterministically**

- identical plaintexts result in identical ciphertexts
- an attacker recognizes if the same message has been sent twice
- plaintext blocks are encrypted independently of previous blocks
- an attacker may reorder ciphertext blocks which results in valid plaintext

Cipher Block Chaining Mode

- The encryption of all blocks are chained together
- The encryption is randomized by using an initialized vector (IV)



CBC-Encryption and Decryption

Encryption and Decryption in CBC mode

Let e() be a block cipher of block size b; let x_i and y_i be bit strings of length b; and IV be a **nonce of length b.**

- **Encryption (first block):** $y_1 = e_k(x_1 \oplus IV)$
- **Encryption (general block):** $y_i = e_k(x_i \oplus y_{i-1}), i \geq 2$
- **Decryption (first block):** $x_1 = {e_k}^{-1}(y_1) \oplus IV$
- **Decryption (general block):** $x_i = {e_k}^{-1}(y_i), i \geq 2$

We do not have to keep IV secret. Why?

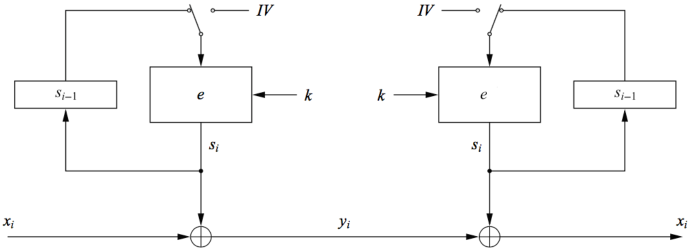
Substitution attack on CBC

Example of electronic bank transfer

- If IV is chosen for every wire transfer, attack will not work
- If IV is kept the same, the attacker would recognize
- Why?

Output Feedback Mode

- Is used to build synchronous stream cipher from a block cipher
- key stream is generated blockwise fashion
- Output of the cipher: key stream bits S_i to encrypt plaintext bits using the XOR operation



OFB - Encryption and Decryption

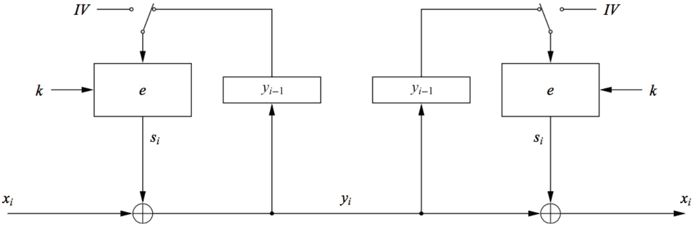
Encryption and decryption in OFB mode

Let e() be a block cipher of block size b; let x_i, y_i and s_i be bit strings of length b; and IV be a nonce of length b.

- Encryption(first block): $s_1 = e_k(IV)$ and $y_1 = s_1 \oplus x_1$
- Encryption (general block): $s_i = e_k(s_{i-1}), i \geq 2$
- Decryption (first block): $s_1 = e_k(IV)$ and $x_1 = s_1 \oplus y_1$
- Decryption (general block): $s_i = e_k(s_{i-1})$ and $x_i = s_i \oplus y_i, i \geq 2$

Cipher Feedback Mode

- Same requirement of plaintext size as ECB
- Uses block cipher as building block for an asynchronous stream cipher
- Key stream S_i is generated in blockwise fashion and function of ciphertext
- **non-deterministic** (if the IV is a nonce)



CFB-Encryption and Decryption

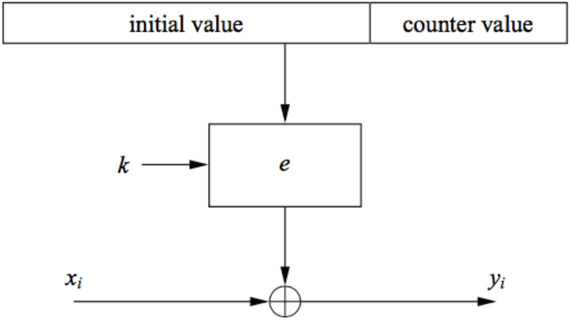
Encryption and Decryption in CFB mode

Let e() be a block cipher of block size b; let x_i and y_i be bit strings of length b; and IV be a nonce of length b.

- Encryption (first block): $y_1 = e_k(IV) \oplus x_1$
- Encryption (general block): $y_i = e_k(y_{i-1}) \oplus x_i, i \geq 2$
- Decryption (first block): $x_1 = e_k(IV) \oplus y_1$
- Decryption (general block): $x_i = e_k(y_{i-1}) \oplus y_i, i \geq 2$

Counter Mode

- Uses a block cipher as a stream cipher
- Key stream is generated in blockwise fashion
- The counter assumes a different value everytime a new key stream block is computed



CTR - Encryption and Decryption

Encryption and Decryption in CTR mode

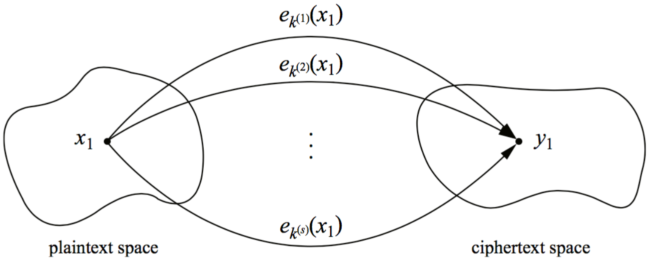
Let e() be a block cipher of block size b, and let x_i and y_i be bit strings of length b. The concatenation of the initialization value IV and the counter CTR_i is denoted by $(IV||CTR_i)$ and is a bit string of length b.

- Encryption: $y_i = e_k(IV||CTR_i) \oplus x_i, x \geq 1$
- Decryption: $x_i = e_k(IV||CTR_i) \oplus y_i, i \geq 1$

Exhaustive key search revisited

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
- brute-force attack is still possible, but several pairs of plaintext–ciphertext are needed



Exhaustive key search revisited

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), ..., (x_t, y_t), the likelihood of false keys which encrypt all plaintexts to the corresponding ciphertexts is:

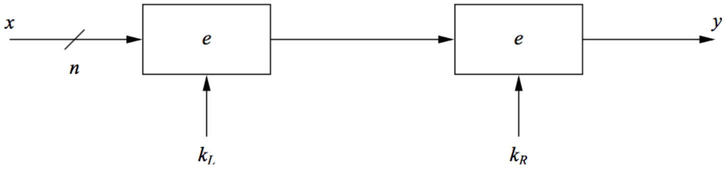
$$2^{k-t*n}$$

Increasing the security of Block cipher

- Multiple Encryption
- Key whitening

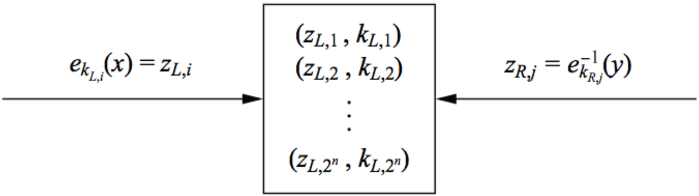
Multiple Encryption- Double Encryption

A plaintext x is first encrypted with a key k_L, and the resulting ciphertext is encrypted again using a second key k_R



Key lengths: k bits, then key space (number of encryptions) =?

Double Encryption- Meet-in-the-middle attack



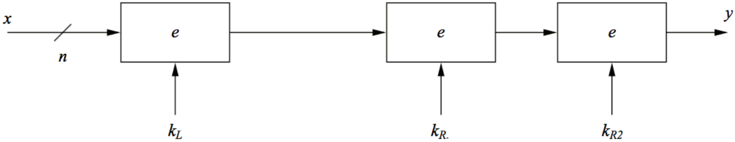
- Phase 1: encryption to compute the look up table $Z_{L,i}$
- Phase 2: decryption (result $Z_{R,i}$) to check whether any $Z_{R,i}$ equal to $Z_{L,i}$

The complexity or number of encryption or decryption = ?

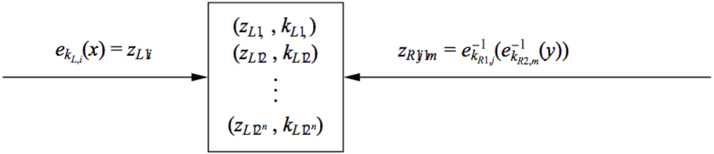
How much it is more secure than single encryption?

Triple Encryption

Encrypt a block three times:



Meet-in-the-middle attack:



The attack reduces the **effective key length** from $3 * k$ to $2 * k$

Key whitening

Make block cipher more resistant against brute-force attack

Key whitening for block ciphers

- Encryption: $y = e_{k,k_1,k_2} = e_k(x \oplus k_1) \oplus k_2$
- Decryption: $x = e^{-1}{}_k, k_1, k_2(y) = e_k{}^{-1}(y \oplus k_2) \oplus k_1$

Security of key whitening: key length k bits, block length n bits

- A naive brute-force attack: $2^{(k+2*n)}$ search steps
- Meet-in-the-middle attack: $2^{(k+n)}$ search steps