ROYAL UNIVERSITY
OF PHNOM PENH

IC: Chapter 2

Symmetric Encryption & Message Confidentiality

19 Feb 2022 by Lecturer Chap Chanpiseth

Outline

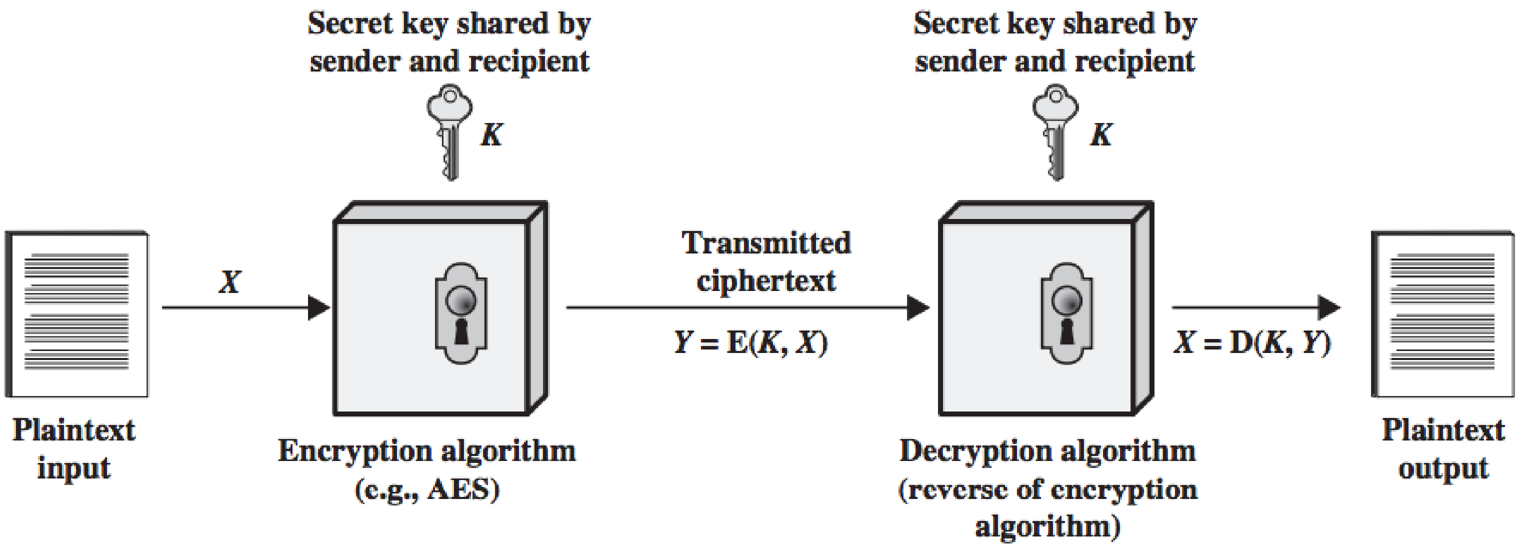
1) Symmetric Encryption Principles
2) Cryptography and Cryptanalysis
3) Perfect Secrecy scheme vs. Computationally secure
4) Feistel Encryption and Decryption
5) Symmetric Block Encryption Algorithms
6) DES: Data Encryption Algorithm

Symmetric Encryption Principles

Symmetric Encryption Scheme has 5 ingredients

1) **Plaintext:** original message or data fed into algorithm as input
2) **Encryption algorithm:** perform various substitutions and transformations on plaintext
3) **Secret Key:** Fed as input for algorithm to encrypt or decrypt the message/data.
4) **Ciphertext:** Scrambled message produced as output. Given 2 distinct keys will generate 2 different ciphertexts.
5) **Decryption algorithm:** A reverse to encryption algorithm. It takes ciphertexts and the same secret key then generates the original message.

Simplified Model of Symmetric Encryption



How to securely use Symmetric Encryption

2 vital requirements for secure use of symmetric encryption:

1) **A strong encryption algorithm**
   - Algorithm is known publicly
   - Ciphertexts could be accessed by everyone

- The adversaries should be unable to decrypt ciphertexts
- OR discover the key if he/she possesses a number of ciphertexts and the plaintext that produced each ciphertext

2) The secret key **MUST be securely shared** between sender and receiver, and the secret key MUST be **kept securely.**

=> The security of symmetric encryption depends on the secret key NOT algorithm.

Cryptography

Cryptographic systems are characterized along three independent dimensions:

1) **Two general Operations for plaintext to ciphertext transformation:**
   i. **Substitution:** each element is replaced/mapped into another element;
   ii. **Transposition:** the position of elements in plaintext is rearranged.
2) **The number of key used** between sender and receiver: a single key or symmetric key
3) **The technique to process plaintext:**
   - A ***block cipher*** processes the input one block of elements at a time and produce an output for each block
   - A ***stream block*** processes the input elements continuously, producing one output at a time

The fundamental requirement is that no information be lost (all operations are reversible). Product systems involve multiple stages of substitutions and transpositions.

Cryptanalysis

Cryptanalysis is the process of attempting to find the plaintext or secret key.

The strategy used by cryptanalysis depends on:

- The nature of the encryption algorithm
- The information in the hand of cryptanalyst

Cryptanalysis

Four Type of Attacks on Encrypted Messages

Type of Attack	Known to Cryptanalyst	Strategy used by Cryptanalyst
Ciphertext only	• Encryption algorithm • Ciphertext to be decoded	Brute-force approach: trying all possible keys => impractical, but possible with weak algorithm
Known plaintext	• Encryption algo • Ciphertext to be decoded • One or more plaintext-ciphertext pair formed with the secret key captured by opponent • Or the analyst may know that certain plaintext patterns will appear in a message.	Deduce the secret key with the basis of the knowledge of known plaintext patterns that appears in the message.
Chosen plaintext	• Encryption algo • Ciphertext to be decoded • Plaintext message chosen by cryptanalyst along with its corresponding ciphertext produced with key	Choose the messages to encrypt => deliberately pick patterns that can be expected to disclosed the structure of key.

Cryptanalysis (Cont.)

Four Type of Attacks on Encrypted Messages

Type of Attack	Known to Cryptanalyst	Strategy used by Cryptanalyst
Chosen ciphertext attack	• Encryption algorithm • Ciphertext to be decoded • Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key	• The attacker is assumed to have a way to trick someone who knows the secret key into decrypting arbitrary message blocks and tell him the result. • The attacker can choose some arbitrary nonsense as an "encrypted message" and ask to see the (usually) different nonsense it decrypts to, and he can do this a number of times.

Type of Attacks on Encrypted Messages: Ciphertext only

❖ Given

- Cipher text of many messages, encrypted with same key
- $C_1 = E_k(P_1), C_2 = E_k(P_2), \dots \dots \dots \dots C_i = E_k(P_i)$

❖ Task

- Find plain text of these messages or even better the key
- Find $P_1, P_2, \dots \dots \dots \dots \dots \dots \dots . . P_i \; or \; K \; or \; P_{i+1}$

❖ Tips

- The opponent must have some general idea of the type of plaintext that is concealed
  - English or French text
  - an EXE file
  - a Java source listing
  - an accounting file, and so on.

Type of Attacks on Encrypted Messages: Known plaintext

❖ Given

- Cipher text and Plain text of the corresponding messages
- $P_1, C_1 = E_k(P_1), P_2, C_2 = E_k(P_2), \dots \dots \dots \dots P_i, C_i = E_k(P_i)$

❖ Task

- **Find key**

❖Tips

- A file that is encoded in the Postscript format always begins with the same pattern,
- Or there may be a standardized header or banner to an electronic funds transfer message, and so on.

Type of Attacks on Encrypted Messages: Chosen plaintext attack

❖ Given

- Plain text and Cipher text pairs
- Can choose plain text that gest encrypted
- $P_1, C_1 = E_k(P_1), P_2, C_2 = E_k(P_2), \dots \dots \dots \dots P_i, C_i = E_k(P_i)\ where\ P_1, P_2 \dots \dots P_i\ can\ be\ chosen.$

❖ Task

- Reveal the structure of the key

❖Tips

- If the analyst is able somehow to get the source system to insert into the system a message chosen by the analyst, then a chosen-plaintext attack is possible.
- In general, if the analyst is able to choose the messages to encrypt, the analyst may deliberately pick patterns that can be expected to reveal the structure of the key.

Perfect Secrecy scheme vs. Computationally secure

Perfect Secrecy means that the ciphertext provides no information about the content of the plaintext.

- No matter how much ciphertext the cryptanalyst has, it does not convey anything about what the plaintext and key were.
- As much key material as plaintext to encrypt. (One-time pad)

An encryption scheme is computationally secure if the ciphertext produced by scheme meets one or both of the criteria:

- The cost of breaking the cipher is more expensive than the value of the encrypted information
- The time required to break the cipher is longer than the useful lifetime of the information

Average time required for exhaustive key search

The below table considers the results for a system that

- Can process 1 million keys per microsecond (μs).
- DES no longer can be considered computationally secure.

Key Size (bits)	Number of Alternative Keys	Time Required at 1 Decryption/μs	Time Required at 10^6 Decryptions/μs
32	$2^{32} = 4.3 \times 10^9$	$2^{31} \mu s = 35.8$ minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	$2^{55} \mu s = 1142$ years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	$2^{127} \mu s = 5.4 \times 10^{24}$ years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	$2^{167} \mu s = 5.9 \times 10^{36}$ years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	$2 \times 10^{26} \mu s = 6.4 \times 10^{12}$ years	6.4×10^6 years

Equivalent key strength between Symmetric key and Asymmetric key

Symmetric Key Size	RSA Key Size	Elliptic Curve Key Size
80	1024	160
112	2048	224
128	3072	256
192	8192	384
256	15360	521

Feistel Encryption and Decryption

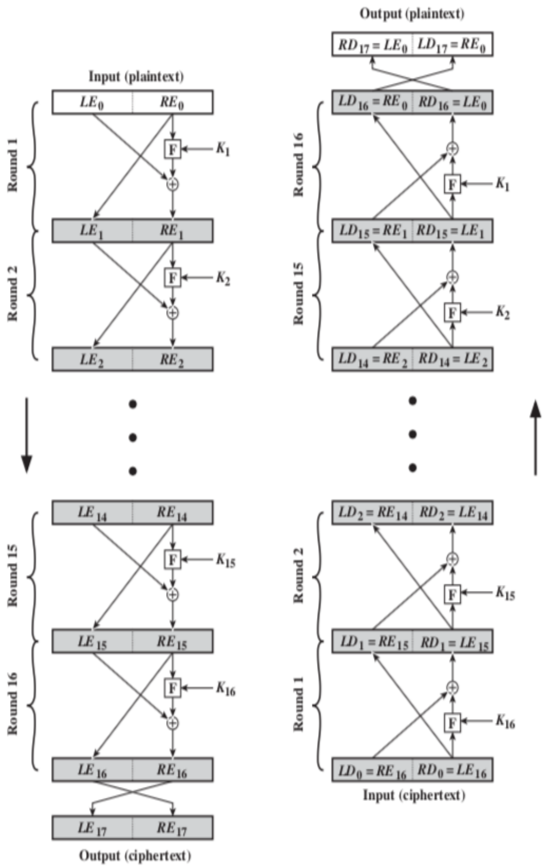
- Developed by IBM in 1973
- Feistel Encryption is used by many symmetric block encryption

- **Encryption algorithm:**
  - Input a plaintext of length ***2w*** and a key ***K***.
  - Plaintext block is divided into two halves, LE_0 and RE_0.
  - The 2 halves of the data pass through ***n rounds of processing*** => *Combine to produce the ciphertext block.*
  - Each round i has as inputs LE_{i-1} and RE_{i-1} derived from previous round & corresponding subkey k_i
    - k_i is ***derived*** from the overall K.
    - The subkeys k_i are different from K
    - k_i is generated from the key by a subkey generation algorithm.

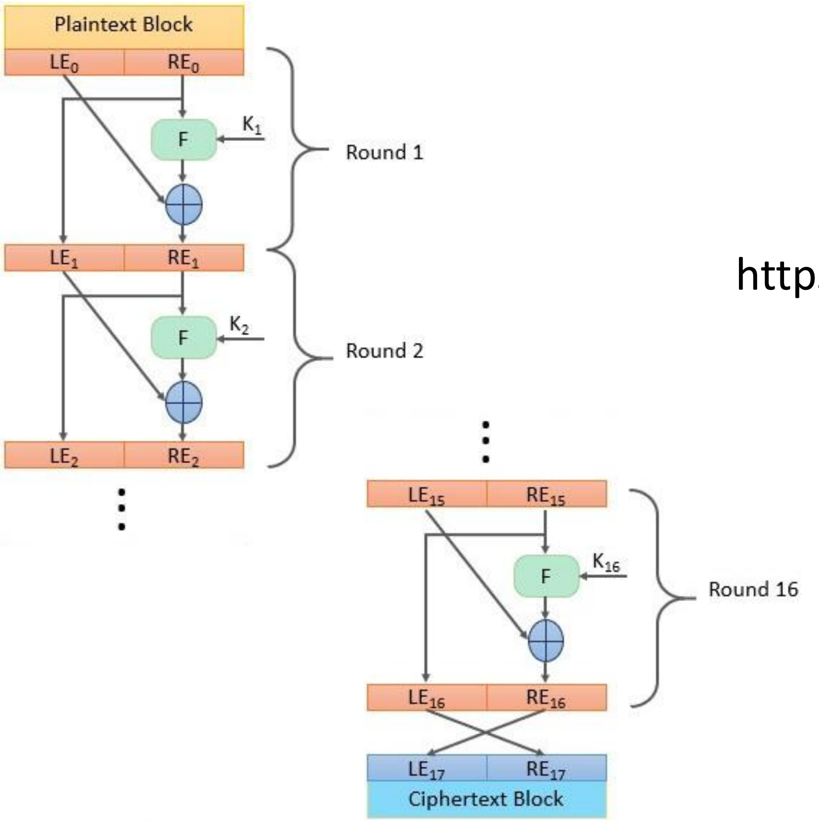
❖ All round have the same structure

❖ Substitution performed on the LE_{i-1} block by applying a *round function F to the* RE_{i-1} *and then taking the exclusive (XOR) of the output and the left half* LE_{i-1}.

Feistel Encryption and Decryption (16 rounds)



Feistel Encryption and Decryption (16 rounds)



https://www.youtube.com/watch?v=drI2shandyk

General design of a symmetric block

The exact realization of a symmetric block cipher depends on the choice of

1) Parameters and
2) Design features.

❖ **Five principle and two additional parameters and features** are considered for designing and evaluating a symmetric block algorithm.

General design of a symmetric block: Parameters and Design features

- **Block size:**
  - Larger block sizes mean greater security
  - but reduced encryption/decryption speed.
  - A block size of 128 bits is a reasonable tradeoff and
  - Nearly universal among recent block cipher designs.
- **Key size:**
  - Larger key size means greater security
  - But may decrease encryption/ decryption speed.
  - The most common key length in modern algorithms is 128 bits.

General design of a symmetric block: Parameters and Design features (Cont.)

- **Number of rounds:**
  - The essence of a symmetric block cipher is that a single round offers inadequate security
  - But that multiple rounds offer increasing security. A typical size is 16 rounds.
- **Subkey generation algorithm:**
  - Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis.
- **Round function:**
  - Greater complexity generally means greater resistance to cryptanalysis.

General design of a symmetric block: Parameters and Design features (Cont.)

There are two other considerations in the design of a symmetric block cipher:

- **Fast software encryption/decryption:**
  - Encryption is embedded in applications or utility functions in such a way as to preclude a hardware implementation.
  - Accordingly, the speed of execution of the algorithm becomes a concern.
- **Ease of analysis:**
  - Although a possible strong algorithm to cryptanalyze is wanted, BUT the easy algorithm for analysis.
  - If the algorithm can be concisely and clearly explained, it is easier to analyze that algorithm for cryptanalytic vulnerabilities and
  - Therefore, develop a higher level of assurance as to its strength. DES, for example, does not have an easily analyzed functionality.

Decryption of a symmetric block cipher

- **In symmetric block cipher, Decryption is essentially the same as the encryption process.**
- **The rule is as follows:**
  - Use the ciphertext as input to the algorithm, but use the subkeys K_i in reverse order.
  - Use K_n in the first round, K_{n-1} in the second round, and so on until K_1 is used in the last round.

❖ ***A nice feature:*** no need to implement two different algorithms i.e., one for encryption and one for decryption.

Symmetric Block Encryption Algorithms

The most commonly used symmetric encryption algorithms are block ciphers.

A block cipher:

- Process the plaintext input: **fixed-sized blocks and**
- Produces a block of ciphertext of **equal size for each plaintext block.**

Three most important symmetric block ciphers:

1) Data Encryption Standard (DES),
2) triple DES (3DES), and
3) Advanced Encryption Standard (AES).

DES: Data Encryption Algorithm

Issued in 1977, Data Encryption Standard (DES) is the most widely used encryption scheme

- As Federal Information Processing Standard 46 (FIPS 46) by the National Bureau of Standards, now known as the National Institute of Standards

Data encryption standard (DES) has been found vulnerable against very powerful attacks and therefore, the popularity of DES has been found slightly on decline.

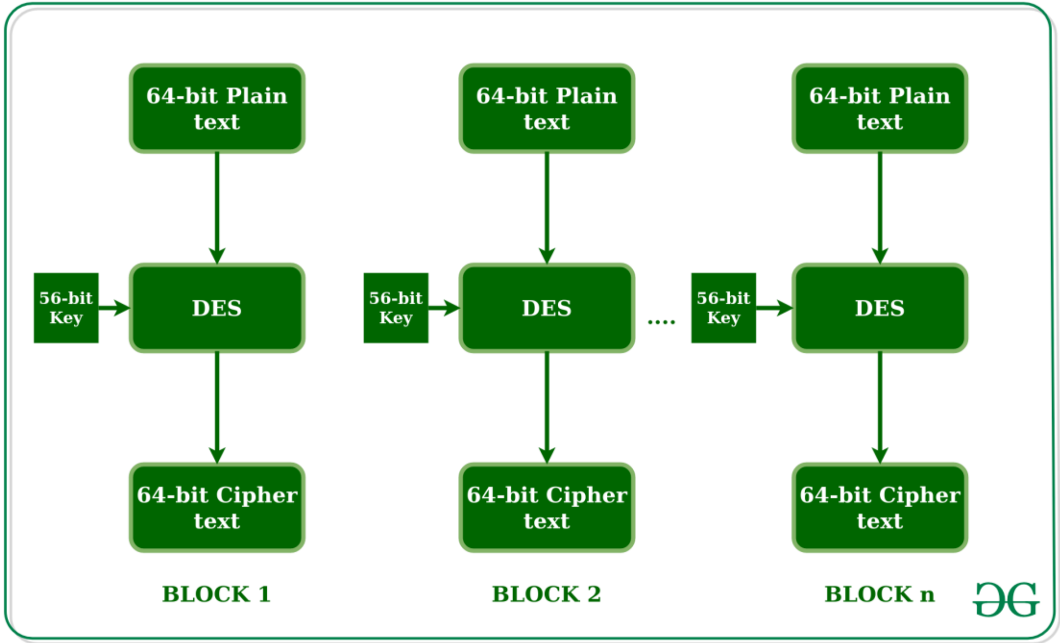
DES: Data Encryption Algorithm (Cont.)

Description of the Algorithm

- DES is a block cipher,
- Encrypts data in blocks of size of 64 bit each, means 64 bits of plain text goes as the input to DES
- Produces 64 bits of cipher text.
- The same algorithm and key are used for both encryption and decryption
- The key length is 56 bits.
- Decryption with DES is essentially the same as the encryption process.
- The rule of decryption is as follows:
    - Use the ciphertext as input to the algorithm BUT use the subkeys K_i in reverse order.
    - Use K_{16} in the first round, K_{15} in the second round, and so on until K_1 is used in the 16th round.

DES: Data Encryption Algorithm (Cont.)

The high-level design concept of DES is show in figure.



DES: Data Encryption Algorithm (Cont.)

The Strength of DES: two categories of concerns

1) **The algorithm:** the possibility that cryptanalysis is possible to find and exploit fatal weaknesses in the algorithm.
2) **The use of a 56-bit key: resistance to brute-force attack?**
   - With a key length of 56 bits => there exists 256 possible keys ~ approximately 7.2×10^{16} keys.
   - *Theory: a brute- force attack appears impractical.*
     - Assuming that on average half the key space has to be searched, a single machine performing one DES encryption per microsecond would take more than a thousand years to break the cipher.
   - July 1998, the Electronic Frontier Foundation (EFF) announced that it had broken a DES encryption
     - Use a special-purpose "DES cracker" machine built for less than $250,000.
     - Within duration of the attack took less than three days.

DES: Data Encryption Algorithm (Cont.)

A final thought

If the only form of attack is **brute force**, then the way to counter such attacks is obvious: **use longer keys**.

References

Network Security Essentials: Applications and Standards, 4th Edition by William Stallings

Introduction to Cryptography with Coding Theory (2nd Edition) by Wade Trappe Lawrence C. Washington(2005-07-25)

Online sources:

https://www.geeksforgeeks.org/data-encryption-standard-des-set-1/

https://www.simplilearn.com/what-is-des-

article#:~:text=The%20DES%20(Data%20Encryption%20Standard,ciphertext%20using%2048%2Dbit%20keys.

https://www.tutorialspoint.com/cryptography/data_encryption_standard.htm

https://www.youtube.com/watch?v=SaZGjQBItBc&t=11s

https://github.com/filgut1/COMP7401-Feistel-Cipher

Thanks for your attention !