Lecture 2

Basic notions on Cryptology

UTAS
Sultanate of Oman
September 2022

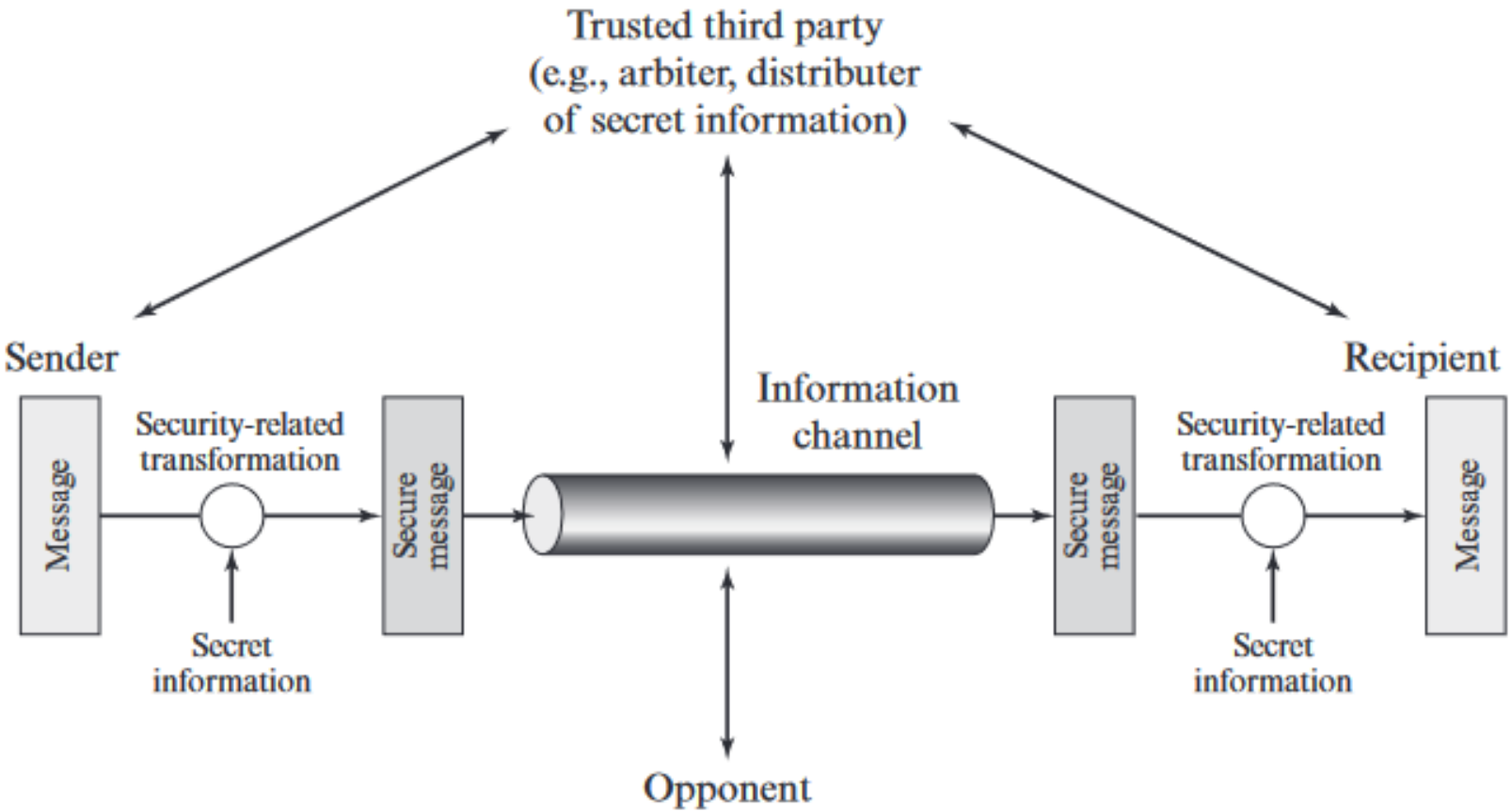
CSSY2201 : Introduction to Cryptography

جامعة التقنية
والعلوم التطبيقية
University of Technology
and Applied Sciences
Salalah صلالة

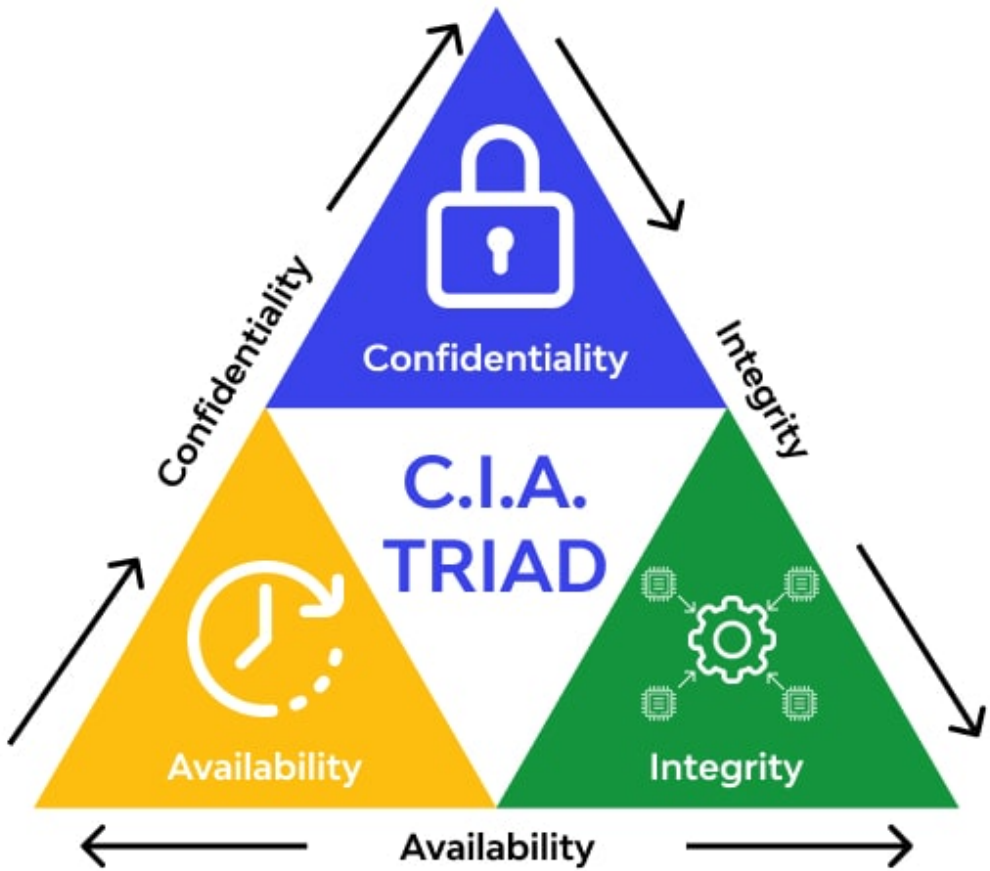
Plan

Plan

Theoretic model of a secure communication



Cryptograhy Objectives : CIA



Others : Authentication, Non-Repudiation

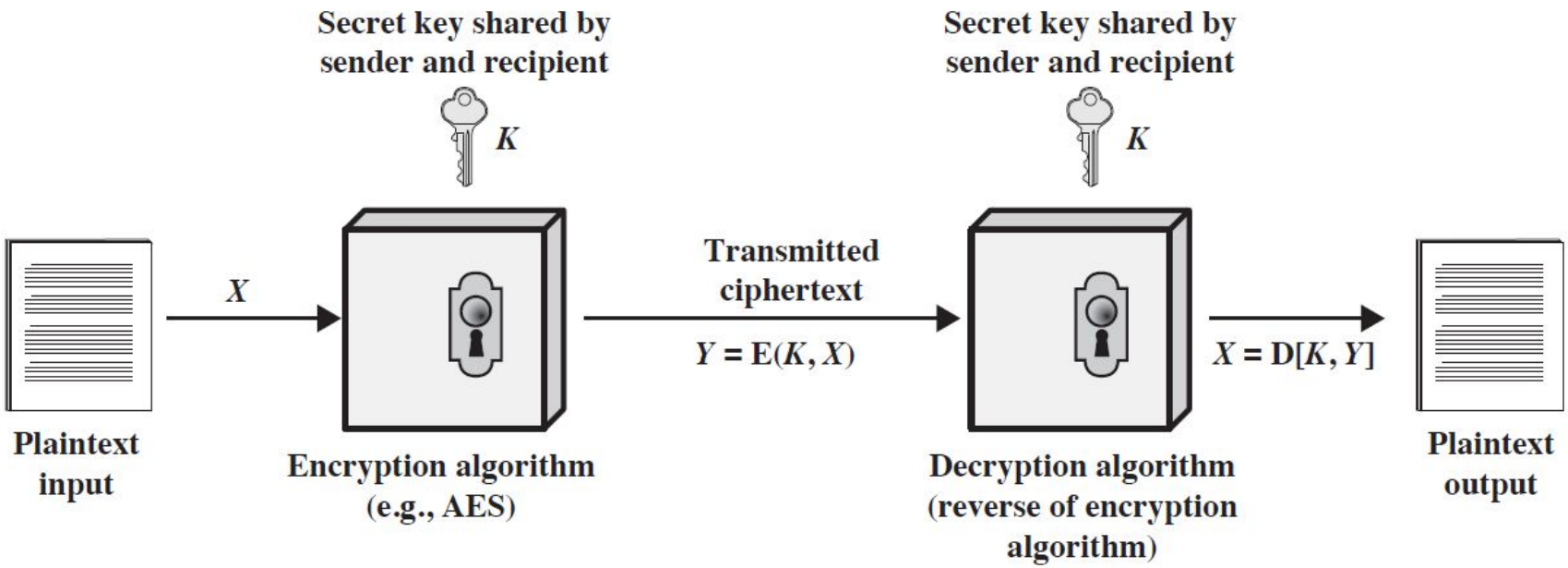
Attacks, services and Mechanisms

- Attack : Any action that compromises information security
- Security Mechanism : A mechanism that is designed to detect, prevent, or recover from a security attack.
- Security Service : A service that improves the security of data processing systems and information transfers. A security service makes use of one or more security mechanisms

Terminology

- Plaintext : original text
- Ciphertext : encrypted text
- Encryption : The process of conversion from plaintext to ciphertext
- Decryption : The process of conversion from ciphertext to plaintext
- Cryptography : Science of secret messages
- Cryptanalysis : Science of breaking secret messages
- Cryptology : science of cryptography & cryptanalysis
- Cryptosystem/cipher : Encryption Algorithm

Symmetric encryption



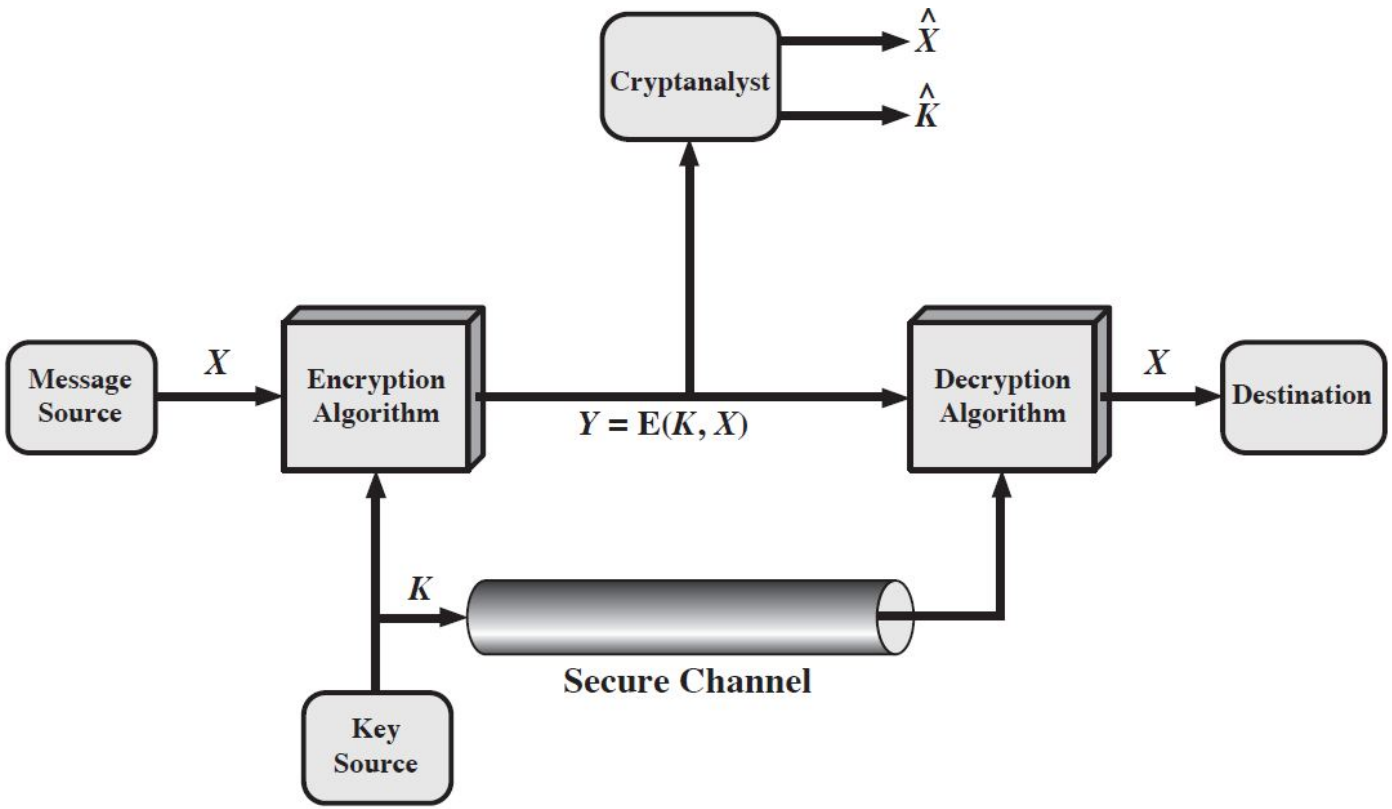
Problems of symmetric cryptography

- symmetric cryptography use only one key for encryption/decryption named (secret key)
- The secret key should be pre-shared between sender and receiver before secret communication.
- if the secret key has been disclosed, then all the communication is compromised.
- does not protect the sender from the receiver which can claim receiving an encrypted message from the sender.
- dose not protect the receiver from someone pretending being the legitimate sender

Plan

Symmetric encryption Model

We introduce the science of breaking secret messages in a symmetric cryptography concept.
here is the model of symmetric cryptography with the presence of a third user (the malicious actor, the hacker, the cryptanalyst)

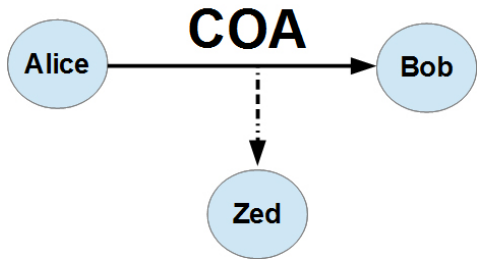


Cryptanalysis

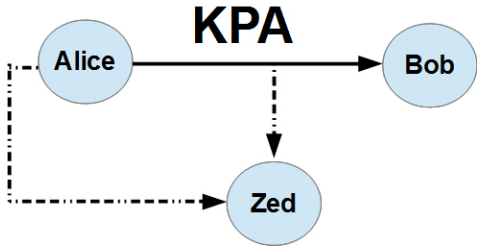
- Objective is to find the secret key not only the plaintext
- brute force attack :
  - Exploit the key length : try all the combinations (on a ciphertext to decrypt it) of the secret key until finding the right one.
  - In average, a hacker needs to try at least half of the key space to break the cryptosystem.
- Cryptanalytic attack : more intelligent, exploits the cryptosystem design vulnerabilities.
- Generally, the cryptanalyst follows 3 steps to break a cryptosystem : (1) Data collection, (2) Analysis, deduction, (3) Exploitation.

Data Collection : Theoretic weaknesses

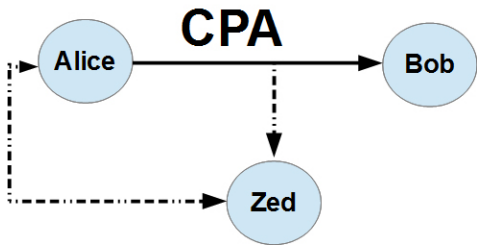
- Observation or action : the hacker is "on-line" connected to the target.
  - Ciphertext-only attack(COA)



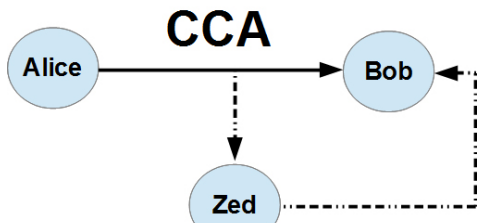
  - Known-plaintext attack(KPA)



  - Chosen-plaintext attack(CPA)

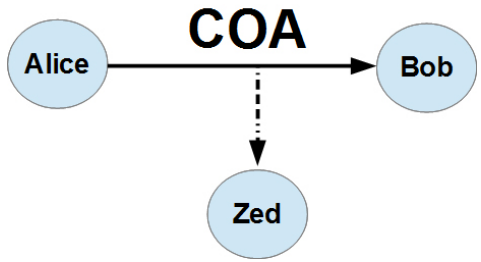


  - Chosen-ciphertext attack(CCA)

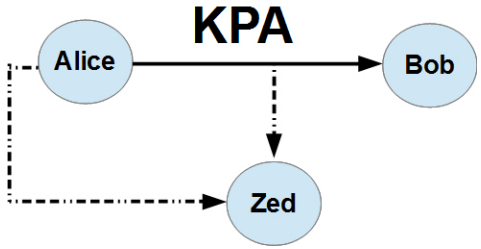


Data Collection : Theoretic weaknesses

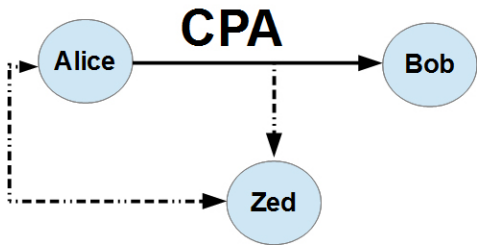
- Observation or action : the hacker is "on-line" connected to the target.
  - Ciphertext-only attack(COA)



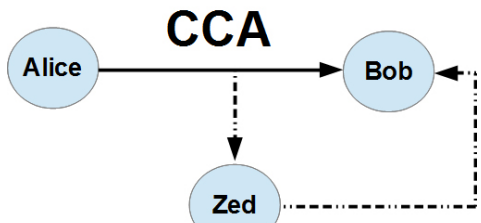
  - Known-plaintext attack(KPA)



  - Chosen-plaintext attack(CPA)

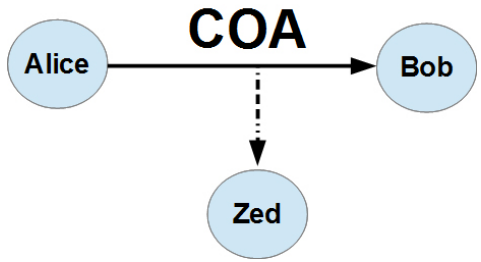


  - Chosen-ciphertext attack(CCA)

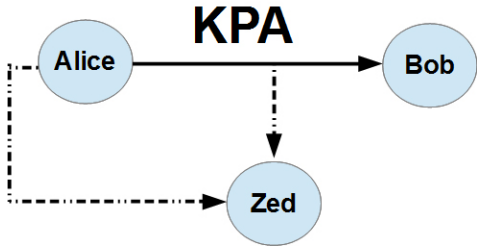


Data Collection : Theoretic weaknesses

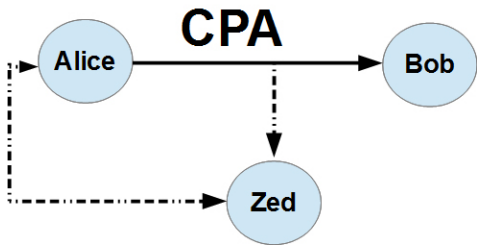
- Observation or action : the hacker is "on-line" connected to the target.
  - Ciphertext-only attack(COA)



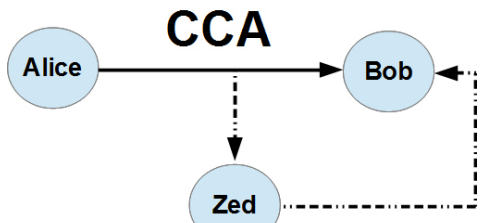
  - Known-plaintext attack(KPA)



  - Chosen-plaintext attack(CPA)

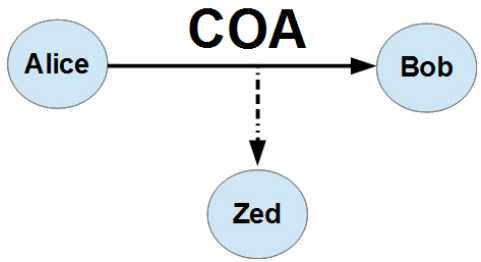


  - Chosen-ciphertext attack(CCA)

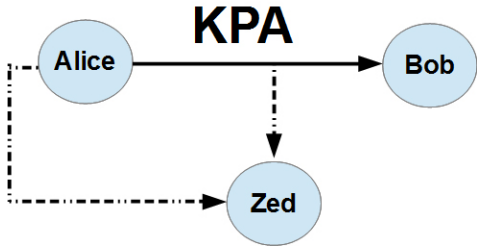


Data Collection : Theoretic weaknesses

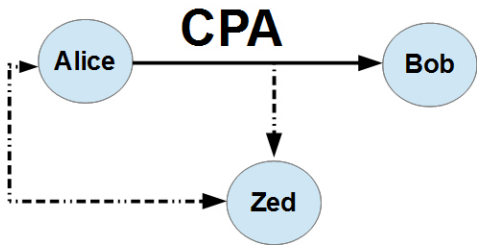
- Observation or action : the hacker is "on-line" connected to the target.
  - Ciphertext-only attack(COA)



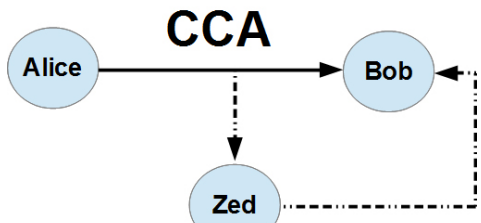
  - Known-plaintext attack(KPA)



  - Chosen-plaintext attack(CPA)



  - Chosen-ciphertext attack(CCA)



Analysis, deduction and Exploit

- "off-line" step : Analysis & Deduction
  - Brute force attack : try all the combinations of the key to find the plaintext from the ciphertext
  - statistical attack : Estimate the occurrence frequency of letters in a text
  - Algebraic attack : try to find equivalent representation of the encryption algorithm to simplify it.
  - Linear cryptanalysis : Linear approximation of the encryption algorithm (which is a non-linear system)
  - Differential cryptanalysis : Study how the plaintexts difference propagate and affect the ciphertext difference to find unbalanced output.
- Exploit : Estimate the decryption key.

Analysis, deduction and Exploit

- "off-line" step : Analysis & Deduction
  - Brute force attack : try all the combinations of the key to find the plaintext from the ciphertext
  - statistical attack : Estimate the occurrence frequency of letters in a text
  - Algebraic attack : try to find equivalent representation of the encryption algorithm to simplify it.
  - Linear cryptanalysis : Linear approximation of the encryption algorithm (which is a non-linear system)
  - Differential cryptanalysis : Study how the plaintexts difference propagate and affect the ciphertext difference to find unbalanced output.
- Exploit : Estimate the decryption key.

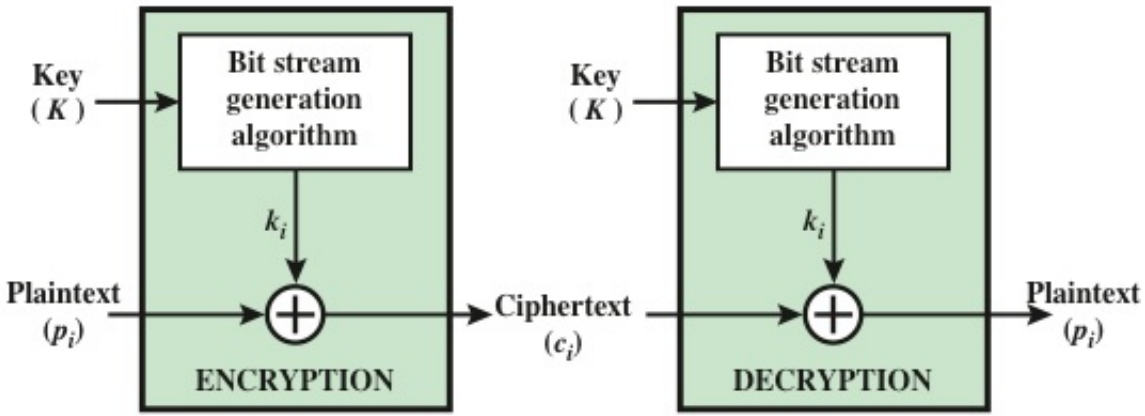
Examples of brute force attack

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} μs = 35.8 minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	2^{55} μs = 1142 years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	2^{127} μs = 5.4×10^{24} years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	2^{167} μs = 5.9×10^{36} years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	2×10^{26} μs = 6.4×10^{12} years	6.4×10^6 years

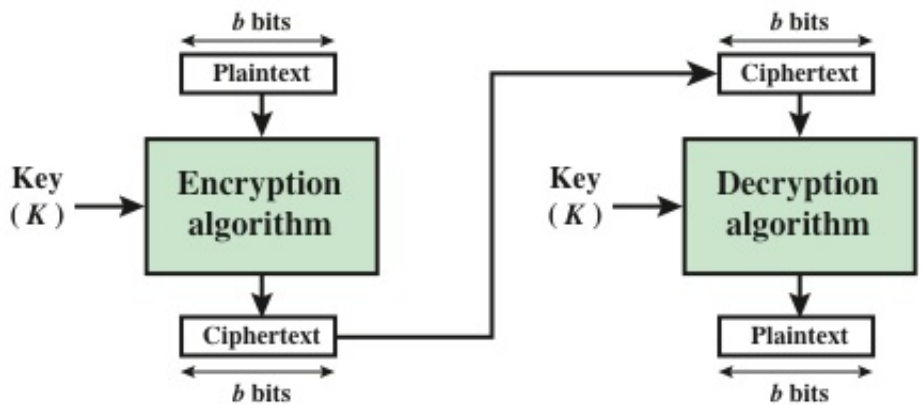
Plan

Block Ciphers and Stream ciphers

- Block Ciphers process the plaintext block by block to be encrypted. and treats the ciphertext block by block too. (ex : DES, AES). -> suitable for strong encryption, encryption of data in rest, encryption of small data amount
- Stream ciphers process the plaintext bit by bit (or byte by byte) to be encrypted. and treat the ciphertext the same way (ex : viginere, vernam) -> suitable for Fast encryption, data in transit encryption, noisy channel like wifi, real-time encryption...



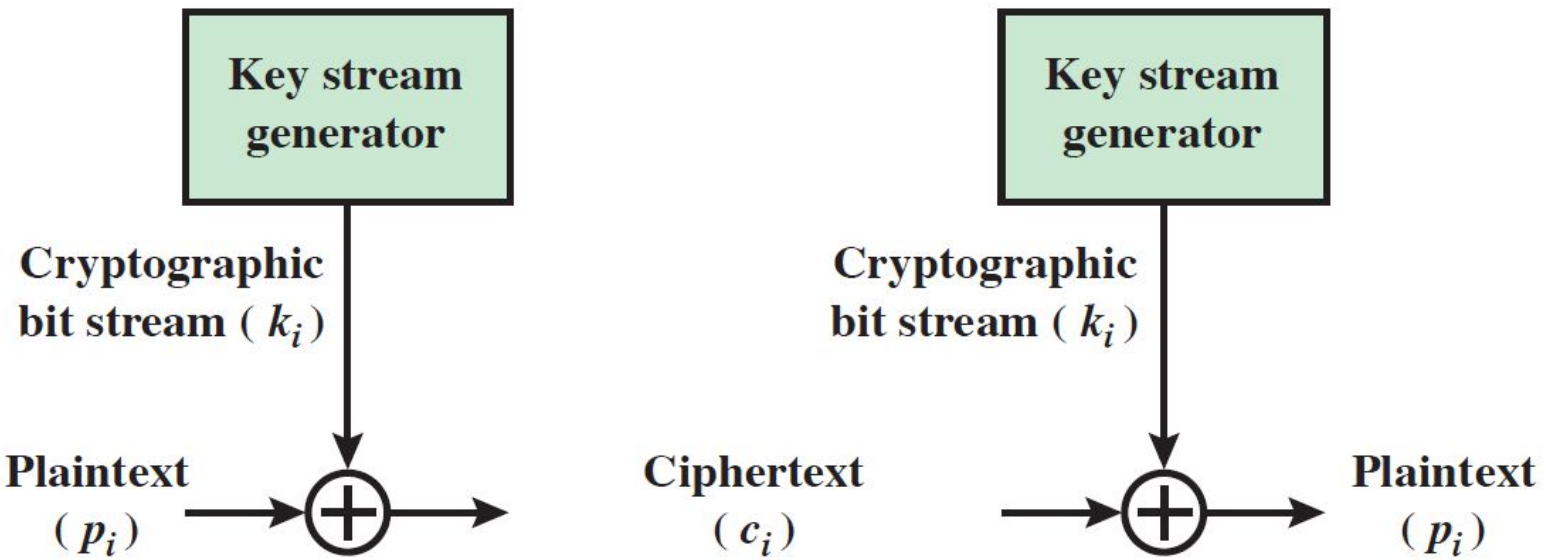
(a) Stream Cipher Using Algorithmic Bit Stream Generator



(b) Block Cipher

Stream cipher : Vernam cipher

- use a key as lomg as the plaintext
- invented by an AT&T enginnner Gilbert Vernam in 1918



One time Pad

- Improvement of Vernam proposed by army officer : Joseph Mauborgne
- Use a random key that is as long as the message in such a way that the key doesn't need to be repeated
- The Key is used to encrypt and decrypt a single message, then it is discarded
- Each new message requires a new key of the same length as the new message
- This cryptosystem is unbreakable
- problems in the production and secure distribution of the key
- Not practical : still used for top-secret and very expensive communications (red phone between Moscow and Washington)

reversible and irreversible functions

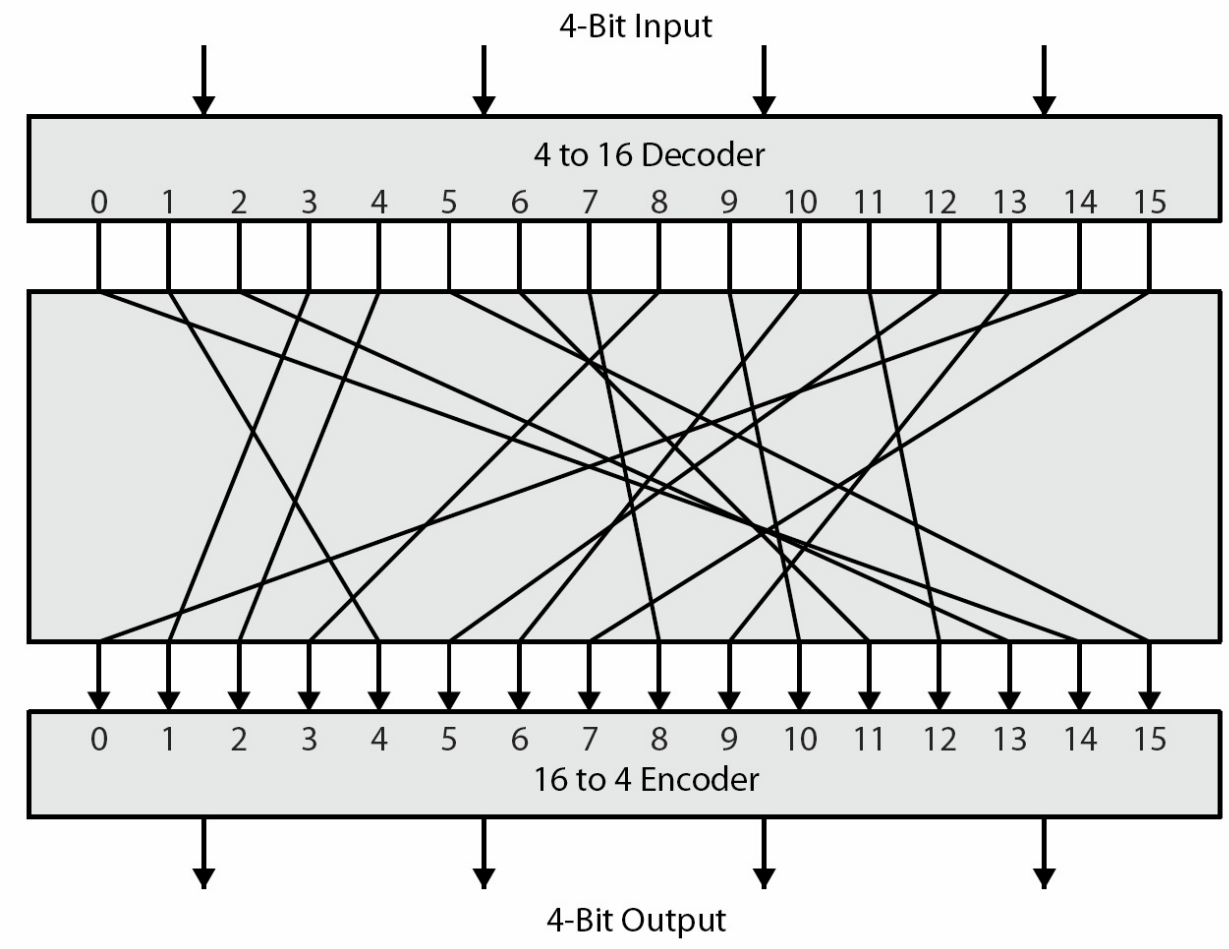
- A block cipher alg takes n bits of plaintext and turns it into n bits of ciphertext
- there are 2^n possible combinations of plaintext
- Encryption must be reversible
- each plaintext block produces a different ciphertext block (bijectivity)
- there are 2^n possible transformations

Reversible Mapping	
Plaintext	Ciphertext
00	11
01	10
10	00
11	01

Irreversible Mapping	
Plaintext	Ciphertext
00	11
01	10
10	01
11	01

جامعة التقنية والعلوم التطبيقية
University of Technology and Applied Sciences
Salalah صلالة

model of a block cipher : 4-bit length word



Look-up tables of the bloc cipher example

Plaintext	Ciphertext
0000	1110
0001	0100
0010	1101
0011	0001
0100	0010
0101	1111
0110	1011
0111	1000
1000	0011
1001	1010
1010	0110
1011	1100
1100	0101
1101	1001
1110	0000
1111	0111

Ciphertext	Plaintext
0000	1110
0001	0011
0010	0100
0011	1000
0100	0001
0101	1100
0110	1010
0111	1111
1000	0111
1001	1101
1010	1001
1011	0110
1100	1011
1101	0010
1110	0000
1111	0101

Claude Shannon and substitution-permutation networks

- Shannon introduced the idea of "Substitution-Permutation" (S-P) networks in 1949
- is the basis of all modern encryption algs
- S-P networks are based on two criteria :
  - Substitution (S-box)
  - Permutation(P-box)
- This provides the criteria for confusion and diffusion of the plaintext and key on the ciphertext

Confusion and Diffusion

- Two terms introduced by Shannon which constitute the basic criteria of an encryption algorithm
- His goal was to design cryptosystems that resist statistical analysis
- confusion : Make the relationship between ciphertext and key as complex as possible (Random appearance)
- diffusion : Every bit in the plaintext affects every bit in the ciphertext (Avalanche effect)

Plan

Password best practices

- No system, encrypted or not, is safe with poor password management.
- Passwords should not be written down.
- Temporary passwords should be used only once and immediately changed once a user logs in.
- Passwords should have length requirements and require the use of special characters to meet a defined complexity.
- Passwords should never be stored in clear text.
- A password for one system should not be used for another.
- Passwords should be changed regularly.

Hashing Passwords

- Hashing functions (details will be seen in Chap7) use specific hashing algorithms but do not use secret keys.
- You can use a number of hashing algorithms, including MD5, SHA-3, SHA-512, HAVAL, and RIPEMD-160.
- the output is not predictable (Random Appearance)
- different messages do not produce the same hash code (Collision resistanance)
- messages are not reversible (One way function)
- If the exact message is entered into a hashing function, the same hash code will be produced (Deterministic)

Example of Hashing very close passwords

Here are the MD5 hash values for the words "password" and "Password". Notice that even though only the first letter was capitalized, the entire message digest is different.

- hash value for "password" : 5f4dcc3b5aa765d61d8327deb882cf99
- hash value for "Password" : dc647eb65e6711e155375218212b3964

A rainbow table, which is a precomputed table for reversing cryptographic hash functions, can be used to crack password hashes. Rainbow tables are used in recovering a plaintext password up to a certain length consisting of a limited set of characters. This is where salting and stretching come in.

Python code for MD5 Hashing

MD5 Haching of "password"

```
import hashlib
plaintext_password = b'password'
hashed_sha512 = hashlib.md5(plaintext_password).hexdigest()
print(hashed_sha512)
```

Plan

- we use hashing to minimize readability when storing passwords, but just hashing alone isn't enough.
- Salting is the process of adding or concatenating a random chunk of bits to the end of the password before it goes through the hashing process.
- You would then save that random chunk of bits along with the hashed password.
- The reason salting is effective is that if bad actors attack your hashing scheme, they are unable to scale their attack to a large number of users or launch brute-force attacks across the enterprise.
- Salting means that a rainbow attack must be recomputed for each individual user. That makes an attacker spend a lot more money per user,

Python code for Salting "password"

MD5 + Salting of "password"

```
import hashlib

def saltPassword_md5(password):
  salt = b'cHp3'
  hashed = hashlib.md5(salt + password).hexdigest()
  print ("%s:%s" % (salt, hashed)) # Store these
  return hashed

plaintext_password = b'Password'
salted_md5 = saltPassword_md5 (plaintext_password)
```

Stretching

- Finally, a tried-and-true method to frustrate attackers is stretching passwords before they're saved to the database. The primary aim of stretching a password is to make deciphering the password more costly-whether with memory, time, or money-than an attacker can afford.
- In stretching, the strength of a password is measured by its bits of key strength. Methods of lengthening the number of bits a password has comes down to the hash function. Usually, hash functions are looped thousands of times, simulating randomness and adding more and more bits to the complexity of a password passed to the database.

Password tools

- One of the most popular libraries is *bcrypt*. Some alternatives include *scrypt* either using the *hashlib* or *cryptography* libraries. *Simplecrypt* also provides libraries for encryption, decryption, and salting.
- To add a salt to your hashed password, examine the following Python code :

bcrypt tool

```
import bcrypt

passwd = b's$cret12'
salt = bcrypt.gensalt()
hashed = bcrypt.hashpw(passwd, salt)

print(salt)
print(hashed)
```