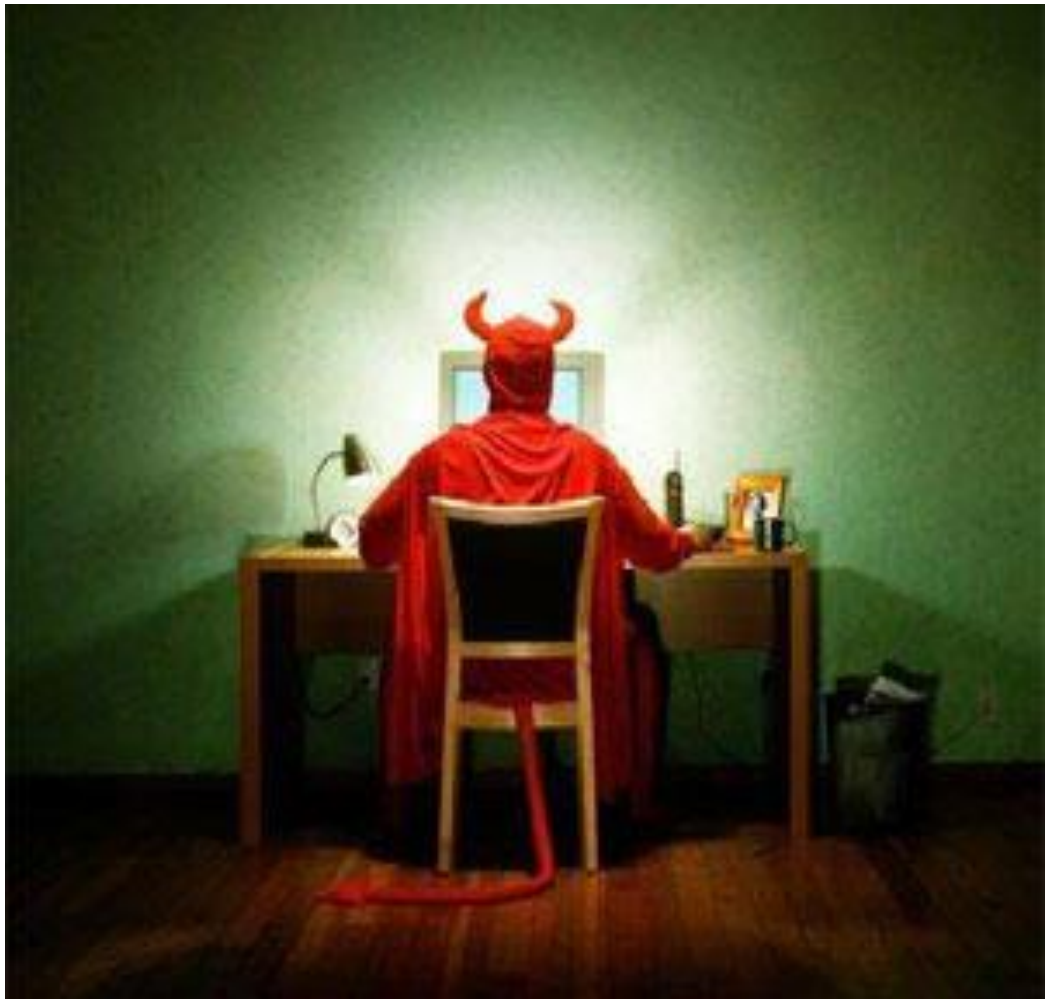
แก้ไขครั้งล่าสุดเมื่อวันที่ 20 พฤศจิกายน 2567

# Security

**Security requires consideration of the external environment.**
**Only human factors, not natural disasters like earthquake.**



Intruder
Hacker
Attacker

A **threat** is the potential for a security violation, such as the discovery of a vulnerability

An **attack** is an attempt to break security.

Threat ภัยคุกคาม
Vulnerable เปราะบาง
Vulnerability ช่องโหว่
Breach ฝ่าฝืน

# รู้ก่อน ระวังก่อน!

## แฮกเกอร์ 5 ประเภท ที่อยู่รอบตัวเรา

### แฮกเกอร์คือใคร

แฮกเกอร์ คือ กลุ่มคนที่มีความรู้ความเชี่ยวชาญเกี่ยวกับคอมพิวเตอร์และเทคโนโลยีเป็นอย่างดี ส่วนใหญ่มักจะเป็นโปรแกรมเมอร์ที่ชำนาญในด้านการเจาะระบบ โดยอาจมีเป้าหมายทั้งในทางที่ดีและไม่ดี เช่น การลักลอบเข้าถึงข้อมูลเพื่อสร้างความเสียหายหรือขโมยข้อมูลไปขาย ตลอดจนการปรับปรุงแก้ไขระบบเครือข่ายให้ปลอดภัยยิ่งขึ้น

**1 Script Kiddies**

แฮกเกอร์มือสมัครเล่นที่ไม่มีความเชี่ยวชาญด้านการเจาะระบบมากนัก แฮกเกอร์ประเภทนี้มักประยุกต์ใช้โปรแกรมของแฮกเกอร์คนอื่น ๆ ส่วนใหญ่ทำไปเพื่อเรียกร้องความสนใจ

**2 Whistleblowers**

บุคลากรภายในบริษัทหรือหน่วยงานรัฐที่ลักลอบขโมยข้อมูลที่เป็นความลับขององค์กรไปเปิดเผยหรือหาผลประโยชน์ เช่น การแบล็กเมล์องค์กรที่ดำเนินกิจการอย่างผิดกฎหมาย

**3 Black Hat Hackers**

บุคคลที่ชำนาญด้านการเจาะระบบคอมพิวเตอร์อย่างผิดกฎหมาย ส่วนมากมักขโมยข้อมูลส่วนบุคคลหรือข้อมูลทางการเงินไปขายในตลาดมืด นับเป็นการก่ออาชญากรรมทางไซเบอร์

**4 Hacktivists**

บุคคลหรือกลุ่มนิรนามที่ลักลอบเข้าถึงข้อมูลบนเว็บไซต์และเครือข่ายคอมพิวเตอร์ของรัฐบาล เพื่อจุดประสงค์ทางสังคมหรือการเมือง ถือเป็นประเภทของแฮกเกอร์ที่พบมากที่สุด

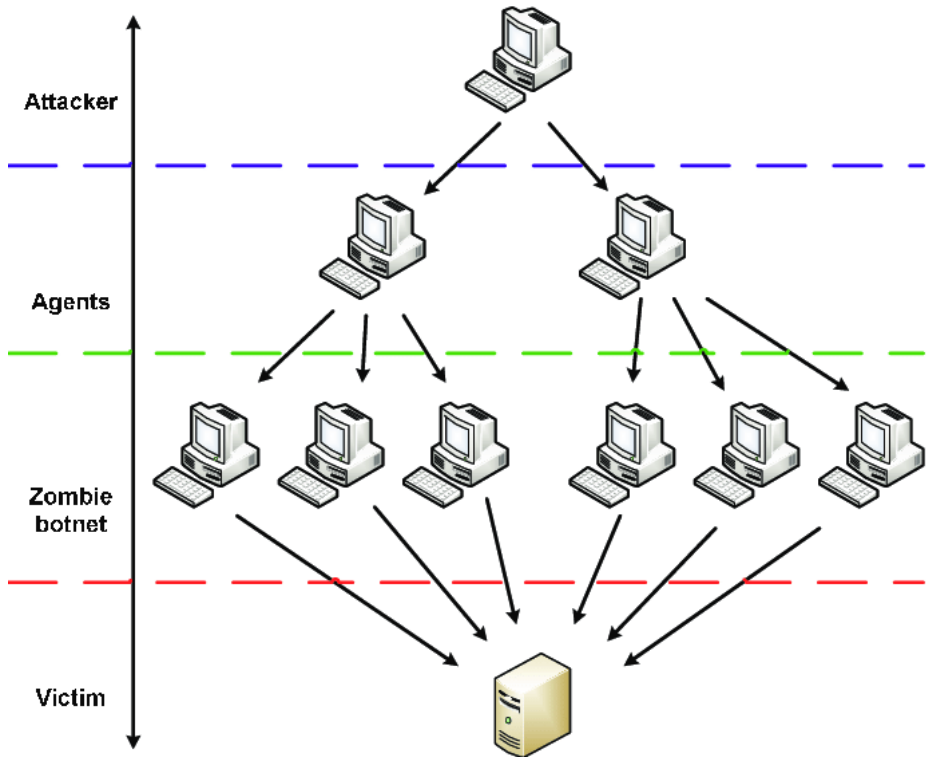
**5 White Hat Hackers**

ผู้เชี่ยวชาญด้านการรักษาความปลอดภัยในโลกไซเบอร์ที่ได้รับอนุญาตให้ละเมิดข้อมูลอย่างถูกกฎหมาย แฮกเกอร์กลุ่มนี้จะให้การช่วยเหลือรัฐบาลหรือบริษัทต่าง ๆ ผ่านการเจาะระบบเพื่อทดสอบระดับความปลอดภัยและแก้ไขช่องโหว่เพื่อป้องกันการโดนโจมตีจากภายนอก

**รู้หรือไม่ :** รายงานประจำปีของ Flashpoint บริษัทชั้นนำที่รวบรวมข้อมูลและทำวิจัยเกี่ยวกับอาชญากรรมทางไซเบอร์ พบว่าในปี **2021** มีการ**ละเมิดและเปิดเผยข้อมูลโดยไม่ได้รับอนุญาต**กว่า **22 พันล้าน** รายการ

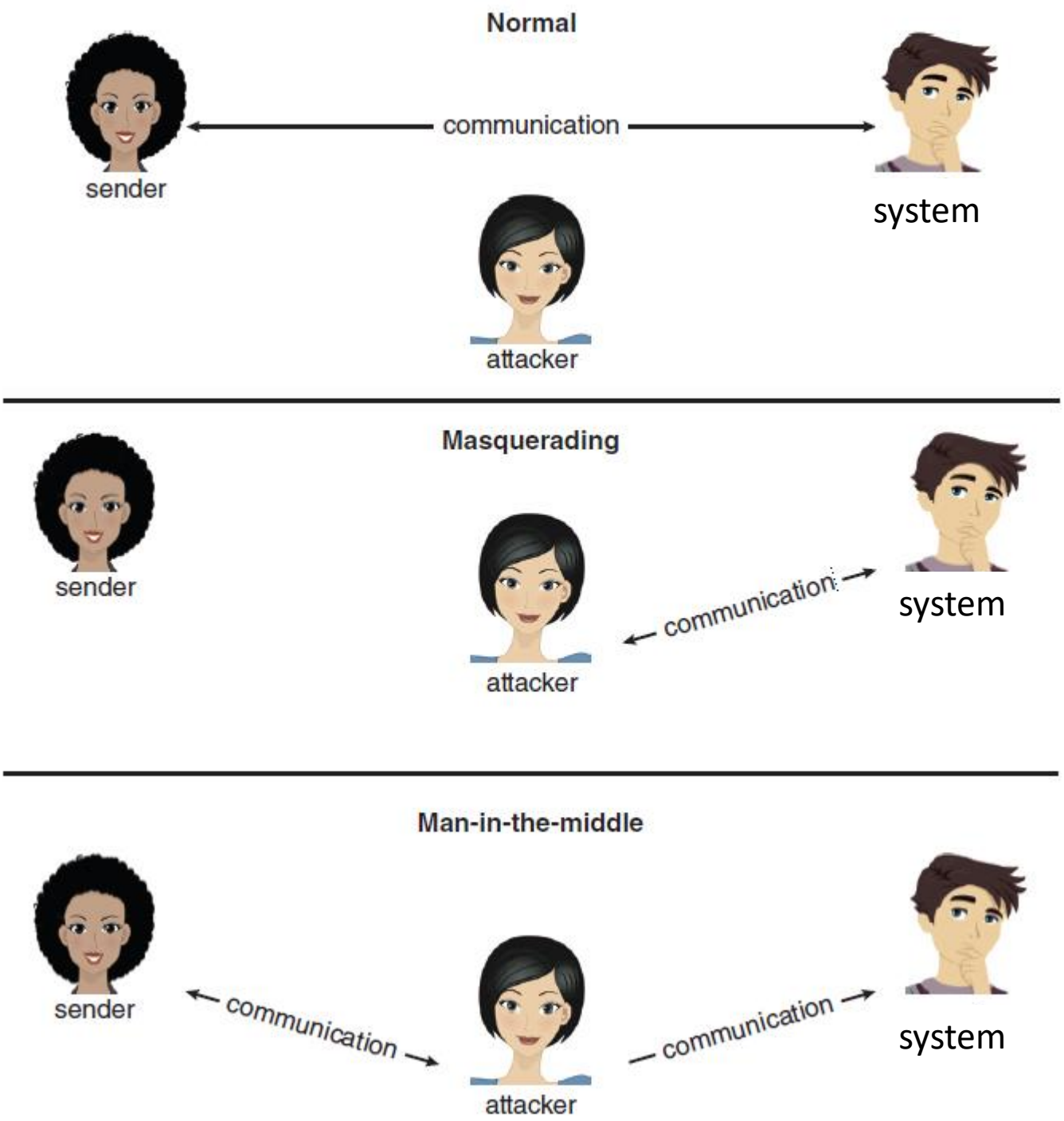
# Attack Types

- **Breach of confidentiality ขโมยข้อมูล เช่น บัญชีผู้ใช้ บัตรเครดิต**
- **Breach of integrity แก้ไขข้อมูล เช่น source code หรือ application**
- **Breach of availability ทำลายข้อมูล เช่น website defacement**
- **Theft of service ขโมยใช้ทรัพยากร เช่น file server, computer, network**
- **Denial of service (DoS) ทำให้ระบบล่ม ให้บริการไม่ได้ เช่น ทำให้เว็บไซต์ล่ม**
  ปัจจุบันพัฒนาเป็น **Distributed DoS (DDoS)**




ถ้าใช้คอมพิวเตอร์เครื่องเดียว หรือหลายเครื่องใน subnet เดียวกัน block ทั้ง subnet ได้ เช่น x.y.z.*

ใช้คอมพิวเตอร์หลายเครื่องจากหลาย ๆ subnet ทำให้ block IP ไม่ทัน มันดูไม่ต่างจากผู้ใช้ปกติ




เมื่อเป็น man in the middle แล้ว ก็ทำได้หลายอย่าง

replay attack เช่น replay คำสั่งโอนเงิน แต่เปลี่ยนบัญชีผู้รับเงิน แล้วให้ sender พิมพ์ OTP ให้

session hijack ถีบ sender ทิ้งไป ทำตัวเป็น sender เองเลย เช่น โอนเงินจากบัญชี sender ไปบัญชีม้า

privilege escalation ใช้ bug ของระบบ ทำให้มี privilege มากกว่าปกติ เช่น เปลี่ยนจาก user เป็น admin

# Social Engineering & Phishing

Social engineering is the term used for a broad range of malicious activities accomplished through human interactions. It uses psychological manipulation to trick users into making security mistakes or giving away sensitive information.


https://www.ktc.co.th/article/knowledge/phishing

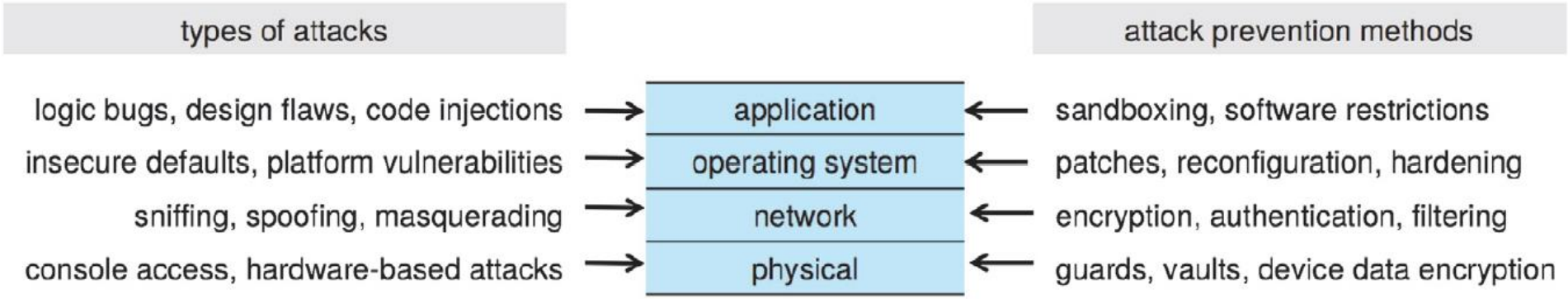
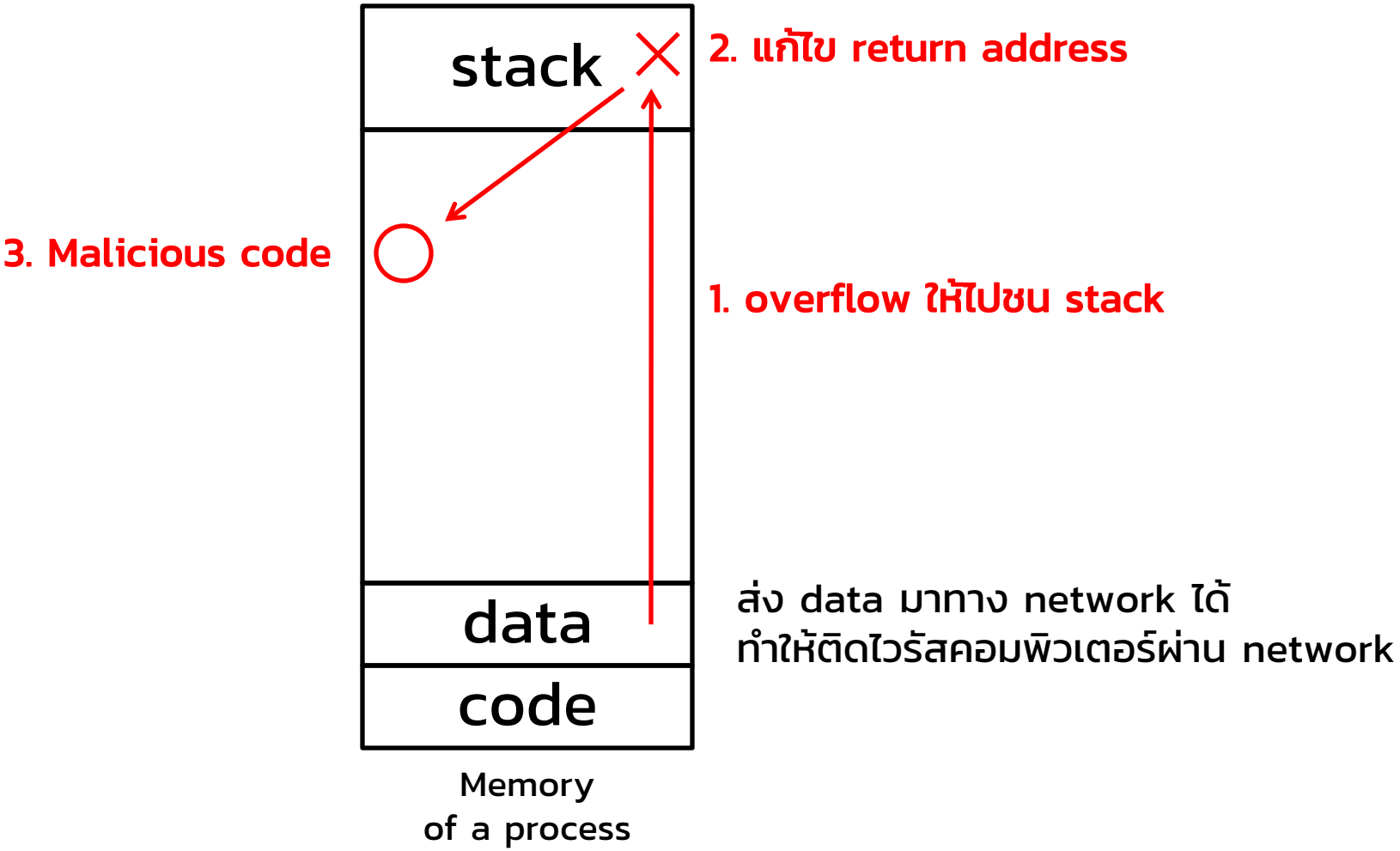



**Figure 16.1** The four-layered model of security.

| | |
|---|---|
| Sniff | ดักฟัง |
| Spoof | หลอกลวง เช่น ปล่อย message ปลอมลงไปใน network |
| Masquerade | ปลอมตัว |

| | |
|---|---|
| Malware | ซอฟต์แวร์ที่ตั้งใจออกแบบมาให้ทำอันตราย |
| Trojan horse | ไม่ได้ทำงานตามที่โฆษณา แต่ทำอันตรายอย่างอื่นด้วย |
| Spyware | แสดงโฆษณา ขโมยข้อมูล ส่ง spam mail (90% in 2010) |
| Ransomware | เข้ารหัสข้อมูล เรียกค่าไถ่ (ransom) |
| Trap door | ผู้พัฒนาระบบ/แอป ทำทางเข้าสำหรับตัวเองไว้ใช้ภายหลัง เช่น แอบเข้ามาปัดเศษสตางค์จากบัญชีอื่น<br>XCodeGhost 2015 เติม malicious code ลงไปในทุกแอปที่คอมไพล์ (ควรติดตั้งแอปจากแหล่งที่เชื่อถือได้เท่านั้น) |
| Logic bomb | ตั้งเวลา เงื่อนไข ให้โจมตี เช่น เมื่อไม่ได้เป็นพนักงานแล้ว |
| Virus | มีการแพร่พันธุ์ (self-replicating) โดยใช้ human activity เช่น การก็อปปี้ไฟล์ข้ามเครื่อง ไวรัสมีหลายประเภท file, boot, macro (spreadsheet), rootkit (easy root access), source code, polymorphic, encrypted, stealth, multipartite, armored |
| Worms | ไม่ใช้ human activity แพร่ไปได้เองใน network |
| Monoculture | ระบบเดียวทั้งโลก เช่น ปี 2004 ระบบปฏิบัติการ Windows Server ติดตั้ง Internet Information Services (IIS) ผู้ท่องเว็บใช้โปรแกรม Internet Explorer ไวรัสติด IIS ก่อน แล้วผู้ท่องเว็บทุกคนที่มาดาวน์โหลดไฟล์จากเว็บนี้จะติดไวรัสไปด้วยทุกครั้ง ไวรัสจะติดตั้ง back door เช่น keystroke logger จำทุกอย่างที่ผู้ใช้พิมพ์ผ่านคีย์บอร์ด |

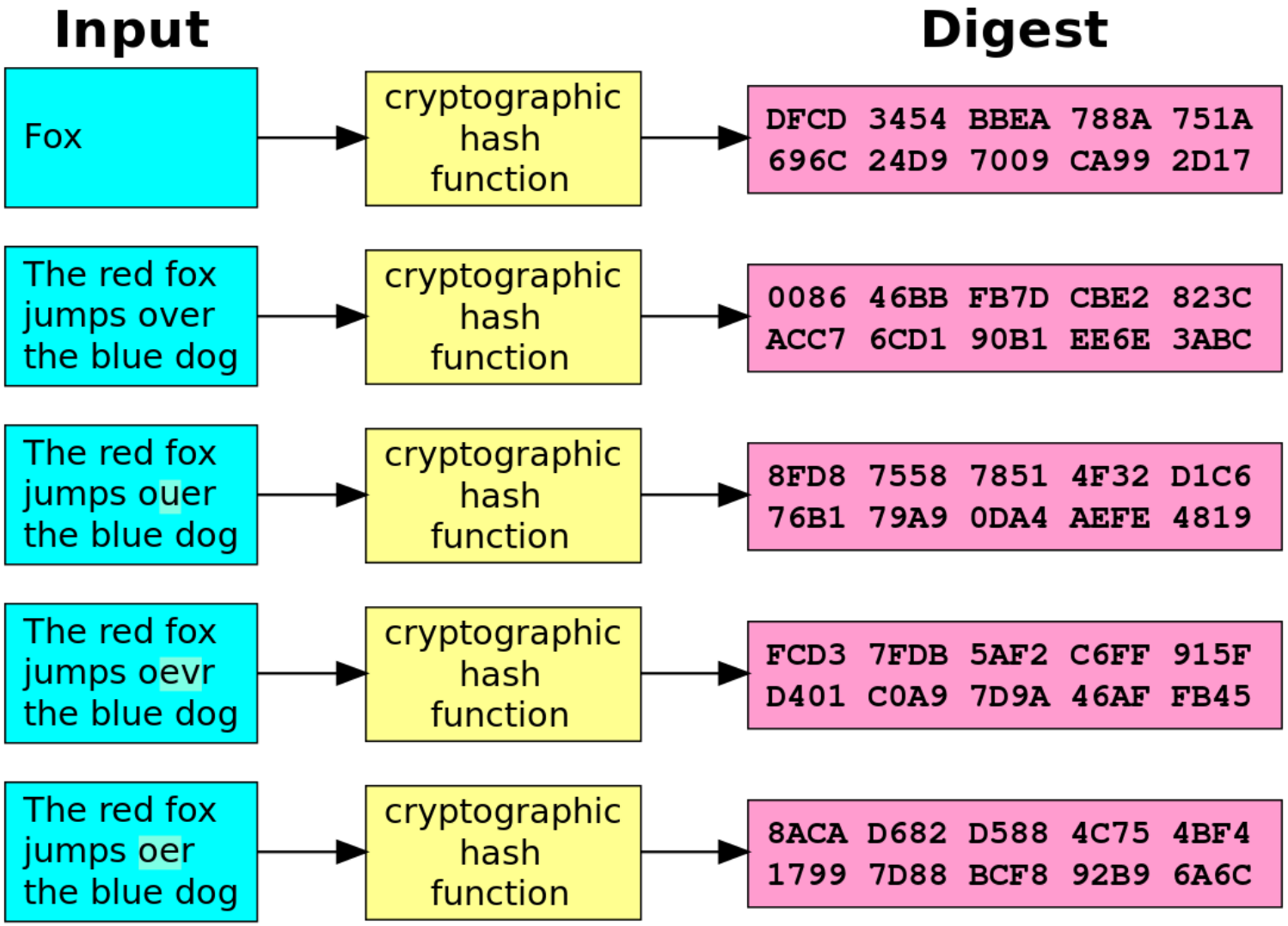
# Stack Overflow หรือ Buffer Overflow



```
#include <stdio.h>
#define BUFFER_SIZE 0

int main(int argc, char *argv[])
{
   int j = 0;
   char buffer[BUFFER_SIZE];
   int k = 0;
   if (argc < 2) {return -1;}

   strcpy(buffer,argv[1]);
   printf("K is %d, J is %d, buffer is %s\n", j,k,buffer);
   return 0;
   }
}
```

**Figure 16.2** C program with buffer-overflow condition.

# Hash Functions




คุณสมบัติที่สำคัญคือ หา inverse function ยากมาก และโอกาสเกิด collision น้อย

# ในทางโปรแกรมมิ่งการเก็บ password ของผู้ใช้ แบบไหนถึงจะปลอดภัย ?

1.

Manee

Password

1234

เก็บ password

ตัวอย่างฐานข้อมูล

| User | Password |
|---|---|
| Manee | 1234 |

Hacker ชอบมองเห็นพาสเวิร์ด

2.

Manee

Password

1234

เก็บ hash value

Hash function

ตัวอย่างฐานข้อมูล

| User | Hash values (password) |
|---|---|
| Manee | a591a6d40bf420404a011733cfb7b1<br>90d62c65bf0bcda32b57b277d9 |

Hacker อ่านพาสเวิร์ดไม่ออก ก็จริง แต่เดาสุ่มได้ไม่ยาก

ดูรหัสผ่านยอดนิยม

3.

Manee

Password

1234

+

Salt (random)

bv5PehSMfV11Cd

เก็บ hash value และ salt value

Hash function

ตัวอย่างฐานข้อมูล

| User | Hash values (password) | Salt values |
|---|---|---|
| Manee | f7b007b41cf9e48d0414<br>]81bb12dbdd88d6d5<a<br>afe8odbd68bja1gGb | bv5PehSMf<br>V11Cd |

Hacker อ่านพาสเวิร์ดไม่ออก แถมเดาสุ่มก็ยากด้วย

**รายชื่อรหัสผ่านยอดนิยมทั่วโลก คัดมาเฉพาะ 10 อันดับแรก เป็นดังนี้**

1. **password** | ผู้ใช้ 4,929,113 บัญชี
2. **123456** | ผู้ใช้ 1,523,537 บัญชี
3. **123456789** | ผู้ใช้ 413,056 บัญชี
4. **guest** | ผู้ใช้ 376,417 บัญชี
5. **qwerty** | ผู้ใช้ 309,679 บัญชี
6. **12345678** | ผู้ใช้ 284,946 บัญชี
7. **111111** | ผู้ใช้ 229,047 บัญชี
8. **12345** | ผู้ใช้ 188,602 บัญชี
9. **col123456** | ผู้ใช้ 140,505 บัญชี
10. **123123** | ผู้ใช้ 127,762 บัญชี

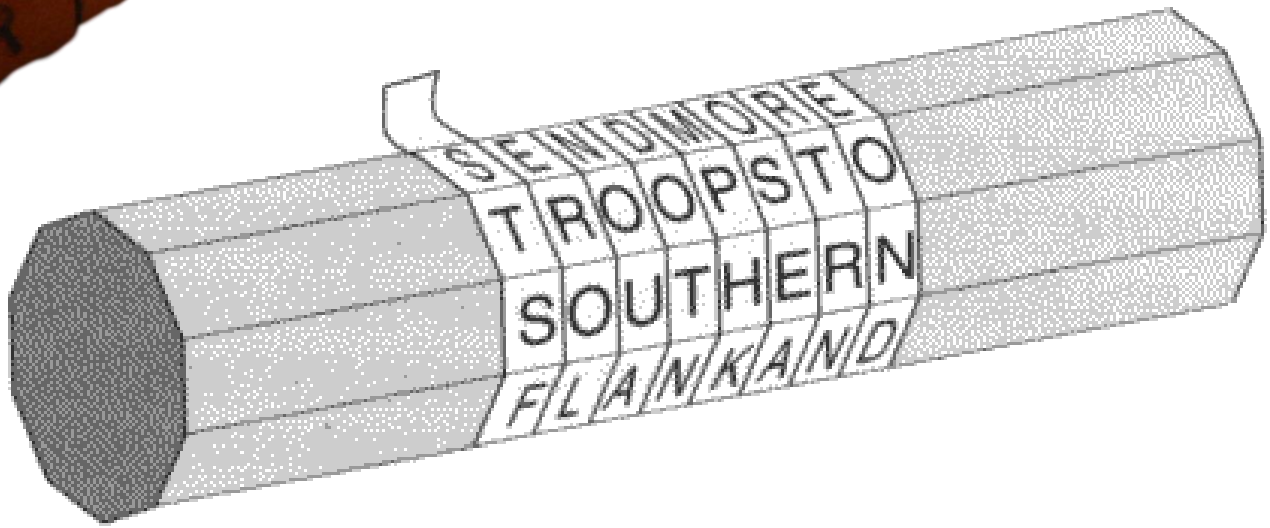
```
using System;
using System.Security.Cryptography;
using Microsoft.AspNetCore.Cryptography.KeyDerivation;
```

```
Console.Write("Enter a password: ");
string password = Console.ReadLine();

// generate a 128-bit salt using a secure PRNG
byte[] salt = new byte[128 / 8];
using (var rng = RandomNumberGenerator.Create())
{
    rng.GetBytes(salt);
}
Console.WriteLine($"Salt: {Convert.ToBase64String(salt)}");

// derive a 256-bit subkey (use HMACSHA1 with 10,000 iterations)
string hashed = Convert.ToBase64String(KeyDerivation.Pbkdf2(
    password: password,
    salt: salt,
    prf: KeyDerivationPrf.HMACSHA1,
    iterationCount: 10000,
    numBytesRequested: 256 / 8));
Console.WriteLine($"Hashed: {hashed}");
```

```
Enter a password: helloworld
Salt: intE98uKsFx7H9Tpk/tTEA==
Hashed: xit0xfNlIfAK2c3T0uogUm/617EoAxUVl4iUSNYpDy0=
```

# Cryptography (hidden/secret to write)


Scytale ไม่ต้องใช้ key ก็ได้
ขอแค่ให้รู้วิธีอ่านที่ถูกต้อง




https://en.wikipedia.org/wiki/Cryptography

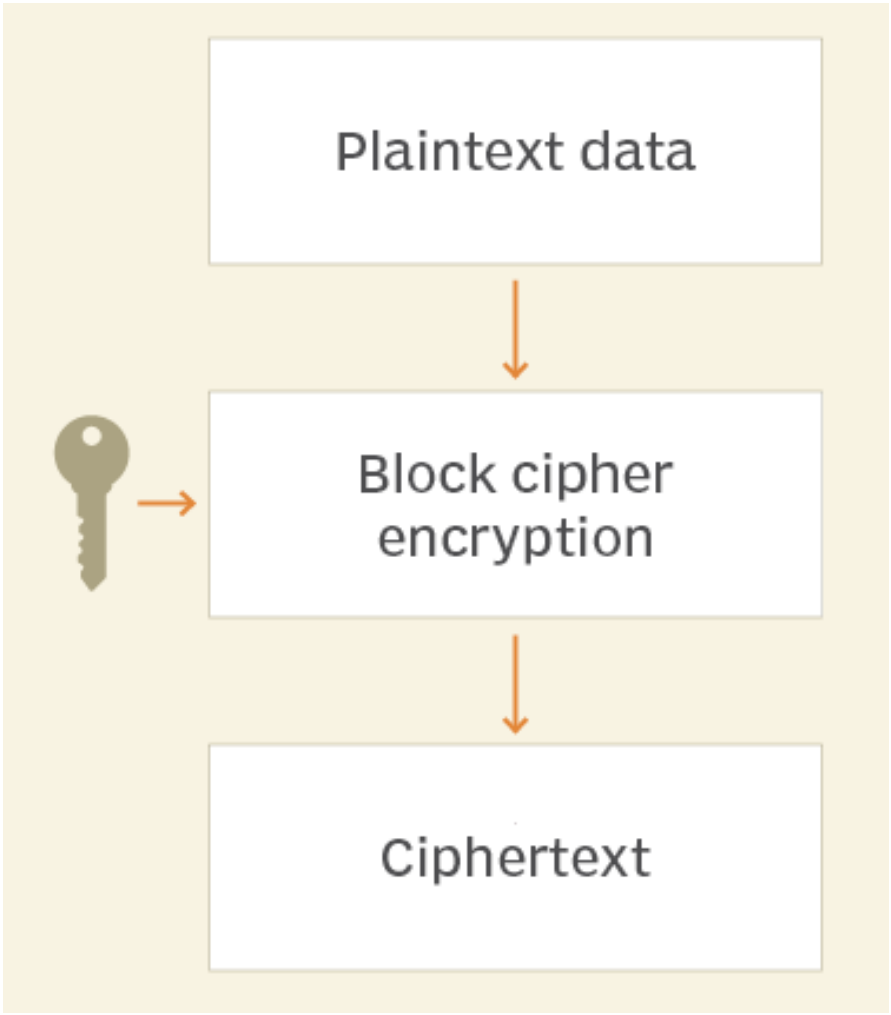
# Encryption

An encryption algorithm consists of the following components:

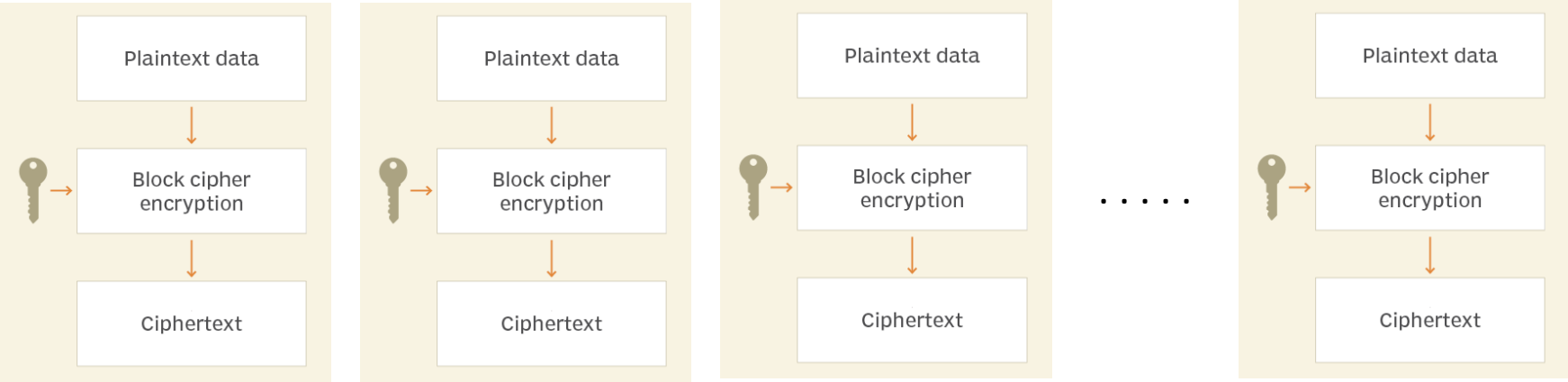
- A set $K$ of keys. ต้องใช้ key
- A set $M$ of messages.
- A set $C$ of ciphertexts.
- An encrypting function $E : K \rightarrow (M \rightarrow C)$. That is, for each $k \in K$, $E_k$ is a function for generating ciphertexts from messages. Both $E$ and $E_k$ for any $k$ should be efficiently computable functions. Generally, $E_k$ is a randomized mapping from messages to ciphertexts.
- A decrypting function $D : K \rightarrow (C \rightarrow M)$. That is, for each $k \in K$, $D_k$ is a function for generating messages from ciphertexts. Both $D$ and $D_k$ for any $k$ should be efficiently computable functions.

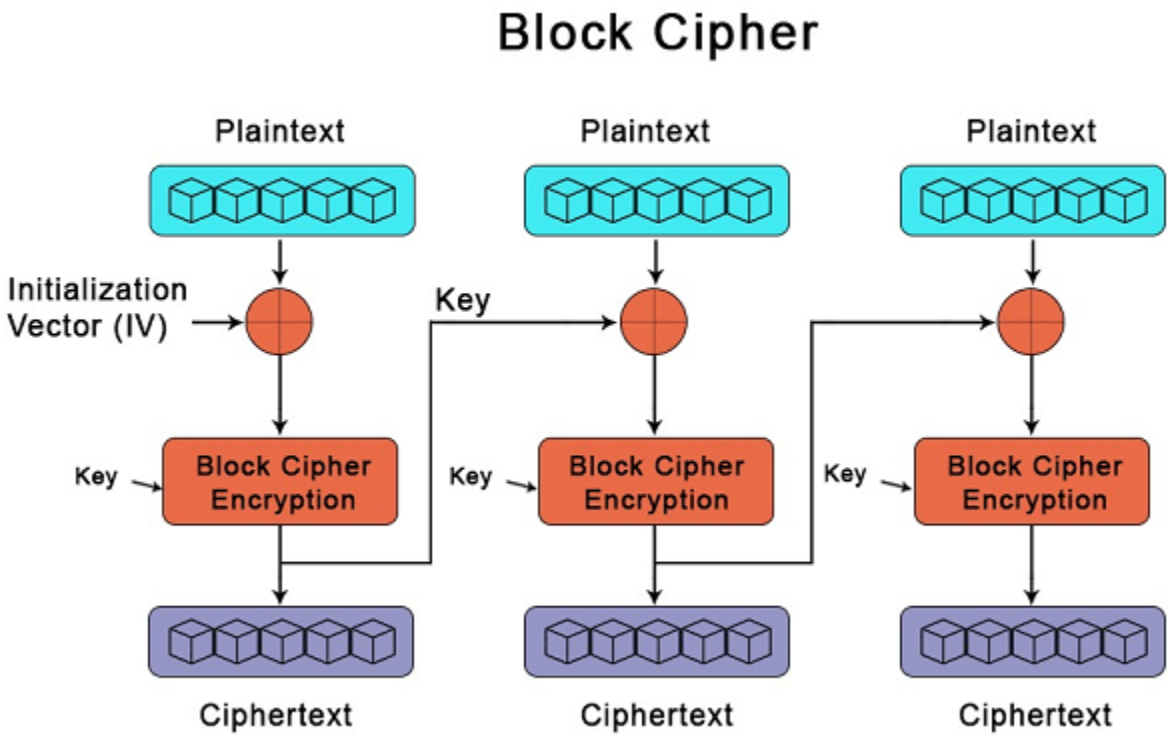
# Symmetric Encryption: Block Cipher

- Data Encryption Standard (DES) 64-bit block, 56-bit key
- Advanced Encryption Standard (AES) 128-bit block, 128/192/256-bit key



- ใช้ block cipher แบบ parallel ไม่ปลอดภัย ถ้า plaintext เหมือนเดิม จะทำให้ได้ cyphertext เหมือนเดิม เป็นการเปิดเผยข้อมูลที่เกิดซ้ำบ่อย ๆ เช่น letter e

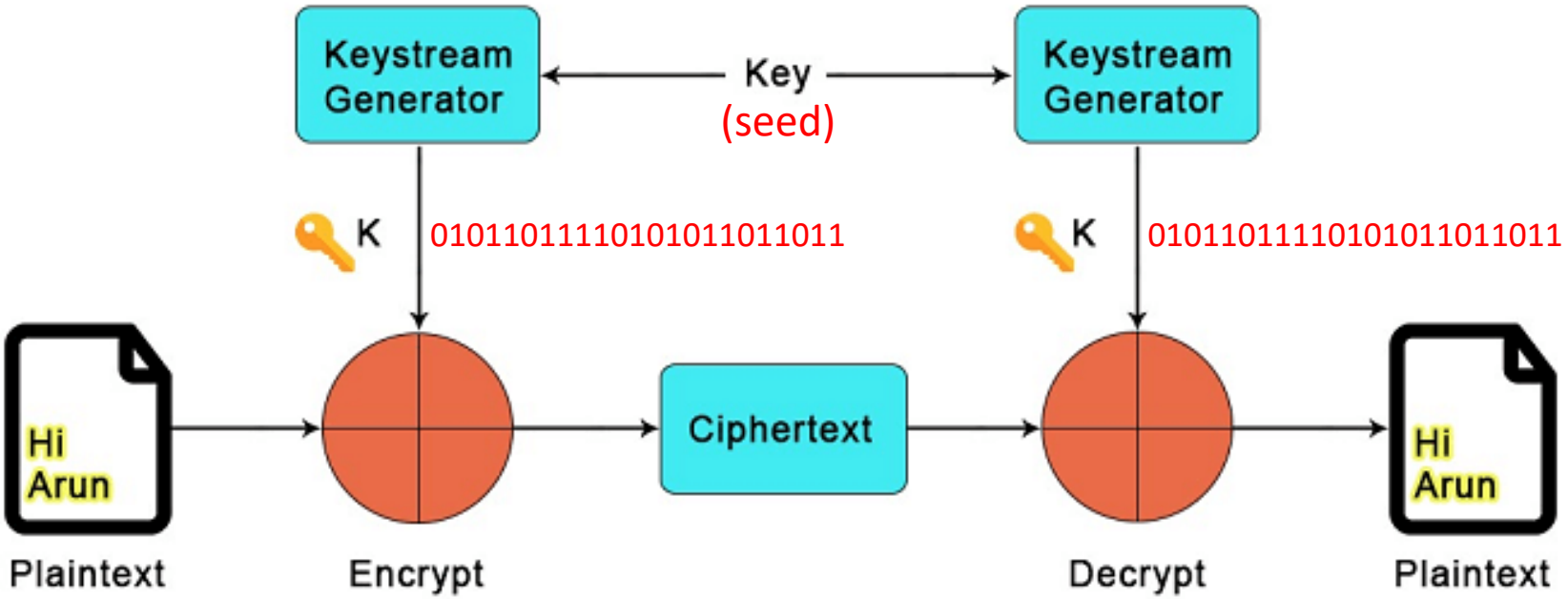



อย่าเขียนโค้ดเอง

ใช้ library มาตรฐาน

# Stream Cipher




Stream ciphers typically execute at a higher speed than block ciphers and have lower hardware complexity. However, stream ciphers can be susceptible to security breaches; for example, when the same starting state (seed) is used twice.

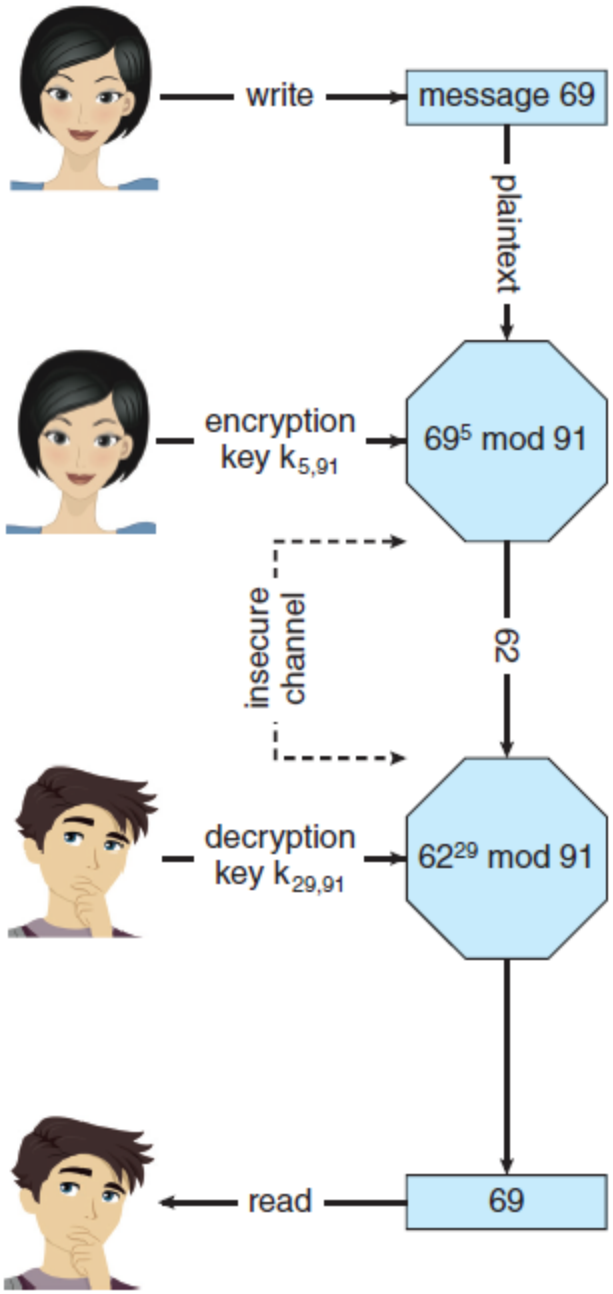
# Asymmetric Encryption

- ข้อเสียของ symmetric encryption คือ sender/receiver ต้องมี secret key ก่อนที่จะเริ่มการสื่อสาร (ต้องไปหาที่ลับ ๆ เพื่อตกลงว่าจะใช้ key อะไร)

# RSA – Rivest, Shamir, and Adleman

In RSA, $k_e$ is the **public key**, and $k_d$ is the **private key**. $N$ is the product of two large, randomly chosen prime numbers $p$ and $q$ (for example, $p$ and $q$ are 2048 bits each). It must be computationally infeasible to derive $k_{d,N}$ from $k_{e,N}$, so that $k_e$ need not be kept secret and can be widely disseminated. The encryption algorithm is $E_{ke,N}(m) = m^{k_e} \bmod N$, where $k_e$ satisfies $k_e k_d \bmod (p-1)(q-1) = 1$. The decryption algorithm is then $D_{k_d,N}(c) = c^{k_d} \bmod N$.

An example using small values is shown in Figure 16.8. In this example, we make $p = 7$ and $q = 13$. We then calculate $N = 7 * 13 = 91$ and $(p-1)(q-1) = 72$. We next select $k_e$ relatively prime to 72 and $< 72$, yielding 5. Finally, we calculate $k_d$ such that $k_e k_d \bmod 72 = 1$, yielding 29. We now have our keys: the public key, $k_{e,N} = 5, 91$, and the private key, $k_{d,N} = 29, 91$. Encrypting the message 69 with the public key results in the message 62, which is then decoded by the receiver via the private key.




ป้องกันข้อมูลสำคัญ เช่น
- password
- credit card no


public key = 5, 91 (ทุกคนบน www ทราบ)
private key = 29 (เป็นความลับ)

**Authentication** พิสูจน์ว่าบุคคลเป็นตัวจริง, พิสูจน์ว่าเอกสารไม่ถูกแก้ไข

**Authorization** พิสูจน์ว่ามีอำนาจหน้าที่ (authority), สิทธิ์ที่จะสั่งให้ทำหรือไม่

เงินกระดาษของราชวงศ์ชิง

ในสมัยโบราณใช้ตราประทับที่ทำจากหยกหรือไม้ไผ่ ผู้อ่านต้องใช้ความชำนาญในการจดจำรอยต่าง ๆ (ต้องเรียนรู้และฝึกฝนก่อน)

# Authentication

An authentication algorithm using symmetric keys consists of the following components:

- A set $K$ of keys.
- A set $M$ of messages.
- A set $A$ of authenticators.
- A function $S : K \rightarrow (M \rightarrow A)$. That is, for each $k \in K$, $S_k$ is a function for generating authenticators from messages. Both $S$ and $S_k$ for any $k$ should be efficiently computable functions.

# JWT.IO

## Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.ey
JzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ikpva
G4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKx
wRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c
```

ข้อมูลนี้ไม่ได้เข้ารหัส (encrypted)
แค่เขียนในรูป base-64 string

มี 3 ส่วน คั่นด้วย . (fullstop)

header . payload . signature

## Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

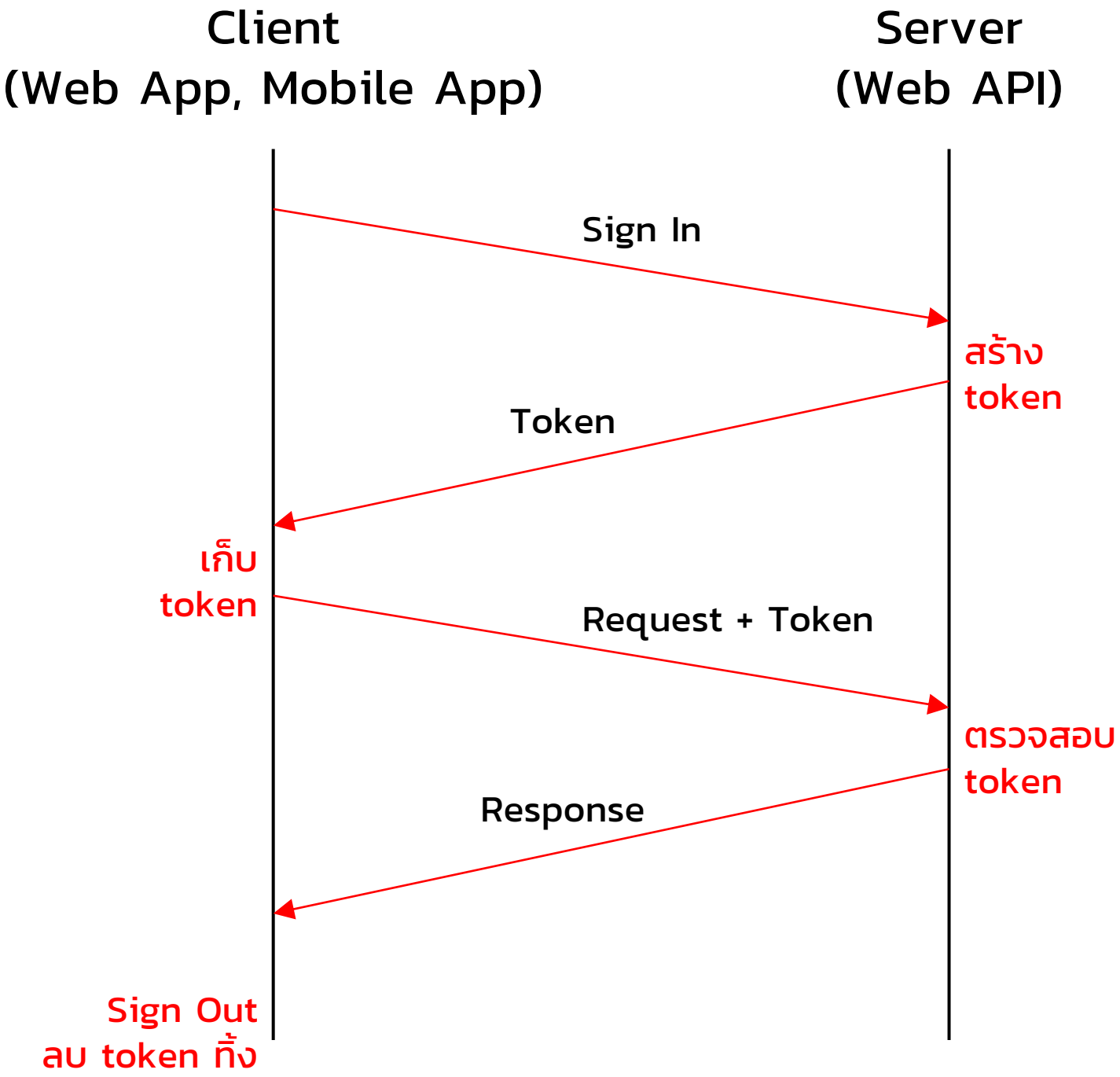
PAYLOAD: DATA

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022
}
```

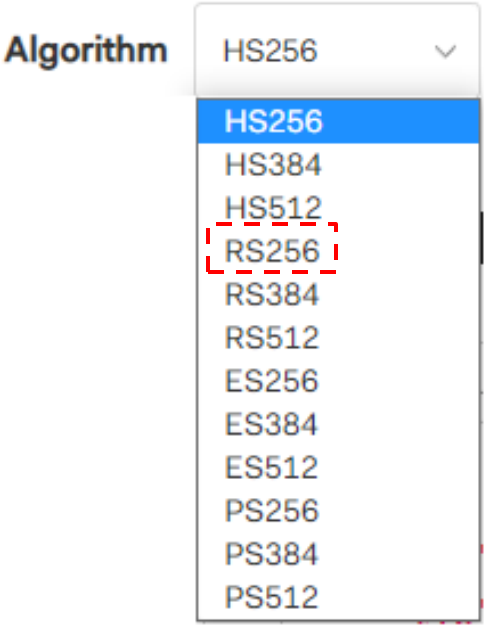
VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) ☐ secret base64 encoded
```

เมื่อล็อกอินสำเร็จ จะได้ Jason Web Token (JWT) ข้อมูลใน payload จะบอกว่า user คือใคร และวันหมดอายุของ token เมื่อจะ request ให้ web app ทำงาน เช่น อ่าน/เขียน db ส่งอีเมล์ อัปโหลดไฟล์ ก็ต้องส่ง token มาด้วยทุกครั้ง เพื่อยืนยันว่าเป็น user คนนี้จริง ๆ


Client
(Web App, Mobile App)
Server
(Web API)
Sign In
สร้าง
token
Token
เก็บ
token
Request + Token
ตรวจสอบ
token
Response
Sign Out
ลบ token ทิ้ง

# HS256 vs RS256 (HMAC using SHA256, RSA using SHA256)




Symmetric key

Asymmetric key

private key ใช้ sign

public key ใช้ verify


วาง public key ไว้บนอินเทอร์เน็ต

เก็บ secret key ไว้เป็นความลับ

token ที่ sign ด้วย private key นำ token นี้ไปใช้กับ web/mobile app อื่น ๆ ได้ เช่น token ออกโดย google บอกว่าเรามีสิทธิ์ใช้งาน service อะไรได้บ้าง หรือจะใช้พิสูจน์ว่าเราเป็นบุคคลนี้โดยอนุโลม (เพราะอาจเป็น token ที่ขโมยมา)

# Chapter 16 Security