Security & Crypto Overview

CSIE 7190 Cryptography and Network Security, Spring 2019

https://ceiba.ntu.edu.tw/1072csie_cns

cns@csie.ntu.edu.tw

Hsu-Chun Hsiao

National Taiwan University

Housekeeping

本學期的reading list

- Available on https://www.csie.ntu.edu.tw/~hchsiao/courses/cns19.html
- 選項可能會增多，但不會移除現有的
- 除非有特別說明，每次選擇一篇即可

Housekeeping

Critique: dos and don'ts

- 請存pdf格式
- one page only!
- 每項都要寫到，特別是reflection
- 假如論文提到 “Crossfire attack is outside the threat model”，而在心得的 weakness / reflection 只寫 “The author does not consider Crossfire.....” 是不夠的。需要補充為什麼這是一個不合理的假設。

Reading critique #2

Write a critique on one of the following:

- MD5 considered harmful today
  - https://www.win.tue.nl/hashclash/rogue-ca
- Abelson, Harold, et al. "Keys under doormats: mandating insecurity by requiring government access to all data and communications." Journal of Cybersecurity 1.1 (2015): 69-79.

Agenda

What is (cyber)security?

Introduction to cryptography

What is Security?

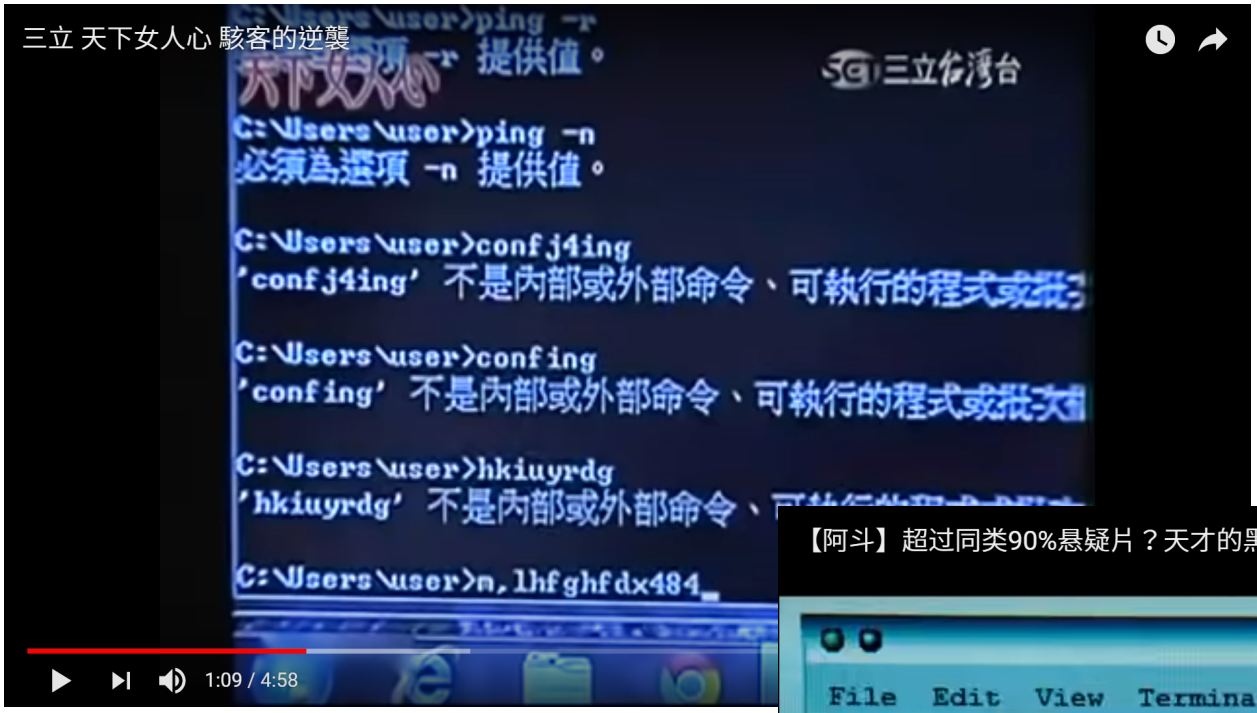
Security requirements

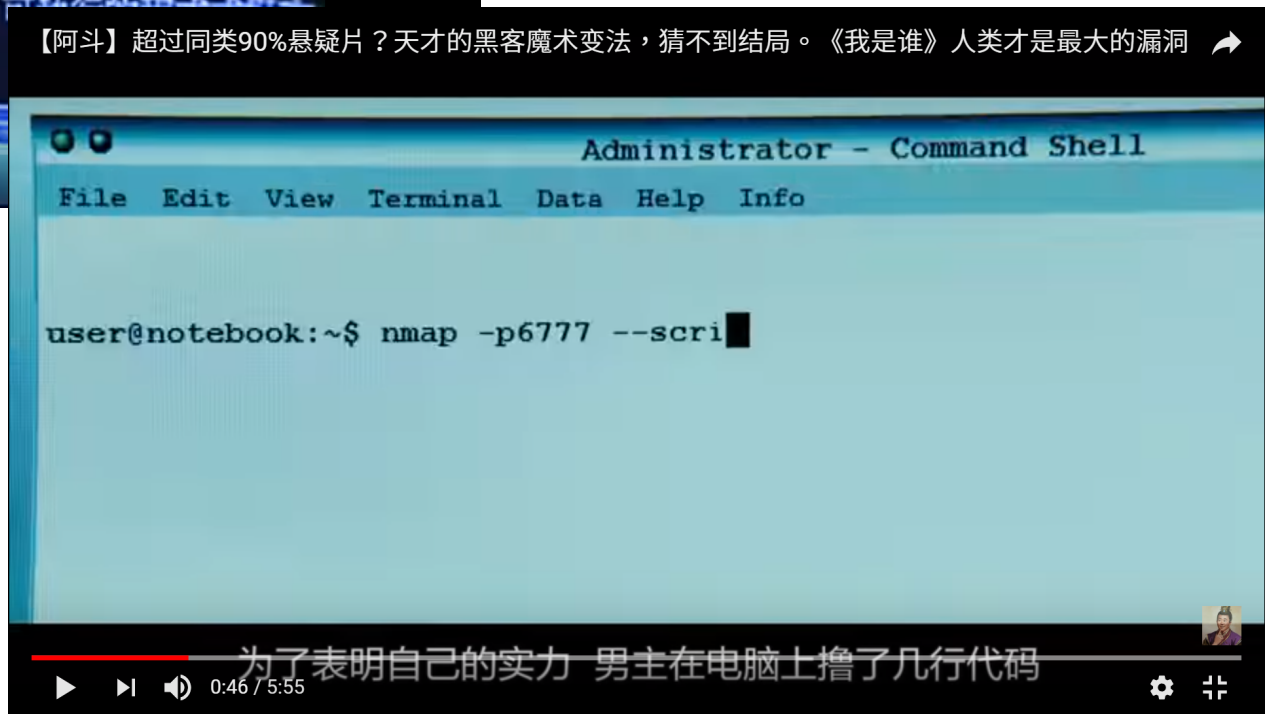
Threat model

Cost of security

從電影認識資安…？

從電影認識資安...？


三立 天下女人心 駭客的逆襲
三立台灣台
C:\Users\user>ping -n
必須為選項 -n 提供值。
C:\Users\user>confj4ing
'confj4ing' 不是內部或外部命令、可執行的程式或批
C:\Users\user>confing
'confing' 不是內部或外部命令、可執行的程式或批
C:\Users\user>hkiuyrdg
'hkiuyrdg' 不是內部或外部命令、
C:\Users\user>n,lhfghfdx484
1:09 / 4:58


【阿斗】超过同类90%悬疑片？天才的黑客魔术变法，猜不到结局。《我是谁》人类才是最大的漏洞
Administrator - Command Shell
File Edit View Terminal Data Help Info
user@notebook:~$ nmap -p6777 --scri
为了表明自己的实力 男主在电脑上撸了几行代码
0:46 / 5:55

What is security?

Protect assets (e.g., data and communication) from unauthorized actions

What is security?

Protect assets (e.g., data and communication) from unauthorized actions

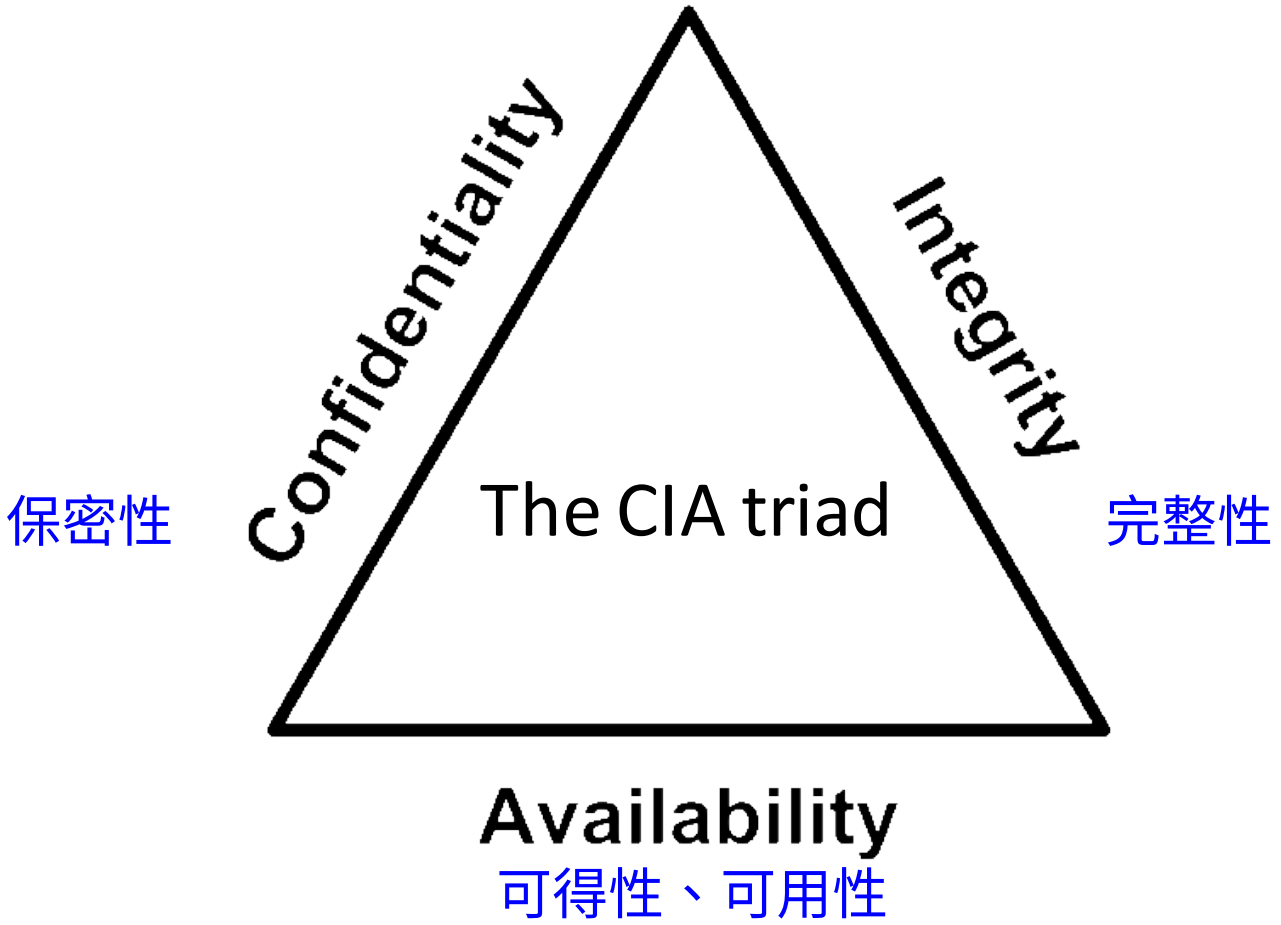
Attackers = entities attempt to do unauthorized actions

Attacker may

- Eavesdrop
- Manipulate
- Denial of service
- …

Security requirements

Properties that the protection should achieve



保密性

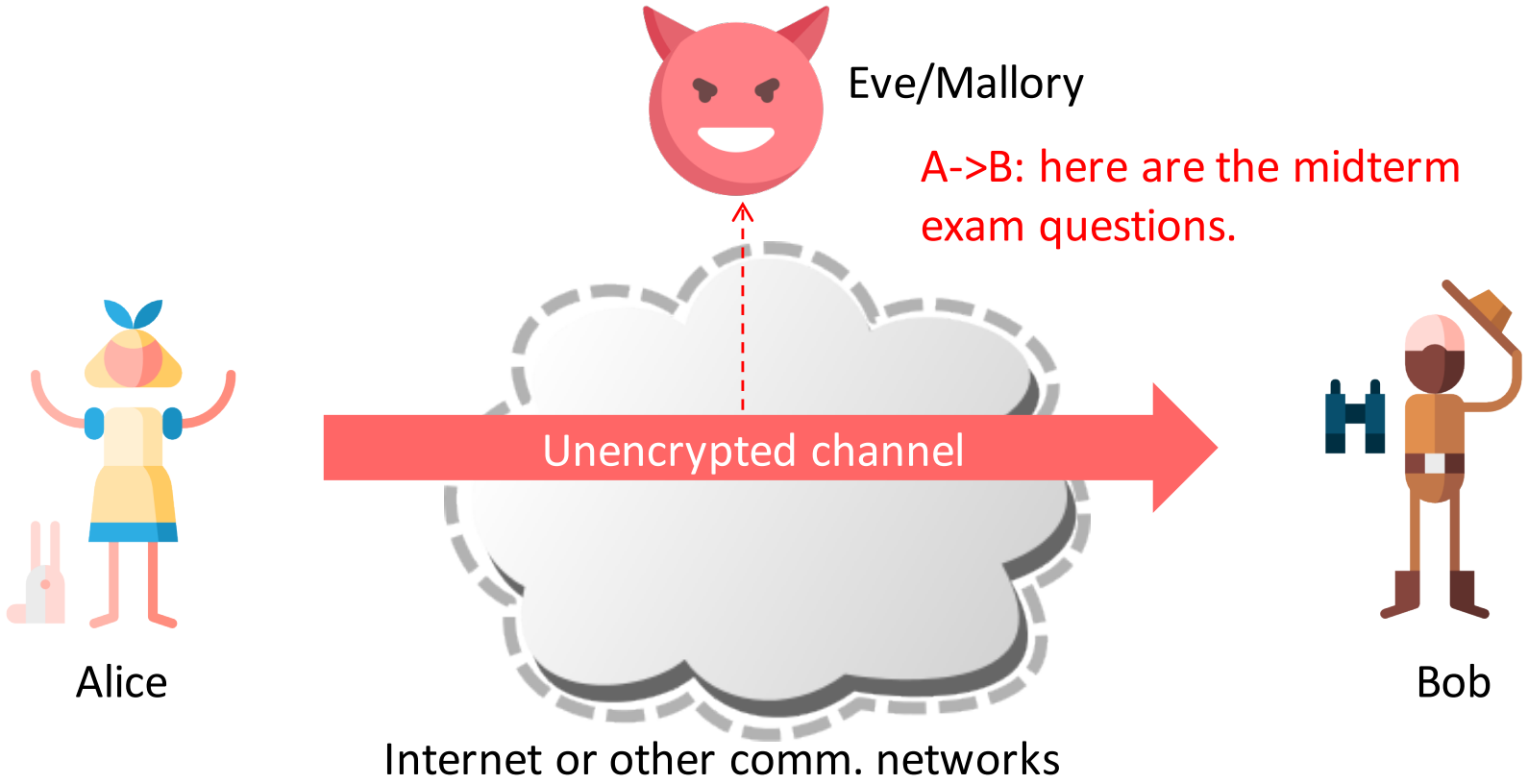
完整性

可得性、可用性

Confidentiality (保密性)

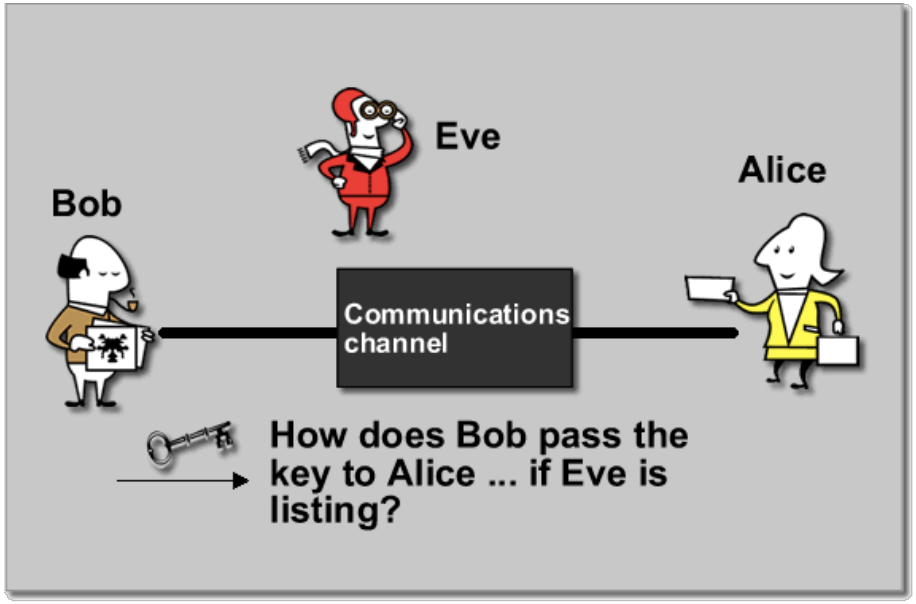
Confidentiality is protection from unauthorized disclosure

Eavesdropping on messages violates confidentiality



Trivia! Alice, Bob, and Eve

Alice and Bob are two commonly used placeholder names in the security field.

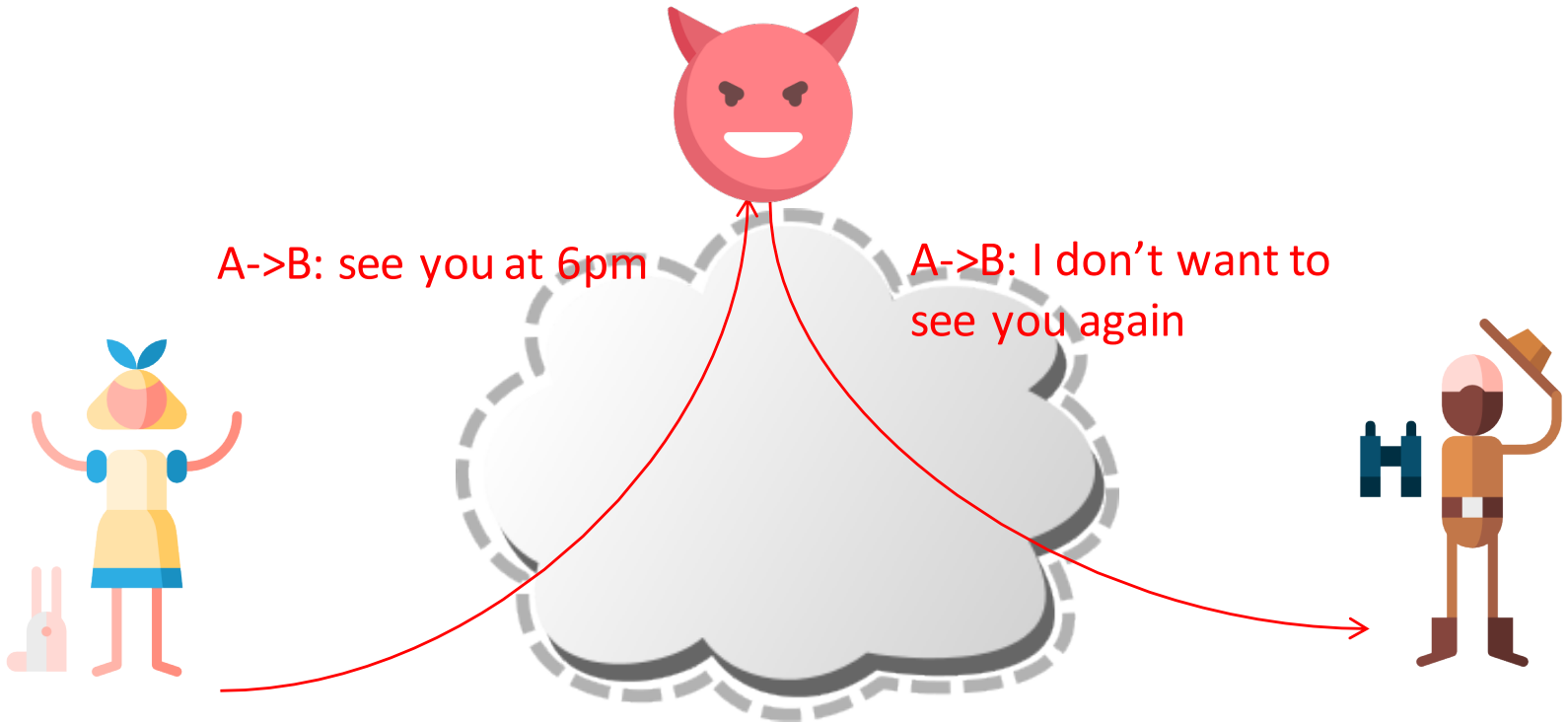


http://billatnapier.com/design_tips240.htm

Integrity (完整性)

Integrity is protection from unauthorized changes

Modification of messages violates integrity

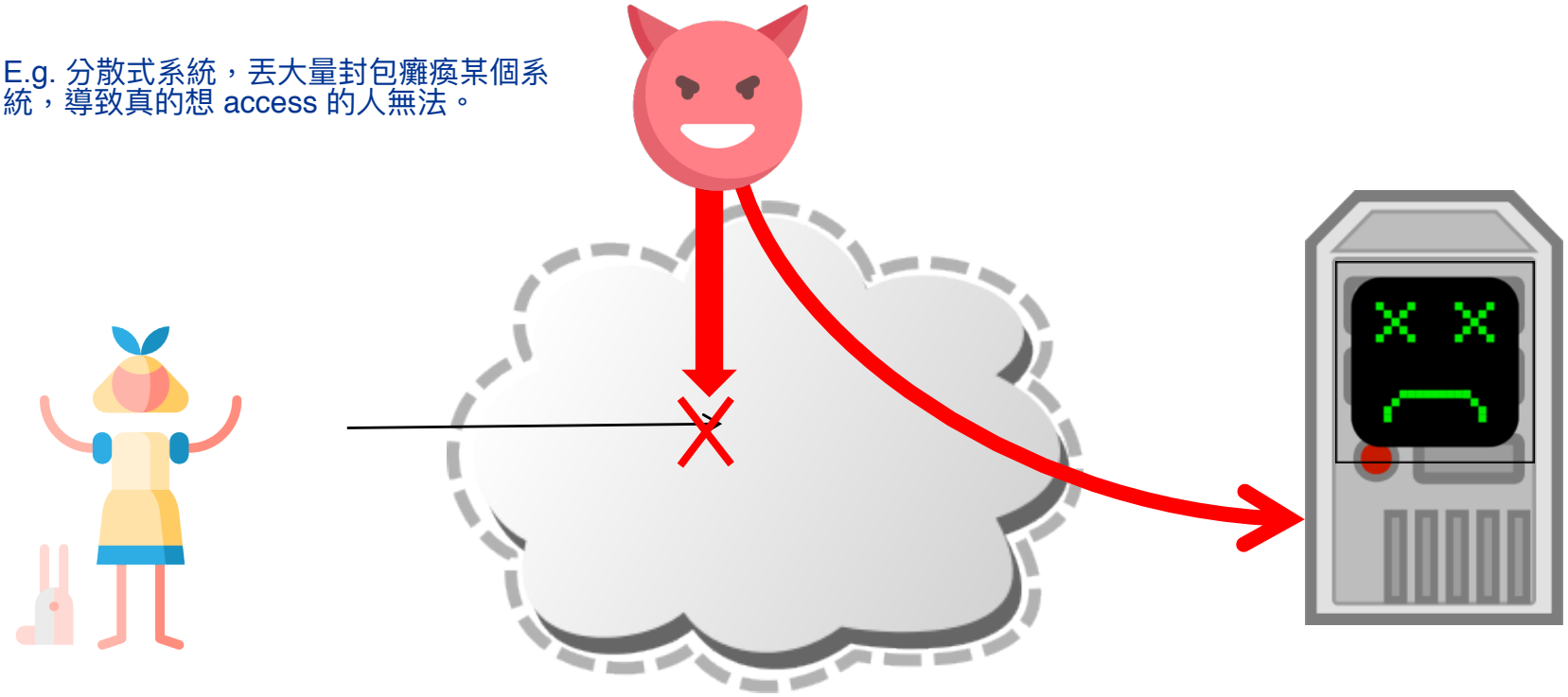


Availability (可用性)

Availability ensures intended users can access service

Denial of Service violates availability

E.g. 分散式系統，丟大量封包癱瘓某個系統，導致真的想 access 的人無法。



Availability：因為沒有連到對的網站
Integrity：DNS 被更改，因此正確的 mapping 被更改了。

Exercise: which security requirement is violated?

cyber.dhs.gov

Home

Blog

ED 19-01 - Mitigate DNS Infrastructure Tampering

Background

Required Actions

Emergency Directive 19-01

January 22, 2019

Mitigate DNS Infrastructure Tampering

This page contains a web-friendly version of the Cybersecurity and Infrastructure Security Agency's Emergency Directive 19-01, "*Mitigate DNS Infrastructure Tampering*". Additionally, see the Director's blog post.

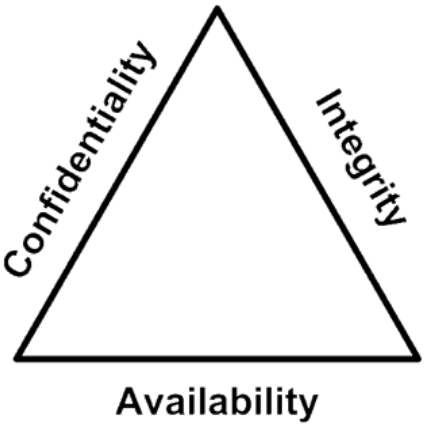
聯邦網站成為DNS挾持目標，美國國土安全部發出緊急指令

數個美國聯邦網站的網域名稱系統（DNS）遭挾持，駭客將使用者流量變更至駭客所控制的架構，再轉回合法服務，所造成的風險高過於短期重新定向使用者流量的作法

文/ 陳曉莉 | 2019-01-24 發表

https://cyber.dhs.gov/ed/19-01/
https://www.ithome.com.tw/news/128433

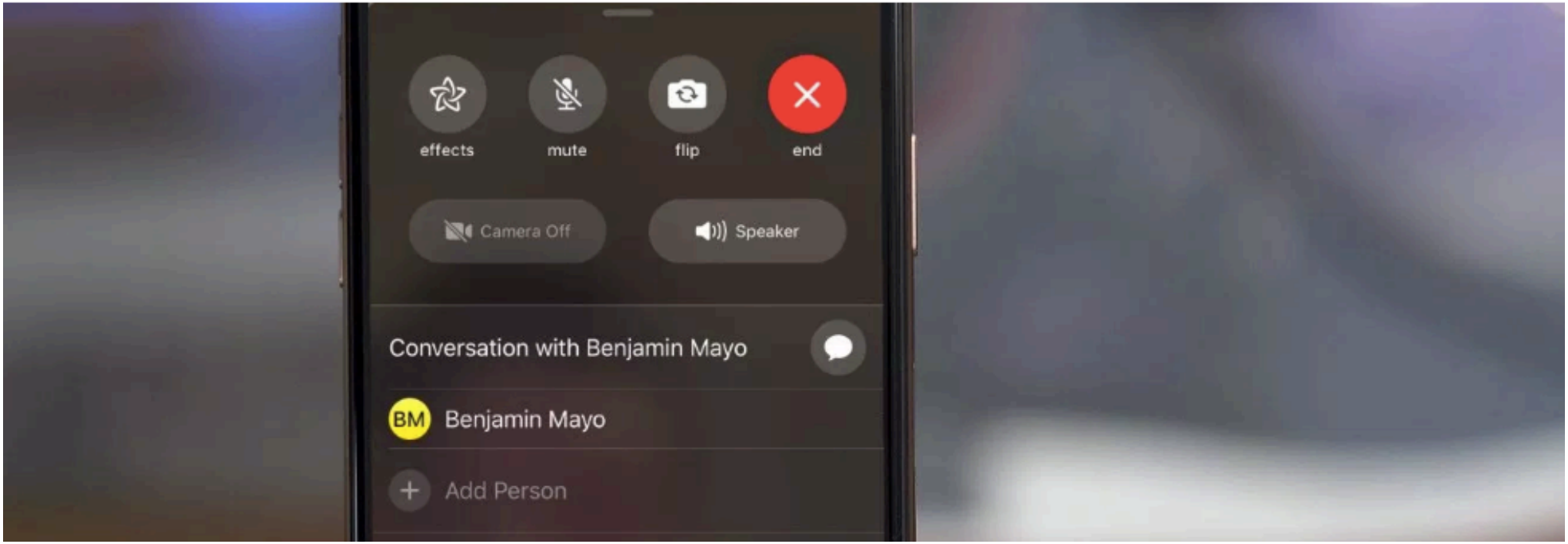
Exercise: which security requirement is violated?



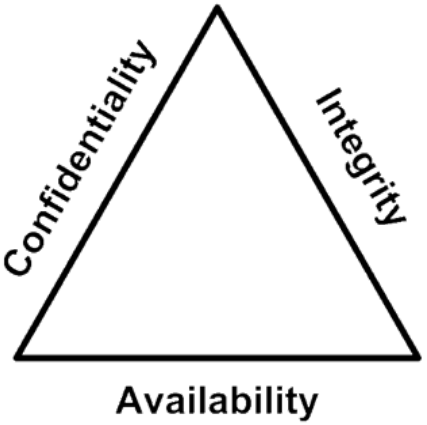
Confidentiality：我沒有要你聽到，你卻聽到了。

Major iPhone FaceTime bug lets you hear the audio of the person you are calling ... before they pick up

Benjamin Mayo - Jan. 28th 2019 3:41 pm PT @bzamayo



Exercise: which security requirement is violated?

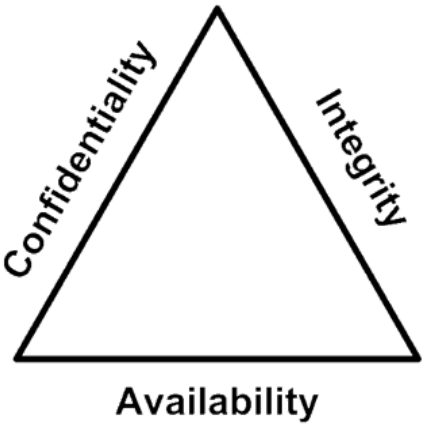


Availability

健保當機2.5小時 全台看診大亂

11504　出版時間：2019/02/19

Exercise: which security requirement is violated?



Availability

Workstation Team <217ta@csie.ntu.edu.tw> Fri, Nov 2, 2018, 3:42 PM

to faculty, Workstation

Chinese (Traditional) > English Translate message Turn off for: Chinese (Traditional)

教授、同仁們，您們好：

因系上IMAP遭受惡意D-DOS攻擊，目前IP已被計中防火牆隔離，故將導致教授群組受到影響（因目前只有教授群組未完全設定信件轉寄，只能透過IMAP管理信件）；
現已將影響程度降到最低，但特定使用族群仍會有信件伺服器無法連線的情況產生（如下表）：

CSIE WEBMAIL	NTU MAIL	GMAIL	手機APP或Outlook等
信件收發不受影響	信件收發不受影響	信件收發不受影響	限制必須於CSIE網域進行連線

如透過CSIE WEBMAIL（https://webmail.csie.ntu.edu.tw）收發CSIE Domain的信件皆能正常使用，藉由CSIE Mail Server轉信至NTU Mail或Gmail之信件亦不受影響。

Other security requirements

Authorization (授權)
Access control (存取控制)
Accountability (可歸責性)
Auditability (可稽核性)
Authenticity (鑑別性)
Non-repudiation (不可否認性)
Anonymity (匿名)
Privacy (隱私)
…

那要怎麼做到滴水不漏？

Wrong question!

100%安全、防禦所有攻擊，實務上是做不到的，為什麼？

- 預算有限
- 效能需求
- 未知的攻擊 (zero-day attacks)
- 難以掌控的因素 (如使用者的使用方式)

~~The system is 100% secure~~

The system provides [Security Requirement] against [Threat Model] under [Assumption]

針對攻擊者的假設：攻擊者的能力、知識、資源等

其他的假設：E.g., 假定所有的客戶都不將新發的提款卡照片po上網或是把密碼告訴別人

例子

The [system] provides [security requirement] against [Threat Model] under [Assumption]

System = ATM提款系統

Security requirement= 身份認證

Threat model = 撿到提款卡並亂試pin碼

很重要，因為要想到要防誰

Assumption = 使用者沒把pin碼寫在卡片套上或是用生日當pin碼

合理的threat model很重要

Threat model

Assumptions about the adversary

- Remember, we can't fight against every possible attack.

Several well-known models exist

- Chosen-plaintext attack (CPA), chosen-ciphertext attack (CCA)
- Honest-but-curious 合不合理要看情境
- Adversary in the Dolev-Yao model
- …

Threat model

Define by attacker's **capability**, **knowledge**, and **resource**

Capability – what can the attacker do?

- E.g., passive vs. active

Knowledge – what does the attacker know?

- E.g., insider vs. outsider

内鬼

Resource – how much resource does the attacker have?

- E.g., script kiddies vs. government-funded groups

What's a reasonable threat model? It depends.

- Risk = impact of the attack × likelihood of the attack

沒有白吃的午餐 – Cost of Security

Security comes with a price

- 開發和維護的成本
- 系統效能降低
- 使用者抱怨

Technical challenge: making security mechanisms cheaper, faster, and more usable

沒有白吃的午餐 – Cost of Security

可能的攻擊這麼多怎麼辦？

沒辦法全防，但可以盡量提升攻擊成功的難度

定義一個合理的threat model

- 如根據risk排序
- Risk = impact of the attack × likelihood of the attack

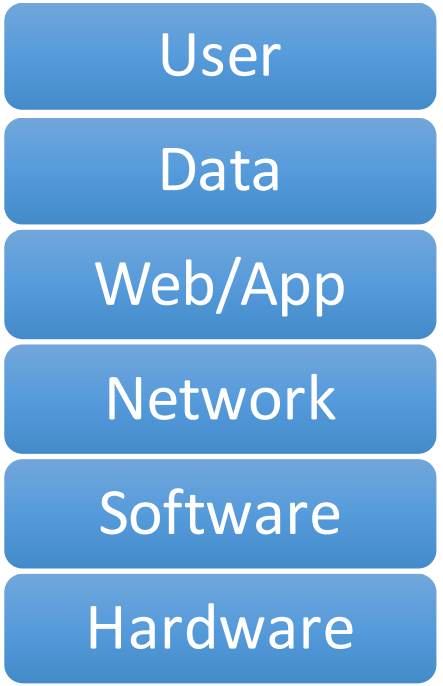
善用共享資源及時修補已知、一般性的漏洞

- Sharing intel to help timely fixes
- Many exploit kits for known attacks; even script kiddies can cause great damage.

把精力放在未知的、針對性的攻擊

安全性取決於最弱的環節

Security is only as strong as the weakest link



釣魚信件

Attack: Find one place to penetrate

Defense: Need to secure every place

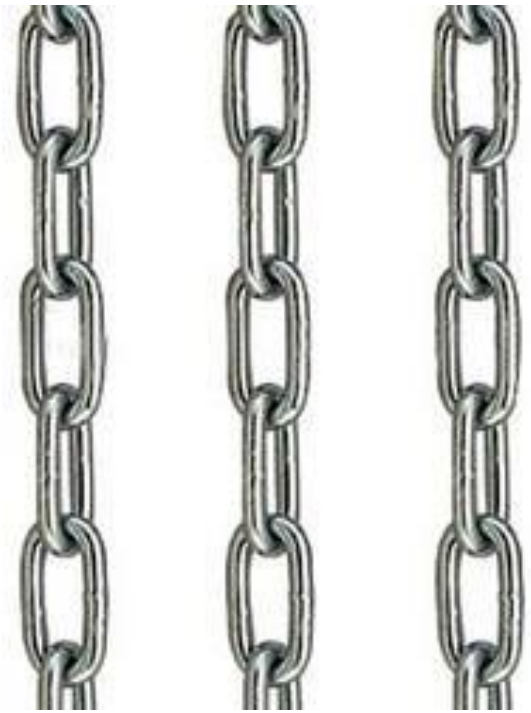
Defense in depth

Examples

- Two-factor authentication
- Anti-virus + firewall + IDS

We can combine multiple strategies

- Prevention
- Detection & Recovery
- Resilience
- Deterrence



Exercise

Security mindset:
Think about how to make it fail instead of how to make it work!

What are the security requirements?

What's the threat model? Is it reasonable?

What might be the weakest link?

通訊軟體

For regular users
For enterprises
For dissidents & journalists in repressive countries

電子投票

For poll
For referendum
For presidential election

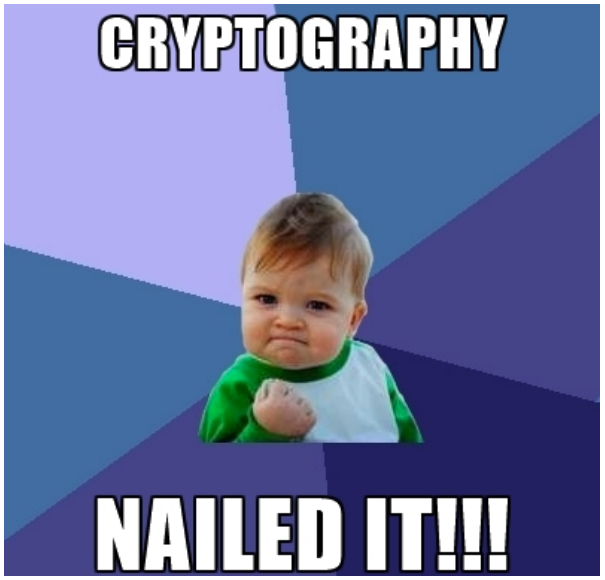
Introduction to Cryptography

密碼學 Cryptography

Mathematical tools to protect data at rest and data in motion from adversaries

(Modern) cryptography is more than encryption.

Security of cryptosystems relies on mathematical modeling and proofs based on plausible assumptions.



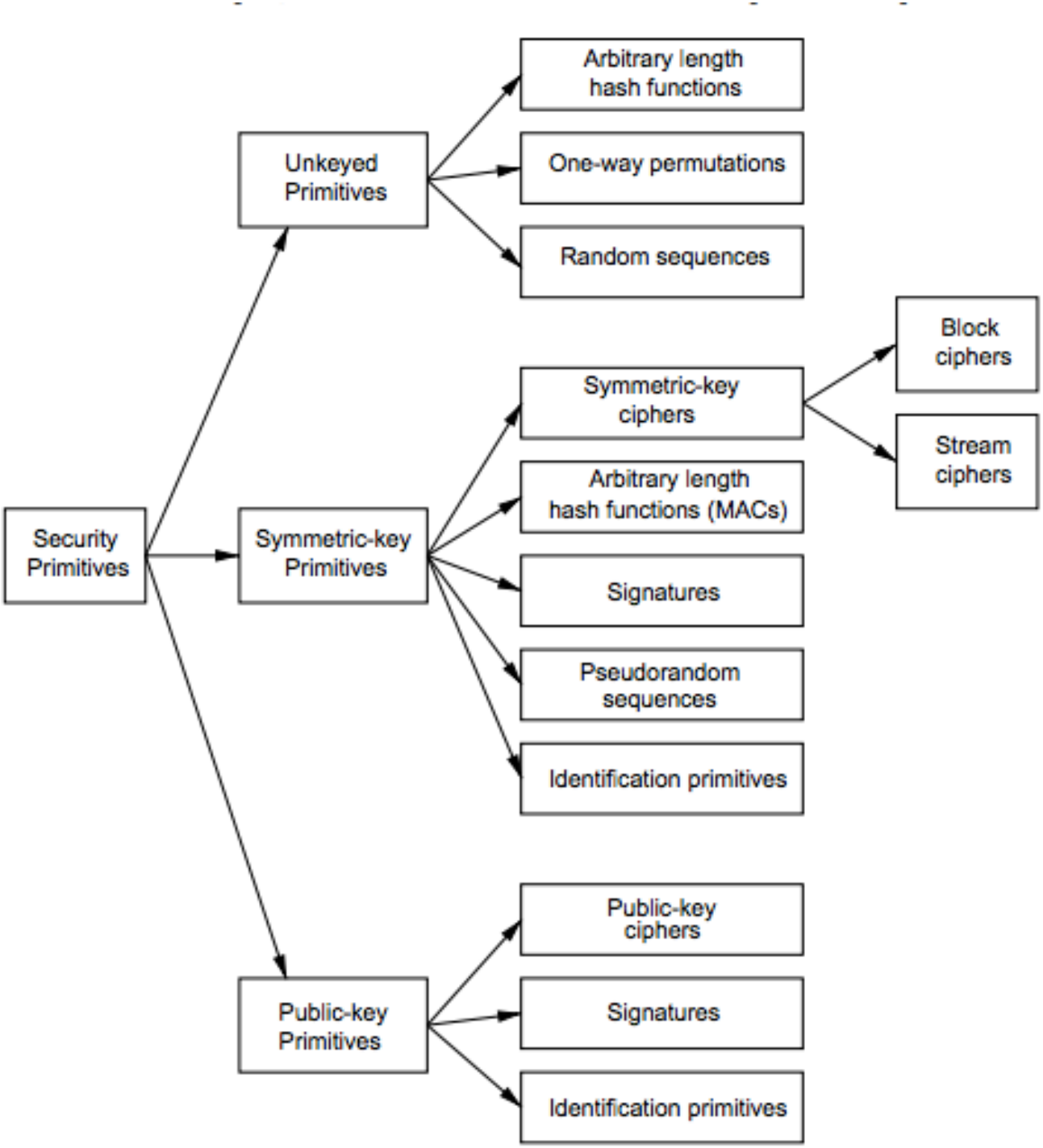


Figure 1.1: *A taxonomy of cryptographic primitives.*

Menezes, van Oorschot, and Vanstone. 1997. **Handbook of Applied Cryptography**. CRC Press

Basic cryptographic primitives

Unkeyed primitives

Cryptographic hash function

- Provides: One-wayness, weak/strong collision resistance

Cryptographically secure pseudorandom number generator

- Provides: computationally indistinguishable from true randomness

Basic cryptographic primitives

Symmetric (shared-key, same-key, secret-key)

Symmetric-key encryption

- Requires: secret key
- Provides: achieve secrecy with parties that share key

Message authentication code (MAC)

- Requires: secret key
- Provides: achieve authentication with parties that share key

Basic cryptographic primitives

Asymmetric (public-private key)

Diffie-Hellman key agreement

- Requires: authentic key from other party
- Provides: both parties can compute secret information

Public-key encryption

- Requires: authentic key from other party
- Provides: achieve secrecy for messages to other party

Digital signature

- Requires: authentic key from other party
- Provides: signature and authentication properties

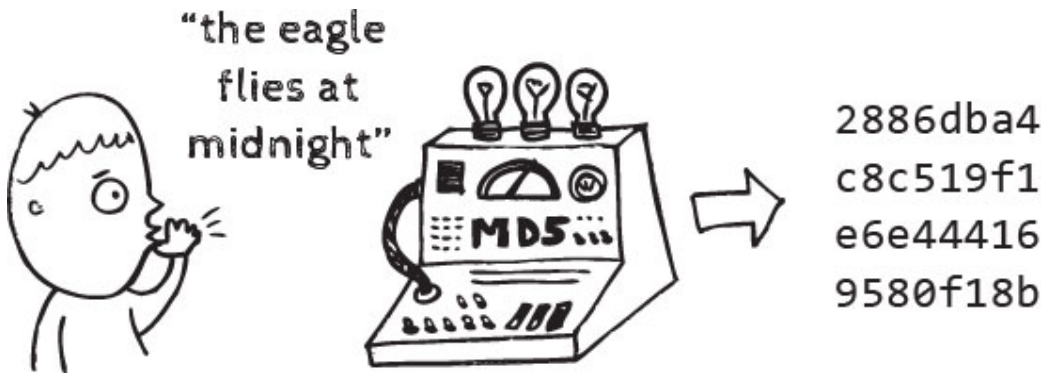
Cryptographic hash function

Maps **arbitrary**-length input to **finite** length output

- $y = H(x)$, y is the *hash* of x, and x is a *preimage* of y
- If $H(x') = H(x)$ and $x' \neq x$, then this is a *collision*
- Ensures one-wayness and collision resistance

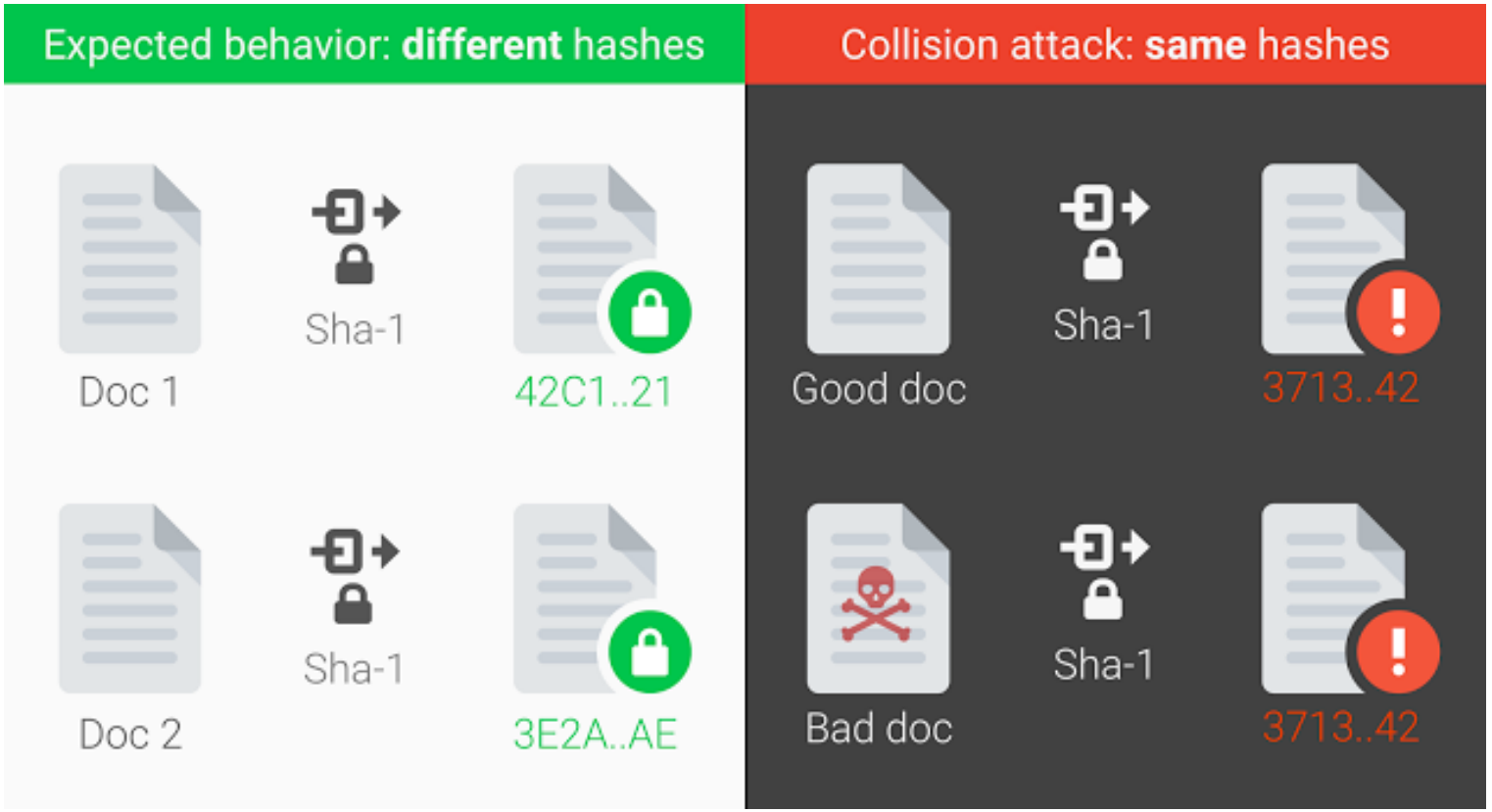
Applications 壓縮之後再比對短一點的 file，不用浪費時間比整份

- Integrity check
- Generating digest
- Commitment
- Password hashing
- Proof of Work



https://blog.varonis.com/the-definitive-guide-to-cryptographic-hash-functions-part-1/

SHA-1 collision found



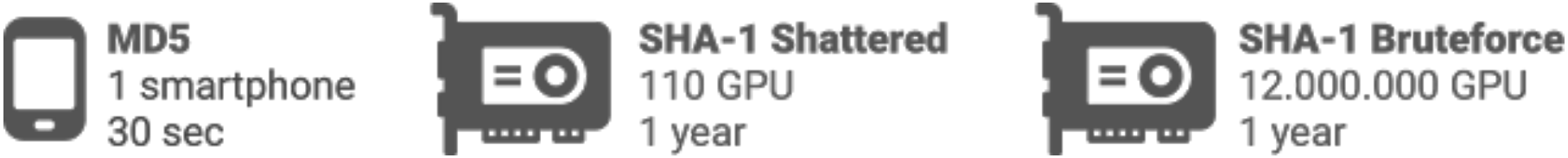
https://security.googleblog.com/2017/02/announcing-first-sha1-collision.html

SHA-1 collision found

SHAttered attack: $2^{63.1}$ SHA-1 evaluations

- Nine quintillion (9,223,372,036,854,775,808) SHA-1 computations
- 6,500 years of CPU computation to complete the attack first phase
- 110 years of GPU computation to complete the second phase
- Cost $110,000 using computing power from Amazon's EC2 cloud
- Still 100,000x faster than the brute force attack

Brute-force attack: 2^{80} SHA-1 evaluations



What's "computationally infeasible" currently?

~2^{33} devices in the world

~2^{30} symmetric cryptographic operations per device per second

~2^{25} seconds per year

~2^{128} operations to brute force AES-128 encryption

=> ~2^{40} years to brute force AES-128 encryption

Well, ~2^{33} years since the beginning of the universe

=> It is infeasible to crack AES-128 using brute force

Random Numbers

Many security schemes rely on the use of random numbers

Common requirements of a sequence of random numbers

- Randomness
  - Uniform distribution
  - Independence
- Unpredictability

"True" randomness implies unpredictability; not vise versa

Random Numbers

However, source of true randomness is rare.

- Hard to create by software alone
- Harvested from physical noises instead

We are settled with pseudorandom numbers in practice.

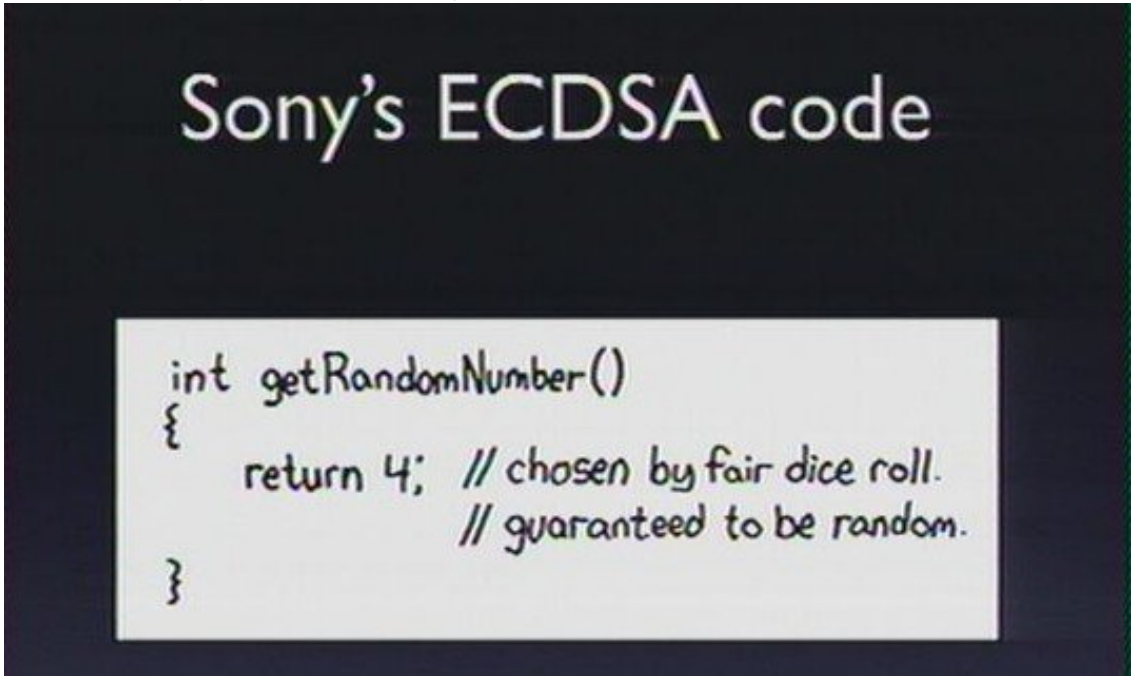
- Not statistically random, but shall pass many statistic tests
- **Pseudorandom number generator**: a deterministic algorithm to generate pseudorandom numbers

實際上只有假 random

Cryptographically secure pseudorandom number generator (CSPRNG)

Computationally indistinguishable from true randomness

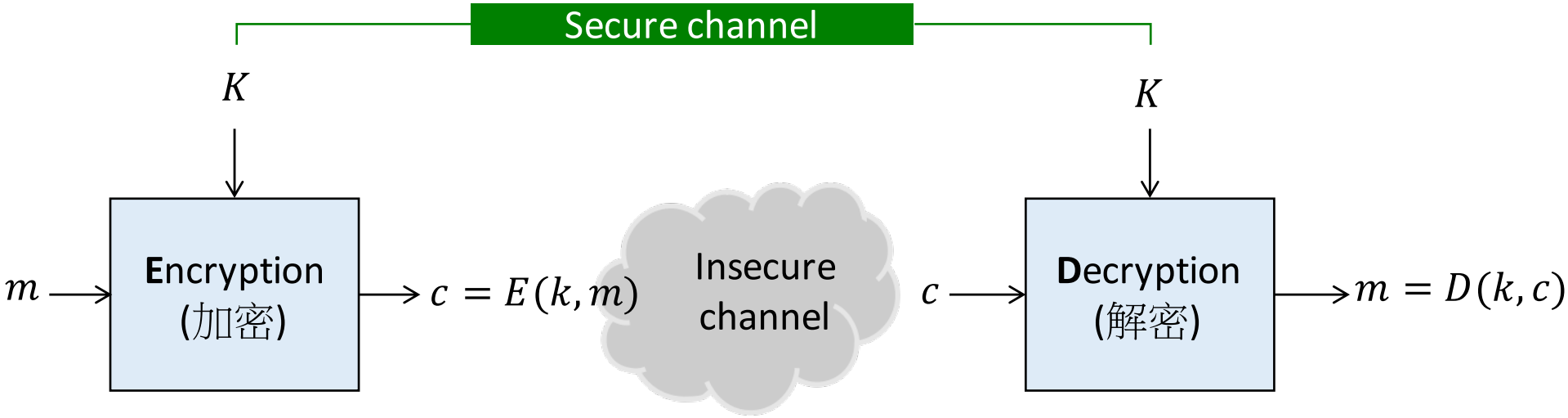
How crypto failed in practice:



https://www.engadget.com/2010/12/29/hackers-obtain-ps3-private-cryptography-key-due-to-epic-programm/

Symmetric-key encryption

Protects data in motion (e.g., communication) and data at rest (e.g., storage) against an *eavesdropper*



Provides confidentiality but not message integrity

Does not say how to securely share a secret key

A simple (yet broken) example: Caesar Cipher

$$E(k, m) = (m + k)\ mod\ 26$$
$$D(k, c) = (c - k)\ mod\ 26$$

m, c are characters

A kind of *substitution* ciphers

Example:

k = 2, m = **`apple`** → c = **`crrng`**

k = 3, c = **`rudqjh`** → m = ? orange

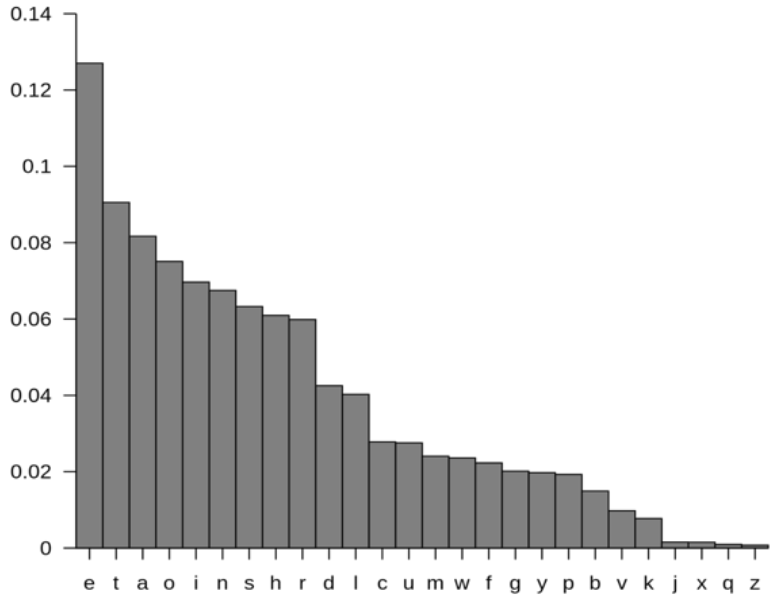
How would you break Caesar cipher?

Breaking Caesar Cipher

Brute force: Try all k (only 26 possibilities)

Frequency analysis: calculate the frequency of unigram, bigram, …

Natural Language Processing



A perfectly secure (yet impractical) example: One-time pad

XOR message with a random key of same length

$$E(k, m) = k \oplus m$$
$$D(k, c) = k \oplus c$$

http://users.telenet.be/d.rijmenants/en/onetimepad.htm

One-time pad achieves perfect security (under certain assumptions)

OTP: XOR message with a **random key** of **same length**

In information theory and cryptography, **one-time pad** is an encryption scheme that is **unconditionally secure**

- $\Pr[E(\boldsymbol{k}, m_0) = c] = \Pr[E(\boldsymbol{k}, m_1) = c]$
  - k is a random variable uniformly drawn from the key space
- $\Pr[\boldsymbol{m} = m | \boldsymbol{c} = c] = \Pr[\boldsymbol{m} = m]$
  - for each $m \in \mathcal{M}$ and each $c \in \mathcal{C}$ with non-zero probability

One-time pad achieves perfect security (under certain assumptions)

Threat model

- An eavesdropper attempts to learn something about the plaintext or key from the observed information

Assumptions

- Assume the sender and receiver share a random secret of length of m
- Assume the key is never reused

Security of encryption schemes

Ideally, we would like to ensure that

- Ciphertext leaks no info about key and/or plaintext
- Plaintext and ciphertext pairs leaks no info about key

However, such **information-theoretical security** is hard to achieve in practice.

- It's proven that keys must be at least as long as messages: if an encryption scheme is perfectly secure, then $|\mathcal{K}| \geq |\mathcal{M}|$.

Instead, we would aim for **computational security**: a computationally bounded adversary cannot recover the key or plaintext in reasonable time.

Modern ciphers

Stream ciphers

用short secret 產生看似random的long key stream

Encrypt one symbol at a time

Example: RC4

Block ciphers

把訊息拆成固定長度的短block，一個一個處理

Encrypt one block (a group of symbols) at a time

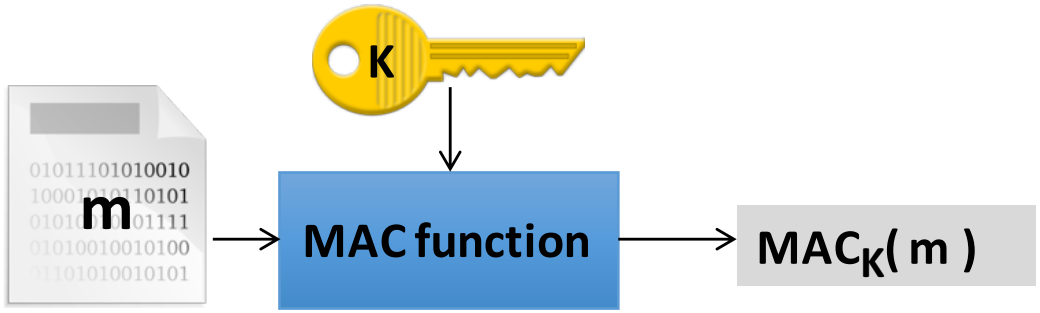
Example: DES, AES

Message Authentication Codes (MAC)

Message authentication codes (MAC), or keyed hash

Provides **integrity** and **authenticity**

- Integrity: *m* was not modified
- Authenticity: *m* was created by the key owner (which implies integrity)



Why public-key cryptography?

對稱式密碼學難以解決：

Key distribution problem

- 對稱式密碼學假設雙方已建立共同的秘鑰
- 秘鑰怎麼安全地建立？

Digital signatures

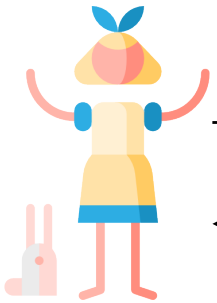
- 數位簽章需提供「不可否認性(non-repudiation)」
- 在對稱式密碼學中，簽名方跟驗證方須有共同的秘鑰
- 雙方知道的都相同，要如何達到不可否認性？

Merkle's Puzzles

Goal: Alice and Bob wants to establish a shared secret in the presence of an eavesdropper, Eve.

1. Pick random k_i, s_i;
Compute n puzzles,
$P_i = (E(k_i, s_i), E(k_i, i), E(k_i, 0))$

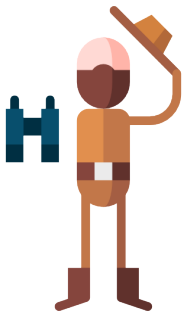
index



Eve

2. n puzzles in random order

4. index, ℓ



3. Pick $P_j = (c_1, c_2, c_3)$; Recover a key k s.t. $D(k, c_3) = 0$ using brute-force search; then

$$\ell \leftarrow D(k, c_2)$$
$$s_\ell \leftarrow D(k, c_1)$$

Established a shared secret s_ℓ

Merkle's Puzzles

An early example of public key cryptography

Computational complexity:

- Alice: $O(n)$
- Bob: $O(|\mathcal{K}|)$
- Eve: $O(n|\mathcal{K}|)$

Complexity gap: 攻擊者要算很久、Alice/Bob不用

然而quadratic complexity gap (when $n = |\mathcal{K}|$) 還是不太夠

Merkle, R. C. (April 1978). "Secure Communications over Insecure Channels". *Communications of the ACM*. **21** (4): 294–299.

More story: http://merkle.com/1974/

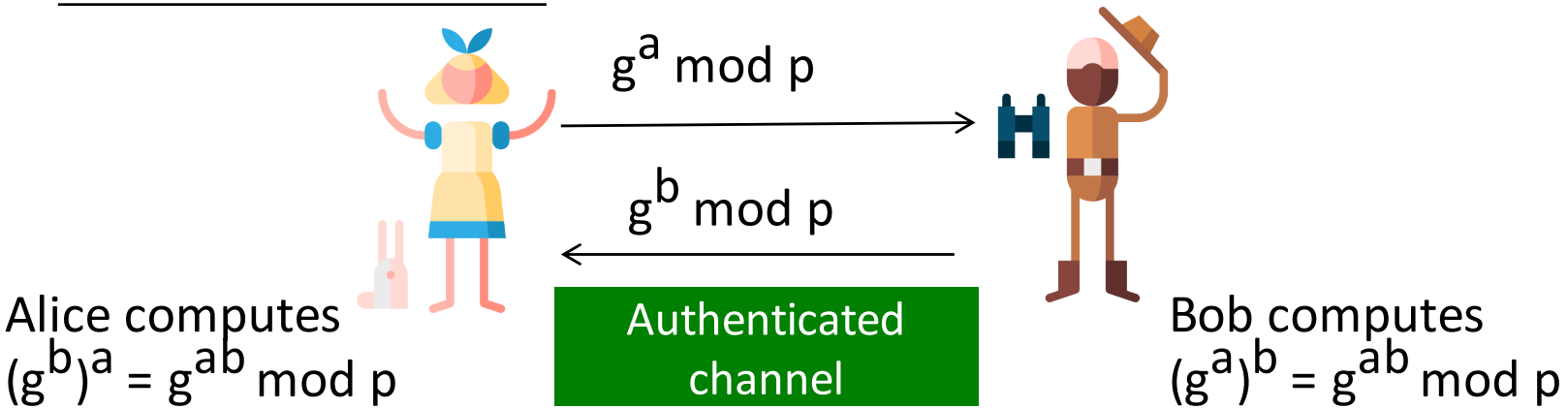
Diffie-Hellman key agreement

Setup:

Public values: large prime p, generator g

Alice picks a secret a, and Bob picks secret b

Key agreement:



Alice and Bob can then use g^{ab} mod p to derive their shared key

2015 Turing Award Winners

Whitfield Diffie and Martin Hellman

For their critical contributions to modern cryptography

- "Diffie and Hellman's groundbreaking 1976 paper, New Directions in Cryptography, introduced the ideas of public-key cryptography and digital signatures, which are the foundation for most regularly-used security protocols on the Internet today."

http://amturing.acm.org/

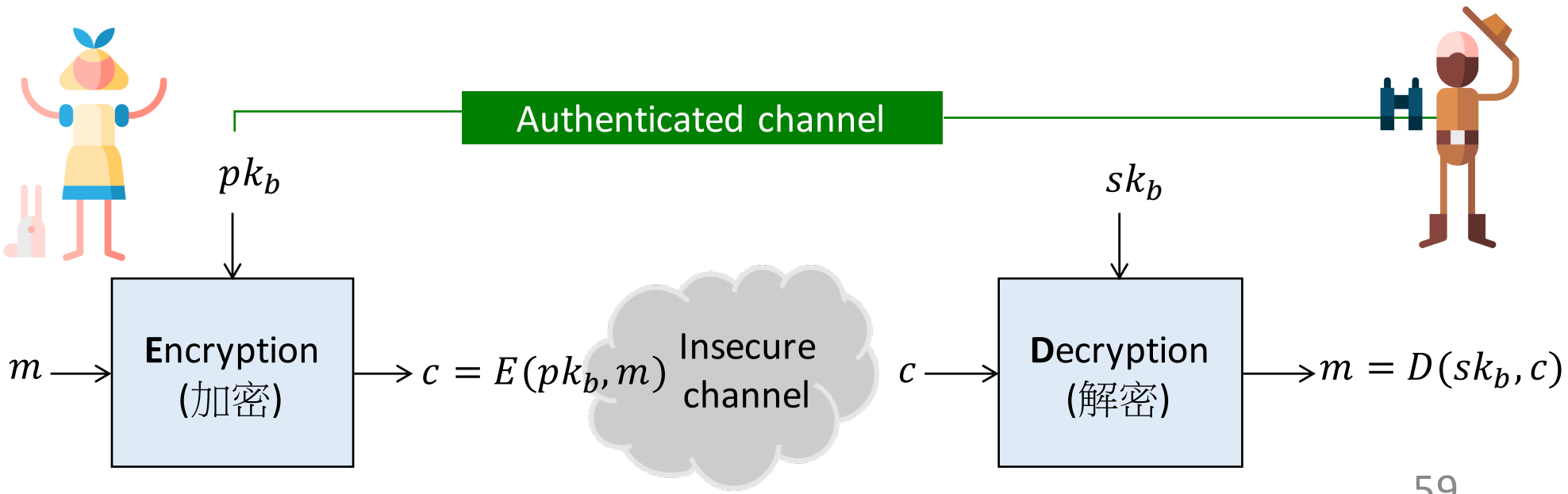
Public-key encryption

Bob has a public/private key pair (pk_b, sk_b).

Assume an authenticated channel to publish pk_b

- Authenticated channel: msgs can't be modified but can be overheard

Bob keeps his own private key sk_b in secret, so only Bob can decrypt c, and only Alice and Bob know m.

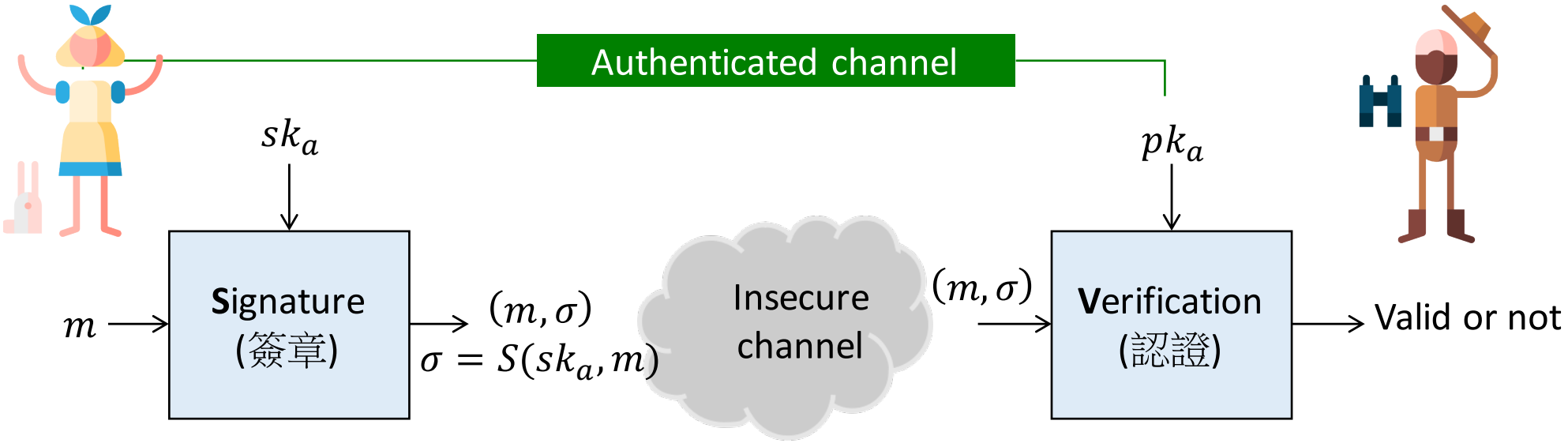


Digital Signatures

Alice has a public/private key pair (pk_a, sk_a).

Only Alice can create this signature (non-repudiation).

Eve can also verify this signature, but any modification will be detected.

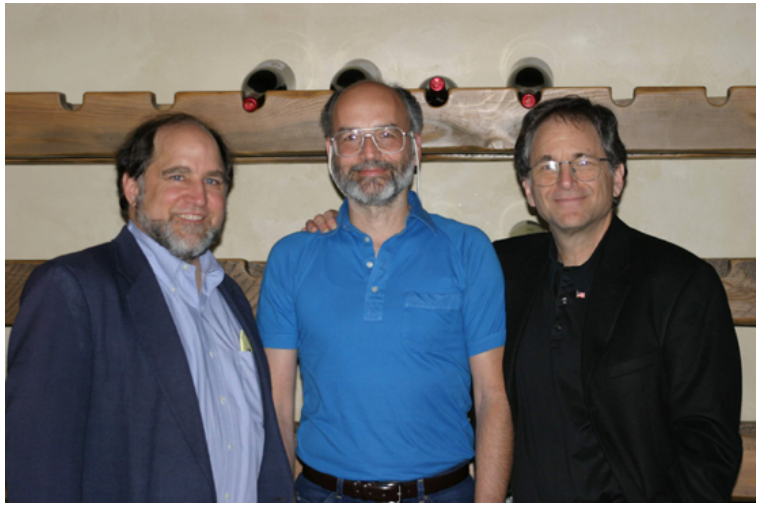


2002 Turing Award Winners

Ron Rivest, Adi Shamir, Leonard Adleman

For their ingenious contribution for making public-key cryptography useful in practice.

- Rivest, Shamir, and Adleman presented practical implementations in their 1977 paper, "A method for obtaining digital signatures and public-key cryptosystems," which showed how a message could easily be encoded, sent to a recipient, and decoded with little chance of it being decoded by a third party who sees it.



Symmetric vs. asymmetric crypto

Symmetric crypto

Both parties share same key

Secret key (or shared key) only known to communicating parties

For secure comm., key should be secret & authentic

Asymmetric crypto

Each party has a public & a private key

Public key known to everyone

Private key only known to owner

For secure comm., private key is secret and public key is authentic

Comparison sym vs. asym crypto

Symmetric crypto	Asymmetric crypto
112 bit key for high security (year 2015)	2048 bit key (RSA) for high security (year 2015)
~1,000,000 ops/sec on 1GHz processor	~100 signatures/sec ~1000 verify/s (RSA) on 1GHz processor
10x speedup in HW	Limited speedup in HW

為什麼密碼學的演算法應該要公開？

Secrets are hard to protect (so should be minimized)

Allows system to be openly examined by many people

Kerckhoffs's principle，重要的資安原則之一

- "A cryptosystem should be secure even if everything about the system, except the key, is public knowledge."
- 以住家安全為例：安全性應仰賴門鎖，而不是把家的格局建得像迷宮。

Security through transparency vs. security through obscurity

很重要所以說三遍

Don’t design or implement your own cryptographic algorithms!

Cryptography is highly brittle; A single specification or programming error can make it completely insecure.

Always use well-developed standards and libraries

In other words, if someone designs his/her own crypto algorithms, you are likely able to break it.

Review

Security requirements: e.g., confidentiality, integrity, availability

Threat model

Security is as strong as the weakest link.

One-time pad and perfect security

Computational infeasibility

Symmetric vs. asymmetric cryptography

“Crypto is bypassed, not penetrated” – Adi Shamir