# Security, Privacy & Integrity

## AS Computer Science ▪ Topic 6

A-Level Computer Science

# What we will cover

1. Security, privacy & integrity
2. Threats
3. Security measures
4. Protecting the data
5. Data integrity
6. Recap

# Three different ideas

**security 安全**

protect data from unauthorised access or change

**privacy 隐私**

who is allowed to see personal data

**integrity 完整性**

data is accurate & complete, not corrupted

# Malware

two ways **malware** 恶意软件 behaves

## self-spreading 自我传播

- **virus** 病毒
- **worm** 蠕虫

spreads copies on its own

## hidden / disguised 伪装

- **Trojan** 木马
- **spyware** 间谍软件
- **ransomware** 勒索软件
- **adware** 广告软件

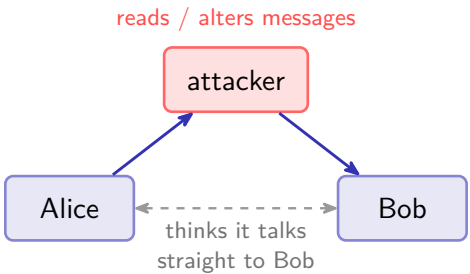
# Social and network attacks

**Tricking people:**

- **phishing** 网络钓鱼 – fake emails/sites
- **pharming** 域名欺骗 – redirect to a fake site

**On the network:**

- **DoS / DDoS** 拒绝服务 – flood a server
- **man-in-the-middle** 中间人攻击 – relay/alter messages



# Security measures


- **standalone:** password, **antivirus** 杀毒软件, updates, **backup** 备份, **encryption** 加密
- **networked:** + **firewall** 防火墙, permissions, **audit logs** 审计日志
- **internet:** **VPN** 虚拟专用网, HTTPS/TLS, **digital signatures** 数字签名

# Match the measure to the threat

interception → **encrypt**

unauthorised access → **authentication**

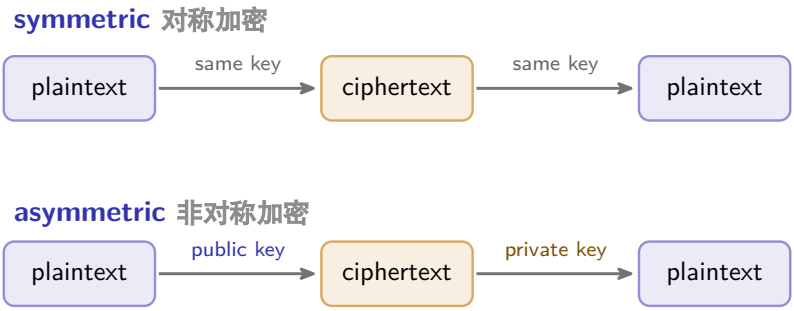
malware → **antivirus + patching**

phishing → **training + filtering**

insiders → **least privilege**

DDoS → **rate limiting**

# Encryption




- **symmetric** 对称加密: one shared key locks & unlocks (AES)
- **asymmetric** 非对称加密: **public key** 公钥 locks, **private key** 私钥 unlocks (RSA)

# Authentication

Three factors – strongest combined (**two-factor** 双因素认证):

- know – a password
- have – a token or phone code
- are – **biometrics** 生物识别 (fingerprint, face)


*"something you have"*

# Validation vs verification

**validation 验证**

*"is the data sensible?"*

- range / type / format
- check digit

catches nonsense data

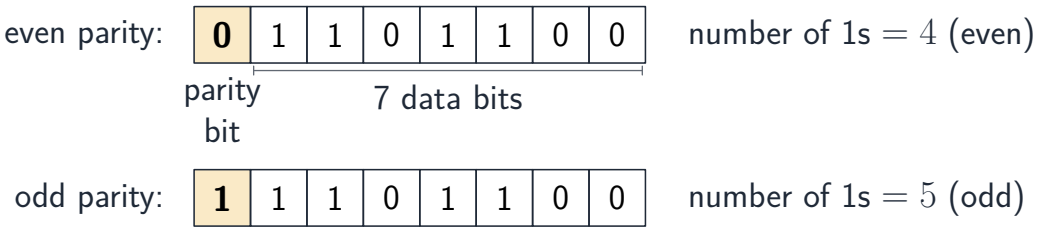
**verification 核对**

*"was it copied correctly?"*

- double entry
- parity / checksum

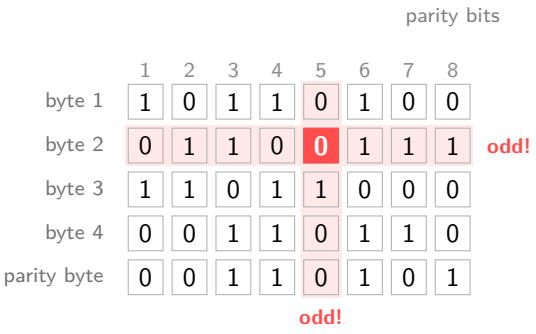
catches copying errors

# Verification on transfer – parity




**parity check** 奇偶校验: count the 1s – a flipped bit changes the count, so the error shows up.

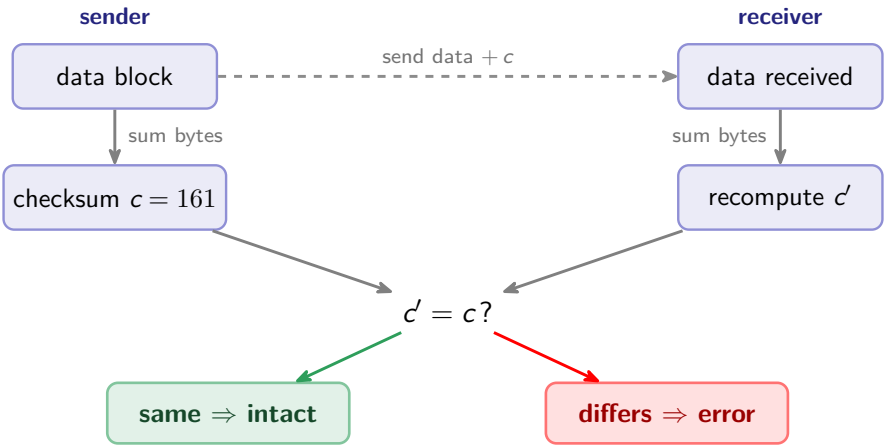
# Worked example: locate the flipped bit




- sent with even parity in every row *and* column
- receiver counts: row 2 odd, column 5 odd
- the error sits where they cross – flip it back to 1

One parity bit only detects an error; a parity block locates it – so it can be corrected.

# Verification on transfer – checksum



# Topic 6 – key takeaways

**1 Three ideas**

security, privacy, integrity differ

**2 Threats**

malware, phishing, DoS, man-in-the-middle

**3 Encryption**

symmetric (one key) vs asymmetric (key pair)

**4 Integrity**

validation = sensible; verification = copied right

# Check yourself

1. Which malware spreads over a network with no user action?
2. Which key decrypts in asymmetric encryption?
3. Is a range check validation or verification?
4. What does a parity bit detect?

**Answers** **1.** a worm **2.** the private key **3.** validation **4.** a single-bit error