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

1 Three ideas

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

Security, privacy and integrity are three different things

security	keeping data safe from unauthorised access, damage or loss
privacy	controlling who is allowed to see data about a person – a legal and ethical question, not a technical one
integrity	keeping data accurate and consistent – uncorrupted in storage and unchanged in transit
they can conflict	a full audit log improves integrity and reduces privacy
DoS/DDoS 拒绝服务	floods a server so real users cannot reach it – an attack on availability, not on secrecy

An encrypted file that has been silently altered is secure and *not* intact. Naming which of the three a scenario threatens is the first mark.

2 Threats

Malware

two ways **malware** 恶意软件 behaves

self-spreading 自我传播

- **virus** 病毒
  attaches to a host file
- **worm** 蠕虫
  needs no host file

spreads copies on its own

hidden / disguised 伪装

- **Trojan** 木马
- **spyware** 间谍软件
  logs keystrokes and passwords
- **ransomware** 勒索软件
  encrypts files, demands payment
- **adware** 广告软件
  pushes unwanted adverts

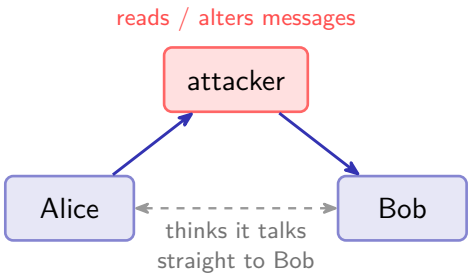
Social and network attacks

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



3 Security measures

Security measures

- **standalone PC:** a strong password, **antivirus** 杀毒软件 kept up to date, prompt updates, **backup** 备份 to separate media, full-disk **encryption** 加密, a locked screen
- **networked PC:** all of that, plus a **firewall** 防火墙, per-user permissions (admin rights only for admins), central **user accounts** 用户账户 and **audit logs** 审计日志
- **across the internet:** a **VPN** 虚拟专用网 encrypting traffic to the gateway, HTTPS/TLS, **digital signatures** 数字签名 proving who sent a message and that it was not altered, and intrusion detection watching for known attack patterns

Match the measure to the threat

interception → **encrypt**

unauthorised access → **authentication**

malware → **antivirus, anti-spyware 反间谍软件, patching**

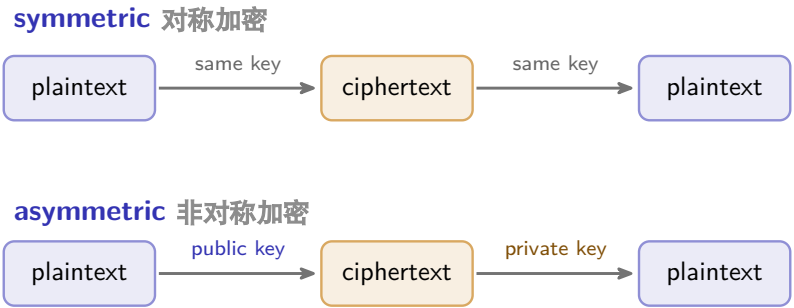
social engineering 社会工程 → **training + filtering**

insiders → **least-privilege 最小权限** + auditing

DDoS → **rate limiting**

4 Protecting the data

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 authentication** 双因素认证. Authentication proves *who* you are; **authorisation** 授权 decides what you may then do.

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

"something you have"

5 Data integrity

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

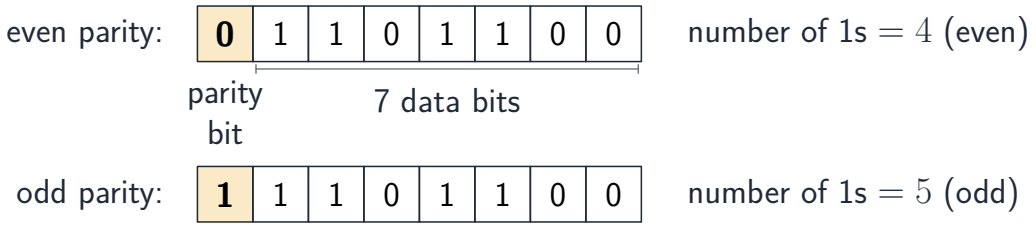
catches copying errors

The validation checks, by what each catches

range check	the value lies between two limits – a month between 1 and 12
type check	the data is of the right type – a number where a number is required
length check	the right number of characters – a 16-digit card number
format check	matches a pattern – an email must contain `@`
presence check	required fields are not empty
existence check	the referenced item exists – a product code is in the table
a check digit 校验位	an extra digit computed from the others (ISBN, card numbers) that spots transcription errors

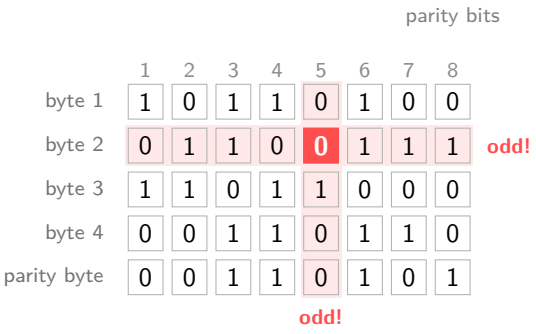
Validation asks *is this sensible*; verification asks *is this what was sent*. A validated wrong value passes every check on this list.

Verification on transfer – parity



parity check 奇偶校验: count the 1s – a flipped bit changes the count, so the error shows up. A **parity block check** 奇偶块校验 goes further: give each byte a row parity bit and compute a column parity byte too, and the crossing row and column locate the flipped bit.

Worked example: locate the flipped bit

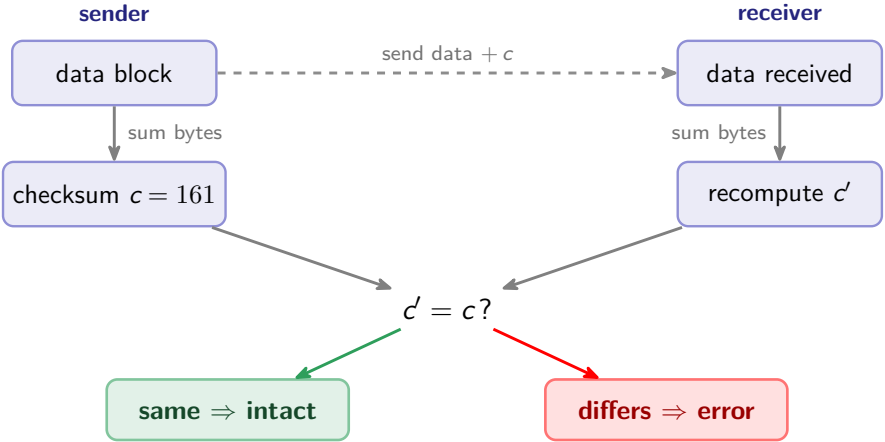


- sent with even parity in every row *and* column
- receiver counts: row 2 odd, column 5 odd
- the error sits where they cross – flip it back to 1

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

Verification on transfer – checksum

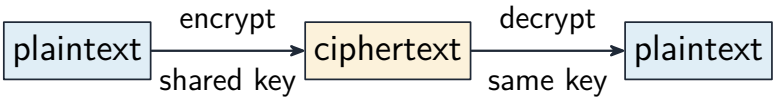
A **cyclic redundancy check** 循环冗余校验 (CRC) is a stronger checksum using polynomial division.



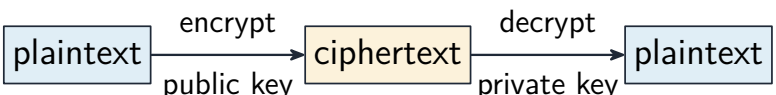
Protecting the data itself

Access control 访问控制 – file permissions enforced by the OS; backups, some off-site; physical security – locked server rooms and cable locks; and encryption: symmetric shares one key, asymmetric encrypts with a public key and decrypts with a private one.

symmetric

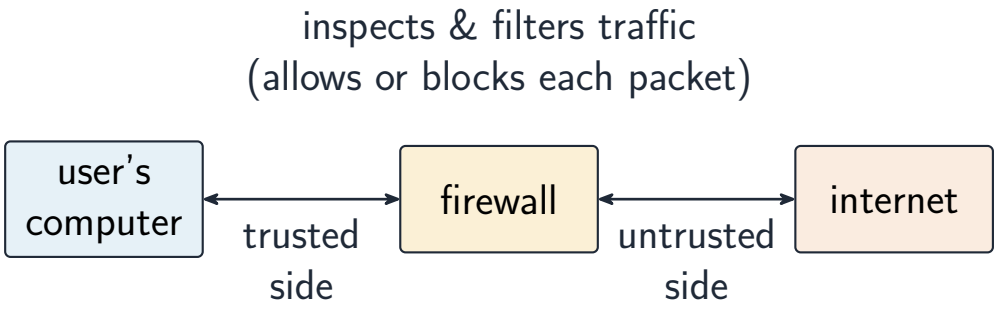


asymmetric



A networked PC

A firewall sits between the user's computer and the internet



Threats come in three groups

malware

software written to harm: viruses and worms that spread on their own; a **Trojan horse** 木马, spyware, ransomware or adware that hides

tricking people

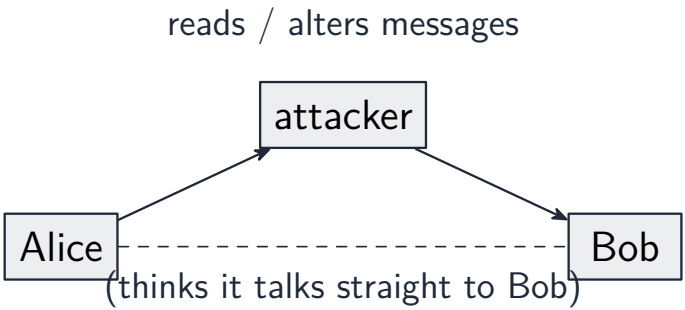
phishing 网络钓鱼 emails and **pharming** 域名欺骗 redirects that send you to a convincing fake

attacking the network

hacking 黑客入侵 through weak passwords or software flaws; **denial of service** 拒绝服务 flooding a server; **eavesdropping** 窃听 on open Wi-Fi

Two things need protecting: the data (confidential, intact, available) and the system itself – a compromised machine goes on to attack others.

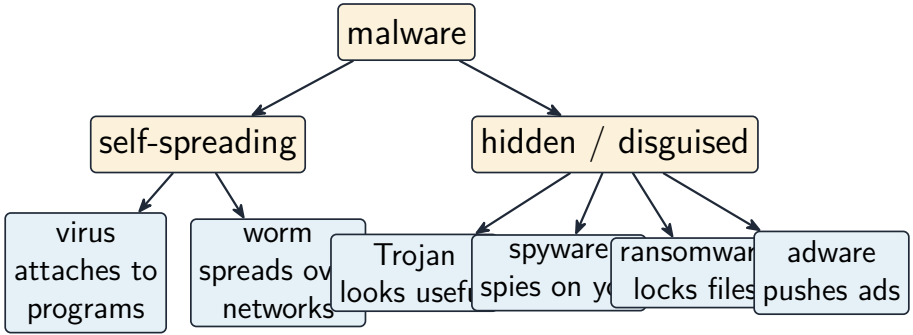
The man in the middle



The attacker relays both sides, so each believes it is talking straight to the other – what HTTPS and **digital signatures** 数字签名 exist to stop.

Threats from networks and the internet, continued

Malware by behaviour: self-spreading (virus, worm) versus hidden/disguised (Trojan, spyware, ransomware, adware)

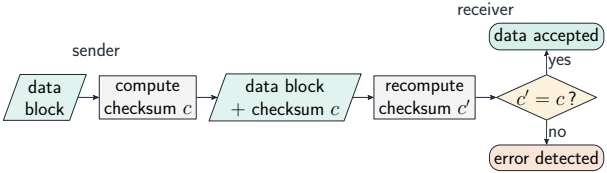


Verification – was the data entered or transferred correct

> Validation checks the data makes sense; verification checks it was copied without change

validation "is this data sensible?"	**verification** "was it copied correctly?"
checks rules *before* storing:	checks the data did not change:
· range / type / format	· double entry
· length / presence	· parity check
· check digit	· checksum / CRC
catches nonsense data	**catches copying errors**

Working out a checksum for a block of data



worked example – checksum = (sum of bytes) mod 256

1 sum of all bytes: $X = 1185$

2 $X \div 256 = 4.629$, round down: $Y = 4$

3 $Z = Y \times 256 = 1024$

4 checksum $= X - Z = 1185 - 1024 = 161$

The sender computes the checksum and sends it with the data; the receiver computes it again and compares. A difference proves the data changed – but a match is not proof it did not.

Worked example – validation or verification?

A user types their date of birth as 31/02/2009, and types their email address twice. Which check catches which error?

- **Validation** asks "is this data *sensible*?" – a format or range check rejects 31/02/2009, because February never has 31 days.
- **Verification** asks "was it *entered correctly*?" – typing the email twice is double entry, and comparing the copies catches a typing slip.
- The limit: validation can never say the data is *right*, only that it is *possible*.

01/02/2009 passes every validation rule even when the user was born in March. That sentence is the answer to most questions on this topic.

Exam tips

- Keep the three ideas separate: **security** (keeping data safe), **privacy** (who may see it), **integrity** (keeping it correct).
- Match each **threat** (malware, hacking, phishing, interception) to a **measure** (firewall, encryption, authentication, access rights).
- Encryption protects **confidentiality, not integrity** – use a checksum, parity or check digit for integrity.
- Distinguish a **virus, worm and Trojan** and how each spreads.

6 Recap

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