

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

protect data from
unauthorised
access or change

privacy

who is allowed to
see personal data

integrity

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

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.



Malware

two ways **malware** behaves

self-spreading

- **virus**

attaches to a host file

- **worm**

needs no host file

spreads copies on its own

hidden / disguised

- **Trojan**

- **spyware**

logs keystrokes and passwords

- **ransomware**

encrypts files, demands payment

- **adware**

pushes unwanted adverts

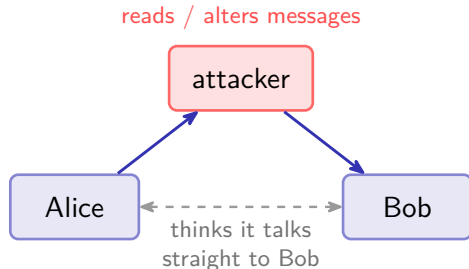
Social and network attacks

Tricking people:

- **phishing** – fake emails/sites
- **pharming** – redirect to a fake site

On the network:

- **DoS / DDoS** – flood a server
- **man-in-the-middle** – relay/alter messages





Security measures



- **standalone PC:** a strong password, **antivirus** kept up to date, prompt updates, **backup** to separate media, full-disk **encryption**, a locked screen
- **networked PC:** all of that, plus a **firewall**, per-user permissions (admin rights only for admins), central **user accounts** and **audit logs**
- **across the internet:** a **VPN** encrypting traffic to the gateway, HTTPS/TLS, **digital signatures** 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**

malware → **antivirus,
anti-spyware , patching**

insiders → **least-
privilege** + auditing

unauthorised access
→ **authentication**

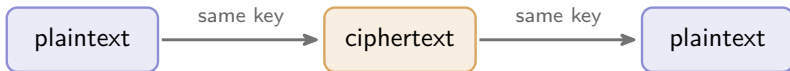
social engineering →
training + filtering

DDoS → **rate limiting**

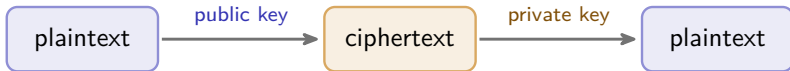


Encryption

symmetric



asymmetric



- **symmetric** : one shared key locks & unlocks (AES)
- **asymmetric** : **public key** locks, **private key** unlocks (RSA)

Authentication

Three factors, strongest combined – **two-factor authentication** . Authentication proves *who* you are; **authorisation** decides what you may then do.

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



“something you have”



Validation vs verification

validation

"is the data sensible?"

- range / type / format
- check digit

catches nonsense data

verification

"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 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

even parity:

0	1	1	0	1	1	0	0
---	---	---	---	---	---	---	---

 number of 1s = 4 (even)

parity bit 7 data bits

odd parity:

1	1	1	0	1	1	0	0
---	---	---	---	---	---	---	---

 number of 1s = 5 (odd)

parity check : count the 1s – a flipped bit changes the count, so the error shows up. A **parity block check** 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

parity bits

	1	2	3	4	5	6	7	8	
byte 1	1	0	1	1	0	1	0	0	
byte 2	0	1	1	0	0	1	1	1	odd!
byte 3	1	1	0	1	1	0	0	0	
byte 4	0	0	1	1	0	1	1	0	
parity byte	0	0	1	1	0	1	0	1	

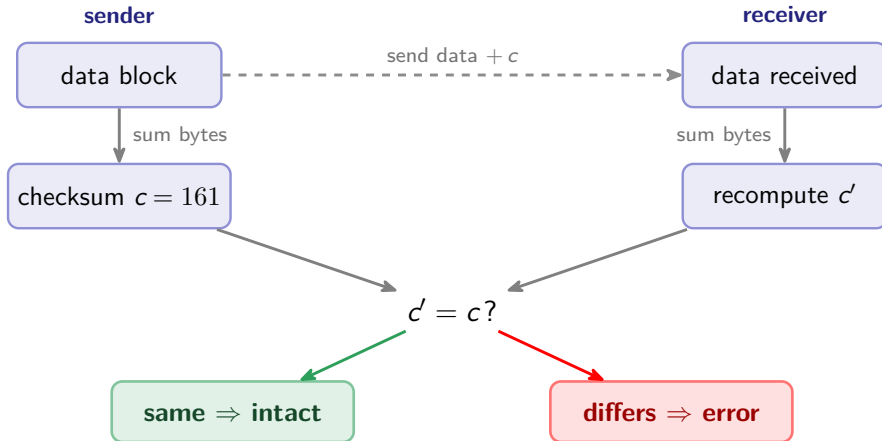
odd!

- 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

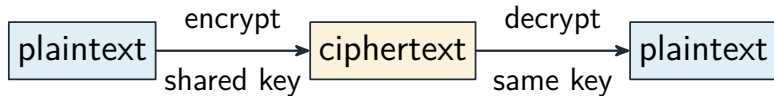
A **cyclic redundancy check** (CRC) is a stronger checksum using polynomial division.



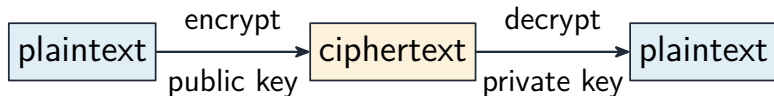
Protecting the data itself

Access control – 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



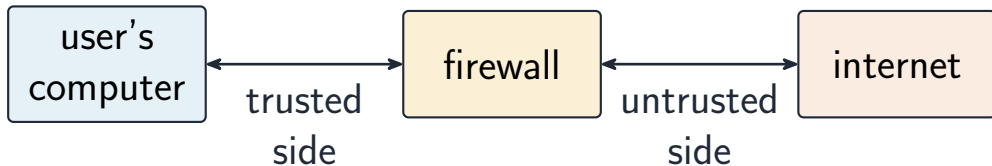
asymmetric



A networked PC

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

inspects & filters traffic
(allows or blocks each packet)



Threats come in three groups

malware

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

tricking people

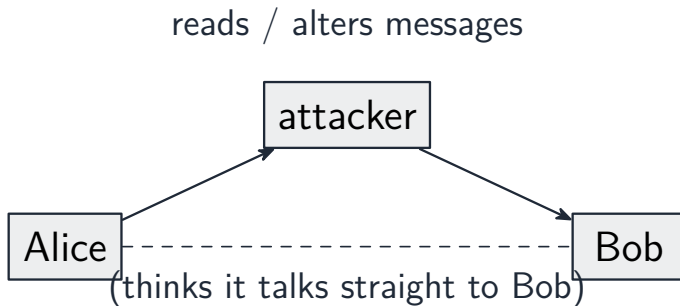
phishing emails and **pharming** redirects that send you to a convincing fake

attacking the network

hacking through weak passwords or software flaws; **denial of service** flooding a server; **eavesdropping** 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.

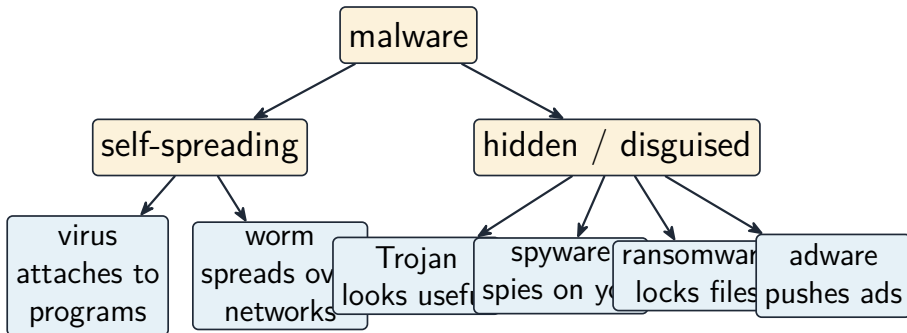
The man in the middle



The attacker relays both sides, so each believes it is talking straight to the other – what HTTPS and **digital signatures** 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:

- range / type / format
- length / presence
- check digit

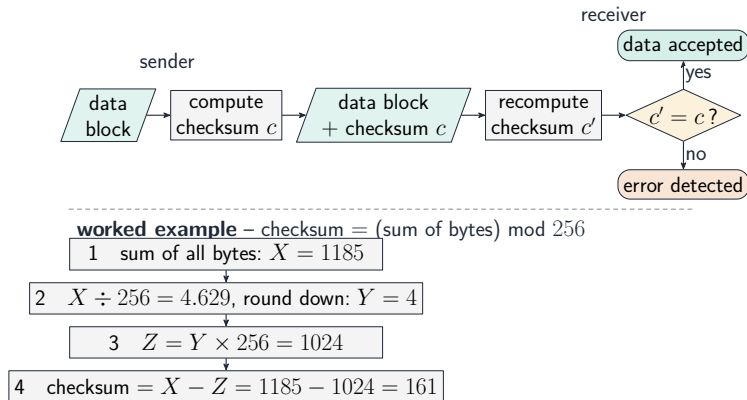
catches nonsense data

checks the data did not change:

- double entry
- parity check
- checksum / CRC

catches copying errors

Working out a checksum for a block of data



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.



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