A-LEVEL COMPUTER SCIENCE • EXERCISE SHEET

17.1 Encryption, Encryption Protocols and Digital Certificates

Name: _____ Class: _____ Date: _____ **Total: 17 marks**

Objective

Build the skills to answer exam questions on **encryption** 加密—symmetric and asymmetric keys, TLS, and digital certificates.

You must be able to:

- explain how **encryption** works and compare **symmetric** and **asymmetric**
- describe the **hybrid** approach and outline **TLS**
- explain a **digital certificate** and how it is checked

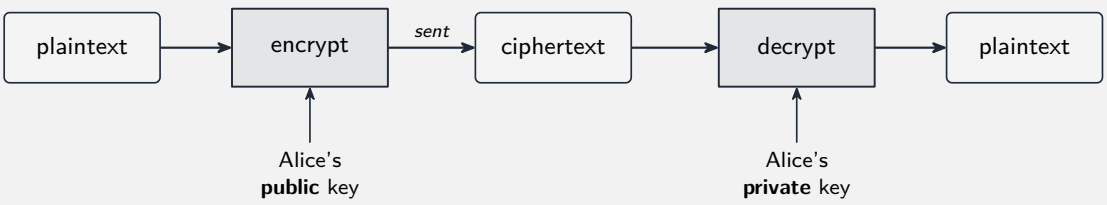
1 Worked examples

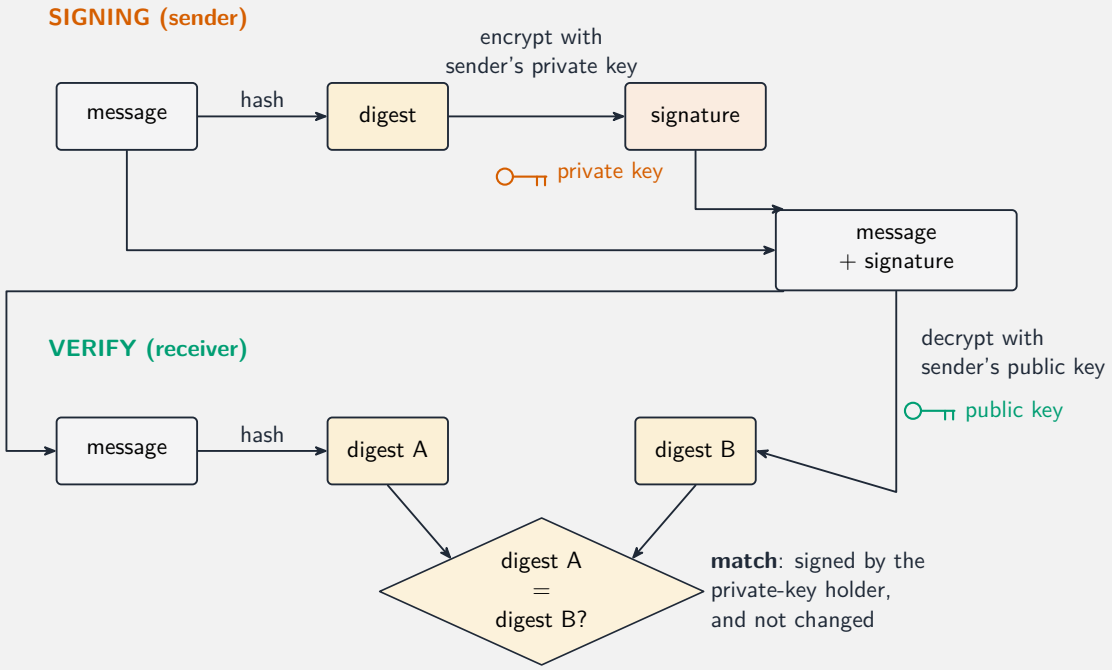
Study these first. Each one shows the method for a question type used later —follow the steps and you can do the Practice and Exam-style questions yourself.

Encryption basics

Encryption turns readable **plaintext** into unreadable **ciphertext** using a **key**; only the right key reverses it (**decryption**).

- **Symmetric** —the **same** key encrypts and decrypts. Fast (good for bulk data), but how do you **share the key** safely?
- **Asymmetric (public-key)** —a **pair** of keys. Encrypt with the recipient's **public** key; only their **private** key can decrypt:





A digital signature authenticates a message

Hybrid (how HTTPS really works)

Asymmetric is **slow**, so it is used only to exchange a fresh **session key**, then fast **symmetric** encryption carries the data. A **TLS** handshake: the server sends its **digital certificate** (with its public key); the client checks it; both agree a **session key**; all later traffic is symmetric.

Digital certificate

A **digital certificate** binds an **identity** to a **public key**, signed by a trusted **Certificate Authority (CA)**. The browser checks the **expiry**, that the **name matches**, and that it is **signed by a trusted CA**.

Digital signature

A **digital signature** 数字签名 proves a message came from the real sender (**authenticity**) and was not changed (**integrity**):

1. the sender **hashes** the message to a fixed-length **digest**;
2. they **encrypt the digest with their *private* key** —this is the signature, sent with the message;
3. the receiver decrypts the signature with the sender's **public key** to recover the digest, and **re-hashes** the received message;
4. if the two digests **match**, the message is genuine and unchanged; if not, it was altered.

2 Practice

Now apply the methods above.

2.1 Define **plaintext** and **ciphertext**. [2]

2.2 State the key difference between **symmetric** and **asymmetric** encryption. [1]

2.3 State the main **problem** with symmetric encryption. [1]

2.4 State what a **digital certificate** binds together. [1]

3 Exam-style questions

3.1 Bob wants to send Alice a secret message using asymmetric encryption. Which of Alice's keys does Bob use to **encrypt**, and which does Alice use to **decrypt**? Explain why this is secure. [3]

3.2 Explain why real systems (like HTTPS) use a **hybrid** approach combining asymmetric and symmetric encryption. [3]

3.3 State **two** things that **TLS** provides. [2]

3.4 State **one** check a browser makes on a **digital certificate**. [1]

3.5 Explain how a **digital signature** lets the receiver check that a message has **not been changed** during transmission. [3]

4 Go further

You are now ready for the real exam questions on this subtopic. Open the **17.1 Encryption and Digital Certificates** past-paper sheet in the Library, or try these in **Practice mode**:

- 9618/32 N25 —Q7 (symmetric and asymmetric encryption)
- 9618/33 J25 —Q7 (digital signatures)
- 9618/31 J25 —Q9 (digital certificates)
- 9618/11 J25 —Q7 (encryption and secure transmission)

Solutions

2.1 Plaintext —the original readable data; **ciphertext** —the scrambled, unreadable form after encryption.

2.2 Symmetric uses the **same** key for both; **asymmetric** uses a **pair** (public to encrypt, private to decrypt).

2.3 Key distribution —sharing the secret key safely with the other party.

2.4 An **identity** (domain/organisation) and its **public key** (signed by a CA).

3.1 Bob encrypts with **Alice's public key**; only **Alice's private key** can decrypt it; since Alice keeps her private key secret, an interceptor with only the public key and ciphertext cannot read the message.

3.2 Asymmetric is **secure for key exchange** but **slow**; symmetric is **fast** but needs a shared key; so asymmetric is used once to share a **session key**, then fast symmetric encryption carries the bulk data —getting both security and speed.

3.3 Any two: **encryption** of data in transit; **authentication** of the server (via certificate); **integrity** (detecting tampering).

3.4 Any one: the certificate is **in date**; the **subject name matches** the site's URL; it is **signed by a trusted CA** / chains to a trusted root.

3.5 The sender hashes the message and encrypts that hash with their **private key** to form the signature; the receiver decrypts the signature with the sender's **public key** to get the original hash, and **re-hashes** the received message; if the two hashes **match** the message is unchanged, and if they differ it was altered in transit.