Exercise walk-through

Encryption, Encryption Protocols and Digital Certificates ▪ 17.1

eXercise

You must be able to

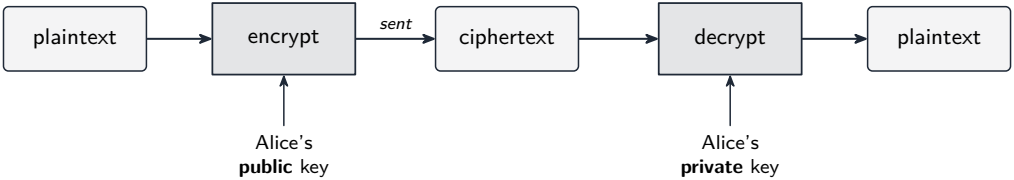
- explain how **encryption** works and compare **symmetric** and **asymmetric**
- describe the **hybrid** approach and outline **TLS**
- explain a **digital certificate** and how it is checked

Method — Encryption basics

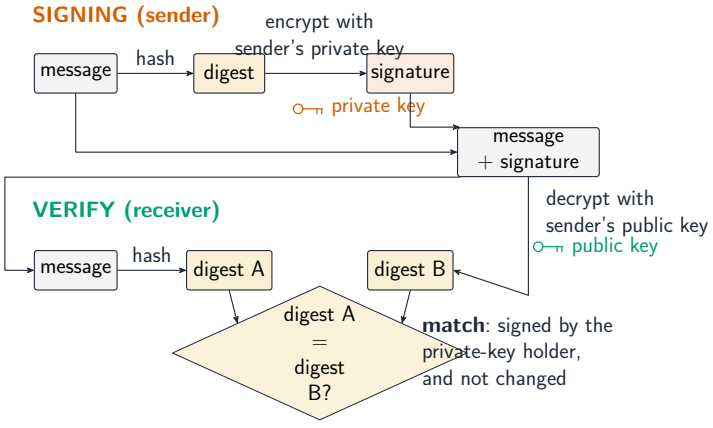
Encryption turns readable **plaintext** into unreadable **ciphertext** using a **key**; only the right key reverses it (**decryption**).

- **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:

Method — Encryption basics



Method — Encryption basics



A digital signature authenticates a message

Method — 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.

Method — 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**.

Method — 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.

Practice 2.1 ▪ 2 marks

Define **plaintext** and **ciphertext**.

Solution 2.1

Method: Encryption basics

Worked steps

Plaintext — the original readable data; **ciphertext** — the scrambled, unreadable form after encryption [B1] [B1].

Practice 2.2 ▪ 1 mark

State the key difference between **symmetric** and **asymmetric** encryption.

Solution 2.2

Worked steps

Symmetric uses the **same** key for both; **asymmetric** uses a **pair** (public to encrypt, private to decrypt) B1.

Practice 2.3 ▪ 1 mark

State the main **problem** with symmetric encryption.

Solution 2.3

Worked steps

Key distribution — sharing the secret key safely with the other party B1.

Practice 2.4 ▪ 1 mark

State what a **digital certificate** binds together.

Solution 2.4

Method: Digital certificate

Worked steps

An **identity** (domain/organisation) and its **public key** (signed by a CA) B1.

Exam-style 3.1 ▪ 3 marks

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.

Solution 3.1

Method: Encryption basics

Worked steps

Bob encrypts with **Alice's public key** B1; only **Alice's private key** can decrypt it B1; since Alice keeps her private key secret, an interceptor with only the public key and ciphertext cannot read the message B1.

Exam-style 3.2 ▪ 3 marks

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

Solution 3.2

Method: Hybrid (how HTTPS really works)

Worked steps

Asymmetric is **secure for key exchange** but **slow** B1; symmetric is **fast** but needs a shared key B1; so asymmetric is used once to share a **session key**, then fast symmetric encryption carries the bulk data — getting both security and speed B1.

Exam-style 3.3 ▪ 2 marks

State **two** things that **TLS** provides.

Solution 3.3

Worked steps

Any two: **encryption** of data in transit; **authentication** of the server (via certificate); **integrity** (detecting tampering) B1 B1.

Exam-style 3.4 ▪ 1 mark

State **one** check a browser makes on a **digital certificate**.

Solution 3.4

Method: Digital certificate

Worked steps

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 B1.

Exam-style 3.5 ▪ 3 marks

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

Solution 3.5

Method: Digital signature

Worked steps

The sender hashes the message and encrypts that hash with their **private key** to form the signature B1; the receiver decrypts the signature with the sender's **public key** to get the original hash, and **re-hashes** the received message B1; if the two hashes **match** the message is unchanged, and if they differ it was altered in transit B1.

Exam-style 3.6 ▪ 3 marks

An online shop uses **asymmetric** encryption and digital certificates to protect its customers.

(a) **Describe** how a pair of asymmetric keys is used so that a customer can send card details that only the shop can read.

(b) **Explain** why the shop must never publish its **private** key, and what it does publish instead.

(c) The shop obtains a **digital certificate** from a certificate authority. **Describe** what the certificate contains and how it proves the shop's identity.

(d) **Describe** the purpose of the **SSL/TLS** protocol, and **state** where it sits relative to the other protocols in use.

(e) **Outline** the steps of the TLS handshake that take place before any card details are sent.

(f) **Explain** why the session then switches to **symmetric** encryption.

Solution 3.6

Worked steps

(a) The shop publishes its **public** key and keeps its **private** key secret B1. The customer's browser encrypts the card details with the shop's **public** key B1. Only the matching **private** key can decrypt them, so only the shop can read the message B1.
(b) The private key is the only thing that can decrypt messages sent to the shop and the only thing that can create its signature B1; publishing it would let anyone read customers' data and impersonate the shop. The shop publishes the **public** key instead, inside its certificate B1.
(c) The certificate contains the shop's **public key**, its domain name and organisation details, the certificate authority's name, and the validity dates B1 B1. The whole lot is **signed** by the certificate authority using the CA's own private key B1. A browser that trusts that CA can verify the signature with the CA's public key, so it knows the

Solution 3.6

Worked steps

public key really belongs to that shop B1.
(d) SSL/TLS provides a **secure, encrypted channel** between the browser and the server B1, giving confidentiality, integrity and authentication of the server B1. It sits **between** the application layer protocol, such as HTTP, and the transport layer — which is why the secure version is called HTTPS B1.
(e) The browser sends a "hello" listing the TLS versions and cipher suites it supports B1. The server replies with its chosen cipher suite and its **digital certificate** B1. The browser verifies the certificate against a trusted CA B1. The two then agree a **session key**, which the browser sends encrypted with the server's public key B1.
(f) Symmetric encryption is far **faster** and less demanding on the processor for large amounts of data B1; asymmetric encryption is used only to exchange the session key safely, after which the bulk of the traffic uses that shared key B1.