AP CYBERSECURITY • EXERCISE SHEET

5.4 Asymmetric Cryptography

Name: _____ Class: _____ Date: _____ **Total: 46 marks**

KEY WORDS private key 私钥 • public key 公钥 • asymmetric encryption 非对称加密 • key pair 密钥对 • keyspace 密钥空间

Objective

Build the skills to answer exam questions on **asymmetric encryption** 非对称加密: choosing the right key, why **key length** 密钥长度 matters, and applying RSA and ECC.

You must be able to:

- explain why asymmetric encryption needs no prearranged shared secret
- decide, for any transfer, **whose** key and **which** key of the pair is used
- explain what happens when a private key is exposed
- calculate a **keyspace** 密钥空间 and the average number of guesses to break it
- state the cost of longer keys and why recommendations change over time
- name common asymmetric algorithms and the applications that use them

1 Worked examples

Study these first. Each one shows the method for a task used later.

■ The problem asymmetric encryption solves

With symmetric encryption, both parties need the **same** key - so the key itself has to be delivered securely before any secure communication can happen. Asymmetric encryption removes that: two people who have never met, and have arranged **no shared secret**, can communicate securely.

Each entity that will **receive** data first generates a **key pair** - two binary strings of equal length generated together. One is published; one is kept.

■ Whose key, and which one

This is the single most-tested decision in the topic, and the rule is short:

> **Encrypt with the receiver's PUBLIC key. Only the receiver's PRIVATE key can decrypt it.**

So work in this order:

1. **Who is receiving?** Not who is sending.
2. **Take that person's PUBLIC key** to encrypt.
3. The receiver decrypts with **their own PRIVATE key**, which never leaves them.

Example. Amir sends a file to Bhavna. Amir uses **Bhavna's public key**. Bhavna decrypts with **Bhavna's private key**. Amir's own key pair is not involved at all in this direction - and for Bhavna to reply, she encrypts with **Amir's public key**.

The common wrong answer is to encrypt with the sender's private key. That does not protect anything: the matching public key is published, so anyone could read it.

■ The private key is the whole security

The private key must be **stored securely**. If it is exposed, shared, stolen, corrupted or compromised, the key pair is no longer secure and must be replaced - and every message ever encrypted to that public key is readable by whoever holds the private key.

■ Key length and keyspace

- An **n-bit** binary key has a keyspace of 2^n - the number of possible keys.
- Guessing randomly, an adversary finds the key on average after $2^n \div 2$, i.e. 2^{n-1}, guesses.

Example. A 16-bit key has $2^{16} = 65\,536$ possible keys, so on average $2^{15} = 32\,768$ guesses. Add one bit and both numbers **double** - which is why key length is quoted in bits and why a small increase is a large gain.

Two qualifications the exam checks:

- **Longer keys cost time** - encryption and decryption both take longer, so length is a trade-off, not a free win.
- **Recommendations change**, because processing power and efficiency keep improving, so software guesses keys faster each year. A length considered safe a decade ago may not be now.
- **Compare like with like.** A key-length comparison is only valid **within the same algorithm**. A 256-bit ECC key and a 256-bit RSA key do not offer comparable security, and saying "longer is stronger" across two different algorithms is wrong.

■ The algorithms

Common asymmetric algorithms: **RSA** and **elliptic curve cryptography (ECC)**. They are used in digital signatures, secure web traffic and key exchange. As with symmetric encryption, the operations can be run from the **command line**, from specialised software, or through web-based tools - and in a CLI a user can generate a key pair and encrypt or decrypt files as needed.

2 Practice

2.1 Explain the advantage asymmetric encryption has over symmetric encryption when two people have never communicated before. [2]

2.2 Priya wants to send a confidential file to Wen.

(a) Whose key is used to encrypt it, and which key of the pair? [2]

(b) Whose key decrypts it, and which key of the pair? [2]

(c) Explain why encrypting with Priya's private key would not keep the file confidential. [2]

2.3 State **three** events that would require a key pair to be replaced. [3]

2.4 Calculate the keyspace and the average number of guesses for each key.

(a) An 8-bit key. [2]

(b) A 20-bit key. [2]

2.5 Explain what happens to the keyspace when one bit is added to a key. [2]

2.6 State **one** disadvantage of using a longer key. [1]

2.7 Explain why key-length recommendations are revised over time. [2]

2.8 A student claims "a 256-bit ECC key is exactly as strong as a 256-bit RSA key, because both are 256 bits". Explain why this reasoning is wrong. [2]

2.9 Name **two** common asymmetric algorithms and **two** applications that use them. [4]

3 Exam-style questions

3.1 To send data securely to a receiver, the sender encrypts with [1]

- **A** the sender's private key
- **B** the sender's public key
- **C** the receiver's private key
- **D** the receiver's public key

3.2 A 12-bit key has a keyspace of [1]

- **A** 12
- **B** 24
- **C** 144
- **D** 4096

3.3 On average, how many guesses are needed to find an n-bit key at random? [1]

- **A** $n \div 2$
- **B** 2^n
- **C** 2^{n-1}
- **D** $2^n - 1$

3.4 Which statement about key length is correct? [1]

- **A** longer keys are faster to encrypt and decrypt
- **B** key-length comparisons are only valid within the same algorithm
- **C** a longer key always removes the need for secure key storage
- **D** key-length recommendations are fixed once published

3.5 A small law firm exchanges case files with three external barristers. Each barrister generates an RSA key pair. The firm's practice manager keeps a folder of every barrister's public key, and also keeps a backup copy of the firm's own private key on a shared network drive "in case the office computer fails". Files are currently encrypted with 1024-bit RSA, a length chosen when the system was installed in 2012.

(a) Describe the correct procedure for the firm to send a confidential case file to one named barrister, and for that barrister to reply. [4]

(b) Explain the risk created by the backup copy of the firm's private key, and state what must happen if that drive is compromised. [3]

(c) The firm's 1024-bit key length was considered adequate in 2012. Explain, referring to processing power, why it may not be adequate now. [3]

(d) A consultant proposes moving to 256-bit ECC and argues this must be weaker because "1024 is bigger than 256". Evaluate this argument. [3]

(e) The practice manager asks why the firm should not simply use one shared symmetric key with all three barristers instead. Give **two** reasons, and state one situation in which symmetric encryption would still be preferable. [3]

Solutions

2.1 asymmetric encryption allows secure communication **without prearranging a shared secret key**; with symmetric encryption the same key must reach both parties first, which is itself a problem when there is no secure channel yet.

2.2 (a) Wen's key, the **public** key. (b) Wen's key, the **private** key. (c) the matching public key is published, so anyone could obtain it and decrypt the file; encrypting with a private key proves origin, but provides no confidentiality.

2.3 any three of: the private key is exposed, shared, stolen, corrupted, or compromised.

2.4 (a) $2^8 = 256$; average $2^7 = 128$ guesses. (b) $2^{20} = 1\,048\,576$; average $2^{19} = 524\,288$ guesses.

2.5 the keyspace **doubles**, because $2^{n+1} = 2 \times 2^n$ - so a single extra bit doubles the work an adversary faces.

2.6 encryption and decryption both take more time.

2.7 computational processing power and efficiency keep improving, so software can guess keys faster than before and a length that was safe when a recommendation was written becomes breakable within a practical time.

2.8 key-length comparison is only valid when comparing keys for the **same** cryptographic algorithm; ECC and RSA achieve their security by different mathematics, so bit counts across the two are not comparable at all - in practice ECC reaches equivalent security at much shorter lengths.

2.9 RSA; elliptic curve cryptography / ECC; any two applications, e.g. digital signatures, secure web traffic, secure key exchange, encrypted email.

3.1 D. Only the receiver's private key can undo it, and only the receiver holds that.

3.2 D. $2^{12} = 4096$.

3.3 C. $2^n \div 2 = 2^{n-1}$.

3.4 B. Longer keys are slower, do not remove the need for secure storage, and recommendations change as processing power improves.

3.5 (a) the firm encrypts the file with **that barrister's public key**, taken from the manager's folder; only that barrister's private key can decrypt it, so only the intended recipient can read it. To reply, the barrister encrypts with the **firm's public key** and the firm decrypts with its own private key.

(b) a private key must be stored securely, and a copy on a shared network drive is reachable by everyone with access to that drive - so an adversary who reaches it can decrypt every file ever sent to the firm, including past traffic they may have captured. If that drive is compromised the key pair is no longer secure: it must be replaced and the new public key distributed to all three barristers.

(c) processing power and efficiency have improved continuously since 2012, so the number of keys that can be tried in a given time has risen by orders of magnitude; 1024-bit RSA is therefore now regarded as too short, and the firm should follow current recommendations rather than the figure chosen when the system was installed.

(d) the argument is **wrong**. A key-length comparison is only valid **within one algorithm**, and RSA and ECC are different algorithms; ECC achieves equivalent security at far shorter key lengths, so 256-bit ECC is considered comparable to - or stronger than - 1024-bit RSA, and it is also faster.

(e) any two of: one shared key means any single barrister's compromise exposes **all** case files, including those sent to the others; the key must first be delivered securely to three separate parties, which asymmetric encryption avoids entirely; revoking one barrister's access requires re-keying everyone. Symmetric encryption is still preferable for **bulk data** - encrypting a large archive, or the firm's own files at rest - because it is much faster.