University of London

Assessment Coversheet

Complete this coversheet and read the instructions below carefully.

Candidate Number: LX0549
Refer to your Admission Notice

Degree Title:
BSc

Course/Module Title:
Computer Security

Course/Module Code:
CM2025

Enter the numbers, and sub-sections, of the questions in the order in which you have attempted them:

Question 2 a b c d e f g
Question 4 a b c d e f g

Date: Sep 15th, 2021

Instructions to Candidates

1. Complete this coversheet and begin typing your answers on the page below, or, submit the coversheet with your handwritten answers (where handwritten answers are permitted or required as part of your online timed assessment).
2. Clearly state the question number, and any sub-sections, at the beginning of each answer and also note them in the space provided above.
3. For typed answers, use a plain font such as Arial or Calibri and font size 11 or larger.
4. Where permission has been given in advance, handwritten answers (including diagrams or mathematical formulae) must be done on light coloured paper using blue or black ink.
5. Reference your diagrams in your typed answers. Label diagrams clearly.

The Examiners will attach great importance to legibility, accuracy and clarity of expression.

Question 2

(a) What is the difference between symmetric and asymmetric cryptography? Give an example of each.

- Asymmetric encryption uses two keys, a set of public and private keys, to encrypt and decrypt data respectively. An example of asymmetric encryption is RSA.
- Symmetric encryption users only one shared key to perform both encryption and decryption. An example of symmetric encryption is the Data Encryption Standard (DES).
- The main difference between the two is that symmetric encryption faces two problems that are solved by asymmetric encryption:
    - Both parties in communication need to share a key, meaning they need to make sure the key is transferred to the other party safely, without someone listening in.
    - It is not possible to prove the authenticity of the sender securely, as both parties use the same key

---

(b) Alice and Bob both have private keys and public keys. How can Alice send Bob a message that only he can read? Explain your answer

Alice can send Bob a message using public key encryption. The process is as follows:

- Alice creates a message
- Alice encrypts the message using Bob's public key and an encryption algorithm such as RSA
- Alice sends Bob the encrypted message
- Bob decrypts the message with his private key
- Bob receives the message

Because only Alice and Bob have their respective private keys, only Bob can read messages that were encrypted using his public key.

(c) Alice and Bob both have private keys and public keys. How can Alice send Bob a message in a way that assures Bob that the message came from Alice. Explain your answer.

Alice can send an authentication message along with her main message as follows.

- Alice creates a hash of her original message, e.g. using MD5
- Alice encrypts the hash using her private key – this is the authentication message
- Alice encrypts the main message and the encrypted hash together using Bob's public key
- Alice sends the message to Bob
- Bob decrypts the message using his private key
- Bob decrypts the authentication message using Alice's public key to verify that the message did indeed come from Alice

---

(d) We have the following encoding system: To encode a message you square it. How do you decode a message? What are the merits and drawbacks of this system.

To decode this message, we can use the inverse of the encryption function, which is to take the square root.

This is essentially a two-way hashing method, or the use of symmetric encryption, as the "key" is known and shared by participants in the communication.

The merits of this system are that it is efficient and computationally cheap. Also, guessing the key is hard because there is no way to know if the right key (square rooting) was guessed correctly.

The drawbacks include that a third party might listen in and identify the method used. It also requires you to first convert the message into a format suitable for squaring, i.e. it must be in a numeric format. Finally, you will run into collisions when encrypting as both negative and positive numbers produce identical squares.

(e) We make the code in part d a bit more robust by the following scheme: Alice has a private key, which is a number, x. She encodes a message by raising it to the x power. She does not tell anybody what x is but she publishes a decoding function. Explain ways that this is better than the code in d but say why it is still inadequate.

This method introduces asymmetric encryption, which makes it stronger by virtue of not having to share the encoding key (Alice's private key). It is also stronger because the potential ciphertext space has been enlarged since we are now using more than just powers of 2, making it harder to guess the key given a set of plaintexts and ciphertexts.

This is still limited to numeric squares though, and can be broken using a brute force attack since guessing squares is not hard.

(f) You are going to encrypt a message uing a playfair code. Make the playfair tableau using compsecexam as the key

C	**O**	**M**	**P**	**S**
E	**X**	**A**	**B**	**D**
F	**G**	**H**	**I**	**K**
L	**N**	**Q**	**R**	**T**
U	**V**	**W**	**Y**	**Z**

(g) Encode the phrase: This is my message using the playfair tableau from the previous answer.

The message can be broken down into two-letter bigrams:

TH IS IS MY ME SS AG E

Using the playfair method, we span rectangles on the cipher grid and get the resulting ciphertext.

QK KP KP PW CA EE XH XA

Question 4

(a) “Secure software development is not all about coding”. Justify this statement with TWO other aspects of secure software development that do not involve coding.

Secure software development is about applying security principles along the entire software development lifecycle. Some methodologies include the Microsoft SDL, which in addition to the coding defines the following two aspects (among others) as part of secure software development:

- Defining and using cryptography standards to ensure all data is protected from unintended disclosure.
- Using approved tools from a published list with associated security checks, striving to use the latest versions of tools

---

(b) The ticket machine in the station will run a Linux operating system. Describe TWO methods you would use to analyse the default security of the operating system.

The first method would be to run a hardening script such as **Harbian**-Audit. This will audit the Linux system for basic security configurations and propose changes based on a hardened version of the Debian distribution.

The second method would be to use **Lynis**, another command line utility that can scan the system for security issues and produces a report.

(c) The sales manager has explained to the train company that your system will use “not one but three totally different types of firewall to maximise the secure experience of customers”. Is this a reasonable statement? Explain your answer.

In general, it is not reasonable to use different firewall types that serve different purposes. For example, the train company might use a stateless firewall, which checks all the traffic coming through. A second firewall with a stateful firewall would be redundant as this only checks traffic at the initial connection packet.

Proxy firewalls could be used to route all traffic through one point (the proxy) but this seems overkill if another firewall is anyway checking all the traffic coming through.

(d) The sales manager has been reading more internet articles about security and has promised that the ticket selling system will use “un-hackable containerisation” on the server side. Describe THREE ways in which containerised systems are vulnerable.

A containerized solution can be compromised in the following ways:

- The **container** can be compromised, and as a result all hosts could be compromised as well from within the container.
- The **host** might be compromised, so all containers on that host can be attacked.
- **Containers** might be communicating with each other, in which case a compromised container can be used to gain access to other adjacent containers.

(e) You decide to use the python language. Name a tool that you can use in Python to audit third-party libraries

A popular tool to audit dependencies in Python is called Safety, and it scans dependencies in a project for known vulnerabilities. It can be installed with the command “pip install safety” and executed with “safety check”.

(f) Does the auditing tool you mentioned carry out passive or dynamic analysis? Justify your answer.

It conducts static / passive analysis because it is not executed at runtime but rather scans the list of dependencies called in the Python code and then checks this against an existing list of issues.

(g) The sales manager has informed the client for the project that you will be using blockchain technology to ensure that the person buying the ticket is the only person that can use that ticket. Explain in some detail how you would go about using blockchain technology to achieve this. Consider the following problems: how would you identify the person, how would you represent a ticket on the blockchain? how would you connect that identity to a given ticket? How might a ticket inspector establish if the person on the train is the person who owns the ticket?

To represent a ticket system on a blockchain, we can tokenize all tickets as **non-fungible tokens** (NFTs).

- Each ticket would exist as an NFT with all related data stored as part of the token.
- Because we can use the ERC-721 token standard, we can capture all information about the owner, the ticket details (route, price, validity, etc.) on the token itself.
- When a customer purchases a ticket, an NFT is created and transferred to them to a digital wallet, e.g. an app provided by the train company. The token could be stored locally or in the cloud with access to it from within the app.
- A smart contract can be used to execute the logic of the ticket, e.g. it can make sure that the ticket is invalidated after a specific date has passed, or if the train conductor validates the ticket, and so on.
- The identity is managed on the token as part of the uniqueID
- These tickets are then represented on the blockchain in blocks of transactions when an NFT (ticket) was created and assigned ownership
- Customers would prove ownership of the NFT by providing a private key that can verify the proof-of-ownership on the blockchain. This might be a password, or handled transparently by an app.