Contents

1. Overview of the Assignment

2. Programming Task Requirements

3. Report Requirements

4. Marking Criteria

5. Deliverables

6. Plagiarism

1. Overview of the Assignment

This assignment requires you to complete 4 programming tasks and to write a brief

report on your findings. The programming tasks carry 80% of the marks whilst the

report carries the remaining 20 marks.

Module learning outcomes are described in the module specification and the ones

that are relevant to this assignment are:

• Understand and manipulate the mathematical and theoretical methods on

which designs are based.

• Implement algorithms and protocols for particular coding schemes,

recognising the need for efficiency in terms of delay, throughput, jitter,

computing resources and quality of service.

• Use cryptographic and coding classes available in modern programming

language environments, such as Java Security, to implement secure

applications

• Evaluate the performance of various coding schemes under application load

and change configuration parameters to optimise them

• Explain the strategies that need to be employed whilst attempting to break a

cipher.

The assignment is worth 75% of the overall mark for the module.

The assignment is described in more detail in section 2.

This is an individual assignment.

Working on this assignment will help you to consolidate your understanding of the

material. You will also improve your general software development and problem

solving skills. If you have questions about this assignment, please post them to the

discussion board on Blackboard.

Cryptography UFCFT4-15-3

2. Programming Task Requirements

Task 1. Submission Date: 22:00 Thursday, 2

nd of November 2023

2.1. Verifying credit card numbers. 5 marks

Your code should accept a 16-digit string and determine whether or not it is a valid

credit card number.

Task 2. Submission Date: 22:00 Thursday, 23rd of November 2023

2.2. BCH Generating and Correcting. 25 marks

For the BCH(10,6) generator, unusable numbers must be identified. For the

BCH(10,6) error corrector, single and double errors must be corrected, and morethan-two-error cases must be detected

Task 3. Submission Date: 22:00 Thursday, 21st of December 2023

2.3. Brute Force Password Cracking. 10 marks

The brute force program should be able to break the following two types of

passwords (see Practical5 worksheet for the testing data):

• Set A - Standard: any password which contains a maximum of six lowercase letters and/or numbers;

• Set B - Special: any password which is a valid BCH(10,6) code.

Task 4. Submission Date: 22:00 Thursday, 21st of December 2023

2.4. A Text Encryption APP Using Stream Cipher and Steganography

30 marks

Write an App which provides the following two functions:

1. Encryption – Taking two input messages:

• ‘message 1’, which is not going to be encrypted, and

• ‘message 2’, which is a secret message needing encryption.

The APP should encrypt ‘message 2’ using a stream cipher, then produce

an output which should contains the original ‘message 1’ and the

encrypted ‘message 2’ hidden within it.

2. Decryption – The APP reads a ‘ciphertext’ (a message output by the

encryption algorithm), and recovers both messages from the ‘ciphertext’.

In summary, this App is a combination of cryptography and steganography. It hides

an encrypted message within an ordinary message. You are not allowed to use any

Cryptography UFCFT4-15-3

existing crypto libraries or other open sources for this task apart from a pseudo

random number generator for the stream cipher. To simplify the task, it is assumed

that both sender and receiver have already obtained a shared key for their stream

cipher. More details on this task can be found in supporting document