THE UNIVERSITY OF HUDDERSFIELD

School of Computing and Engineering

ASSIGNMENT SPECIFICATION

Module Code	CIS2201
Module Title	Cyber Security
Course Title/s	MComp/BSc(Hons) Computing MSci/BSc(Hons) Computing Science (including with variants) MEng/BSc(Hons) Software Engineering BSc(Hons) Web Programming (including with variants) BSc(Hons) Web Design

Assignment Details		
Title	Assignment 1	
Weighting	50%	
Mode of working for assessment task.	☒ Individual There should be no collusion or collaboration whilst working on and subsequently submitting this assignment.	
Module Leader	Saad Khan	Contact details: saad.khan@hud.ac.uk Office: SJ3/18
Module Tutor/s	Saad Khan	

Hand-out date	28/02/2022
How to submit your work	Brightspace TurnitIn electronic submission
Submission date/s and times	08/04/2022

Expected amount of independent time you should allocate to complete this assessment	20 Hours
Submission type and format	Formal report
Date by which your grade and feedback will be returned	3 weeks after the deadline

Your responsibility	It is your responsibility to read and understand the University regulations regarding assessment. http://www.hud.ac.uk/registry/regulationsandpolicies/studentregs. Please pay special attention to the assessment regulations (section 4) on Academic Misconduct In brief: ensure that you; 1. DO NOT use the work of another student - this includes students from previous years and other institutions, as well as current students on the module. 2. DO NOT make your work available or leave insecure, for other students to view or use. 3. Any examples provided by the module tutor should be appropriately referenced, as should examples from external sources. Further guidance can be found in the SCEN Academic Skills Resource and UoH Academic Integrity Resource module in Brightspace. If you experience difficulties with this assessment or with time management, please speak to the module tutor/s, your Personal Academic Tutor, or the School's Guidance Team. (sce.guidance@hud.ac.uk).
Requesting an Extension	You are reminded to 'back-up' your work as extensions will not be given for lost work, which includes work lost due to hardware and software failure/s. Extension requests will only be approved if you can demonstrate genuine, unexpected circumstances along with independent supporting evidence (e.g medical certificate) that may prevent you submitting an assessment on time. Submit an extension request via Student Portal within 2 working days of the due date. Extension requests, up to a maximum of 10 working days, but typically 1-5 working days, will be considered provided that there is appropriate evidence which clearly indicates reasons for the request. You will have 5 working days after submitting a request to provide the evidence. Failure to submit evidence will result in the request being rejected and your work being marked as a late submission. If you are unable to submit work within the maximum extension period of 10 days, contact the School's Guidance team (sce.guidance@hud.ac.uk), as you may need to submit a claim for Extenuating Circumstances (ECs).

Extenuating Circumstances (ECs)	An EC claim is appropriate in exceptional circumstances, when an extension is not sufficient due to the nature of the request, or it concerns an examination or In-Class Test (ICT). You can access the EC claim form via MyHud or Registry website; https://www.hud.ac.uk/registry/extenuatingcircumstancesfaqs where you can also find out more about the process. You will need to submit independent, verifiable evidence for your claim to be considered. Once your EC claim has been reviewed you will get an EC outcome email from Registry. If you are unsure what it means or what you need to do next, please speak to the Student Support Office – SJ1/01 An approved EC will extend the submission date to the next assessment period (e.g July resit period).
Late Submission (No ECs approved)	Late submission, up to 5 working days, of the assessment submission deadline, will result in your grade being capped to a maximum of a pass mark. Submission after this period, without an approved extension, will result in a 0% grade for this assessment component.
Tutor Referral available	☐ Yes ☒ No

Resources	Please note: you can access free office software and you have 1 Tb of free storage space available on Microsoft's OneDrive system. https://students.hud.ac.uk/it/unimail/office365/

Enigma Simulator

1. Assignment Aims

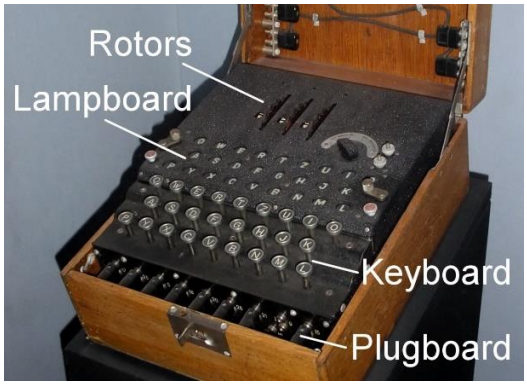
In this assessment you are required to develop a working Enigma machine. You are free to use any programming language. The developed software and executable will be used for marking purposes. A demonstration session of 10 minutes may be needed if the software cannot easily be executed on the examiner's PC.

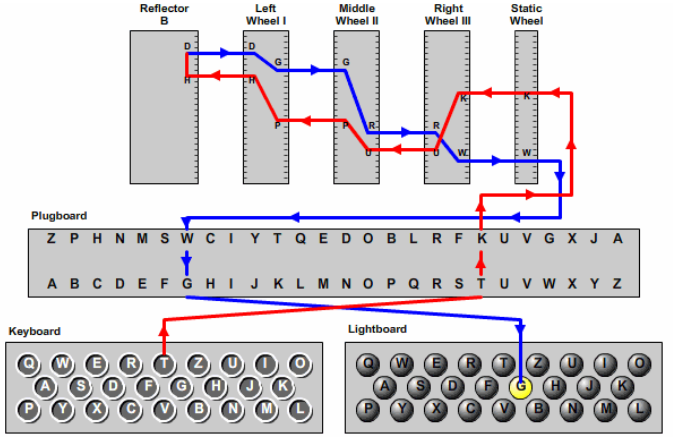
2. Learning Outcomes:

- Understand core concepts of implementing cryptography in software.
- Design and develop trustworthy software with suitable testing.
- Discuss techniques and challenges involved during the development process.

3. Assessment Brief

The Enigma machine was a revolutionary device capable of encrypting and decrypting messages for transmission over open communication channels. The Enigma machine implements a Symmetric key algorithm where the key is encoded by positions on the plug-board and rotor wheels.





The encryption algorithm was designed and built through the use of clever circuit design and was not implemented as a software algorithm, unlike most modern techniques.

This assignment requires you to develop a software version of an Enigma machine, both in terms of implementing the fundamental algorithm as well as a graphical user interface for ease of use.

It is your choice as to what software architecture you use to implement your Enigma simulator; however, your system must have the following high-level functionality:

- Capability to encrypt and decrypt messages constructed of alphabet characters (a-z) and to a minimum length of 200 characters.
- Allow for the Enigma machine to be configured (selecting start-rotor positions and plug-board).
- Provide a graphical user interface for interacting with the Enigma machine.

Note that there are many example Enigma software projects available on the internet. Although you are welcome to review other implementations to seek inspiration, you must design and implement your own system and reference any resources in your report. You are required to:

- Provide a thorough design of your software, which at minimum will consist of a Class diagram, an Activity diagram for the encryption process, and a Sequence diagram showing how individual aspects of the code work together.
- Submit your code for analysis by a source code plagiarism checker, capable of identifying code available in the public domain as well as identifying inconsistencies in programming style.
- Provide comprehensive notes in your code to demonstrate understanding of how you have implemented the key algorithm.

You are required to submit:

- A report detailing the design, development, testing and reflective analysis of your software; and
- The source code and a compiled Executable.

4. Marking Scheme

Criteria (Report)	Marks
System design UML Models and interface designs (wireframes, etc.) **Must have:** 1 class, 1 activity/sequence diagram, and mock-up design. **Measured:** Correctness of UML and quality of interface designs	15
Implementation notes Log of development challenges and fixes **Must have:** At least three detailed entries **Measured:** Completeness and quality	15
Testing Both functional and usability testing **Must have:** Unit test functionality testing table **Measured:** Completeness of testing	10
Reflective analysis A personal reflection detailing the success of the project and what you would do different next time **Must have:** A reflective evaluation of 300-500 words in length **Measured:** Level of detail and reflection	10
Total	50

Criteria (Source code and Executable)	Marks
Configuration of Enigma Configuration of the virtual plug-board and rotor wheels **Must have:** Ability to accurately configure a working Enigma software machine **Measured:** Efficiency and Accuracy	10
Encrypt and Decrypt The user can perform encryption and decryption characters (a-z) **Must have:** A mechanism to encrypt and decrypt 200 characters at minimum **Measured:** Accuracy and Efficiency	20
Presentation of the system The system controls are presented in an easy-to-use and intuitive manner **Must have:** A GUI to interact with the machine **Measured:** Quality and ease of use	10
Additional functionality Functionality that was not specified. Optional for a higher mark **Measured:** Usefulness and impact on the software system	10
Total	50

5. Grading Rubric

	Marks Available							
Criterion	0 to 29%	30-39%	40-49%	50 - 59%	60 - 69%	70 - 79%	80-89%	≥ 90%
System Design	Severely flawed, or non-functional system design	Minimal, or mostly inaccurate system design	Incomplete system design, falling short of the requirement	Acceptable system design but has significant room for improvement	Good system design, meeting assignment criteria with some room for improvement	A complete system design, demonstrating some critical understanding, meeting assignment criteria to a high standard	Excellent, complete system design, demonstrating critical understanding, meeting assignment criteria to a very high standard	A high degree of originality and complete system design, demonstrating critical understanding, meeting assignment criteria to a professional standard, with very little room for improvement
Implementation notes	Severely flawed notes, or non-submission	Minimal, or incomplete notes, falling short of the requirement	Presents notes but leaves significant room for improvement	A coherent set of notes but can be improved by providing more details	A complete and coherent set of notes. Communicated well for the assignment brief	A complete and coherent set of notes, clear and appropriate for the assessment brief	Excellent, complete, clear, and coherent set of notes. Shows critical understanding of the implementation and challenges involved	A high degree of originality. Excellent, polished, complete, clear, and coherent set of notes. Shows understanding of the implementation and challenges involved to a professional standard
Testing	Severely flawed testing, or non-submission	Minimal testing, falling short of the requirement	Adequate testing with significant room for improvement	Good attempt at testing, meeting criteria with good room for improvement	Very good attempt at testing, meeting criteria fairly well but can be improved	Excellent testing, meeting all criteria to a high standard	Outstanding testing, meeting all criteria to a professional standard	A high degree of originality, complete and outstanding testing, meeting all criteria to a professional standard

Reflective analysis	Little to no reflection, or non-submission	Disorganised, incoherent, and very little reflection	Shows some attempt to reflect but leaves significant room for improvement	Clear reflective analysis, with good room for improvement	Clear reflective analysis, considering weak points and suggestions for further development	Good, clear reflective analysis, considering weak points and suggestions for further development	Strong reflective analysis, considering weak points and suggestions for further development	Very detailed, strong reflective discussion of the product with clear ideas and plans for further development
Configuration of Enigma	Severely flawed configuration, or non-submission	Very little to practically no configuration	Poor, flawed configuration, falling short of the requirement	Incomplete or inflexible configuration, meeting criteria with good room for improvement.	Complete, good attempt at configuring the system, meeting criteria well.	Complete and sound configuration, with reasonable flexibility, meeting criteria to a high standard.	Excellent, complete and sound configuration, with full flexibility, meeting all criteria to a very high standard.	Excellent, complete, and sound configuration, with full flexibility and additional controls, meeting all criteria to a professional standard.
Encrypt and Decrypt	Non-submission, or non-functional	Very little to practically no required functionality	Poor, flawed functionally, falling short of the requirement	Incomplete or partial functionality implemented, leaving significant room for improvement	Complete, good attempt at implementing the functionality, meeting criteria well	Complete and sound implementation of functionality, fully operational, meeting criteria to a high standard	Complete and sound implementation of functionality, fully operational, meeting criteria to a very high standard	Excellent and sound approach for implementation, fully functional, meeting all criteria to a professional standard
Presentation of the system	Severely flawed, non-functional GUI	Presentation seriously impedes comprehension	Disorganised and incoherent GUI	Shows some attempt to organise the developed in a logical manner	Good attempt at organisation and coherence of GUI. Easy to follow and understand	Good, organised, coherent and constructed to a very high standard. Easy to follow and understand	Outstanding approach, coherent, and well-organised GUI. Easy to follow and understand	Polished, imaginative approach, coherent, well-organised, and signposted GUI.
Additional functionality	Severely flawed, does not function, or non-submission	Minimal implementation, falling short of the requirement	Adequate implementation with significant room for improvement	Good attempt at implementation, meeting criteria well with room for improvement	Very good attempt at implementation, meeting criteria	Excellent implementation, meeting all criteria to a high standard	Outstanding implementation, meeting all criteria to a professional standard	A high degree of originality