MODULE SPECIFICATION

Part 1: Information			
Module Title	Cryptography		
Module Code	UFCFT4-15-3	Level	Level 6
For implementation from	2018-19		
UWE Credit Rating	15	ECTS Credit Rating	7.5
Faculty	Faculty of Environment & Technology	Field	Computer Science and Creative Technologies
Department	FET Dept of Computer Sci & Creative Tech		
Contributes towards	Information Technology [Sep][FT][Frenchay][1yr] BSc (Hons) 2018-19		
Module type:	Standard		
Pre-requisites	Introduction to OO Systems Development 2018-19, Programming in C 2018-19		
Excluded Combinations	None		
Co- requisites	None		
Module Entry requirements	None		

Part 2: Description

Overview: Pre-requisites: students must take one out of UFCFC3-30-1 Introduction to OO Systems Development or UFCFF6-30-1 Programming in C

Educational Aims: See Learning Outcomes.

Outline Syllabus: The syllabus includes:

Review of background mathematics: modular arithmetic, gcd algorithms, factorization algorithms, functions, pseudo random generators;

Historical ciphers: substitution, vignere & permutation ciphers, rotor machines and Enigma;

Perfect secrecy, semantic security;

Stream ciphers, block ciphers, modes of operation, symmetric and asymmetric encryption, RSA.

cryptographic hash functions, data integrity and digital signatures, authenticated encryption, authenticated key exchange;

Error correcting principles. Some important codes: Hamming codes, Reed-Solomon, BCH, and Turbo codes;

Basic compression techniques: Huffman coding, LZW Compression.

Teaching and Learning Methods: Students will learn through a combination of lectures, tutorials and practical activities in a computer laboratory with approximately one third of the contact time being devoted to each of these. Lectures will explore and illuminate the theoretical material. In the tutorials, students will have the opportunity to discuss the theory and tease out its implications for the practical work. In the practical activities, students will work in the computer labs and construct encryption and related software.

Students will also be expected to learn independently and carry out reading and directed study beyond that available within taught classes, as suggested in the table above.

Students will be given weekly practical tasks and problem sets. These weekly practical tasks are programming tasks which will be integrated into the assessed coursework. The weekly problem sets are designed to consolidate the learning and help student to check if they have fully understood the contents. These weekly problems are not formally assessed.

Contact Hours:

Activity:
Contact time: 36 hours
Assimilation and development of knowledge: 74 hours
Exam preparation: 20 hours
Coursework preparation: 20 hours
Total study time: 150 hours

Part 3: Assessment

The module is assessed by a 3 hour examination at the end of the teaching and also by coursework. These components of assessment are equally weighted. The exam assesses the students' understanding of the theoretical aspects of the module whereas the coursework is an opportunity for students to demonstrate their grasp of the application of these concepts to a problem in the area of cryptography.

The assessed practical work will involve at least one activity from the following list:

The construction of :
Encryption & decryption,
Error coding, and,
Compression software.

Benchmarking, performance analysis and optimisation of:
Encryption/decryption algorithms,
Error correction codes, and,
Compression coding schemes.
Attacking software to attempt the breaking of the ciphers given encrypted texts. An interesting measure of the success of both encryption and attacking software.

First Sit Components	Final Assessment	Element weighting	Description
Written Assignment - Component B		75 %	Practical coursework involving, for example, the production of encryption or benchmarking software
Examination - Component A	✓	25 %	Exam (2 hours)
Resit Components	**Final Assessment**	**Element weighting**	**Description**
Written Assignment - Component B		75 %	Practical coursework involving, for example, the production of encryption or benchmarking software
Examination - Component A	✓	25 %	Exam (2 hours)

Part 4: Teaching and Learning Methods

Learning Outcomes

On successful completion of this module students will be able to:

	Module Learning Outcomes
MO1	Recognise the importance and need for coding data streams in computer science for security, error correcting, compression
MO2	Understand and manipulate the mathematical and theoretical methods on which designs are based
MO3	Implement algorithms and protocols to for particular coding schemes, recognising the need for efficiency in terms of delay, throughput, jitter, computing resources and quality of service
MO4	Use cryptographic and coding classes available in modern programming language environments, such as Java Security, to implement secure applications
MO5	Evaluate the performance of various coding schemes under application load and change configuration parameters to optimise them
MO6	Determine modern encryption techniques appropriate for a variety of applications
MO7	Explain the strategies that need to be employed whilst attempting to break a cipher

Contact Hours

Contact Hours	
Independent Study Hours:	
Independent study/self-guided study	114
Total Independent Study Hours:	114
Scheduled Learning and Teaching Hours:	
Face-to-face learning	36

	Total Scheduled Learning and Teaching Hours:	36
	Hours to be allocated	150
	Allocated Hours	150
Reading List	*The reading list for this module can be accessed via the following link:* https://uwe.rl.talis.com/modules/ufcft4-15-3.html	