Universitatea Naţională de Ştiinţă şi Tehnologie Politehnica Bucureşti
Facultatea de Electronică, Telecomunicaţii şi
Tehnologia Informaţiei

COURSE DESCRIPTION

1. Program identification information

1.1 Higher education institution	**National University of Science and Technology Politehnica Bucharest**
1.2 Faculty	**Electronics, Telecommunications and Information Technology**
1.3 Department	**Telecommunications**
1.4 Domain of studies	Electronic Engineering, Telecommunications and Information Technology
1.5 Cycle of studies	Bachelor/Undergraduate
1.6 Programme of studies	Technologies and Telecommunications Systems

2. Date despre disciplină

2.1 Course name (ro) (en)				Bazele criptologiei 1 Fundamentals of Cryptology 1			
2.2 Course Lecturer				Colaborator Dr. Adriana Clim			
2.3 Instructor for practical activities				Colaborator Dr. Gabriel Negara			
2.4 Year of studies	3	2.5 Semester	II	2.6. Evaluation type	V	2.7 Course regime	Op
2.8 Course type		S	2.9 Course code	04.S.06.A.229		2.10 Tipul de notare	Nota

3. Total estimated time (hours per semester for academic activities)

3.1 Number of hours per week	3	Out of which: 3.2 course	2.00	3.3 seminary/laboratory	1
3.4 Total hours in the curricula	42.00	Out of which: 3.5 course	28	3.6 seminary/laboratory	14
Distribution of time:					hours
Study according to the manual, course support, bibliography and hand notes Supplemental documentation (library, electronic access resources, in the field, etc) Preparation for practical activities, homework, essays, portfolios, etc.					8
Tutoring					8
Examinations					2
Other activities (if any):					2

3.7 Total hours of individual study	8.00
3.8 Total hours per semester	50
3.9 Number of ECTS credit points	2

4. Prerequisites (if applicable) (where applicable)

4.1 Curriculum	Programming language.

4.2 Results of learning	General knowledge of mathematics and computer science (beginner/intermediate level in an object-oriented programming language, ease of use of the computer).

5. Necessary conditions for the optimal development of teaching activities (where applicable)

5.1 Course	None.
5.2 Seminary/ Laboratory/Project	Mandatory attendance to understand, through implementation, the concepts taught.

6. General objective *(Reffering to the teachers' intentions for students and to what the students will be thought during the course. It offers an idea on the position of course in the scientific domain, as well as the role it has for the study programme. The course topics, the justification of including the course in the currcula of the study programme, etc. will be described in a general manner)*

The discipline ensures the acquisition of fundamental knowledge in the field of classical and modern cryptology. This includes:
Theoretical Foundations:

- Acquiring the basic mathematical concepts essential for an introduction to cryptography.
- Understanding the types of fundamental classical systems and the evolution of cryptographic systems up to the transition to modern cryptology.
- Exploring the influence of information theory and the principles of modern cryptology.
- Classifying and understanding the concepts of computer security, common attacks, and security level assessment

Practical Application:

- Practically acquiring the concepts taught in the course by implementing software programs in the C# language
- Solving concrete practical problems that involve aspects such as estimating the level of resistance of a cryptographic system, attack methods, data security concepts, and the verification/validation of the security level provided by the system.

7. Competences *(Proven capacity to use knowledge, aptitudes and personal, social and/or methodological abilities in work or study situations and for personal and proffesional growth. They refflect the empolyers requirements.)*

Specific Competences	Developing the skills to apply general knowledge of fundamental mathematics and algorithmics in the analysis of cryptographic system resistance. The possibility of developing software applications that implement attacks on cryptographic systems.
Transversal (General) Competences	Developing the competence to objectively evaluate a system's security level.

8. Learning outcomes *(Synthetic descriptions for what a student will be capable of doing or showing at the completion of a course. The learning outcomes reflect the student's acomplishments and to a lesser extent the teachers' intentions. The learning outcomes inform the students of what is expected from them with respect to performance and to obtain the desired grades and ECTS points. They are defined in concise terms, using verbs similar to the examples below and indicate what will be required for evaluation. The learning outcomes will be formulated so that the correlation with the competences defined in section 7 is highlighted.)*

Knowledge	*The result of knowledge aquisition through learning. The knowledge represents the totality of facts, priciples, theories and practices for a given work or study field. They can be theoretical and/or factual.* Acquisition of theoretical and practical knowledge regarding the cryptographic component of system security used for data protection.
Skills	*The capacity to apply the knowledge and use the know-how for completing tasks and solving problems. The skills are described as being cognitive (requiring the use of logical, intuitive and creative thinking) or practical (implying manual dexterity and the use of methods, materials, tools and intrumentation).* The ability to estimate the impact level of theoretical or practical vulnerabilities that may affect a system with cryptographic components.
Responsability and autonomy	*The student's capacity to autonomously and responsably apply their knowledge and skills.* The competence to realistically evaluate the security strength offered by a system incorporating cryptographic components.

9. Teaching techniques *(Student centric techniques will be considered. The means for students to participate in defining their own study path, the identification of eventual fallbacks and the remedial measures that will be adopted in those cases will be described.)*

Active learning methods, interactive participation in the course and laboratory.

10. Contents

COURSE		
Chapter	**Content**	**No. hours**
1	Mathematical foundations, elements of higher algebra, number theory, elements of information theory, aspects of crypto history, classifications	6
2	Classical cryptology. Substitution type systems - simple, non-uniform, multiple, polyalphabetic systems.	6
3	Transposition type systems - with complete or incomplete square	4
4	Combined systems, cipher machines, perfect systems.	6
5	Introduction to modern cryptology. Types of systems, advanced concepts of modern systems security, standards and evaluation.	6
	Total:	28

Bibliography:

Applied cryptography – Bruce Schneier, ed II
Design, Principle and Practical Applications – Cryptography Engineering – Niels Ferguson, Bruce Schneier, Tadayoshi Kohno
A guide to computer Network Security, Joseph Migga Kizza
Fundamentals of cryptology, Henk C.A. van Tilborg
Handbook of applied cryptography – Alfred Menezes, Oorschot, Vastone.
Understading Cryptography – Cristofor Paar, Jan Pelzl

LABORATORY		
Crt. no.	**Content**	**No. hours**
1	Introduction to the C# language - programming principles, syntax, types of Visual Studio applications, code structuring, examples of .NET and custom classes.	4
2	Presentation of advanced language elements – instruction types, operators, .NET and user-defined data types, work scenarios.	2
3	Design and implement software applications that use and exemplify the concepts, notions and types of cryptographic systems presented in the course.	6
4	Implementation component within the evaluation tests.	2
	Total:	14
Bibliography: www.msdn.com Materials created by course/application owners		

11. Evaluation

Activity type	11.1 Evaluation criteria	11.2 Evaluation methods	11.3 Percentage of final grade
11.4 Course	Acquisition of fundamental theoretical concepts. Understanding how to apply theory to specific problems. Comparative analysis of theoretical techniques and methods.	Written tests conducted during the semester, covering the entire curriculum. These tests will synthesize the comparative theoretical study of the subject matter with the practical application models explained through exercises and problems.	50%
11.5 Seminary/laboratory/project	Acquiring the necessary concepts for analyzing the security level offered by specific applications, as well as the software implementation of these applications. Programming methodology and the type of solutions identified. Active engagement throughout the laboratory work.	The evaluation of practical implementation skills demonstrated during partial assessments. Bonus points for active involvement during the classes. A minimum of one evaluation focusing on application and practical skills.	50%
11.6 Passing conditions			

- Understanding and internalizing the basic course notions. - Basic knowledge of C#, including the development of console and Windows Forms applications. - Design and implementation of applications that apply the cryptographic concepts and systems covered in the course.

12. Corroborate the content of the course with the expectations of representatives of employers and representative professional associations in the field of the program, as well as with the current state of knowledge in the scientific field approached and practices in higher education institutions in the European Higher Education Area (EHEA)

Cryptology is a key component of any method used to ensure the security of electronic communications and data.

Date	Course lecturer	Instructor(s) for practical activities
	Colaborator Dr. Adriana Clim	Colaborator Dr. Gabriel Negara
Date of department approval	Head of department	
Date of approval in the Faculty Council	Dean	