COMPUTER SYSTEMS SECURITY

COURSE OUTLINE

1. GENERAL			
SCHOOL	School of Engineering		
ACADEMIC UNIT	Department of Electrical and Computer Engineering		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	0811.9.015.0	**SEMESTER**	1st
COURSE TITLE	Computer Systems Security		

INDEPENDENT TEACHING ACTIVITIES if credits are awarded for separate components of the course	**WEEKLY TEACHING HOURS**	**CREDITS**
	0	4
Total	**0**	**4**

COURSE TYPE general background, special background, specialised general knowledge, skills development	Specialization
PREREQUISITE COURSES	Operating Systems
LANGUAGE OF INSTRUCTION and EXAMINATIONS	English
OFFERED TO ERASMUS STUDENTS	Yes (in English)
COURSE WEBSITE (URL)	https://eclass.hmu.gr/courses/ECE150

2. LEARNING OUTCOMES

Learning outcomes

The **knowledge** which students acquire upon successful completion of the course relates to understanding the design of multilayer protection mechanisms for computing systems, with an emphasis on embedded systems security. Security primitives are examined in detail, including lightweight cryptographic software libraries and hardware security devices (programmable crypto engines, crypto ICs). In addition, security patterns/protocols for efficient access control, data privacy, anonymity, confidentiality, integrity, and availability are examined. Case studies range from device security (cryptos), to memory protection/isolation (ARM Trustzone), to operating system kernel and file system support, to application and system/network security, including high-level security event tracing, correlation, and visualization.
The **skills**, which students develop upon successful course completion, relate to:
• Understanding the design and use of public key and symmetric cryptography (lightweight
• Understanding the design and use of digital certificates and signatures
• Designing and implementing protocols and techniques for security and data privacy at device, system/network, and application level
The **abilities**, which students develop upon successful course completion, enable problem-solving abilities that relate to
• Integrating security/trust in system/platform design and implementation
• Implementing secure embedded systems using lightweight security primitives/protocols
• Validating security functions and evaluating overheads of at device-, system-, and network-level

General Competences

- Search, analysis and synthesis of data and information, using the necessary technologies
- Adapt solutions to new situations (resource sharing, congestion, contention etc)
- Autonomous work
- Teamwork
- Decision making
- Work in an interdisciplinary environment
- Promoting liberal, creative and inductive/deductive thinking

3. SYLLABUS

Theoretical Lectures

- History – Classical Cryptography- Mathematical Preliminaries
- Authentication, Authorization
- SW Vulnerabilities - Memory Errors/Buffer Overflows, Viruses, Worms
- Side Channel, Energy Profiling, Covert Channel
- Buffer Overflows
- Operating System and Network Security (DDoS, Firewall, IPSec, OpenSSL/TLS, OpenVPN, syslog/IDPS)
- Symmetric Cryptography, NIST-approved Operating Modes
- Public Key Cryptography (RSA, Diffie Hellmann) & Elliptic Cryptography
- Security Primitives, Protocols, and Services
- Digital Certificates & Signatures
- Message Authenticity - Merkle Trees
- Application Security – Web/Ηλεκτρονικό Ταχυδρομείο (HTTPS, SMTP)
- Embedded Security, Cybersecurity & Safety, e.g. Smart Vehicles, e-Health platforms
- Special Topics (e.g. Blockchains, Steganography, Kerberos, Secret Sharing, Zero-Knowledge Proofs, Oblivious Transfers, Commit Protocols, HW Security (Crypto ICs, ARM Trustzone, Secure Boot/File Systems), Homomorphic Security, Quantum Cryptography, Data Privacy, Anonymity, Onion Routing/Tor, Legal Framework, GDPR, HIPAA etc)

Lab

The student lab focuses on open source hardware/software and Linux system security. Students gain experience in cryptographic mechanisms (AES encryption/decryption, integrity), authentication (SHA3, one-way hash functions), domain isolation, data privacy and anonymity by applying well-established security patterns for device, system/network, and application security. The lab also examines practical use of software tools, cryptographic security libraries, programmable crypto engines, and crypto ICs in experimental platforms and real embedded systems, such as healthcare and automotive.

4. TEACHING and LEARNING METHODS - EVALUATION

DELIVERY Face-to-face, Distance learning, etc.	Eclass for optional exercises. Project presentations and demonstrations in the Lab.
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY Use of ICT in teaching, laboratory education, communication with students	Eclass
TEACHING METHODS The manner and methods of teaching are described in detail.	

Activity	**Semester workload**
Course total	

STUDENT PERFORMANCE EVALUATION Description of the evaluation procedure	All announcements related to the syllabus, including grading, and complementary reading material, solved exercises, and optional homeworks, are permanently posted in the course web page (ECLASS). The course grade incorporates the following evaluation procedures: 1. Final Oral Exam on theoretical/practical problems (50%) 2. Term Project Presentation and Demonstration (50%) The project usually relates to systems/network programming, Linux drivers & kernel modules, RTOS, real-time systems or small software stacks. Students provide weekly reports on their progress, and a final presentation and demonstration at the end of their project.

5. ATTACHED BIBLIOGRAPHY

Recommended Bibliography:

• P. C. Pfleeger, S. L. Pfleeger, J. Margulies, "Security in Computing", 5th edition, 2018. Prentice Hall, 2018.
• D. Basin, P. Schaller, M. Schlaepfer, "A Hands-on Approach", Springer, 2011.

Other Important Sources

• Eclass - http://eclass.hmu.gr (notes, examples, open source coce)
• Development boards, pirate devices, virtual machines accompanied with open source software and manuals for examining attack and devising protection mechanisms

Relevant Scientific Journals & Conferences

• ACM Transactions on Privacy and Security
• IEEE Transactions on Dependable and Secure Computing
• IEEE Security & Privacy
• IEEE Transactions on Information Forensics & Security
• IEEE Transactions on Intelligent Transportation Systems
• IEEE Transactions on Vehicular Technology
• USENIX Security Symposium
• IEEE Symposium on Security and Privacy
• DEFCON and BLACKHAT conferences
• Embedded Security-related conferences, e.g. Embedded Security in Cars (ESCAR), Linux Security Summit, Automotive Linux Summit, Automotive Manufacturing Summit, Automotive World