Course Number, Title, Credits, Contact hours

CYBR 463L/563L; 4 Credits; 5 Contact Hours

Course Description

This course covers the general principles of modern cryptography, including symmetric cryptosystems, asymmetric cryptosystems, secure hash functions, and cryptographic level randomness. Other topics may include historic cryptosystems and their cryptanalysis, information entropy, zero knowledge proofs, trusted computing architectures, and information theory as it relates to cryptography. Programming assignments are required, writing and class presentations may be required.

Required, Elective or Selected elective

This is a required course for CYBR degree.

Prerequisites or Co-requisites

Prerequisites: MATH 225 or MATH 301 or equivalent.

Reference Text, Title, Author

Cryptography and Network Security, Behrouz A. Forouzan, McGraw-Hill, Inc

Cryptography and Network Security, William Stallings, Pearsons.

Course Learning Objectives

1. Understand the basic principles, goals, and foundational concepts of cryptography.
2. Learn to implement cryptographic techniques such as encryption, decryption, and hashing.
3. Explore the strengths, limitations, and security implications of cryptographic systems.
4. Study advanced topics like secure communication protocols, zero-knowledge proofs, and post-quantum cryptography.
5. Apply cryptographic methods to real-world security challenges and key management practices.
6. Analyze the ethical considerations and the impact of emerging technologies on cryptography.

Weekly Lab Exercises

Week	Lab Title	Description
1	Cryptanalysis and Classical Ciphers	Implement Traditional substitution Ciphers like Caesar and Vigenère ciphers using Java/Python/C
2	Number Theory for Cryptography	Implement modular arithmetic, GCD (Euclidean algorithm) and modular exponentiation
3	Implementing DES	Write a simplified DES encryption function focusing on S-box substitution, permutation, and key scheduling.

Week	Lab Title	Description
4	AES Key Expansion and Encryption	Implement AES key expansion and a single encryption round without using libraries. Validate against a Python cryptographic library.
5	Block Cipher Modes of Operation	Explore ECB, CBC, and CTR modes using PyCryptodome. Encrypt and decrypt files and visualize image encryption results.
6	Implementing RSA	Implement RSA key generation, encryption, and decryption. Compare with OpenSSL-generated RSA keys. Other Public-Key Cryptosystems: Performing an MITM Attack
7	Cryptographic Hash Functions	Create a custom hash function, compute SHA-256 hashes using Python, and analyze collision resistance with test cases.
8	Digital Signatures; Cryptographic Key Management and Distribution	Implement RSA-based digital signatures manually and verify them. Use OpenSSL to validate signatures. Cryptographic Key Management and Distribution : Examining a Self-Signed Certificate, Examining PKI Certificates
9	Post-Quantum Cryptography and Blockchain	Use a post-quantum cryptography library to explore algorithms like Kyber or SPHINCS+. Building a simple Blockchain
10	Building a PKI and Trusted Computing	Generate X.509 certificates, validate a certificate chain, and simulate a secure communication environment.