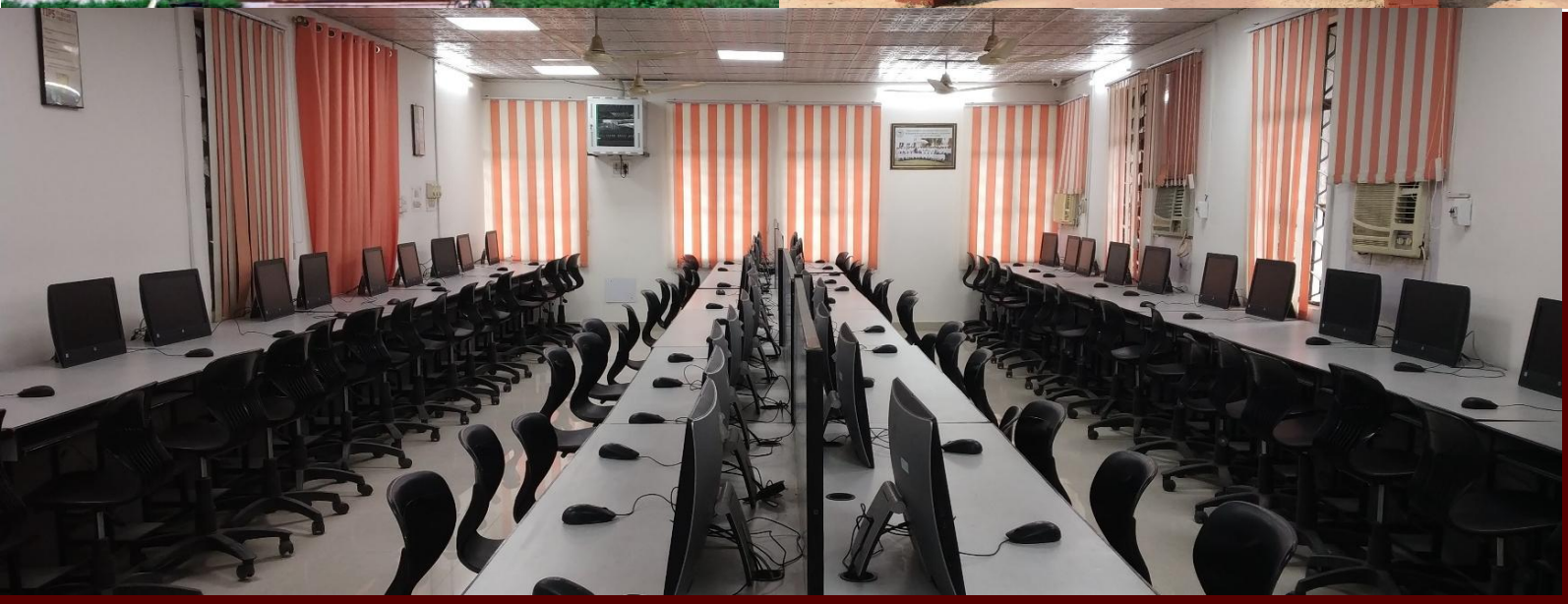
جب کوئی قوم فن اور علم سے عاری ہو جاتی ہے تو وہ غربت کو
دعوت دیتی ہے اور جب غربت آتی ہے تو وہ ہزاروں جرائم کو جنم

"When a nation becomes devoid of art and learning, it invites poverty and when poverty comes it brings in its wake thousands of crimes."

-Sir Syed Ahmad Khan

B.Sc. (CA) VI Semester Lab



Laboratory Course-VI

Course Code- CABSXO6P04

DEPARTMENT OF COMPUTER SCIENCE

ALIGARH MUSLIM UNIVERSITY, ALIGARH

2025-2026

Credits

The following lab manual up-gradation committee updated the lab manual:

- ❒ *Prof.Arman Rasool Faridi (Chairperson)*
- ❒ *Prof. Mohammad UbaidUllah Bokhari*
- ❒ *Prof. Aasim Zafar*
- ❒ *Prof. Suhel Mustajab*
- ❒ *Dr. Faisal Anwar*
- ❒ *Dr. Mohammad Sajid*
- ❒ *Dr. Mohammad Nadeem*
- ❒ *Dr. Shabbir Hassan*
- ❒ *Dr. Faraz Masood*

The following committee member originally design the lab manual:

- ❒ *Prof. Mohammad Ubaidullah Bokhari*
- ❒ *Dr. Arman Rasool Faridi*
- ❒ *Dr. Faisal Anwer*
- ❒ *Prof. AasimZafar (Convener)*

Design & Compilation:

- ❒ *Dr. Faraz Masood*

Revised Edition: Jan, 2026

Department of Computer Science, A.M.U.,

Aligarh (U.P.) India

COURSE TITLE: Laboratory Course-VI **COURSE CODE:** CABSXO6P04
CREDIT: 02 **PERIODS PER WEEK:** 03
CONTINUOUS ASSESSMENT: 40 **EXAMS:** 60

COURSE DESCRIPTION

This course is designed to help students in learning cyber security concepts using available online tools in a controlled lab environment. Approach in this course is to take teach the theory of cyber security, mathematics behind it and hands-on implementation to achieve security measures and withstand against several cryptographic attacks.

COURSE CONTENT

This course is designed to provide the students the opportunity to learn the the concepts of cyber, cyber space and cyber security. How to procect and maintain the cyber space against various cyber attacks.

OBJECTIVES

This course is designed to help students in:

- Utilize the SageMath tool for solving mathematical problems, graphing, simulations, and modeling.
- Leverage the Python Runtime Environment (PRE) for problem-solving and modular programming using functions and libraries.
- Implement number theoretic concepts like closures, totatives, and orders, essential for cryptography and cryptanalysis.
- Explore the intrinsic properties of numbers that support the logic behind cryptographic algorithms.

- ❒ Develop classical ciphers such as Caesar, Playfair, and Hill and apply substitution and permutation techniques to Feistel Networks.
- ❒ Understand hash functions, their implementation, and performance comparisons in SageMath.
- ❒ Implement cryptographic algorithms, including asymmetric key systems, Message Authentication Code (MAC), and Digital Signature.
- ❒ Perform security analysis of IT setups, web application audits, penetration testing (using Burp Suite), and vulnerability assessments with Kali Linux tools.
- ❒ Study cybercrimes, the Indian IT Act 2000, and ITAA 2008, focusing on web application security measures like client/server-side validation.
- ❒ Identify and resolve issues like buffer overflow, TOCTOU bugs, and other critical software vulnerabilities.
- ❒ Conduct ethical hacking, real-time network traffic analysis, and secure communication setup using OpenSSL.
- ❒ Enhance backend server management through certificate generation, signing, and configuration.

OUTCOMES

After completing this course, the students would be able to:

- ❒ Proficiency in using SageMath for solving mathematical problems, graphing, simulations, modeling, and implementing number-theoretic concepts.
- ❒ Capability to use the Python Runtime Environment (PRE) for modular programming and problem-solving using libraries and functions.
- ❒ Understanding of intrinsic number properties and concepts like closures, totatives, and orders for cryptographic logic building and cryptanalysis.

- ❒ Implementation of classical ciphers, including Caesar, Playfair, and Hill, and substitution and permutation techniques in Feistel Networks.
- ❒ Knowledge of hash functions, their implementation, performance comparisons, and applications in cryptographic algorithms.
- ❒ Practical experience in developing cryptographic algorithms, including asymmetric key systems, Message Authentication Code (MAC), and Digital Signatures.
- ❒ Skills in conducting security audits, penetration testing using Burp Suite, and vulnerability assessments with tools like Kali Linux.
- ❒ Awareness of cybercrimes, the Indian IT Act 2000, and ITAA 2008, and their relevance to web application security.
- ❒ Competence in identifying and addressing software vulnerabilities like buffer overflow and TOCTOU bugs.
- ❒ Application of ethical hacking techniques and real-time network traffic analysis using open-source tools.
- ❒ Ability to establish secure communications using OpenSSL, including managing certificates and backend server configurations.
- ❒ Strengthened understanding of client-side and server-side validations in web application security.

RULES AND REGULATIONS

Students are required to strictly adhere to the following rules.

- ❒ The students must complete the weekly activities/assignments well in time (i.e., within the same week) that need to be checked and signed by the concerned teachers in the lab in the immediate succeeding week. Failing which activities/assignments for that week will be treated as incomplete.

- ❒ The students must maintain the Lab File of their completed activities/assignments in the prescribed format (**Appendix-1**).
- ❒ At least **TEN (10)** such timely completed and duly signed weekly activities/assignments are compulsory, failing which students will not be allowed to appear in the final Lab Examination.
- ❒ The students need to submit the following three deliverables for each exercise duly signed by the Teacher:
  - ❖ Coding
  - ❖ Input /Output
- ❒ Late submissions would not be accepted after the due date.
- ❒ Cooperate, collaborate, and explore for the best individual learning outcomes, but copying is strictly prohibited.
- ❒ The Continuous Lab assessment will be based on two sessionals, each carrying 30 marks.
- ❒ Marks distribution for each sessional:
  - ❖ 20 Marks: For a duly signed lab report by the respective lab teacher.
  - ❖ 5 Marks: For solving a lab question/program on the day of the sessional.
  - ❖ 5 Marks: For the viva conducted during the sessional.
- ❒ This distribution ensures a balanced evaluation of students' practical work, problem-solving skills, and conceptual understanding.

APPENDIX-1

Template for the Index of Lab File

WEEK NO.	PROBLEMS WITH DESCRIPTION		PAGE NO.	SIGNATURE OF THE TEACHER WITH DATE
1	1#			
	2#			
	3#			
2	1#			
	2#			
	3#			
3	1#			
	2#			
	3#			

Note: *The students should use Header and Footer mentioning their roll no. & name in footer and page no in header.*

WEEK #1

OBJECTIVE

- ❒ To explore the available cryptographic Tool.
- ❒ To learn basic of Python programming, libraries and runtime environment.
- ❒ To learn the properties of number Theory.

OUTCOMES

After completing this, the students would be able to:

- ❒ Use the SageMath Tool and their available features.
- ❒ Use Python Runtime Environment (PRE) for problem solving.

PROBLEMS

1# Prepare a report on SageMath tool and its different functionalities.

2# Write a Python program to detect whether given integer is perfect or not ?

3# Write a Python program to detect whether given integer is Armstrong Integer or not?

WEEK #2

OBJECTIVE

- ❒ To learn the various Primality Tests methods based on deterministic and probabilistic algorithms.
- ❒ To learn implementation of various number theoretical concepts.
- ❒ To explore the intrinsic properties of numbers.

OUTCOMES

After completing this, the students would be able to:

- ❒ Implement code, libraries, and program modularity (functions).
- ❒ Implement various number theoretic concepts.

PROBLEMS

1# Write a program to detect whether a number is prime or not.

2# Write a function to compute GCD of two integers.

3# Write a program to detect whether two numbers are relatively prime or not?

4# Write a function to display all'n' narcissistic number. For example, 153 is a 3 narcissistic number because $1^3 + 5^3 + 3^3 = 153$, and 1634 is 4 narcissistic number because $1^4 + 6^4 + 3^4 + 4^4 = 1634$.

e.g.,

Input: Enter value of 'n': 4
Expected output: 1634, 8208, and 9474

WEEK #3

OBJECTIVE

- To gain practical experience on numbers used in computation.
- To delve deeper into the properties of Prime numbers.

OUTCOMES

After completing this, the students would be able to:

- Learn the intrinsic features of numbers that play a major role in logic building of various cryptographic algorithms.
- Implement concepts such as Closures, Totatives, Orders, etc., that play a vital role in cryptanalysis.

PROBLEMS

1# Write a function to check whether a number is in the form of 2^k or not.

e.g.,

Test-1	Input:	Enter a number: 13
	Expected output:	13 is NOT in the form of 2^k
Test-2	Input:	Enter a number: 16
	Expected output:	16 is in the form of 2^k (for k = 4)

2# Write a function to check whether a prime number is Mersenne prime or not?

e.g.,

Test-1 Input: Enter a prime number: 13

Expected output: 13 is NOT Mersenne prime

Test-2 Input: Enter a prime number: 7

Expected output: 7 is a Mersenne prime

3# Write a program to evaluate Euler Totients (Totatives) of a given integer.

e.g.,

Input: Enter a number: 10

Expected output: Euler Totients of 10 is 4

Note: Similarly, another for set of inputs 2, 3, 4, 7, and 31the output will be 1, 2, 2, 6, and 30 respectively.

4# Implement a function that takes two arguments (r, n) and computer order of r under modulo(n) operation.

e.g.,

Test-1 Input: Enter value of r and n: 2 7

Expected output: Order of 2modulo(7) is 3

Test-2 Input: Enter value of r and n: 5 29

Expected output: Order of 5modulo(29) is 14

Note: Similarly, another for set of input (r: 3), and (n: 8, and 9) the output will be 2, and 'NOT defined' respectively, (because 3modulo(8)==2 and 3modulo(9) can't be defined).

WEEK #4

OUTCOMES OBJECTIVE

- To explore the essential security parameters, vulnerablilities and attack in web applications.
- To get aware of different types of cybercrimes.
- To learn the Indian IT Act.

After completing this, the students would be able to:

- Identify the underlying security measures, vulnerabilities and attack immune in web applications.
- Understand the different types of cybercrimes, Indian IT Act 2000, and IT Act Amendment 2008 (ITAA 2008).

PROBLEMS

1# Illustrate major potential security vulnerabilities in web applications. Use a checklist to document the vulnerabilities and suggest basic countermeasures.

2# Categorize the following hypothetical scenarios into different types of cybercrimes (e.g., hacking, identity theft, online fraud) and suggest how they can be prevented.

a. ***Hacking***
b. ***Identity Theft***
c. ***Online Fraud***
d. ***Cyberstalking and Harassment***
e. ***Malware Attacks***
f. ***DDoS***

g. ***Eavesdropping***

h. ***Cyber Espionage***

i. ***Intellectual Property Theft***

j. ***Online Defamation***

k. ***Child Exploitation***

l. ***Cyberbullying***

3# Create a timeline of key events in the development of the Indian IT Act. Discuss any one section of the act (e.g., Section 46, 66, 66A) with its implications in combating cybercrime.

4# Draft a basic security policy for a small office setup, focusing on password policies, data backup policies, and internet usage guidelines.

5# Write a program to test password strength based on length, complexity, characters, and patterns. Use it to evaluate common passwords.

WEEK #5

OBJECTIVE

- To develop skill to implement security in organizations.
- To get hands-on practice of performing brure force attack in a controlled lab environment.
- To explore the installation process of Kali Linux via bootable USB.
- To get an overview and working principle of network sniffing tools.

OUTCOMES

After completing this, the students would be able to:

- Perform security analysis of organisations' IT setup..
- Perform experiments on tools available in Kali Linux.

PROBLEMS

1# Perform a basic SWOT analysis (Strengths, Weaknesses, Opportunities, Threats) for a sample organization's IT security setup.

2# Write a program to simulate simple brute force attack. Try to crack numeric passwords of length 4,6 and 8 digits.

3# Install Kali Linux in your USB drive and demonstrate booting of the system using it. Write all the steps with proper screenshots.

4# Demonstrate the working of Wireshark tool to capture a sample TCP/IP packet details in Kali Linux. Write all steps with proper screenshots.

WEEK #6

OBJECTIVE

- To learn foundation of symmetric cryptosystem.
- To gain practical experience on implementation of classicical ciphers.

OUTCOMES

After completing this, the students would be able to:

- Implement classical ciphers such as Caesar, Playfair and Hill.
- Apply substitution and permutation techniques to implement the Feistel Network.

PROBLEMS

1# Write a program to perform encryption and decryption using XOR operation on plaintext 'Cyber Security' with value 0, 1 and 5.

2# Implement the Caesar cipher in Python to encrypt and decrypt a given text.

3# Write a program to implement the Playfair cipher for text encryption and decryption.

4# Write a program to encrypt and decrypt a simple text file using Hill Cipher technique in Python.

5# Write a program to implement a simplified version of the Feistel structure used in block ciphers.

WEEK #7

OBJECTIVE

- To implement various Hash functions.
- To grasp the understanding of key exchange scheme.

OUTCOMES

After completing this, the students would be able to:

- Know about Hash functions, their implementation and performance comparison.
- Implement cryptographic algorithms.

PROBLEMS

1# Write a program to implement a Hash function.

2# Write a program to implement BlowFish algorithm.

3# Write a program to implement TwoFish algorithm.

4# Implement the Diffie-Hellman Key Exchange scheme.

5# Calculate the message digest of a text using the SHA-1 algorithm.

WEEK #8

OBJECTIVE

- ❒ To understand the concepts of modular arithmetic.
- ❒ To learn basic theory and implementation of algebraic structures.
- ❒ To analyze the outcome of cryptosystems on SageMath platform.

OUTCOMES

After completing this, the students would be able to:

- ❒ Implement number theoteric concepts on SageMath.
- ❒ Implement assymetric key cryptosystem.

PROBLEMS

1# Find Inverse of a number under *modulo(p)* (whwrw p is a prime number) using SageMath.

2# You have been given a set and you are required to check whether it is Group, Ring or Field using SageMath.

3# Implement Elliptic Curve cryptography using SageMath.

4# Perform a simulated elliptic curve Diffie-Hellman (ECDH) key exchange using the SageMath's elliptic curve functionality.

5# Write a program for RSA algorithm using SageMath.

WEEK #9

OBJECTIVE

- ❒ To gain practical experience on implementation of secure hash algorithm.
- ❒ To analyze and compare the integrity and authenticity of messages or digital documents.

OUTCOMES

After completing this, the students would be able to:

- ❒ Implement various Hash functions in SageMath environment.
- ❒ Comprehend the concepts behind Message Authentication Code (MAC), and Digital Signature.

PROBLEMS

1# Implement SHA-1, SHA-2 and SHA-3 using SageMath.

2# Implement MD-5 using SageMath.

3# Demonstrate Hash-based Message Authentication Code (HMAC) implementation using Python.

4# Compare Message Authentication Code (MAC) and Digital Signatures on parameters like how they use keys, security level, and computational cost by simulating both in Python.

WEEK #10

OBJECTIVE

- To gain practical experience on Digital Sinature.
- Gain insign on software tool for security assessment and penetration testing.

OUTCOMES

After completing this, the students would be able to:

- Know the implication of Digital Signature and their implementation.
- Perform web application security audits and penetration testing by using Burp Suite.

PROBLEMS

1# Research and implement a digital signature verification process using a cryptographic library. Sign a document with a private key and verify the signature using a public key.

2# Write a report on working of Burp Suite and its different functionalities.

3# Perform a brute-force attack on a dummy login page using Burp Suite. Document the findings and mitigation strategies.

WEEK #11

OBJECTIVE

- To analyze the security vulnerabilities of some dummy web application.
- To learn working of code injection techniques on unsecure web applications.

OUTCOMES

After completing this, the students would be able to:

- Apply security concepts and attack immune of web applications DVWA.
- Know about the significance of client side validation and server side validations.

PROBLEMS

1# Explore Damn Vulnerable Web Application (DVWA) to understand how various vulnerabilities work and practice securing them.

2# Set up a basic vulnerable web application (e.g., DVWA) and identify at least two common vulnerabilities (e.g., XSS or SQL injection).

3# Create a web form with input fields and demonstrate how improper validation can allow SQL injection. Propose a solution using proper input sanitization.

WEEK #12

OBJECTIVE

- ❒ To learn the basis of buffer overflow and their solution.
- ❒ To get aware about software bug.

OUTCOMES

After completing this, the students would be able to:

- ❒ Identify the issue of buffer overflow, their pros, cons and solution.
- ❒ Rectify the serious software bug TOCTOU.

.

PROBLEMS

1# Write a C program that demonstrates a simple buffer overflow. Analyze the overflow behavior and discuss its implications.

2# Write a program in C to demonstrate integer overflow during arithmetic operations. Explain its potential consequences.

3# Simulate a Time-of-Check to Time- of-Use (TOCTOU) vulnerability using a simple file access program in C. Discuss how to mitigate such vulnerabilities.

WEEK #13

OBJECTIVE

- ❒ To learn the basic of hacking and ethical hacking.
- ❒ To explore the vulnaribility of applications running with http protocol.
- ❒ To gain a practical hands-on on cryptographic attacks.

OUTCOMES

After completing this, the students would be able to:

- ❒ Implement Ethical Hacking procedures.
- ❒ Analyzes network traffic in real time by using a free & open-source tool.

PROBLEMS

1# List some vulnerable websites which are used to learn Ethical hacking like http://www.itsecgames.com/.

2# Find out user id and password of any website hosted as HTTP protocol using Wireshark tool.

3# Attack atleast five vulnerabilities of vulnerable websites (like http://www.itsecgames.com/) using various tools.

4# Simulate a packet sniffing attack using Wireshark and analyze the captured packets. Write steps.

WEEK #14

OBJECTIVE

- ❒ To get familiar about the advanve features of free, open-source mathematical modelling softwares.
- ❒ To gain practical experience on how OpenSSL is used.

OUTCOMES

After completing this, the students would be able to:

- ❒ Know about the working principle of a free, open-source mathematical software tool SageMath. How SageMath is used to solve a Simple to Complex Calculations, Graphing, Simulations, and Modeling.
- ❒ Know how to established a secure communication between network endpoints using OpenSSL. Also, how to configure servers, managing certificates, generating certificate signing requests, and other backend service management.

PROBLEMS

1# Prepare a report on SageMath tool and its different functionalities.

2# Use OpenSSL to inspect the details of an HTTPS connection.

3# Configure a local server (e.g., Apache or Nginx) to use HTTPS with a self-signed SSL certificate.

4# Set up an email client (e.g., Thunderbird) with PGP encryption and send a secure email.

5# Set up a basic firewall using iptables or Uncomplicated Firewall (UFW) in Linux and configure rules for blocking and allowing traffic.