5. Stallings W. *Network security essentials Applications and Standards*, 6th edition, Pearson Education, 2018.
6. Whitman M.E., Mattord H.J., *Principle of Information Security*, 6th edition, Cengage Learning, 2017.
7. Bishop M., *Computer Security: Art and Science*, 2nd Revised edition, Pearson Education, 2019.
8. Anderson R.J., *Security Engineering: A guide to building Dependable Distributed Systems*, 2nd edition, John Wiley & Sons, 2008.

Suggested Practical List

1. Demonstrate the use of Network tools: ping, ipconfig, ifconfig, tracert, arp, netstat, whois.
2. Use of Password cracking tools : John the Ripper, Ophcrack. Verify the strength of passwords using these tools.
3. Use nmap/zenmap to analyze a remote machine.
4. Use Burp proxy to capture and modify the message.
5. Implement caesar cipher substitution operation.
6. Implement monoalphabetic and polyalphabetic cipher substitution operation.
7. Implement playfair cipher substitution operation.
8. Implement hill cipher substitution operation.
9. Implement rail fence cipher transposition operation.
10. Implement row transposition cipher transposition operation.
11. Implement product cipher transposition operation.

GE8c/DSE: INTRODUCTION TO PARALLEL PROGRAMMING

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lectur e	Tutorial	Practical/ Practice		

Introduction to Parallel Programming	**4**	**3**	**0**	**1**	**Pass in Class XII**	**Computer System Architecture/A course in C++at class XII/Data Structures, Operating Systems**

Course Objective

The course introduces the students to the basic concepts and techniques of parallel programming. It enables them to design and implement parallel algorithms. The course would give the students hands-on practice to write parallel programs using shared and distributed memory models using OpenMP and Message Passing Interface (MPI).

Course Learning Outcomes

On successful completion of this course, the student will be able to:

1. Appreciate the need of Parallel algorithms
2. Describe architectures for parallel and distributed systems.
3. Develop elementary parallel algorithms in shared memory models.
4. Develop elementary parallel algorithms in distributed memory models.

Syllabus

Unit 1

Introduction to Parallel Computing: Trends in microprocessor architectures, memory system performance, dichotomy of parallel computing platforms, physical organization of parallel platforms, communication costs in parallel machines, SIMD versus MIMD architectures, shared versus distributed memory, PRAM shared-memory model, distributed-memory model.

Unit 2

OpenMP programming for shared memory systems: Thread Basics, Controlling Thread and Synchronization Attributes, Multi-thread and multi-tasking, Context Switching, Basic OpenMP thread functions, Shared Memory Consistency Models and the Sequential Consistency Model, Race Conditions, Scoping variables, work-sharing constructs, critical sections, atomic operations, locks, OpenMP tasks, Introduction to tasks, Task queues and task execution, Accessing variables in tasks, Completion of tasks and scoping variables in tasks.

Unit 3

MPI programming for distributed memory systems: MPI basic communication routines (Introduction to MPI and basic calls, MPI calls to send and receive data, MPI call for broadcasting data, MPI Non-blocking calls, Introduction to MPI Collectives, Types of interconnects (Characterization of interconnects, Linear arrays, 2D mesh and torus, cliques)

Unit 4

Applications: Matrix-matrix multiply, Odd-Even sorting, distributed histogram, Breadth First search, Dijkstra's algorithm.

References

1. Grama, A., Gupta, A., Karypis, G., Kumar, V., *Introduction to Parallel Computing*, 2nd edition, Addison-Wesley, 2003.
2. Quinn, M., *Parallel Programming in C with MPI and OpenMP*, 1st Edition, McGraw-Hill, 2017.
3. Revdikar, L., Mittal, A., Sharma, A., Gupta, S., *A Naïve Breadth First Search Approach Incorporating Parallel Processing Technique For Optimal Network Traversal*, International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 5, May 2016

Additional references

(i) B. Parhami, Introduction to Parallel Processing: Algorithms and Architectures, Plenum, 1999, Springer.

Suggested Practical List

1. Implement Matrix-Matrix Multiplication in parallel using OpenMP
2. Implement distributed histogram Sorting in parallel using OpenMP
3. Implement Breadth First Search in parallel using OpenMP
4. Implement Dijkstra's Algorithm in parallel using OpenMP

DSC17/GE7d/DSE8e: CLOUD COMPUTING

Credit distribution, Eligibility and Pre-requisites of the Course

	Credits	Credit distribution of the course		