Assignment #2

Parallel and Distributed Computing

Instructions:

- Need to submit .zip of C files only (MPI in C only)
- Strict plagiarism policy applies to the code submitted
- Viva will be taken for all submissions
- No viva means zero marks
- Non running code will be awarded zero marks

Q1: Simplified AES (S-AES) is to AES as Simplified DES(S-DES) is to DES. In fact, the structure of S-AES is exactly the same as AES. The differences are in the key size (16 bits), the block size (16 bits) and the number of rounds (2 rounds.

Design a parallel algorithm using **MPI (Message Passing Interface)** to crack the **Simplified AES (S-AES)** encryption scheme. Your algorithm should divide the brute-force key search across multiple MPI processes, leveraging the computational benefits of parallel programming on an MPI cluster. [50 marks]

Details of S-AES:

https://www.rose-hulman.edu/class/ma/holden/Archived_Courses/Math479-0304/lectures/s-aes.pdf

Steps to implement:

1. **Key Search Problem**:
    - Assume that a known plaintext-ciphertext pair is available.
    - Use a brute-force approach to search for the correct 16-bit key that was used to encrypt the plaintext.
2. **Parallelization Strategy**:
    - Divide the 16-bit key space (65,536 possible keys) evenly among the MPI processes. For example, with 4 processes, each would handle 16,384 keys.
    - Each process should work on a subset of the key space and check if the correct key produces the given ciphertext from the plaintext.
    - Once a process finds the correct key, it should notify all other processes to terminate.
3. **Benefits of MPI**:
    - Explain how the **parallelization** of the brute-force key search using MPI improves the performance compared to a serial implementation.
    - Discuss how dividing the key space across multiple processes reduces the overall computation time.

4. **Code Implementation**:
    - Write the parallel code using MPI that performs the brute-force attack.
    - Ensure that the program uses efficient communication and synchronization among processes.

Expected Output:

- The correct 16-bit key found by the MPI cluster through parallel brute-force search.
- An analysis of the time taken for the brute-force key search compared to a serial approach.

Implementation Steps:

- **Initialize MPI Environment:** Set up MPI with MPI_Init, determine the number of processes (MPI_Comm_size), and identify each process's rank (MPI_Comm_rank).
- **Assign Key Ranges:** Calculate the range of keys each process will handle based on its rank.
- **Brute-Force Attack:** Each process iterates through its assigned key range, performing decryption and checking against the known plaintext.
- **Communication:** Once a process finds the correct key, it can broadcast the result to all other processes using MPI_Bcast or MPI_Send/MPI_Recv, prompting them to terminate early.
- **Finalize MPI:** Clean up the MPI environment with MPI_Finalize.

Q2: Add a simple function in your MPI program to validate that you have successfully find out the correct key from give combinations? Each process independently attempts to decrypt the ciphertext using its subset of keys and compares the result with the known plaintext to identify the correct key. [50 marks]