SYNOPSIS OF MINI PROJECT- Team 6

From Classical to Quantum: Parallelizing AES, RSA Algorithm on CPU, GPU and Shor's Algorithm on Quantum Computers

Introduction: This project investigates the efficiency and feasibility of quantum computing in solving cryptographic problems by implementing and analyzing the AES (Advanced Encryption Standard) and RSA (Rivest-Shamir-Adleman) algorithms on CPU and GPU, as well as executing Shor's algorithm on a quantum backend. The aim is to explore the potential benefits of these critical algorithms across different computing architectures.

Project Goals:

1. Implement AES and RSA algorithms on CPU and GPU.
2. Execute Shor's algorithm on a quantum backend.
3. Compare the performance and efficiency of these implementations.
4. Recommend the most efficient computational platforms for various types of algorithms.

Methodology:

- **Implementation of Algorithms:** AES and RSA algorithms were implemented and tested on both CPU and GPU using Google Colab.
- **Quantum Execution:** Shor's algorithm was executed on a quantum backend using IBM's Qiskit framework.

Technologies and Tools:

- **Google Colab:** Used for CPU and GPU computations.
- **Qiskit Library:** An open-source quantum computing framework by IBM, used for creating, manipulating, and running quantum circuits on quantum processors and simulators.

Results:

AES Algorithm:

- **CPU Execution:** Moderate performance for encrypting data, showing efficiency but limited by the sequential nature of CPU operations.
- **GPU Execution:** Significantly faster due to parallel processing capabilities, ideal for large-scale data encryption tasks.

RSA Algorithm:

- **CPU Execution:** Slower due to the computational intensity of asymmetric encryption, particularly for large data sets.
- **GPU Execution:** Improved performance, though still computationally heavy, making it more suitable for scenarios where secure key exchange is critical.

Shor's Algorithm:

- **Quantum Backend Execution:** Successfully executed on IBM's quantum processors, demonstrating the algorithm's potential to factorize integers efficiently. This showcases the power of quantum computing in potentially breaking RSA encryption.

Conclusion:

The project demonstrates the potential of using GPUs to enhance the computational efficiency of cryptographic algorithms like AES and RSA. Shor's algorithm highlights the promise of quantum computing in factorizing large numbers, which is crucial for breaking RSA encryption. In summary:

- **AES:** Used for symmetric key encryption, suitable for securing data transmission due to its speed, especially when parallelized on GPUs.
- **RSA:** Used for asymmetric key encryption, ideal for secure key exchange but computationally intensive, benefiting from GPU acceleration.
- **Shor's Algorithm:** Quantum algorithm used for factorizing large integers, crucial for breaking RSA encryption, showcasing the potential of quantum computing.