Semester Project Documentation (DSA – CS – 221)

Semester Project Title: PHASE (Post Quantum Cryptographic High Assurance Encryption Model)

Student Details:

	Student Name	**Student Reg #**	**Student Degree**
Student-1	Asad Shayan	2023629	BSCS
Student-2	Yahya Qadeer Dar	2023759	BSCS
Student-3	Rayyan Hassan Salman	2023601	BSCS

1. Main Features

1. Quantum-Resistant Encryption
2. Secure Key Exchange Mechanism
3. High Performance and Efficiency
4. Layered Security Architecture
5. Scalability and Adaptability
6. SMS Chat Application:
   Real-time messaging between two users
   Encrypted communication using PHASE_C encryption model

2. Types of Users & Requirements

User: Client (Sender)

- Must be able to connect to the server.
- Should be able to input and send encrypted text messages.
- Receive and decrypt responses from the server.

User: Server (Receiver)

- Listen to incoming messages.
- Decrypts the received messages and sends them to the recipient.
- Responds back with an acknowledgment encrypted using the same encryption model.

System Administrator

- Manage server operations and ensure system security.
- Can monitor real-time messages sent and received between clients.

3. Requirements Breakdown

1. Quantum-Resistant Encryption
   1.1 Use post-quantum cryptographic algorithms like lattice-based cryptography.
   1.2 Ensure resistance to both classical and quantum attacks.
   1.3 Provide secure API endpoints for data encryption and decryption.

2. Secure Key Exchange Mechanism
    2.1 Implement a quantum-safe key exchange protocol.
    2.2 Generate keys that cannot be derived even with quantum computing power.

3. High Performance and Efficiency
    3.1 Optimize encryption algorithms to handle large datasets with minimal latency.
    3.2 Utilize advanced data structures like heaps and hash maps to improve computational efficiency.

4. Layered Security Architecture
    4.1 Combine symmetric and asymmetric encryption for added resilience.
    4.2 Ensure multiple layers of defense to mitigate attack vectors.

5. Scalability and Adaptability
    5.1 Design the framework to integrate easily with diverse applications (e.g., finance, healthcare, Social).
    5.2 Allow modular updates to adapt to new cryptographic developments. (newer theories other than lattice one)

6. SMS Chat Application
    6.1 Client should encrypt messages before sending them.
    6.2 Server should decrypt the received messages and send a response.
    6.3 The communication should be real-time and secure.
    6.4 Implement socket programming to establish connections between client and server.

5. Features to Codding Matrix

This matrix is currently a work in progress and will be updated as the project advances and additional tasks are completed.

Sr #	Feature Name	DSA Concept Used	Operation Performed	Complexity Analysis (Approximate)	No. of Variables & Objects Created	Functions Created	Line of Code Written
1	Quantum-Resistant Encryption	Vectors	Key Generation, Encryption	O(n)	3	3	40
2	Layered Security Architecture	String, XOR Operations	Symmetric Encryption & Decryption	O (n log n)	2	2	30
3	Secure Key Exchange Mechanism	Graphs, Hashing	Key Exchange, Verification	O(n^2)	8	4	120

6. Project Screenshots

The project is still in its development stages, and as such, the screenshots provided are not final. They will be updated in parallel with the ongoing progress of the project.

```
65  public:
71      int getSharedKey() {
73      }
74  };
75
76  class LayeredSecurity {
77  private:
78      string symmetricEncrypt(const string &message, const string &key) {
79          string encrypted = message;
80          for (size_t i = 0; i < message.size(); ++i) {
81              encrypted[i] = message[i] ^ key[i % key.size()];
82          }
83          return encrypted;
84      }
85
86      string symmetricDecrypt(const string &ciphertext, const string &key) {
87          return symmetricEncrypt(ciphertext, key);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR

Code

```
PS C:\Users\Asad Shayan\Documents\C++\icpc> cd "c:\Users\Asad Shayan\Documents\C++\icpc\" ; if ($?) { g++ p1.cpp -o p1 } ; if ($?) { .\p1 }
Encrypted Lattice Message: 16 79 48 43
Decrypted Lattice Message: 9 78 25 35
Shared Key (Simplified Key Exchange): 41
Encrypted Message: "►Ø↕►↑§
Decrypted Message: QuantumSafe
PS C:\Users\Asad Shayan\Documents\C++\icpc>
```

0 0 BLACKBOX Chat Add Logs Improve Code Share Code Link Search Error Ln 82, Col 1 Spaces: 4 UTF-8 CRLF {} C++ AI Code Chat Win32