Introducing QASCS – Quantum-Aware Secure Communication System

WHAT IS QUANTUM SECURITY?

PREPARING FOR THE POST-QUANTUM ERA

Andeta Zeqiri & Orgito Leka (pr0f3550r1)

LEARNING OBJECTIVES

Why quantum computing matters for security

Why current cryptography is at risk

What “quantum security” really means

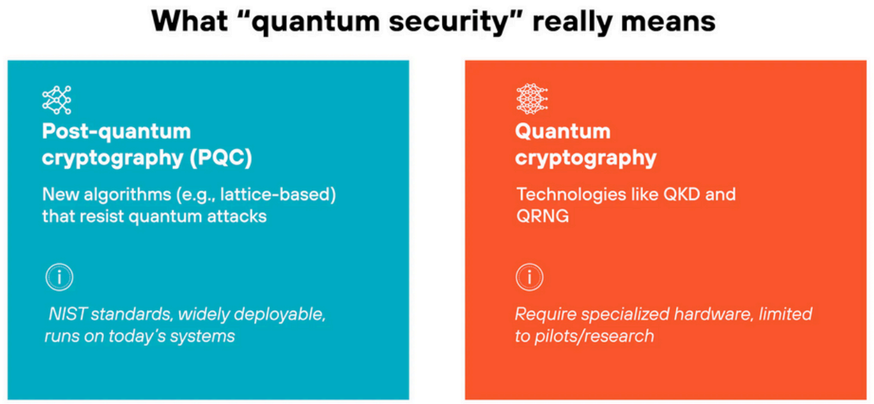
Two approaches: PQC vs Quantum Cryptography

Introducing QASCS

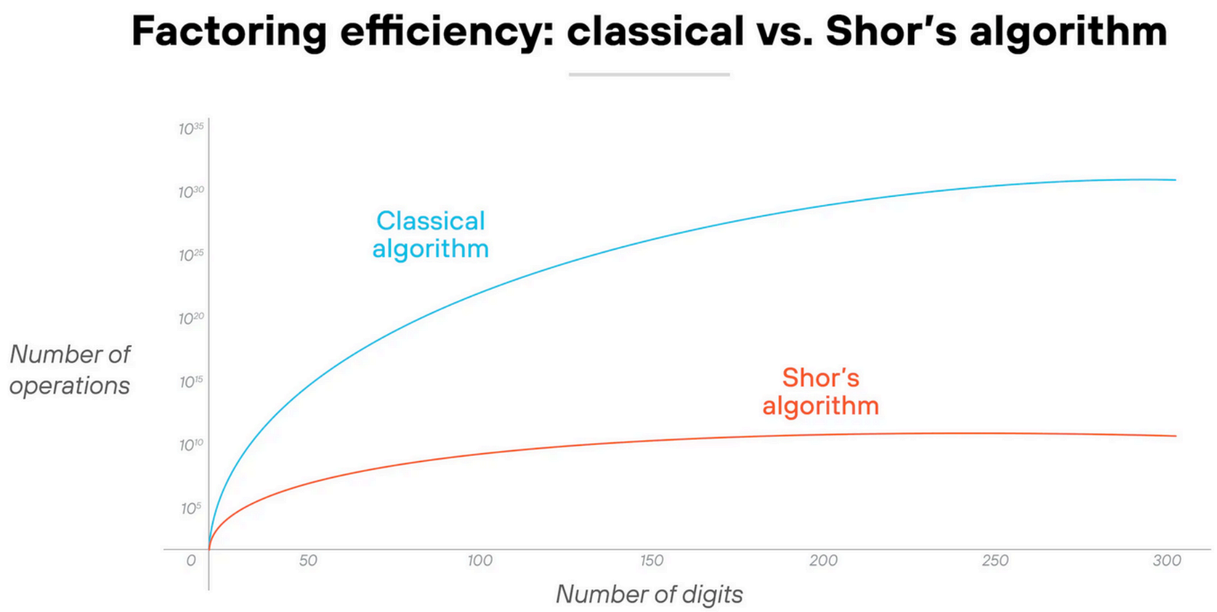
WHAT DOES THE INDUSTRY REALLY MEAN BY “QUANTUM SECURITY”?

“Quantum security” is often used loosely. In practice, it usually means post-quantum cryptography (PQC)—the move away from RSA and ECC, which could be broken by large quantum computers using algorithms like Shor’s.

PQC focuses on new cryptographic methods designed to resist both classical and quantum attacks. While quantum technologies like QKD and QRNG exist, standards bodies such as NIST are prioritizing PQC to protect today’s data from future quantum threats.



When people say “quantum security,” they usually mean PQC — the primary path forward for securing data against quantum computers.

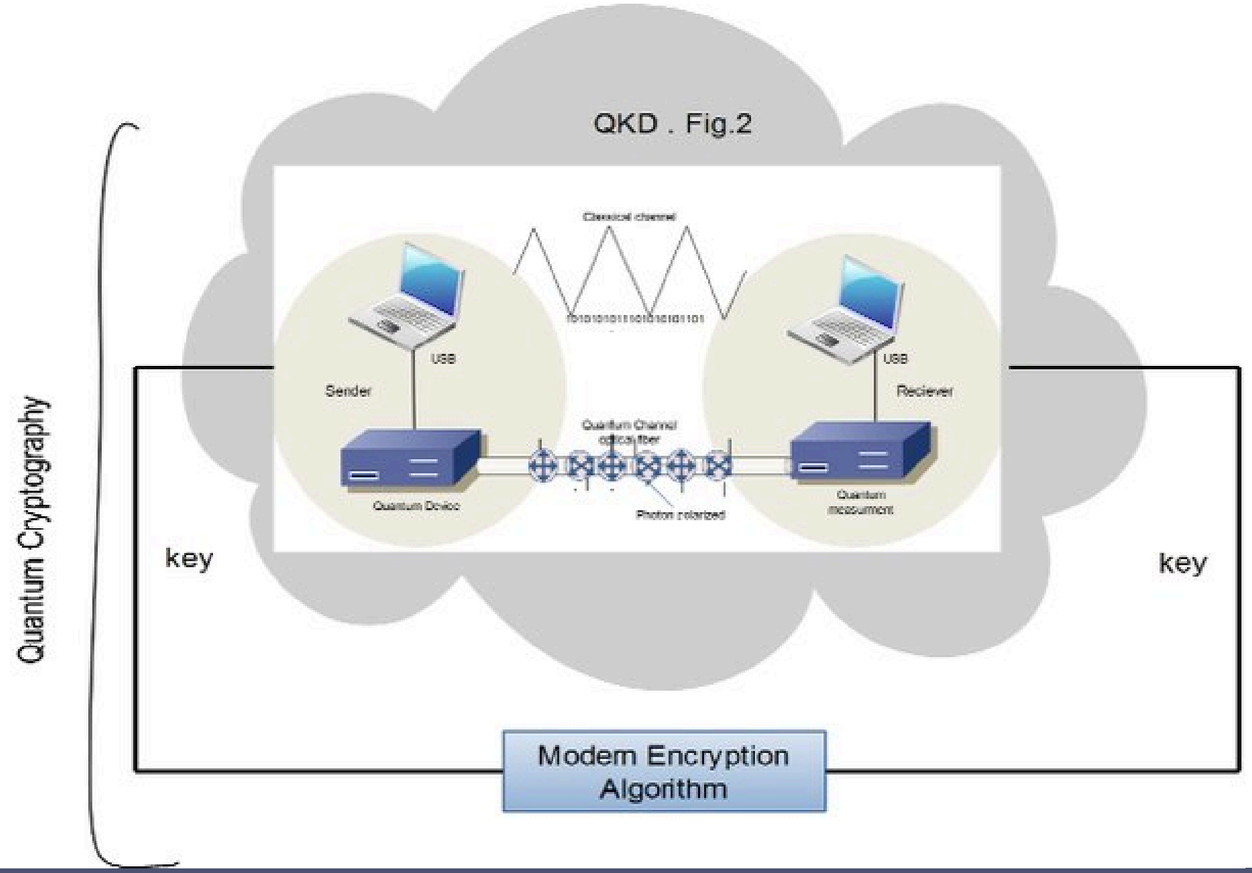


Shor's algorithm factors large numbers far faster than classical methods, threatening RSA and ECC once quantum computers scale.

A sufficiently powerful quantum computer could use Shor's algorithm to break RSA and ECC, immediately exposing encrypted traffic, digital signatures, and authentication systems, while symmetric encryption like AES is impacted differently by Grover's algorithm, which effectively halves key strength and speeds up brute-force attacks. Despite this, AES remains secure with larger keys—AES-256 is expected to withstand quantum attacks—making symmetric cryptography more resilient than RSA or ECC, though it still requires updated key management. The threat is not purely theoretical: attackers can already collect encrypted data and decrypt it later when quantum computers mature, a strategy known as harvest now, decrypt later, putting long-term sensitive information at risk.

WHY WON'T TODAY'S ENCRYPTION HOLD UP AGAINST QUANTUM COMPUTERS?

Quantum cryptography (also known as quantum encryption) refers to various cybersecurity methods for encrypting and transmitting secure data based on the naturally occurring and immutable laws of quantum mechanics.



QUANTUM CRYPTOGRAPHY

The Quantum Threat: Quantum Attack Models

Current Encryption Methods at Risk

Traditional encryption methods like RSA and ECC are fundamentally vulnerable to quantum computing due to their reliance on mathematical problems that quantum algorithms can solve efficiently. As quantum computers develop, they pose an imminent threat to data security across various sectors.

Timeline Estimates for Quantum Computers

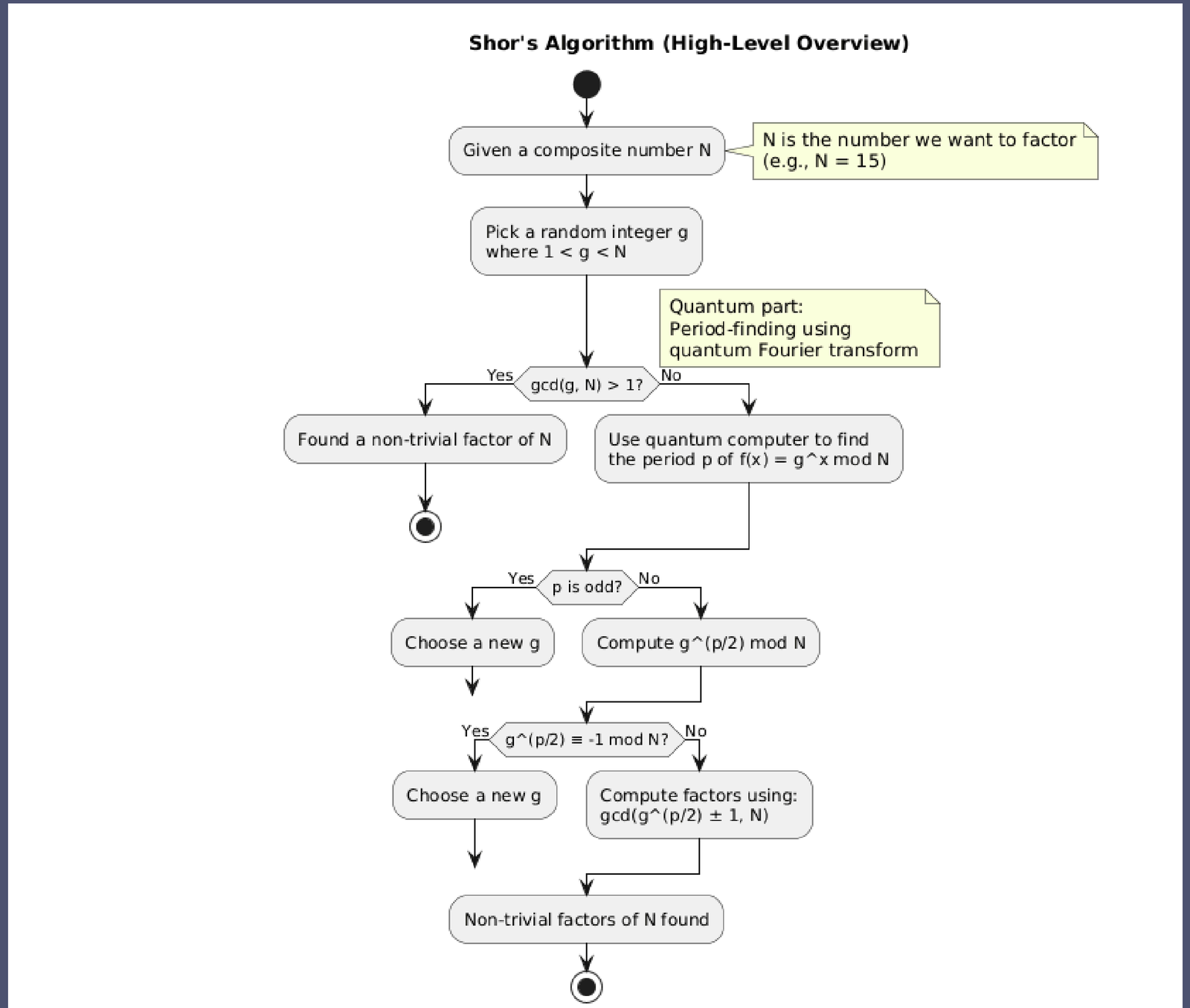
Experts estimate that large-scale quantum computers capable of breaking current encryption methods could be available within the next 5 to 10 years. This timeline raises urgent questions about the preparedness of industries relying on traditional cryptographic systems.

"Harvest Now, Decrypt Later" Attacks

"Harvest now, decrypt later" is a strategy where adversaries collect encrypted data today with the intention of decrypting it in the future when quantum computers become available. This tactic emphasizes the need for immediate action to protect sensitive information.

What is Shor's Algorithm?

Shor's Algorithm is a quantum algorithm that efficiently addresses the integer factorization problem, threatening the security of public-key cryptosystems such as RSA and ECC. The advancement of quantum computers could severely impact the confidentiality and integrity of communications protected by these methods.



An In-Depth Look at Quantum-Safe Communication Techniques and Architecture

This presentation introduces the Quantum-Aware Secure Communication System (QASCS), which aims to provide future-proof encrypted communication solutions in the evolving landscape of quantum computing. Explore how QASCS integrates quantum threat modeling with secure communication methods to prepare for the quantum era.

Python 3.8+

Ensure you have Python 3.8 or a newer version installed on your system. This is essential for compatibility with the QASCS implementation and its dependencies.

Virtual Environment Setup

Set up a virtual environment to isolate the project dependencies from other Python projects on your machine. This helps in managing packages and versions efficiently without conflicts.

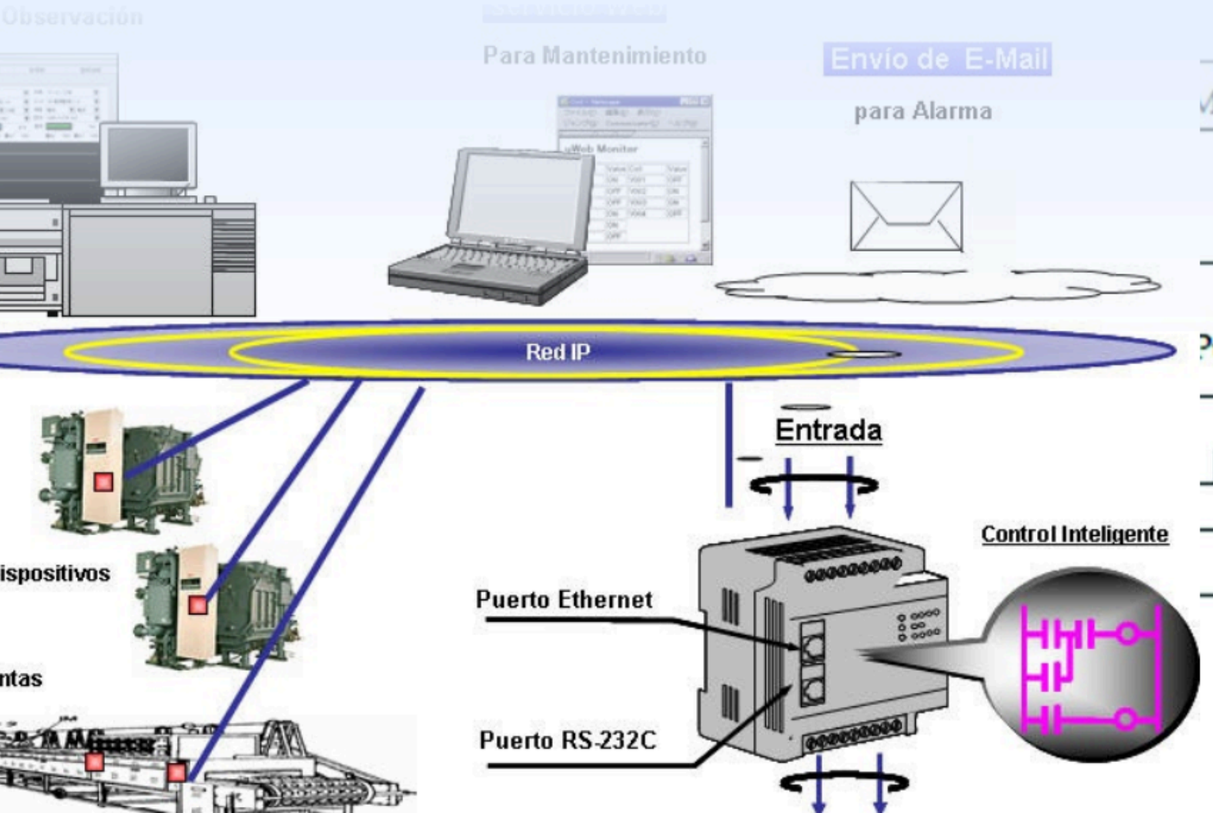
Dependencies Installation

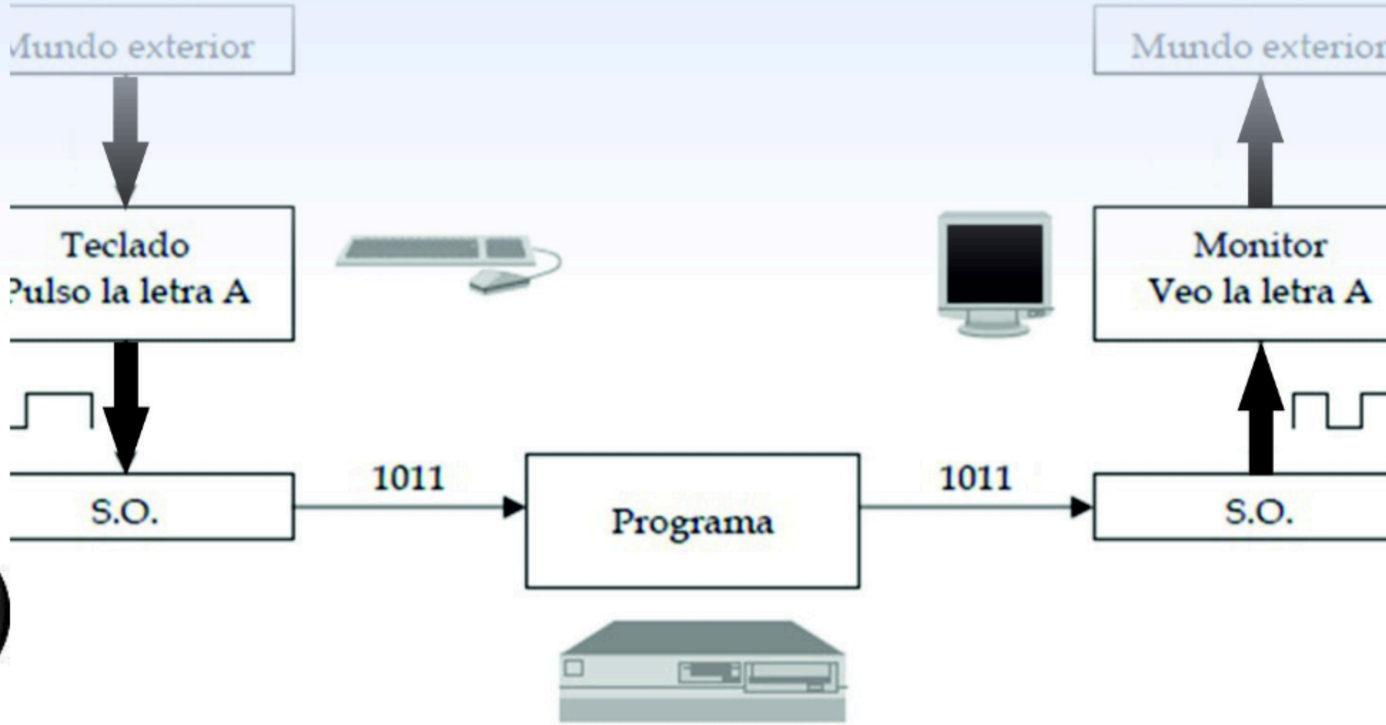
Install the necessary dependencies as outlined in the project requirements. This typically involves using pip to fetch and install packages that the QASCS project relies on to function correctly.

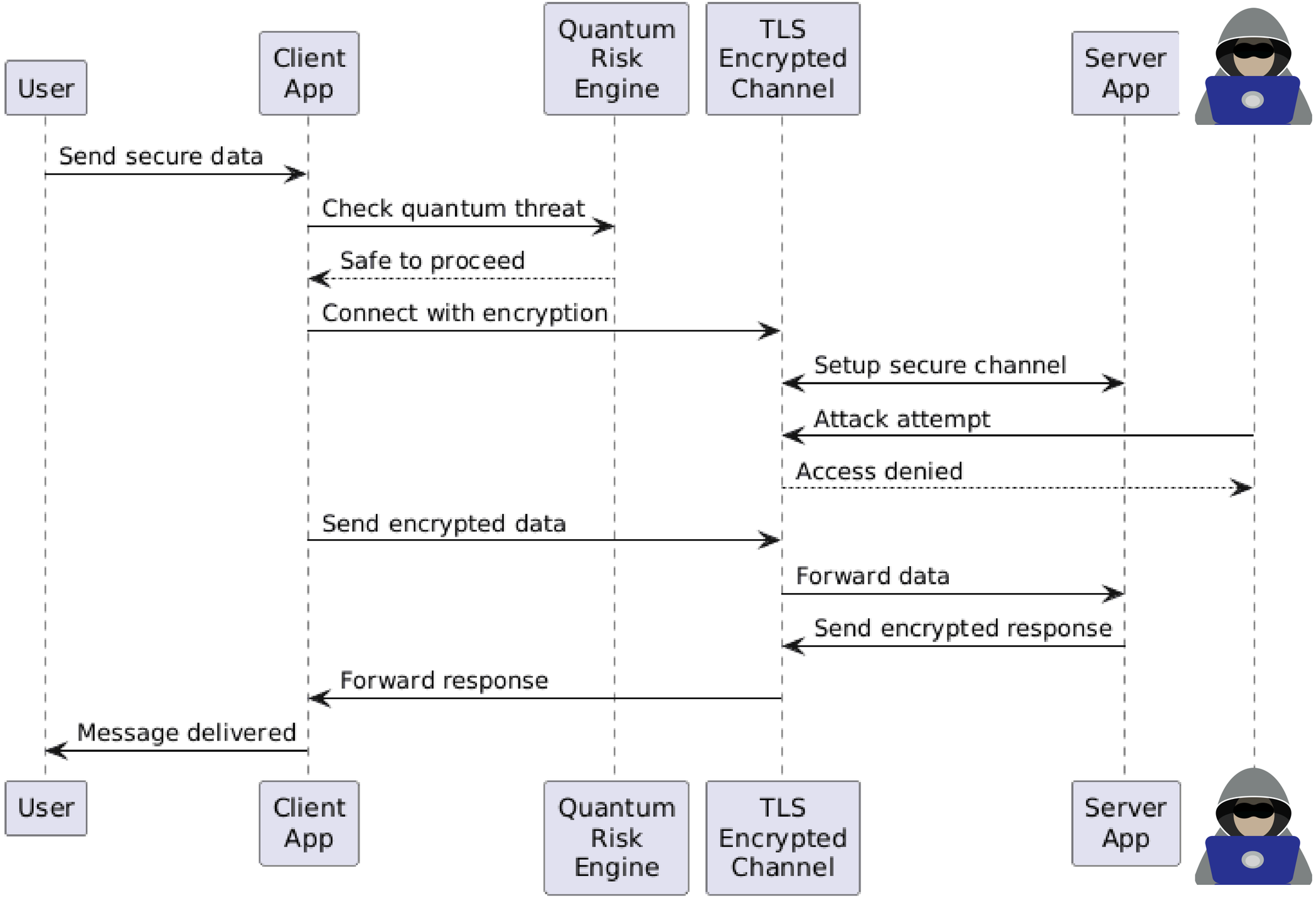
Getting Started - Quick Demo: Prerequisites

Technical Deep Dive: Client-Server Communication

This section delves into the fundamental aspects of client-server communication within the QASCS framework, focusing on the implementation of TLS and the message exchange processes. Understanding these components is crucial for grasping how secure communication is established and maintained in a quantum-aware environment.






User
Client App
Quantum Risk Engine
TLS Encrypted Channel
Server App
Send secure data
Check quantum threat
Safe to proceed
Connect with encryption
Setup secure channel
Attack attempt
Access denied
Send encrypted data
Forward data
Send encrypted response
Forward response
Message delivered
User
Client App
Quantum Risk Engine
TLS Encrypted Channel
Server App

Thank you for your attention! We welcome any questions you may have about the Quantum-Aware Secure Communication System (QASCS). For further inquiries or technical discussions, please feel free to reach out using the provided contact information or visit our GitHub repository.