Quantum-Safe Communication: Implementing BB84 QKD

This presentation explores the implementation of the BB84 Quantum Key Distribution protocol, a critical step towards secure communication in a quantum computing era.

The Looming Threat: Why Quantum Cryptography Matters

Problem Statement

Classical key-exchange protocols like Diffie-Hellman and RSA are vulnerable to attacks from future quantum computers. This poses a significant threat to current encryption standards.

Core Challenge

How can two parties establish a shared secret key with provable security, even against a quantum-powered eavesdropper?

Our Approach

We implement and demonstrate the BB84 quantum key distribution protocol using a simulated quantum backend. This showcases how basis randomness and qubit superposition prevent interception.

Our Quantum Development Environment

Python Ecosystem

Chosen for rapid prototyping and its rich scientific libraries.

Qiskit + AerSimulator

Qiskit builds quantum circuits; AerSimulator provides efficient, noise-free quantum simulation.

Matplotlib & Qiskit Visualization

Used for plotting histograms and basis-agreement charts.

Google Colab

A zero-install cloud notebook for easy sharing and collaboration.

BB84 Protocol: A High-Level Overview

1. **Alice Encodes**

   Generates random bits and bases, then encodes qubits.

2. **Qubits Traverse**

   Qubits travel through the quantum channel, where Eve might intercept.

3. **Bob Measures**

   Measures in random bases; they publicly compare bases.

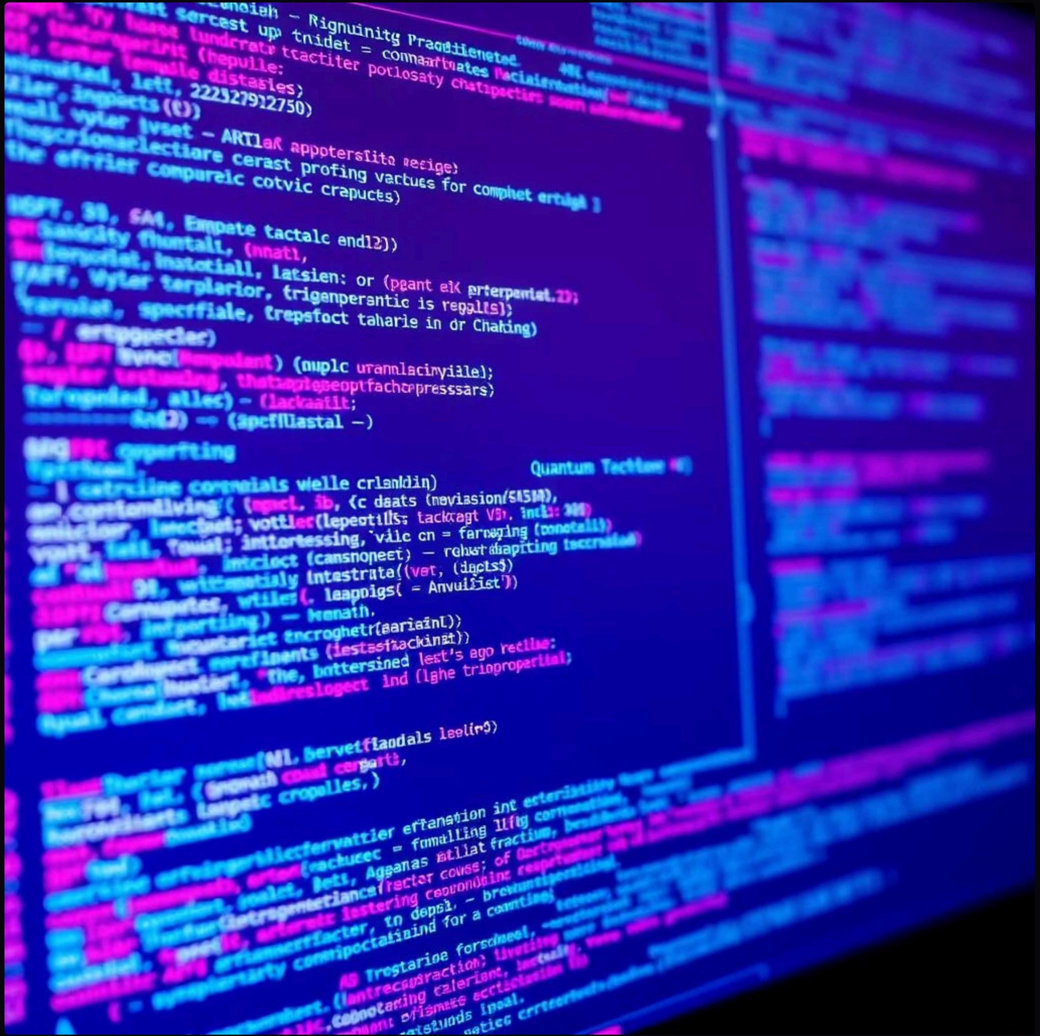
4. **Key Derivation**

   Only matching-basis positions form the shared secret key.

Behind The scenes : Code Structures

1. Utility Functions

- **Generate random bits(n):** Makes a list of n random 0s and 1s.
- **Generate random bases(n):** Picks n random “Z” or “X” labels.
- **Why it matters:** True randomness keeps eavesdroppers from guessing your qubits.



2. Quantum Operations

- **Encode bits(bits, bases):**
  - If bit = 1, flip the qubit.
  - If basis = “X”, put it into superposition (Hadamard gate).
- **Measure qubits(qubits, bases):**
  - If Bob’s basis is “X”, undo the superposition before measuring.
  - Measure in the standard (Z) basis to get a 0 or 1.
- **Why it matters:** Measuring in the wrong basis gives random results and reveals tampering

3. Protocol Logic

- Alice makes bits & bases, encodes qubits, and sends them.
- Bob picks his bases and measures each qubit.
- They publicly share their bases (not the bit values).
- They keep only bits where bases matched—this is the shared key.

Why it matters: Any eavesdropper disturbs qubits, creating errors that Alice & Bob can spot.

4. Visualization

- **Bit Histogram:** Shows how many 0s vs. 1s ended up in the key.

Why it matters:
A balanced 0/1 split confirms good randomness.

Project Demonstration and Key Metrics

70

Qubits Sent

Total qubits transmitted in the simulation.

~35

Sifted Key Length

Approximate length of the final secure key.

~25%

BER (Eve Present)

Bit-error rate when an eavesdropper intervenes.

~0%

BER (Clean)

Bit-error rate without any eavesdropping.

Overcoming Quantum Implementation Challenges

1

Understanding Quantum Gates

Mastered Hadamard and Pauli-X gates, basis impact on measurement.

2

Optimizing Circuit Simulation

Overcame slow simulation by reusing the AerSimulator instance.

3

Reproducible Randomness

Ensured testing consistency by optionally setting random.seed().

4

Debugging Quantum Circuits

Used qc.draw() and histograms for circuit correctness verification.

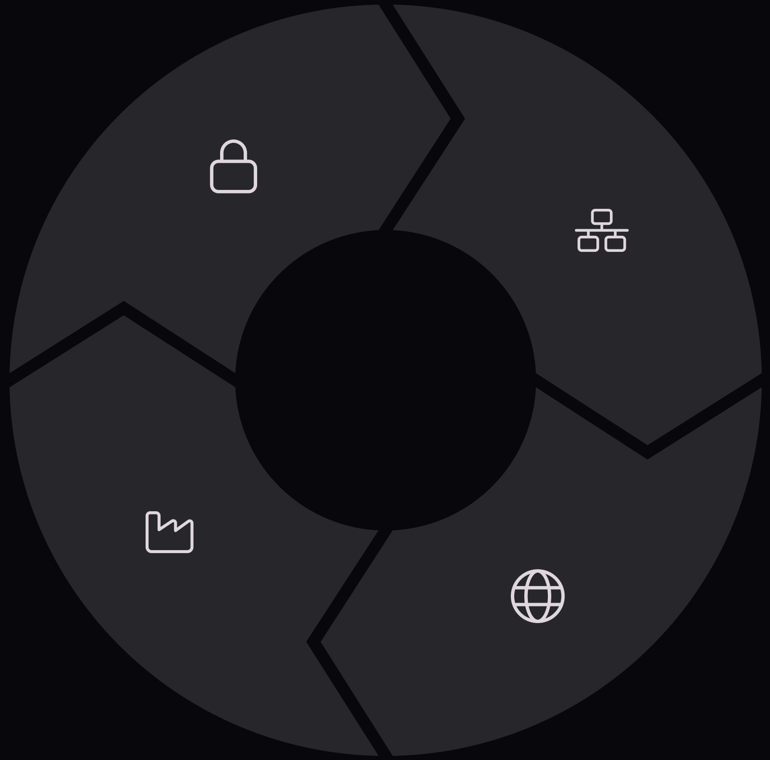
Future of Secure Communication

Optimization Engines

Speed up complex scheduling, routing, and resource-allocation problems (logistics, supply-chain, cloud resource scaling)

Quantum-Safe Cryptography

Develop and test post-quantum (quantum-resistant) encryption algorithms to protect data against future quantum attacks.



Quantum Simulation

Model molecules, materials, and chemical reactions far more efficiently—accelerating drug discovery, battery design, and advanced materials R&D.

Quantum-Enhanced Machine Learning

Explore hybrid models (classical + quantum) for faster feature selection, clustering, and solving certain linear algebra tasks underpinning AI.

The BB84 implementation is a foundational step towards building truly quantum-secure communication channels. The journey continues with further research and development in quantum-safe algorithms and their real-world applications.