Securing the Future: BB84 Quantum Key Distribution and Post-Quantum Security

PROBLEM:

Shor's algorithm running on a sufficiently large quantum computer can efficiently break classical public-key cryptography (like RSA and ECC/Ed25519).

THE SOLUTION:

Quantum Key Distribution (QKD): A system that uses quantum physics to create and share a secret key

BB84 Quantum Key Distribution Process

Alice Chooses Secret Bit

Alice selects a random bit for each qubit.

Alice Chooses Basis

Alice selects a random basis for each qubit.

Bob Chooses Basis

Bob selects a random basis for each incoming qubit.

Bob Measures Qubit

Bob measures each qubit using his chosen basis.

Bases Match

Bob gets the correct bit if bases match.

Bases Mismatch

Bob gets a random bit if bases mismatch.

Alice and Bob Compare Bases

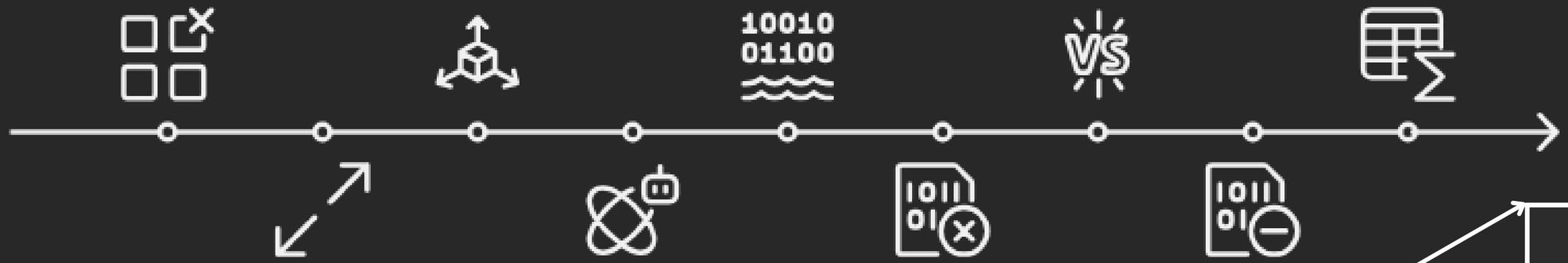
Alice and Bob publicly compare their chosen bases.

Discard Mismatched Bits

Alice and Bob discard bits where bases did not match.

Form Sifted Key

Alice and Bob form the Sifted Key from matching bits.



- Alice and Bob choose different basis.
- High Natural Noise
- Eavesdropping

Simulating Quantum Security: Our BB84 Implementation

To validate the BB84 protocol's efficacy, we implemented an end-to-end simulation using Qiskit, IBM's open-source quantum computing framework. This allowed us to model the behavior of qubits and the protocol's mechanics under various conditions.

We ran simulations on the Qiskit Aer Simulator, treating the encoded photon polarizations as qubits. This approach provides a practical environment for analyzing the protocol's error rates and the impact of simulated eavesdropping attacks.

The simulation confirmed the protocol's ability to detect eavesdropping by observing the increased quantum bit error rate (QBER), a direct consequence of the No-Cloning Theorem.

BB84 vs. Classical Cryptography: A Resilience Comparison

Parameter	BB84 Quantum Key Distribution	RSA	Ed25519
Foundational Security	**No-Cloning Theorem**	**Computational Hardness** (Integer Factorization)	**Computational Hardness** (Elliptic Curve Discrete Logarithm)
Quantum Resilience	Information-Theoretically Secure (Immune). Future-proof security guaranteed by physical laws.	Vulnerable. Easily broken by **Shor's Algorithm**.	Vulnerable. Broken by a quantum computer running **Shor's Algorithm**.
Primary Function	**Key Distribution Only**. Generates and shares a secret key.	Used for **Encryption, Key Distribution, and Digital Signatures.**	Primarily used for **Digital Signatures** and Key Exchange.
Time Complexity	O(n) - Linear in key length	O(exp(n^(1/3))) - Sub-exponential to break	O(exp(n^(1/2))) - Exponential to break

Eavesdropper Detection Probability

In the BB84 protocol, Eve's presence is detected by an elevated Quantum Bit Error Rate (QBER). Assuming Eve randomly guesses the correct encoding/measurement basis 50% of the time, she introduces an error that is not caught during sifting 75% of the time (3/4).

0.75	$(3/4)^n$	$1-(3/4)^n$
Undetected Error Probability	**Non-Detection After n Bits**	**Detection Probability P(n)**
Probability of Eve introducing an error that goes initially unnoticed.	The chance of Eve's presence not being detected across n checked bits.	The final probability of detecting the eavesdropper after checking n bits.

The security of the BB84 protocol relies on the rapid increase in detection probability as more bits are publicly compared:

- n=10 checked bits: P ≈ 0.94. A small sample bit string yields a 94% chance of detection.
- n=25 checked bits: P ≈ 0.99. Increasing the sample bit string provides near-certain detection.