BB84 Quantum Key Distribution prototype

Background and Motivation

As our classical public-key cryptography (RSA, ECC) is threatened by advances in quantum computing, we have looked to novel methods of encryption to solve this dilemma. Quantum offers us new ways to secure information, one of which is Quantum Key Distribution. Unlike factorization-based schemes, QKD protocols use the unique properties of quantum computing to generate shared secret keys that eavesdroppers cannot intercept undetected.

In the 1980s, Charles Bennett and Gilles Brassard created BB84 (Bennett & Brassard, 1984). In BB84, Alice and Bob exchange qubits in random bases, measure them, then sift and process measurement results to agree on a secret key. Any attempt by an eavesdropper (Eve) to measure the qubits introduces detectable errors.

Getting Started

Using Qiskit, you will build a fully functional BB84 Quantum Key Distribution prototype that runs on real or simulated quantum hardware. The goal is demonstrating how two parties can establish a shared secret key with information-theoretic security. Your implementation should include end-to-end key generation, measurement and sifting procedures, and error-rate estimation to detect any eavesdropping. You'll collect and plot key metrics and illustrate how these values shift when you introduce noise. By comparing performance under different conditions, you will showcase the practical feasibility of BB84 on today's quantum platforms and highlight the utility of the approach.

After you validate the implementation, create an informative presentation that teaches the fundamentals of QKD and places it in the broader context of post-quantum encryption. Make sure you compare BB84 to at least one other emerging approach, discussing how these algorithms resist quantum attacks.