Asymmetric Cloning to Eavesdrop on BB84 Protocol

Exposing fault-lines in the BB84 protocol in practice.

Elizabeth G. Campolongo, Ph. D.

Brian Pigott, Ph. D.

Hardik Routray

Mentor: Alex Khan

Overview

What is BB84?

The BB84 Protocol is a method of Quantum Key Distribution (QKD) that—theoretically—has perfect security. However, actualization of quantum computers with current technology does not guarantee the conditions necessary for such a level of security. The fidelity of the GPU (noise inherent in the system) reduces the accuracy of transmissions.

Context

This reduction can be leveraged by an eavesdropper (“Eve”) in intercepting bitstreams sent between two parties (eg., “Alice” sends bits to “Bob”). Due to the inherent loss expected in the system, Eve may obtain usable information in a manner nearly undetectable.

Coming up…

In the following slides we will demonstrate how this is possible and the tradeoff between information gain and detectability.

Understanding the BB84 Protocol

Step 1

Alice generates a random string of bits (“0” and “1”), which she then encodes through random choice of the X or Y basis to send to Bob through a quantum channel.

Step 2

Bob also generates a random string of bases (“X” and “Y”) with which he measures the qubits sent by Alice.

Step 3

Over a classical channel, Alice shares the bases with which she encoded the bitstring. Alice and Bob keep only the bits measured with the same bases.

Step 4

Alice and Bob then compare a subset of these agreed upon bits to ensure their accuracy at a high-enough level.

The problem

Goal

Eve wants to maximize the information gained from eavesdropping while minimizing the chance of detection.

Context

Alice and Bob are communicating over a public lossy channel—noise is expected.

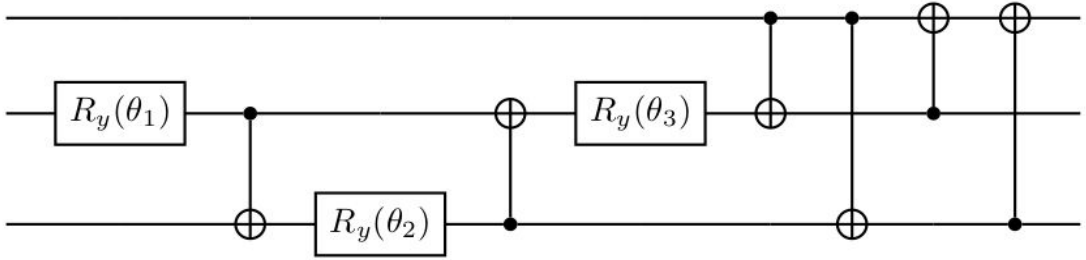
Problem statement

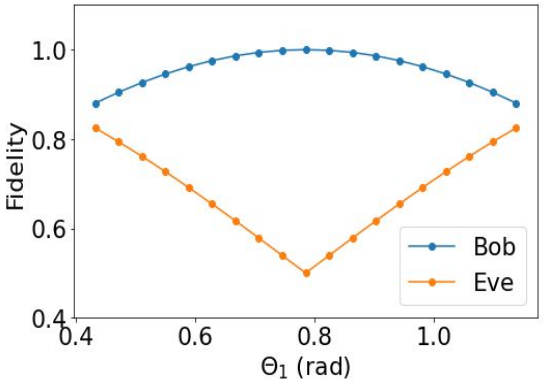
The no-cloning theorem tells us that a qubit (quantum bit) cannot be perfectly duplicated, meaning any clone will necessarily have some nonzero probability of being measured incorrectly despite using the proper basis.

The Intercept-Resend Attack

- Eve measures each “flying” qubit, randomly choosing either the X or Y basis in which to measure.
- If Eve guesses the correct basis, she gets the information about Alice’s bit.
- If Eve guesses the incorrect basis, she doesn’t get any information about Alice’s bit AND she introduces an error in Bob’s measurement.
  - If Bob measures in a different basis than what Alice sent, he won’t notice this error.
  - But, if Bob measures in the same basis as Alice, he will notice this error.
- On average, Eve will guess the correct basis about 50% of the time, thus obtaining Alice’s bit. However, Bob will incur an error rate of about 25% in the qubits that Eve interferes with.

Phase-Covariant Cloning

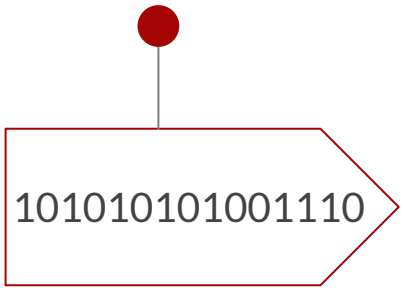


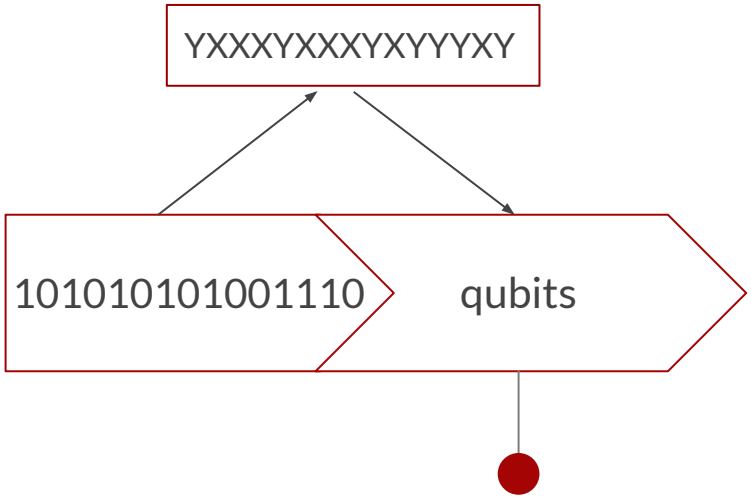


- Eve opts to make a clone – an imperfect copy – of the flying qubit. She retains the second qubit in the circuit to measure later on.
- The clone is phase covariant: fidelity of the clone is independent of the relative phase.
- The flying qubit is represented by the top wire; the second and third qubits are contained in Eve's ancilla.
- Three angles (θ_1, θ_2, and θ_3) are involved in the cloning circuit.
  - For the clones to be phase-covariant, θ_3 can be written as a function of θ_1 and θ_2.
  - The value of θ_1 determines the fidelity of Bob's qubit.
  - We choose the value θ_2 of to optimize the fidelity of Eve's clone; thus θ_2 depends on θ_1.
- If Eve chooses θ_1 close to $\pi/4$, Bob's clone is nearly perfect.
- The fidelity of Eve's clone never falls below 0.5, meaning her error rate is at most 0.5. That's better than guessing the basis!

Implementation

Alice generates a random string of 0’s and 1’s to establish a new secret key with Bob.

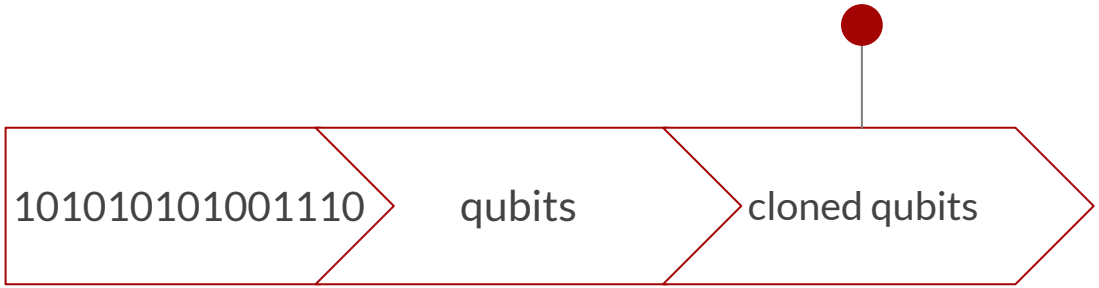


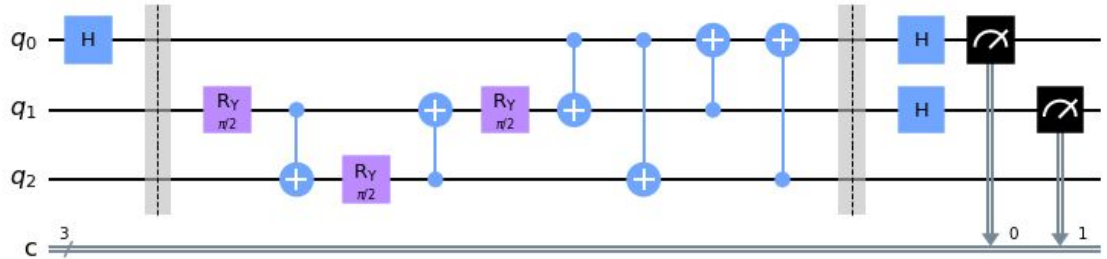


Alice randomly chooses a basis (X or Y) to encode each bit of the string and sends the resulting qubits over the unsecured quantum channel.

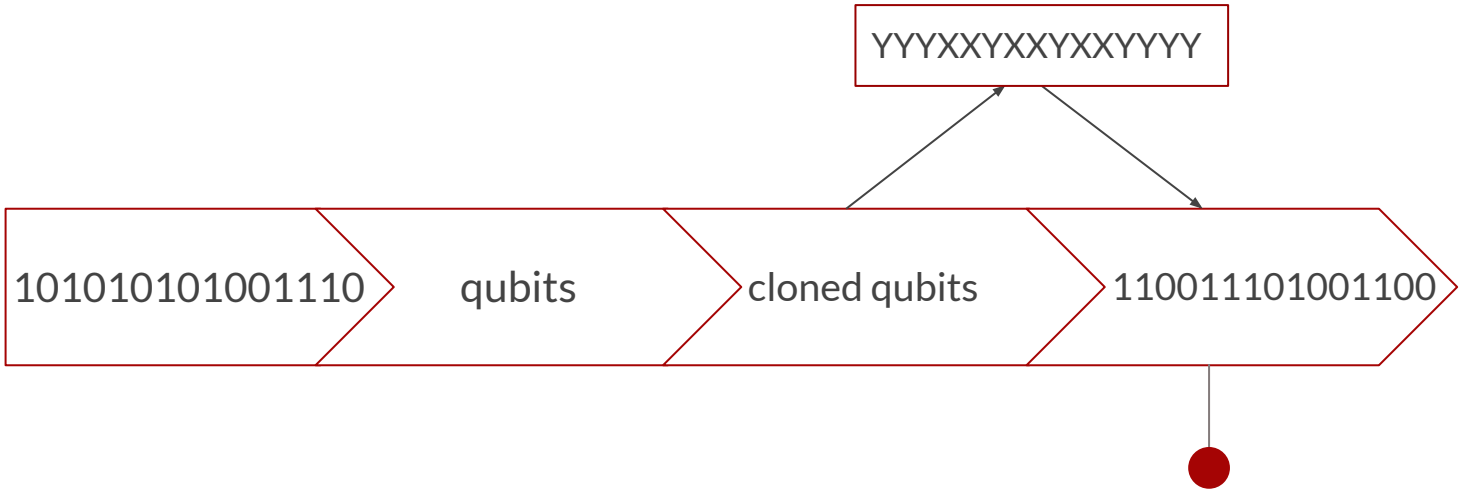
Eve intercepts the qubitstring and clones each qubit asymmetrically with her chosen θ_1.

She sends the better clone to Bob and keeps the lesser for herself.





Bob randomly chooses bases to measure the [cloned] qubits he receives.



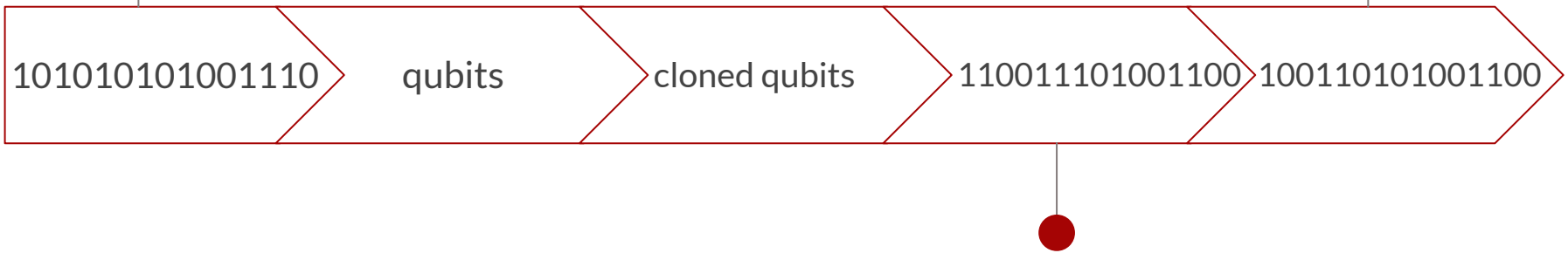
Bob measures the bit string with his random string of X and Y bases.

He has a 50% chance of measuring the proper output from the qubits he measures with the opposite basis of Alice.

Alice and Bob share their measurement basis strings to select their secret key.

Eve measures her clones based on this information.

Alice's Message



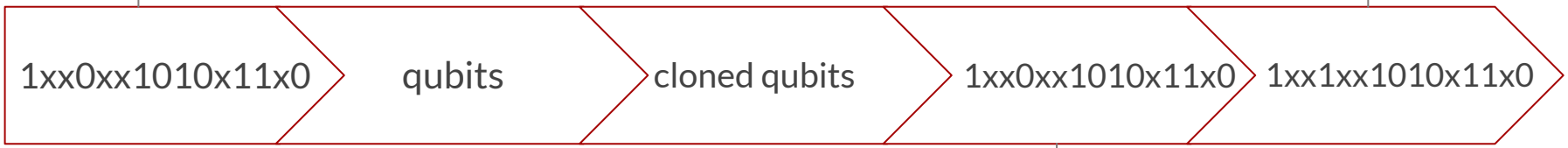
Bob's Measurement

Alice Bases:

YXXXYXXXYXYYYXY

Alice's Key

Eve's Copy of the Key



Bob's Key

Bob's Bases:

YYYXXYXXYXXYYYY

Example Process: Conclusion

Alice and Bob's joint key: 101010110

Eve's copy of the key: 111010110

The Quantum Bit Error Rate (QBER) for Bob: 0.0

The Quantum Bit Error Rate (QBER) for Eve: 0.1111

Alice and Bob's strings match, so the eavesdropping goes undetected.

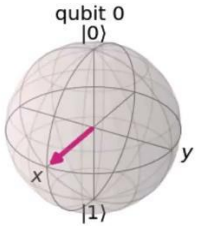
BB84 protocol with phase covariant cloning

Step 1: Alice performs encoding

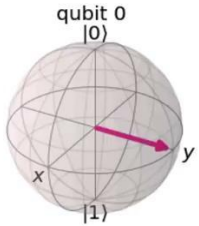
Input the bitstring length (accepts only positive integers) to generate a random bitstring alongwith randomly generated bases for Alice

Only equatorial bases are used instead of Z basis as in intercept-resend BB84 attack

X basis



Y basis



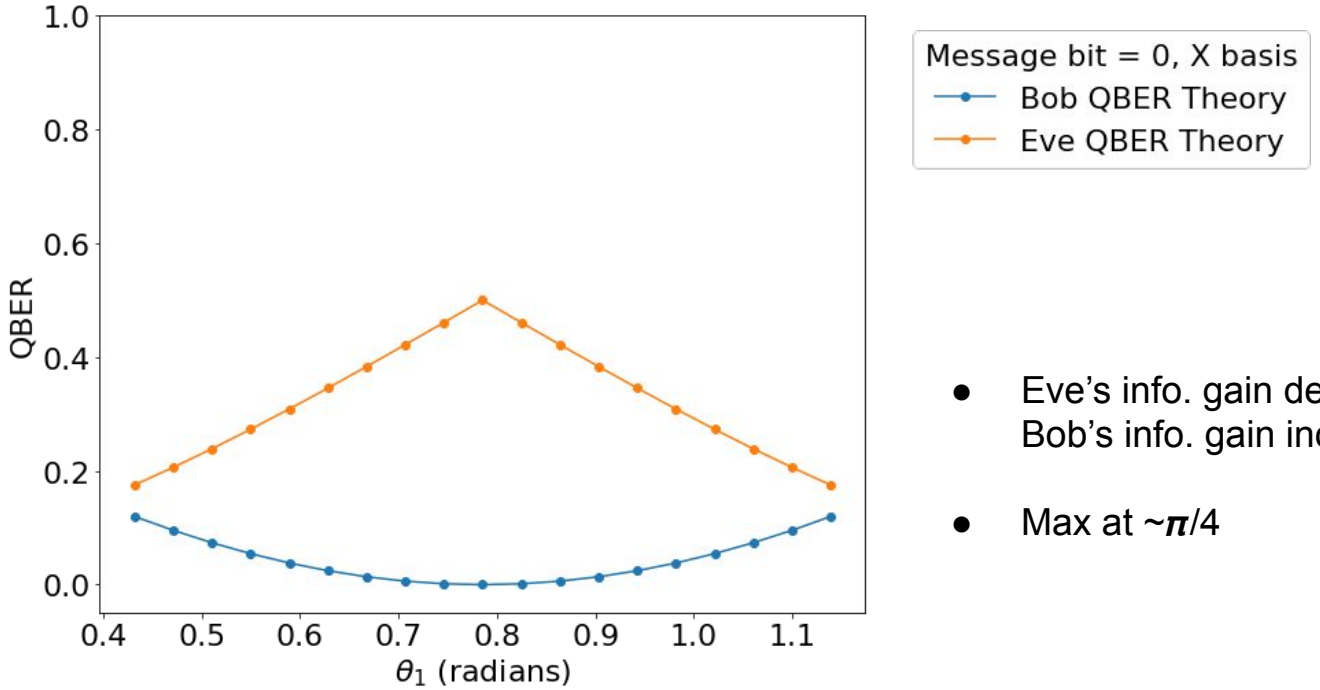
length 5

Alice bits	Alice bases
0	Y
1	X
0	X
1	X
1	X

Try it for yourself on our webapp: BB84Cloner

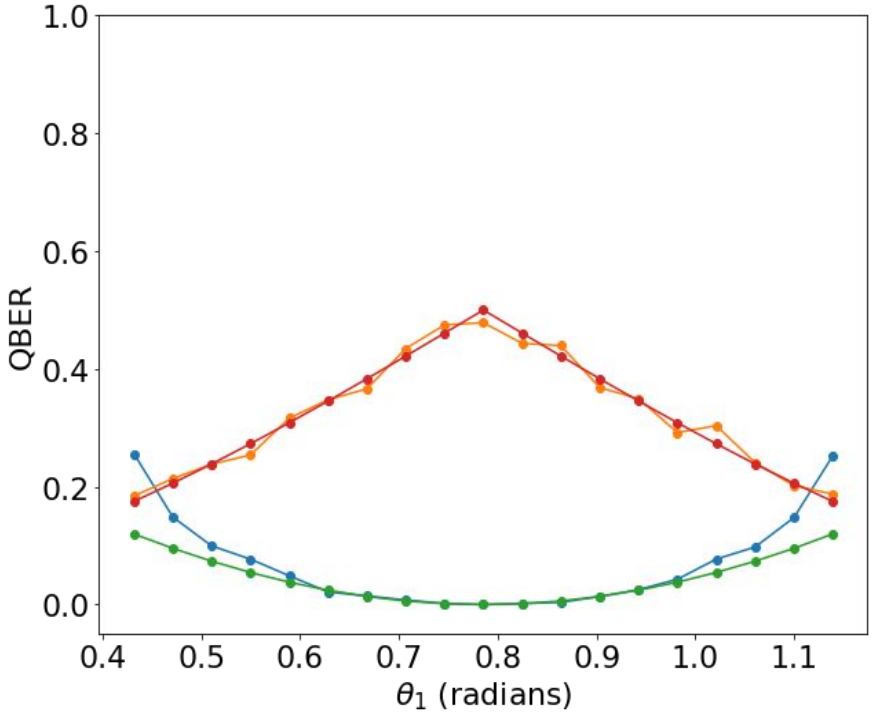
See our work on GitHub: QuForce BB84 Project

Result: Theoretical Expectation



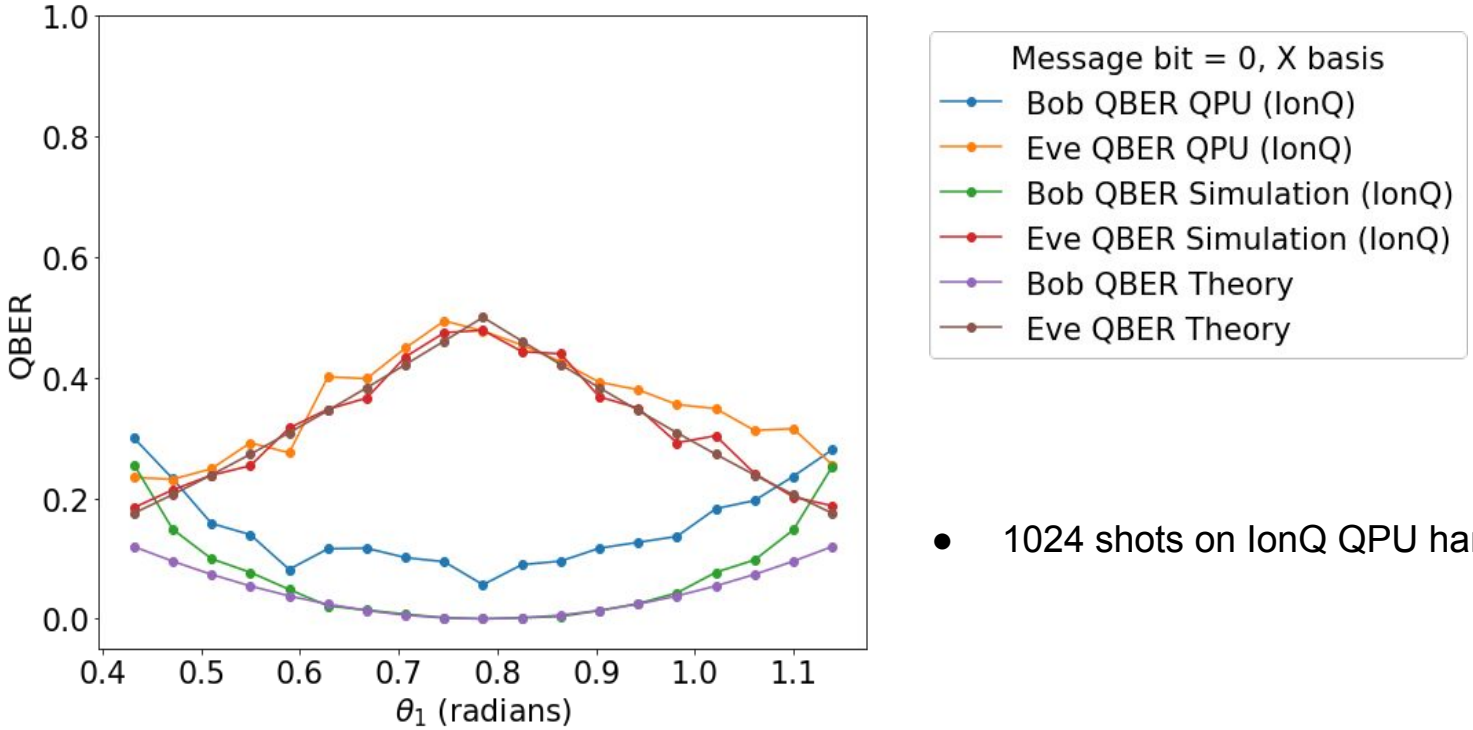
- Eve's info. gain decreases as Bob's info. gain increases
- Max at ~π/4

Result: Simulation



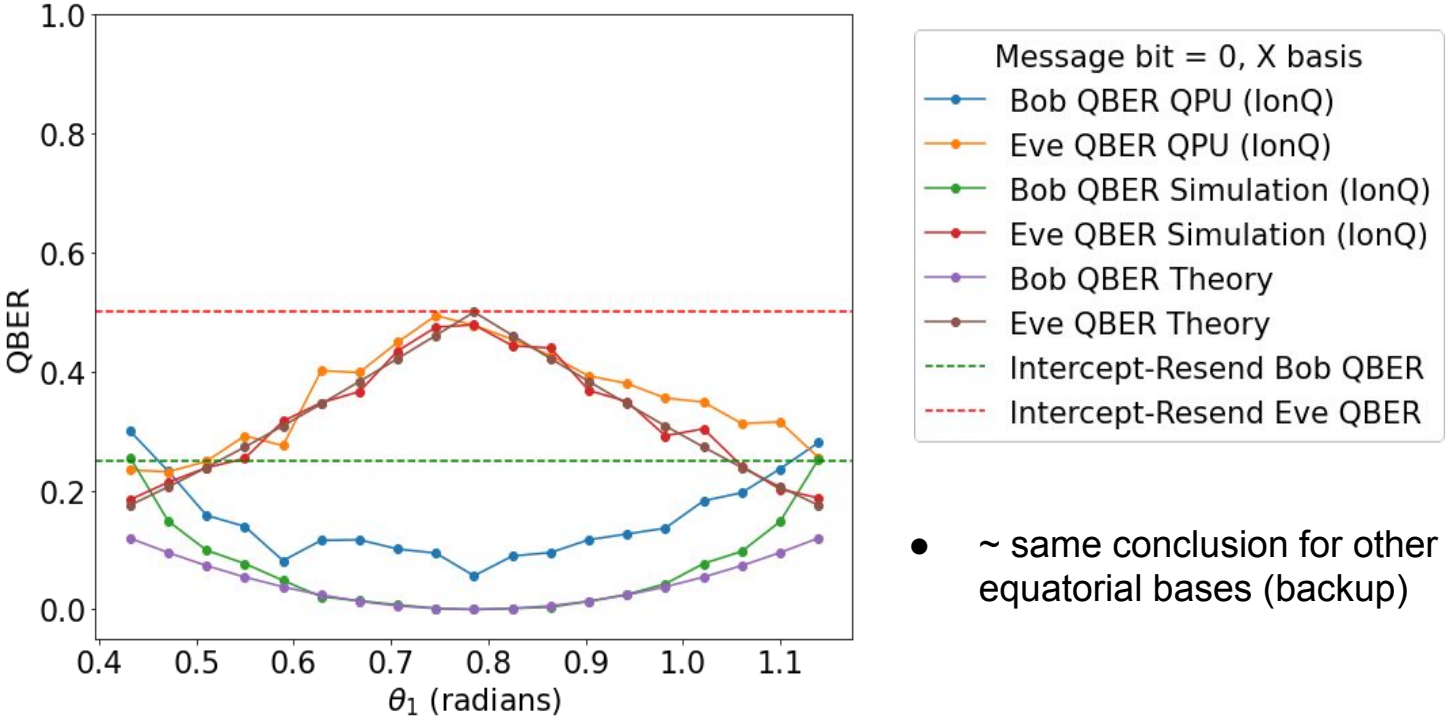
- 1024 shots on IonQ simulator

Result: IonQ Implementation



- 1024 shots on IonQ QPU harmony

Cloning vs. Intercept-Resend



- ~ same conclusion for other equatorial bases (backup)

Future Direction:
IonQ Implementation in Native Gates

- This is not the circuit IonQ actually implements, but a transpiled version which has far more gates, and thus more noise.
- Apply the Native Gates directly to gain a better understanding of the residual noise.

Impact on the Team

We learned…

- about quantum key distribution methods and their implementation on quantum computers.
- how to read and design a quantum circuit, then interpret its output.
- about the impact on programming of different types of quantum computers, eg. IBM's superconducting vs. IonQ's iontrap.
    - This requires a transpiling step to the appropriate Native Gates when implementing circuits.

We found a community of Quantum enthusiasts—Thank you, QuForce!

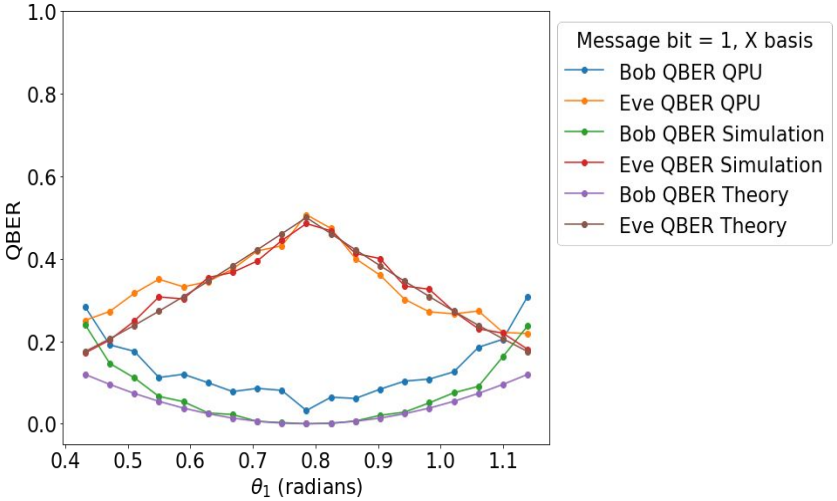
References

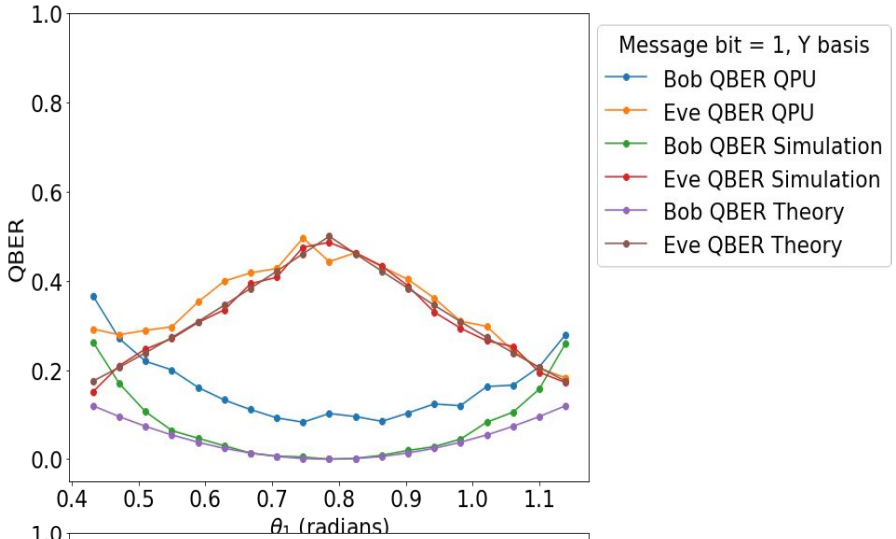
1. C.H. Bennett and G. Brassard. “Quantum Cryptography: Public Key Distribution and Coin Tossing.” *International Conference on Computers, Systems & Signal Processing*, Bangalore, India, December 1984.
2. V. Bužek, S. L. Braunstein, M. Hillery, and D. Bruß. “Quantum copying: A network.” *Phys. Rev. A*, 56:3446–3452, Nov 1997.
3. H. Fan, K. Matsumoto, X. Wang, and M. Wadati. “Quantum cloning machines for equatorial qubits.” *Phys. Rev. A*, 65:012304, Dec 2001.
4. D. Maslov, “Basic circuit compilation techniques for an ion-trap quantum machine.” *New Journal of Physics*, 19(2), 2017.
5. A.T. Rezakhani, S. Siadatnejad, and A.H. Ghaderi. “Separability in asymmetric phase-covariant cloning.” *Physics Letters A*, 336(4):278–289, 2005.

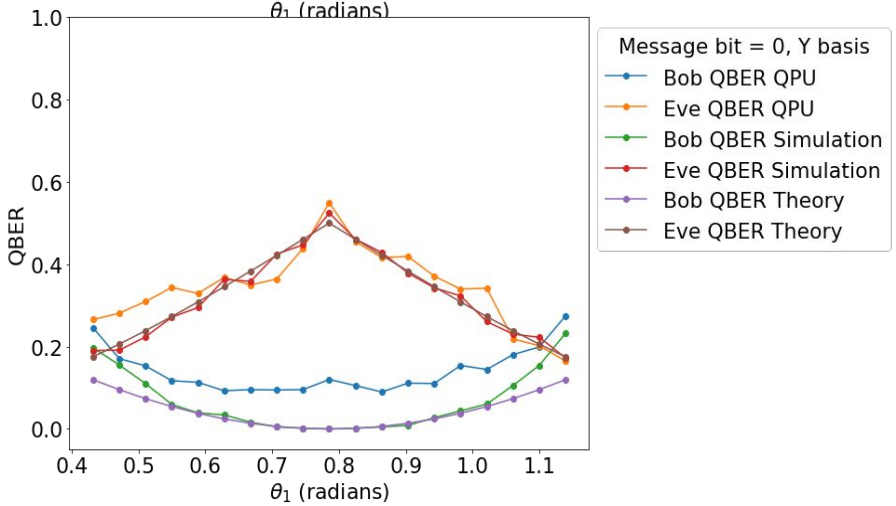
Thank you!

Try it for yourself on our webapp: BB84Cloner	See our work on GitHub: QuForce BB84 Project

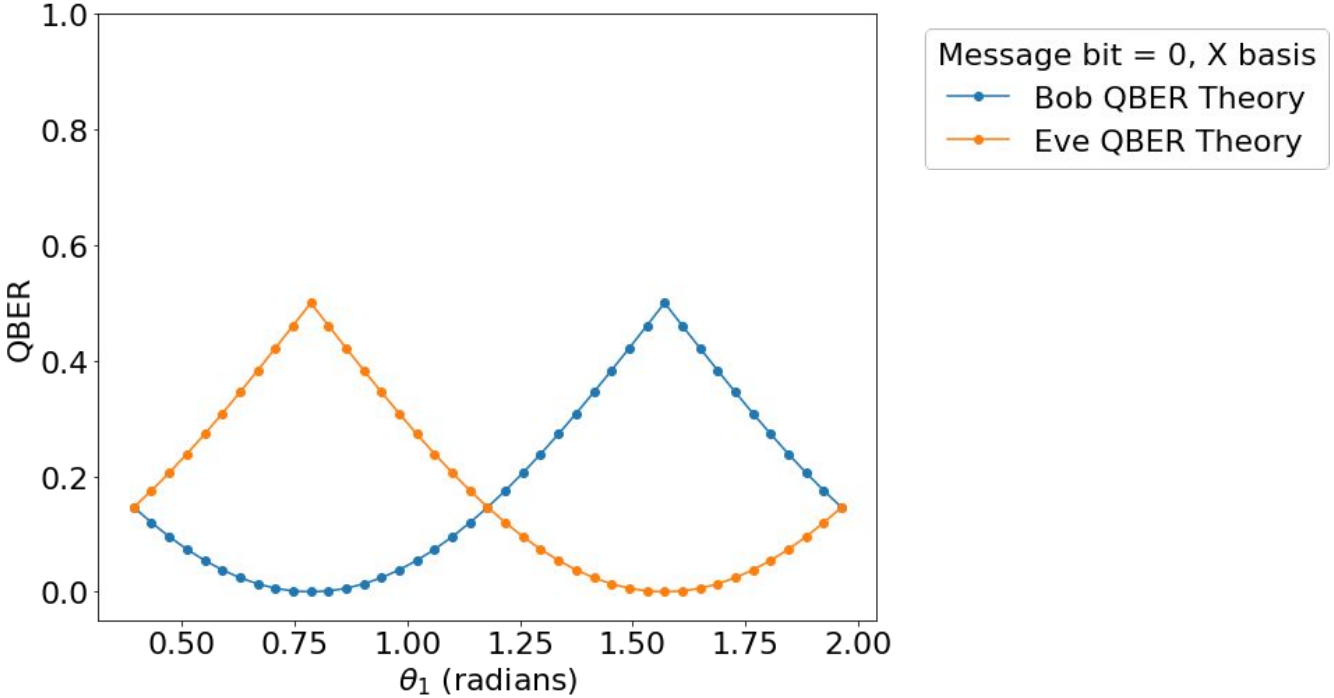
Backup







QBER results inside the nominal range
(for bits, bases not shown in main slides)



QBER results for outside the nominal range