© 2023 The Coding School

Day 3 Summary of Key Concepts

Lecture: Quantum Algorithms and Protocols

Summary: In lecture today, we learned about the basis of quantum computing applications - algorithms and protocols. We answered the following questions:

- What is the difference between an algorithm and a protocol, both classically and in quantum?
- What is NISQ?
- What is QKD? Why is it important? Where is the QKD research currently?
- What is BB84?

Algorithms vs. Protocols

Algorithms:
A specific procedure for solving a computational problem. Algorithms are like recipes - they outline the steps required to compute/perform something.

Examples of classical algorithms: searching, sorting, ranking.

Protocol
A set of standard rules that allow electronic devices to communicate with each other. Protocols are like contracts - agreements between several devices to follow specific standards of communication.

Examples of classical protocols: HTTP (P stands for Protocol) - a protocol any device connected to Internet uses.

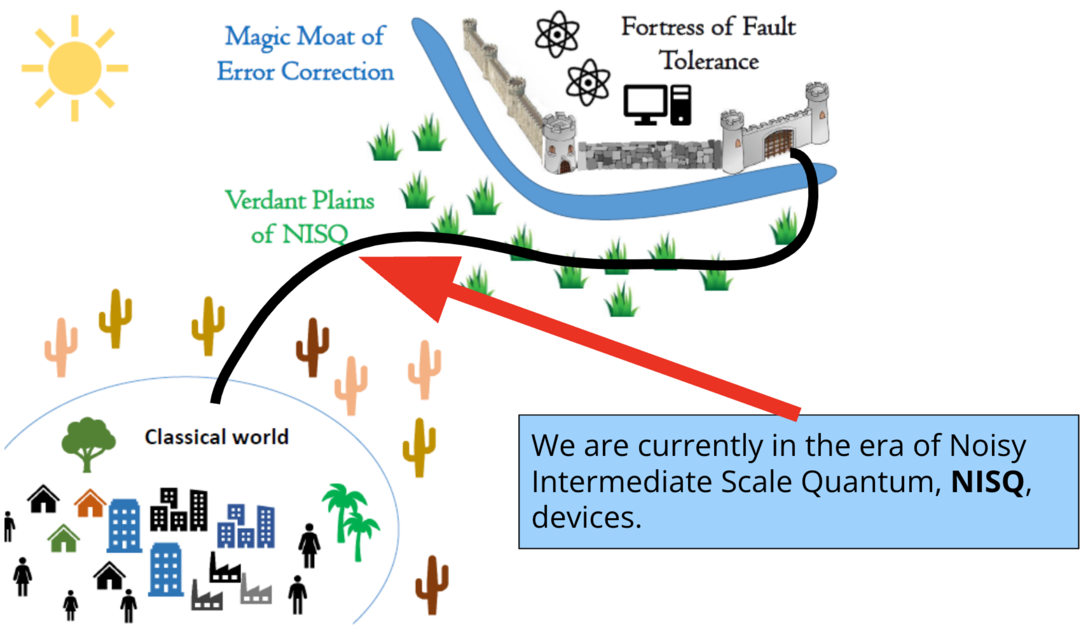
Quantum Algorithms

- All quantum algorithms boil down to **quantum circuits**, but **not every quantum circuit is a useful quantum algorithm**.

© 2023 The Coding School

- Quantum algorithms won't be useful unless they do **something classical algorithms cannot** that also does something of value.
- Quantum algorithms need to leverage our **three quantum resources** and do something clever with them:
    - Superposition
    - Entanglement
    - Interference

Where are we in our search for useful quantum algorithms?



NISQ
An important goal set by scientists in the NISQ era is to demonstrate quantum algorithms that can do something we could not do classically, even if the end result isn't useful.

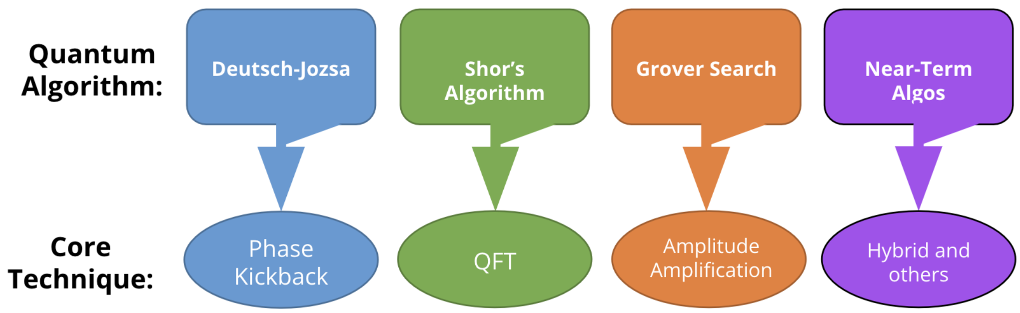
Near-term Algorithms are algorithms that we hope will start being useful soon even with only small, faulty devices available.

- These tend to be classical-quantum hybrid approaches where we use the best of classical and quantum computing together to do something neither one could do alone.

© 2023 The Coding School

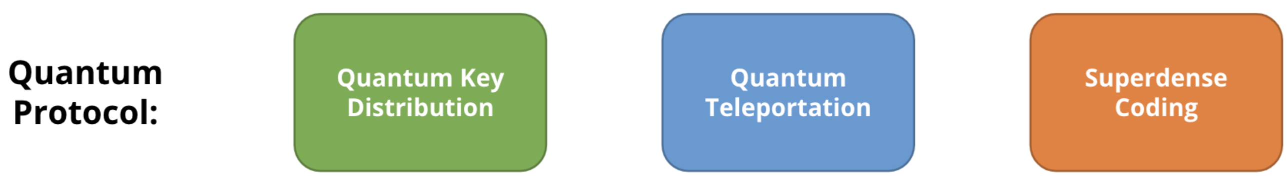
Landscape of Quantum Algorithms

As quantum devices get more sophisticated, it is valuable to know what the possible uses are or will be. Here are some examples:



<u>Quantum Protocols.</u> A set of standard rules that use the properties of quantum physics to allow electronic devices to communicate with each other.

Here are some of the quantum protocols that **we know** have a significant advantage over any classical alternatives:



<u>Quantum Key Distribution (QKD)</u>

Background and Motivation

Quantum Key Distribution is a quantum protocol with applications in **cybersecurity**.

© 2023 The Coding School

What is cybersecurity?

- Cybersecurity is an emerging field of technology that protects our computer and network systems from bad actors
- As we digitize assets like money, media files, & sensitive data, the opportunity arises for people to breach & steal these assets.
- Cybersecurity measures protect our digital belongings of value.

Quantum Key Distribution (QKD) is a cryptographic quantum protocol.

- Cryptography is a subset of cybersecurity. It is the practice of techniques for secure communication in the presence of third party adversaries.
- The goal of cryptography is to construct protocols that prevent third parties from reading private messages.
- Two important concepts in cryptography:
    - <u>The Channel:</u> We send encrypted messages over a channel.
        - Public channels could include phone, email, or social media direct message.
        - Private channels could include an optic fiber.
        - Channels can be classical or quantum.
    - <u>The Key:</u> We use a key to encrypt & decrypt a message.
        - The key is a critical piece of information.
        - If eavesdroppers can intercept the key, they can gain access to our valuable digital assets.

Alice and Bob: We will learn about QKD by imagining a communication between two imaginary people. It is a convenient convention to call them Alice and Bob (from A and B). When we want to model a malicious interceptor, we call her Eve (from Eavesdropper).

Basics of QKD

The Problem: Alice and Bob are trying to send a secret message. In order to protect their messages, Alice and Bob share a private key to encrypt/decrypt their messages. A third party, Eve, is trying to listen in on their conversation.

© 2023 The Coding School

Further explanation:

- In cryptography, Alice and Bob need some way to communicate so that even if Eve is listening she cannot understand what they're saying.
- One way to do this is that they agree upon a random set of bits that they can encode and decode messages with, but will make the messages look like total randomness to Eve.
- We call this random set of bits a key.

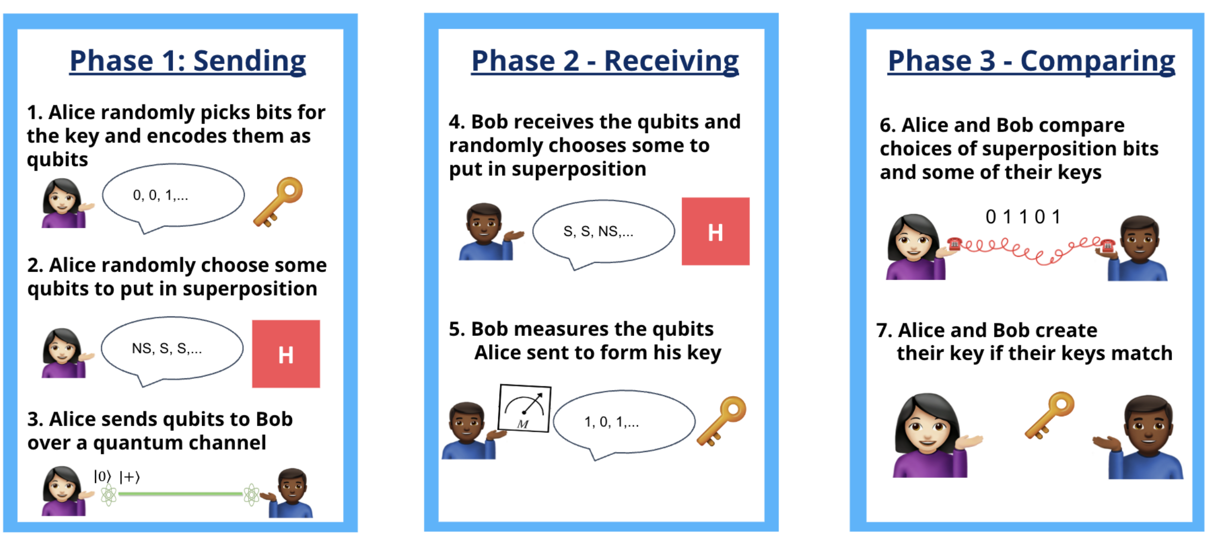
The goal: Confirm that our communication channel is secure - that our key (the secret password) was only shared between the sender and intended recipient.

QKD performs Key Distribution by leveraging quantum properties. Different QKD protocols use quantum properties differently.

In lecture and lab, we looked at the QKD protocol called BB84 that uses **superposition** to "hide" choices made by Alice and Bob and **quantum measurement** to detect the presence of Eve in a way that classical protocols cannot.

The BB84 Protocol

The following picture summarizes BB84 Protocol process **without Eve**:



© 2023 The Coding School

With Eve:

Phase 1: Sending

This is the same no matter what.

Phase 2 - Receiving

Eve goes through Phase 2 first and sends qubits to Bob.

😈

Bob goes through Phase 2 next without realizing Eve intercepted.

💁🏿‍♂️

Phase 3 - Comparing

This is the same no matter what. Alice and Bob hope they notice if Eve intercepted by seeing differences in the bits of their keys that they share.

Quick Summary of BB84

- Alice and Bob need:
    - A qubit for each bit of their intended key.
    - A quantum channel to communicate over.
- Alice and Bob both choose random qubits to apply an H gate to (create/destroy superposition for), hoping **most of the qubits they each chose match each other.**

Further Reading:

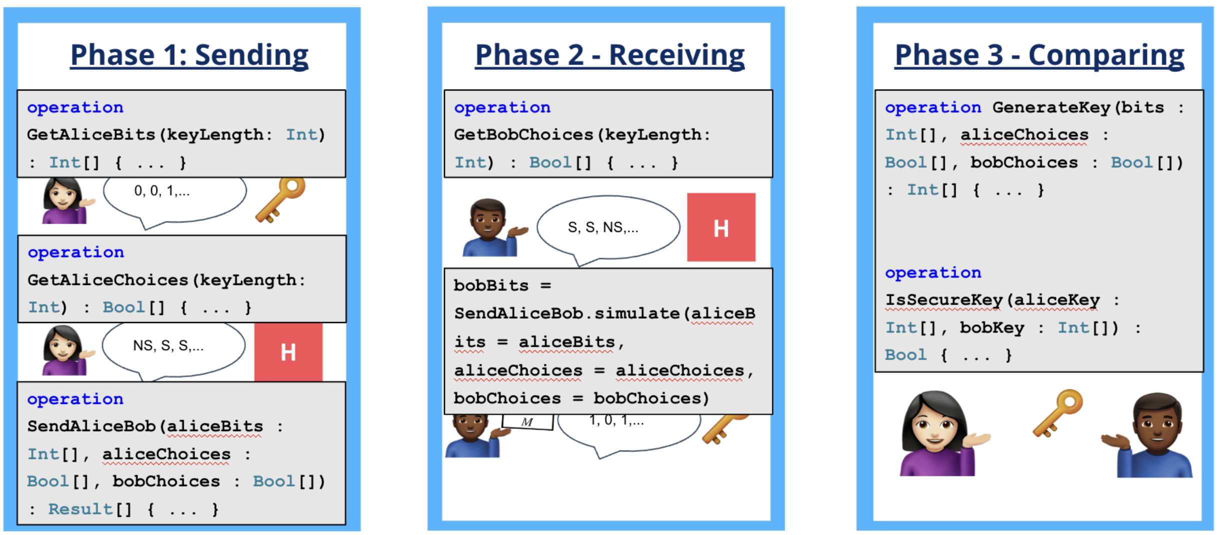
- Alternative QKD protocols
    - B92: https://www.st-andrews.ac.uk/physics/quvis/simulations_html5/sims/cryptography-b92/B92_photons.html
    - E91: https://www.ux1.eiu.edu/~nilic/Nina's-article.pdf
- Interpreting really small probabilities
    - https://www.youtube.com/watch?v=8Ko3TdPy0TU
- Original QKD paper
    - https://arxiv.org/abs/2003.06557
- QKD commercial implementation
    - https://www.idquantique.com/quantum-safe-security/overview/quantum-key-distribution/

© 2023 The Coding School

Lab: Implementing BB84

In lab, we learned how we can implement and demonstrate BB84 in Q#. We also used both the local Microsoft simulator as well as **remote, real hardware from IonQ!**

Here's the summary of operations we implemented:



Summary of BB84:

- Alice encodes the list of the classical bits of her key into qubits and applies an H gate to any she chooses to put into superposition
- Bob guesses which qubits Alice put into superposition and applies an H gate to them. Then, he measures all the qubits he received from her.
- Alice and Bob compare which qubits they put into superposition and only keep the ones they agree upon.
- Alice and Bob compare some of their measured bits to see if Eve has intercepted, since this could change the state of the qubits from Alice to Bob.