Quantum Criptograhy

With a look at Quantum Key Distribution

Michele Beretta

UniBG
https://github.com/micheleberetta98/qkd-presentation

2021

Table of Contents

1. Introduction
   - Basics
   - Notation
   - No-cloning theorem
2. Quantum Cryptography
   - Why
   - Quantum Key Distribution
   - In real life

The Qubit

The *bit* is the fundamental concept of classical computation - it can be either 0 or 1.

In a quantum world, where *superposition of states* is a thing, we use an analogous concept - the *quantum bit*, or **qubit**.

A qubit can be in any *linear combination* of two base states.

The Qubit

A classical bit is like a coin - either *head* or *tail*.

A qubit can be both *head* or *tail* at the same time - that is, until observed.

Observing a qubit makes it *decay* in one of the base states. Hence, measurement *changes* the real world.

Making a Quantum Computer

Making a qubit is hard - for example, nuclear spin can be mantained for long, but it's hard to measure.

A good quantum computer has to be *well isolated*, but its qubits have to be *accessible* in order to be manipulated.

A Mathematical Representation

Qubit

Given two states $|0\rangle$ and $|1\rangle$ a qubit is defined as

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \quad \alpha, \beta \in \mathbb{C}$$

For our purposes, it's safe to assume $\alpha, \beta \in \mathbb{R}$.

Some Linear Algebra

A qubit can be thought as a *vector* in a 2-dimensional vector space. The states $|0\rangle$ and $|1\rangle$ are the basis of this space.

One example could be

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix}, \quad |1\rangle = \begin{bmatrix}0\\1\end{bmatrix} \implies |\psi\rangle = \begin{bmatrix}\alpha\\\beta\end{bmatrix}$$

The *dot product* between two qubits $|\psi\rangle$ and $|\varphi\rangle$ is $\langle\psi|\varphi\rangle$, and the *tensor product* is given by $|\psi\rangle \otimes |\varphi\rangle$, or the shorter variant $|\psi\rangle\,|\varphi\rangle$.

Note that $\langle\psi| = |\psi\rangle^T$.

Measuring

When measuring a qubit, we can get:

- A 0 with probability $|\alpha|^2$
- A 1 with probability $|\beta|^2$

Since the probabilities must sum to 1, it has to be

$$|\alpha|^2 + |\beta|^2 = 1$$

Or, in other words, the qubit's state must be normalized.

So what is a qubit?

Mathematical Representation of a Qubit

A qubit is a *unit vector* in a *two-dimensional complex vector field*.

For example

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

is a qubit that, when measured, gives either 0 or 1 fifty-percent of the time.

More qubits?

We can combine multiple qubits. For example, a two qubit system has four *computational basis*

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle$$

In this system, a qubit can be in a superposition on 4 states.

In general, if we have *n* qubits, then the system can be in a superposition of 2^n states.

Gates

How do we modify qubits? With *quantum gates*.

Quantum Gate

A quantum gate is a *complex matrix* which must be unitary.

For example, suppose we define the **NOT** gate as

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Then it's easy to show that

$$X \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix}$$

Hence, a **NOT** gate inverts the probabilities of measuring 0 and 1.

More gates

Two other important gates are:

- The **Z** gate, which flips the sign of the $|1\rangle$ state

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

- The **H** gate, or *Hadamard* gate, used to bring the qubit in a superposition

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

A multi-qubit gate: CNOT

A *controlled-not* or **CNOT** gate is a two-qubit input gate, the *control* qubit and the *target* qubit.

The target qubit is flipped if the control qubit is set to 1. Its matrix is

$$U_{CN} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

And its effect is such that

$$|A, B\rangle \rightarrow |A, B \oplus A\rangle$$

Where $\oplus$ is addition modulo two.

Can we copy a qubit?

If we measure a qubit we destroy its superposition - but can we copy the qubit itself?

The answer is **no**. It is impossible to make a copy of an unknown quantum state.

Proof/1

Suppose we have a copying machine and we want to copy a qubit $|\psi\rangle$ into another qubit $|s\rangle$. Thus, the initial state of this machine is

$$|\psi\rangle \otimes |s\rangle$$

Suppose we have a unitary evolution U that effects the copying procedure, ideally

$$|\psi\rangle \otimes |s\rangle \rightarrow U(|\psi\rangle \otimes |s\rangle) = |\psi\rangle \otimes |\psi\rangle$$

Now, suppose this procedure works for two particular states, $|\psi\rangle$ and $|\varphi\rangle$. We have

$$U(|\psi\rangle \otimes |s\rangle) = |\psi\rangle \otimes |\psi\rangle$$
$$U(|\varphi\rangle \otimes |s\rangle) = |\varphi\rangle \otimes |\varphi\rangle$$

Proof/2

Taking the inner product of these two equations gives us

$$\langle\psi|\varphi\rangle = (\langle\psi|\varphi\rangle)^2$$

But $x = x^2$ has only two solutions - 0 and 1.

So $|\psi\rangle = |\varphi\rangle$ or $|\psi\rangle$ and $|\varphi\rangle$ are orthogonal.

Hence we can only clone orthogonal states, making general quantum cloning impossible.

Table of Contents

1. Introduction
   - Basics
   - Notation
   - No-cloning theorem
2. Quantum Cryptography
   - Why
   - Quantum Key Distribution
   - In real life

Why do we care about this?

Because of what quantum computers could be capable of - namely, efficiently perform some tasks that *are not feasible on a classical computer*.

In particular, **public key cryptography** is based on the infeasibility of some problems, such as prime factorization or discrete logarithm.

Fast prime factorization

For example, finding the prime factorization of an *n*-bit integer requires

$$\exp \Theta \left(n^{\frac{1}{3}} \log^{\frac{2}{3}} n \right)$$

operations using the *number field sieve*.

A quantum algorithm can be shown to accomplish the same task using

$$O \left(n^2 \log n \log \log n \right)$$

operations - so it's *exponentially faster*.

The demonstration of this is quite lengthy, but can be found in [NC10] or in [DW99].

Private Key Cryptography

In a private key cryptosystem, if Alice and Bob wish to exchange information they both must have a *matching key*, used to encrypt and decrypt the message.

As long as the keys are truly secret, this method is provably secure.

But how to keep the keys truly secret? The problem is the *secure distribution* of the keys.

Quantum Key Distribution (QKD)

It's a *provably secure* protocol by which *private keys* can be created between two parties over a *public channel*, exchanging qubits.

Basic idea - an external observer (Eve) cannot gain any information from the qubits transmitted without disturbing the state, because

1. Qubits cannot be copied (no-cloning theorem)

Quantum Key Distribution (QKD)

It's a *provably secure* protocol by which *private keys* can be created between two parties over a *public channel*, exchanging qubits.

Basic idea - an external observer (Eve) cannot gain any information from the qubits transmitted without disturbing the state, because

1. Qubits cannot be copied (no-cloning theorem)
2. Information gain is possible at the expense of introducing disturbance (provable)

The BB84 Protocol

Suppose Alice and Bob want to distill a shared secret key

Alice has two strings a and b of $(4+\delta)n$ bits. She encodes them as a block of $(4+\delta)n$ qubits

$$|\psi\rangle = \bigotimes_{k=1}^{(4+\delta)n} |\psi_{a_k b_k}\rangle$$

where a_k is the k^{th} bit of a (similarly for b).

The BB84 Protocol

So every qubit is one of these four states

$$
\begin{aligned}
|\psi_{00}\rangle &= |0\rangle \\
|\psi_{10}\rangle &= |1\rangle \\
|\psi_{01}\rangle &= |+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \\
|\psi_{11}\rangle &= |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}
\end{aligned}
$$

We can think of these as qubits encoded in one of two basis - X or Z - based on the value of the bit b_k.

The BB84 Protocol

If Eve is eavesdropping, then Bob receives a “disturbed” version of the qubits $\mathcal{E}(|\psi\rangle)$, where $\mathcal{E}$ is a quantum operation used to denote the effect of Eve on the channel.

Bob then publicly announces the fact that he has received the qubits.

Bob measures each qubit in basis X or Z, based on a $(4+\delta)n$ bits, string b', which he creates randomly. Let’s call the measurement result a'.

The BB84 Protocol

Alice publicly transmits b, and Alice and Bob discard any bits where Bob measured a different basis than Alice prepared.

With high probability, there are at least $2n$ bit left (if not, abort). Let's keep only the first $2n$ bits.

Alice chooses a subset of n bits as a check, and tells Bob which bits she selected. They both compare these, and if more than an acceptable number of bits disagree the protocol is aborted.

We can obtain a m-bits key from the remaining n bits by performing

- *Information reconciliation*, a form of error correction
- *Privacy amplification*, a method for reducing Eve's partial information about the key

Quick demo

Quick demo!

The BB84 protocol simulated in Elixir

But does any of this work?

Yes, Quantum Key Distribution has already been tested in real life. For example

- In 2015 the Unversity of Geneva and Corning Inc. achieved a secret key rate of $12.7\mathrm{kbit/s}$ over $307\mathrm{km}$ [K+15]
- In 2017 the Institute for Quantum Compuing and the University of Waterloo (Canada) achieved quantum key distribution between a ground transmitter and a moving aircraft [P+17]

References and further reading

Ronald De Wolf.
Quantum computation and shor's factoring algorithm, 1999.

Boris Korzh et al.
Provably secure and practical quantum key distribution over 307 km of optical fiber, 2015.

Michale Nielsen and Isaac Chuang.
Quantum Computation and Quantum Information.
Cambridge University Press, 2010.

Cristopher Pugh et al.
Airborne demonstration of a quantum key distribution receiver payload, 2017.