Quantum Key Distribution

1 Introduction

When we are dealing with sensitive data (e.g. banking details, personal data, etc.) over an insecure or public channel, there is no doubt that some kind of security measure is needed to protect them from getting stolen or breached. One way to secure our data is to encrypt them before sending them and then decrypt them again at the destination. This can be achieved, for example, with a symmetric key, meaning that both endpoints (source and destination) share the same secret key to be used for encryption as well as decryption. However, before this can be done, both endpoints need to acquire the symmetric key, ideally without another unwanted person getting a hold of it. In symmetric key cryptography, this process is called *Key Distribution*.

In the classical world, we can meet in person or physically transport the key (not practical), or we can use a physically secure channel to distribute the key. The disadvantage is that that we have to trust that the channel is secure enough and we will never know if the key got compromised.

In the quantum world, we distribute the key by means of a quantum channel, therefor called *Quantum Key Distribution*. Unlike before, we are now able to recognize if another person tried to steal this key during the distribution process (in terms of a probability, i.e. how likely that is).

To understand how this is possible, we need to understand a few basic principles of the quantum world, or more specific a quantum computer.

The normal computers we use every day are based on bits which can either be 0 or 1 with (always) 100% certainty, this however is different for bits in a quantum computer, called qubits. A qubit is in a quantum state, which describes a probability of possible outcomes (in this case 0 or 1) when measuring this quantum state. This means that a qubit can have for example a 50% chance to be a 0 and a 50% chance to be a 1 and only measuring it will determine which one it is. On the other hand, a qubit can also have the probability of 100% to be only one outcome and so on.

What we also need to know is that quantum states can be described in different bases and that measuring a quantum state in the wrong basis can change the result. For example, we have a state $|0\rangle$, which is a 0 in the Z-basis. When we also measure it in the Z-basis, it is guaranteed to be 0, whereas if we would measure it in the X-Basis, there is a 50% chance for it to be either a 0 or a 1. The same holds the other way around, e.g., $|+\rangle$, which is a 0 in the X-basis, will return a 0 when measured in the X-Basis, or either 0 or 1 when measured in the Z-Basis.

This property is helpful in the case of key distribution, since a third party has to guess which bases the qubits are in to measure the correct output for the secret key (the longer the key the harder it gets to correctly guess the basis for each qubit). If a wrong basis is used, the measuring of the qubit has a 50% chance to be wrong. The third party cannot be sure if the qubit they then forward to the receiving endpoint is actually correct or not. By looking at the qubits where the receiving endpoint used the correct basis for measuring, we are able to check for results that are different to the sending endpoint. This would indicate that it is likely someone tried to steal the key, used the wrong basis and then measured and forwarded the wrong qubit, otherwise the qubit would have the correct value 100% of the time if it is measured in the correct basis. For normal key distribution, if a third party has access to the used channel, they can simply read the correct bits of the secret key without anybody knowing.

To use qubits, we need a quantum channel, for example by sending individual photons through an optic-fibre cable and use the polarisation of a photon, which can be in one of two states, as our qubit to represent either a 0 or a 1.

As the name already indicates, quantum key distribution only involves the distribution of the key through a quantum channel, whereas this key is then used to encrypt and decrypt the data with an arbitrary algorithm.

In summary, the advantage of quantum key distribution is the possibility of knowing (at least in terms of a probability) whether or not someone tried to steal the key during the distribution process. This however comes with the cost of needing a quantum channel and the ability to manipulate and send, as well as receive and measure a quantum state. Since there are also secure classical ways of distributing a symmetric key, the question is if the benefits outweigh the costs.

2 Protocol

Alice wants to send Bob a key for them to use as a symmetric key to encrypt and decrypt their messages.

Alice prepares her qubits to represent the secret key in randomly chosen bases and sends them through a quantum channel to Bob.

Bob receives the qubits and measures them in randomly chosen bases.

Alice and Bob then publicly share what bases they used for each qubit. They discard a qubit if they used different bases for it (because there is a chance that Bob's result could be wrong) and keep the qubit if they used the same bases.

To check if somebody tried to interfere in the distribution process, Alice and Bob choose a sample of qubits from the key and compare their results. If nobody interfered in the distribution process, Alice and Bob will have the same result for the whole sample. If the results from the sample differ, Alice and Bob know that somebody tried to measure and steal the key, since in theory it is not possible that Bob would get a different result for a qubit even though it was measured in the right basis. However, because practically there can occur errors when measuring, Alice and Bob have to decide what margin of error they allow before assuming the error is actually big enough to be caused by another person. If the error in the sample is too big, Alice and Bob assume that the key was compromised and repeat the distribution process.

At last, if Alice and Bob are happy with the results from the sample, they discard the sample, since it is now public, and use the rest as their shared secret key to encrypt and decrypt their messages in the future.

2.1 Example

Alice wants to use the following string as a secret key:

110011100011111110

For each bit she then prepares a corresponding qubit in the following bases:

xzzxxxxzzxxzzxxxzzx

Her quantum state to send to Bob therefor would look like this:

|-⟩|1⟩|0⟩|+⟩|-⟩|-⟩|1⟩|0⟩|+⟩|+⟩|1⟩|1⟩|-⟩|-⟩|-⟩|1⟩|1⟩|+⟩

Bob randomly chooses bases to measure the qubits Alice sends to him:

xzxzzxzxxzzxzzxzxx

2.1.1 No Interception

Bob's measurement results could look like the following (bits where he used the correct bases are marked):

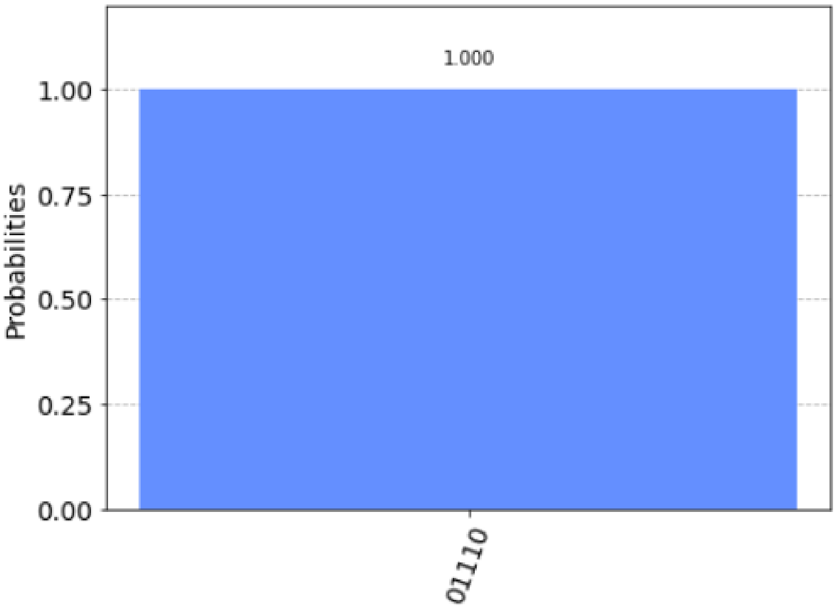
11100**11**0**0**1**1**101**11**0**0**

Alice and Bob share the bases they used and discard the qubit where they used a different basis. Their key now looks like this:

Alice: 111101110

Bob: 111101110

They decide to use the first 5 qubits as a sample and see that they both have "01110", which means it is probably not likely that somebody tried to steal the key. They then discard this sample and use the rest as their shared secret key, i.e. "1111".



No matter how many times we repeat this process (here 1024 times), we see that Bob always gets the same result for the sample as Alice as depicted in the graph above. (See notebook for code).

2.1.2 With Interception

Bob’s measurement results could look like the following (bits where he used the correct bases are marked):

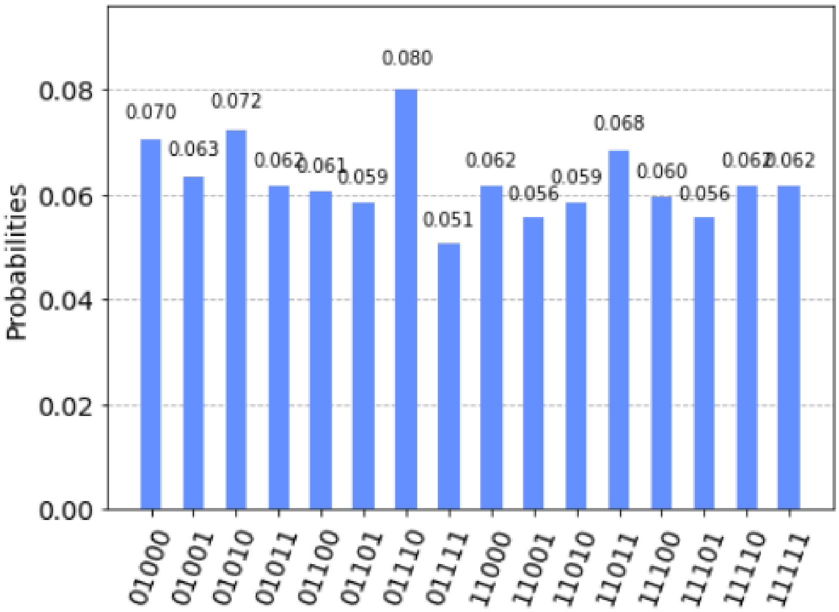
10100**01**0**1**1**1**101**01**0**0**

Alice and Bob share the bases they used and discard the qubit where they used a different basis. Their key now looks like this:

Alice: 111101110

Bob: 100111010

They decide to use the first 5 qubits as a sample and see that Bob’s sample “11010” is different to Alice’s sample “01110”, which means it is likely that somebody tried to steal the key. They discard their keys and repeat the whole process again.



In this example with interception (where the third party always alternated between the Z- and X-basis), if we also repeat this process numerous amounts (here 1024 times), we see that Bob can have different outcomes, and only in 8% of the attempts are the samples the same (“01110”) so that the interference would go undetected. (See notebook for code).

3 Risk of Quantum Key Distribution

Since the test to check if a person tried to steal the key assumes that the error occurs because a wrong basis was used for a measurement, the possibility exists that they used the correct bases or that the measurements in the wrong basis luckily results in the correct value. This would then go unnoticed in the distribution process. We can calculate the probability with which an interference would go undetected by Alice and Bob for a key of a certain length.

The probability that a third party uses the correct basis to measure a qubit is 50% (either the Z-Basis or the X-Basis), but even if the wrong basis was used, there is still another 50% chance that Bob measures the correct value Alice prepared. This means that overall the probability to go undetected is 50% for choosing the correct basis plus 25% (50% to choose the wrong basis times 50% for Bob to still measure the correct value), which overall makes up a 75% chance to go undetected for a single qubit. The longer the secret key is, i.e., the more qubits are used, the harder it gets to go undetected for each one (probability gets lowers). This can overall be calculated by $p = 0.75^x$, whereas *x* is the length of the secret key.

The example above compares a sample of length 5, which would mean the probability to go undetected is $0.75^5 \approx 23.7\%$.

For a probability of roughly 1% to go undetected, Alice and Bob would need to compare a sample with 16 qubits ($0.75^{16} \approx 1.002\%$).