BB84 Project

BB84 Overview

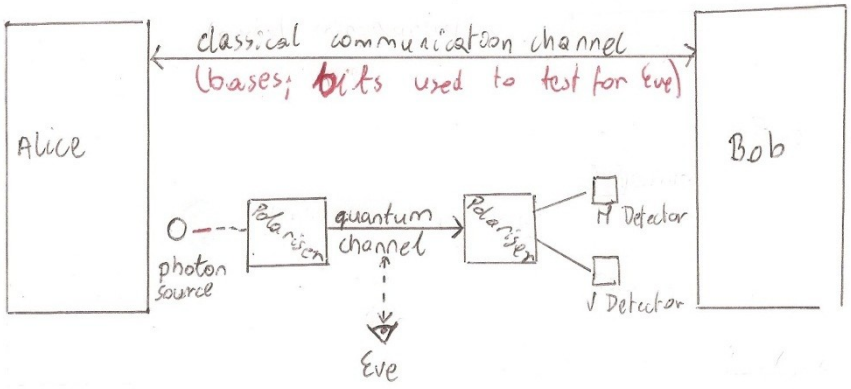
BB84 is a quantum algorithm that relies on the security provided by the wave function collapse and the associated “no-cloning theorem”.

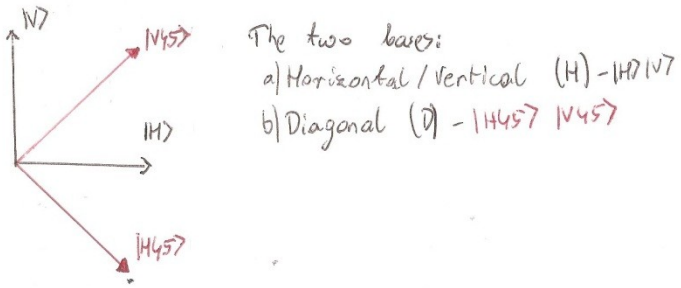
The algorithm allows two parties to agree on a secret key over an unsecured channel. It is important to note that the algorithm does not prevent an intruder from listening in, but instead provides a statically reliable way of telling that the key has been intercepted.

Operation

The algorithm involves two parties – Alice and Bob – who wish to share/agree on a secret key.

In addition to classical computing resources, the physical setup that enables the transfer requires a public quantum channel and two sets of bases on both ends of the communication. The bases that are used are orthogonal – see the diagrams below:





Alice, who has a public quantum channel that connects her to Bob, uses a random number generator to generate bits of the key. For every generated bit, she randomly chooses one of her two bases and transmits that bit as a photon that is polarised in the chosen basis (in other words, she transmits a qubit).

The qubit goes along the quantum channel to Bob, who measures the polarization using a randomly chosen basis. If Bob uses the same basis to measure as Alice used to polarize, the qubit will collapse

to the original bit value (in other words, collapse to the original value with a probability of 1). On the other hand, if Bob randomly chooses the other basis, the qubit will collapse to either 0 or 1 with equal probability (probability of ½). This process will be repeated until the entire key has been sent.

Here comes Eve, the eavesdropper. It is possible for Eve to listen in and intercept qubits as they are travelling along the quantum channel from Alice to Bob, but there is a catch – Eve does not know which of the two bases to use when collapsing the intercepted qubit. Because of this Eve has to rely on the same tactic as Bob – pick a random basis. But this brings the danger of using the wrong basis and reading in a random, incorrect value (incorrect value => incorrect key). See section - "Probabilities and Error Rates" - for more details.

To cover her tracks, Eve has to polarise the intercepted bits just like Alice did, using the correct basis (the same basis that Alice used) and send them on to Bob. This highlights another one of her problems – Eve cannot be certain that she used the right basis – she can only assume/hope that she did. If she sends an incorrectly polarised photon to Bob, there is a chance that her presence will be revealed.

Let's come back to Bob. The entire key has now been transferred from Alice to Bob. For every bit that was sent using the technique outlined above, there is a 50% probability that Bob used the wrong basis and recorded a random-value bit. To make Bob's readings meaningful, Alice and Bob tell each other what basis each of them used for every bit that was sent. This is done using a classical, public channel. Whenever the bases differ, the corresponding bit is discarded. This eliminates about 50% of the original key (there is a 50% chance that an incorrect basis was used for each qubit).

Bob now has a portion of the key that he knows he has read correctly because of the information on bases that he received from Alice. To ensure that Eve was not involved in the exchange, Alice and Bob use a portion of that remaining key (50% in the case of this assignment), share that portion between them over a public, classical channel and check whether their parts of the key are the same.

Since Alice and Bob used the same bases for that portion of the key, they can expect their keys to be the same. If Eve were to intercept and randomly trample through the data, some bits that Bob has recorded would be different from Alice's bits (the no-cloning theorem and the probabilistic nature of the two orthogonal bases ensure that "random trampling" is the only method of eavesdropping). Eve got unlucky with those bits – she used the wrong basis and when Bob made the reading, the qubit randomly collapsed to an incorrect value.

At any rate, if an intruder is detected Alice and Bob cannot use the key from this session. On the other hand, if no intruders are found, they can use the remaining secret bits as a shared key. Note that in the case of this assignment the secret key will be ~25% of the size of the original key – ~50% of the original key was first discarded because of incorrect random bases, and another ~50% of the remaining key was then sent over a public channel to test for Eve and now has to be discarded as well => 1 * ½ * ½ = ¼.

Probabilities and Error Rates

For every bit that was intercepted by Eve, there is a 25% probability that an incorrect value was forwarded on to Bob.

Wrong basis used (50%) AND wrong random bit recorded (another 50%):

⇨ ½ * ½ = ¼ = 25%

If Eve is looking at every qubit that is sent, when that qubit (or rather its equivalent that was created by Eve) is checked by Bob, there is a 25% chance that Bob will notice that Eve was involved.

Eve used the wrong basis (50%) AND Bob got a random, incorrect result (another 50%) – note that it doesn't matter what Eve has sent if she used the wrong basis.

⇨ ½ * ½ = ¼ = 25%

In the assignment, a 1024 bit key was used. What is the probability that Bob will notice Eve's influence if ½ of the remaining key is used in the test?

1. We start off with 1024-bit key, around ½ of that will be discarded due to the difference in bases => 1024 * ½ = 512-bit key remaining.
2. A half of that will be used to test for the presence of an eavesdropper => 512 * ½ = 256-bit key to test.
3. For every bit, there is a 25% probability of detecting Eve's presence => 75% probability of not finding anything wrong (The 25% was established in the previous calculation).
4. For the entire 256-bit "test key" there is a ¾ ^ 256 chance that Eve will not be detected => 1.0367724e-32.
5. Conclusion - as the size of the test key increases, the probability of detecting Eve tends to 1, with 256-bit test key offering a one in a nonillion chance of Eve remaining undetected.

The Implementation

A 1024-bit key is sent by Alice to Bob in the simulation, which implies that the final key will be around 256 bits long (~50% discarded because of incorrect bases, another ~50% of the remaining 512 bits used to test for the presence of Eve) ~256 bits are used to test for Eve.

This simulation is implemented in Java using network Sockets. Alice acts as the server and Bob is the client from the network's point of view. Eve is simulated on the server-side – before each qubit is officially sent to Bob using the socket connection, Eve gets a chance to intercept and re-polarise it. The decision to simulate Eve on the server, rather than the client-side is a purely arbitrary – it would've been just as easy to simulate Eve on Bob's side of things. In real implementations of BB84, Eve eavesdrops over the quantum channel.

To run the program, the Server class should be run first. Then, a decision should be made whether to simulate the program with or without Eve. After that, the Client class can be run. When the program completes, each step of the algorithm will be displayed from Alice's (server) and Bob's (client) point of view, each in their respective windows.