Alice 1

Mission 2 : Setting Up Quantum Key Distribution

70 minutes of gameplay [60/200 points]

As you progress through this company start-up, you start to suspect from quantum physicists that classical communication is probably not very secure. Well, you have gone through a full week of Quantum Wierdness and whatnot lectures.

You want to develop a scheme from the lectures: The BB84 QKD scheme. However, as you are just a start-up company, you do not have a lot of money, so you can not buy single photon detectors. Also, your friend, who is an expert in popular science, convince you that you don't need single photons. Well, as you don't have much choice anyway, you will just try it lorr...
The goal of this mission is to establish the quantum link between Alice and Bob.

This mission is divided into smaller tasks, which consists of compulsory and optional tasks. The compulsory tasks are marked with either [Checkpoint], [Final Task], or [Secret Task] flags, while the unmarked tasks are sort of optional. It is thus a priority to complete all the flagged tasks before the optional tasks, as one will not be able to revisit these tasks after the deadline. The compulsory tasks are very important for the upcoming missions. It is also highly advisable to split the tasks among your teammates.

[20 points] [Checkpoint] Align the polarisation

> *Why would you want to use less to send information. Wouldn't more photons guarantee a stronger signal?*
>
> *- Your good friend, Dr. Emmett Brown*

Objectives:

1. Align the polarisation between Alice and Bob, and
2. Obtain the intensity matrix, with signal degradation less than 0.2.

Very important notes: Read the safety precautions in Section 1.3.1 of the technical documents. This is *extremely important*, as you will get tested about this at the end of the session (no joking, confirm plus chop!).

Point allocation scheme:

- [Full] points upon completion of the objectives
- [80%] of total points upon completion of the objectives, but with the signal degradation higher than 0.2.

Step by step walkthrough:

1. Upload the Arduino program `ArduinoQuantum.ino` to the correct device.
2. Run the polarisation alignment `GUI` programs (in the folder `Alignment GUI`): For Alice, `runSender.py` and for Bob, `runReceiver.py`. You might want to refer to `Section 3.4.3` for more information regarding the GUI. The goal of this exercise is to align Alice and Bob polarisation axis with respect to each other, i.e. if Alice sends `H` polarised light, and Bob measures in `H` polarisation basis, the measurement result should give maximal intensity. *You need to close the GUI (or at least stop the device) before moving on to other tasks.*
3. Set the correct device address in `devloc_quantum.txt`. Run the calibration program `send_calibrate.py` (for Alice) and `recv_calibrate.py` (for Bob). This program measures the `Intensity Matrix` between different polarisation states of Alice and Bob. Make sure that the signal degradation is lower than 0.2 (or repeat the previous step if not), and note down the mean value of the intensity.

[10 points] Polarisation basis handout

> *Give me six hours to chop down a tree and I will spend the first four sharpening the axe.*
>
> *- Abraham Lincoln*

Objective: Complete the `Polarisation Basis` handout. No cheating or copying with Bob allowed [insert stern warning].

Point allocation scheme:

- Based on the number of correct responses in the handout.

[10 points] Randomness handout

> *Creativity is the ability to introduce order into the randomness of nature.*
>
> *- Charles Bennett and Gilles Brassard*

Objective: Complete the `Randomness` handout. No cheating or copying with Bob allowed [insert stern warning].

Point allocation scheme:

- Based on the number of correct responses in the handout.

Note: Only do this when there is a free time or there is a member in your group who happens to be free.

[20 points] [Final Task] Establishing secure key

> *There is an interesting similarity between probabilistic encryption and quantum cryptography: both rely on the notion of reduction. However, whereas the former reduces the unproved computational complexity of some outstanding problems of number theory to the difficulty of breaking the schemes, the latter relies on the most fundamental beliefs of quantum physics.*
>
> *- Dave Barry*

Objectives:

1. Successfully construct a secure key derived from 16 bit polarisations choices and bases made by Alice and Bob, and keysifted via a public channel.
2. Construct 3 secure keys with those methods, but with Alice not communicating with Bob nor looking at his paper or computer (vice versa). The only communication allowed is the `public channel` paper that is passed back and forth between them.

Point allocation scheme:

- [Full] points by completing all the objectives **within 60 minutes** from the start of Mission 2 (leaving 10 more minutes to wrap up other tasks), or if fails,
- [80%] of total points by completing all the objectives within the time limit, or if fails,
- [10%] of total points reduction for each failed attempt in constructing the secure key (in Objective 2).

Step by step walkthrough:

1. Alice runs `send_key.py` and Bob runs `recv_key.py`. Note that Bob needs to run his program first. After running the programs, Alice will obtain the unsifted key (16 bits) and basis choices (16 bits), while Bob will obtain the measurement result (16 bits) and basis choices (16 bits). For Bob, remember to set the value in `threshold.txt` to be the mean of the intensity matrix.
2. Alice and Bob fills in the keysift form according to the procedures. Supposedly, they are only allowed to pass the `public channel` paper back and forth. However, in the trials Alice and Bob can work together and help each other. They might also benefit with the program `keysift_hint.py`.
3. Alice and Bob attempt the second objectives, i.e. without any other communication channel except the `public channel` paper. This session will be guided by the `GameMaster`, and after each attempt you will submit the keysift form to the `GameMaster`. He will then confirm whether the key are correctly constructed.