Implementation-

We have written three program files -send.py, receive.py, eve.py for the transmitter, receiver, and eavesdropper node, respectively. These codes are then implemented in the terminals of the raspberry pi with the functionalities. We have used three Raspberry Pi to run these programs. send.py is responsible for generating, modulating and transmitting the signal. This runs on the 1st R pi. Receive.py is responsible for receiving the signal and demodulate it. This runs on the 2nd R pi. Eve.py is executed on a 3rd R pi, which is trying to receive and demodulate the signal but is unsuccessful in doing so.

Method-

1. We chose the size of the bit-set to be 10 for simplicity purposes although it can be modified.
2. Next, a random bit sequence is generated using in-built python libraries.
3. During each transfer a phase shift is added to the constellation diagram, which belongs to a sequence known only by the source and the receiver.
4. Now, to improve the security we used variable phase shift keying so that different number of bits are converted from the original sequence to form the modulated signal.
5. Afterwards, we used socket communication for broadcasting the data so that every receiver can get the data but only the legitimate receiver is aware of the phase change and psk sequence.
6. After the signal is received, the true receiver can convert it back to its original form, while the eavesdropper would end up with a garbage value each time.