Secure Quantum Data Communication Simulation

This script demonstrates how the encryption scheme is used for sending data from one party "Alice" to another party "Bob". in this simulation Alice is sending the message "hello" to Bob, we can see the original binary representation of this message which will then get fed into our quantum circuit. In the quantum circuit the encryption results in blocks where the quantum state is unrecognizable from the original data. Then Bob takes in this data and decrypts it to get the original binary data then converts it back into plain text to display the original message.

```
%File for demo
%Initialization (RNG seed, block size, signature size, size of clifford operator
set)
Alice = sender(42,4,2,50);
Bob = receiver(42,4,2,50);

codeword = Alice.messageToBinary("hello")
```

```
codeword = 10×1 string
"0110"
"1000"
"0110"
"0101"
"0110"
"1100"
"0110"
"1100"
"0110"
"1111"
```

```
encrypted_states = Alice.encrypt(codeword);
%Show the first 2 blocks of encrypted data
for i=1:2
    X = ['Block number ',num2str(i),':'];
    disp(X)
    disp(formula(encrypted_states{i}));
end
```

```
Block number 1:
(-0.125-0.125i) * |000000> +
(0.125-0.125i)  * |000001> +
(-0.125+0.125i) * |000010> +
(0.125+0.125i)  * |000011> +
(0.125-0.125i)  * |001000> +
(-0.125-0.125i) * |001001> +
(-0.125-0.125i) * |001010> +
(0.125-0.125i)  * |001011> +
(-0.125-0.125i) * |010000> +
(0.125-0.125i)  * |010001> +
(0.125-0.125i)  * |010010> +
(-0.125-0.125i) * |010011> +
(0.125-0.125i)  * |011000> +
(-0.125-0.125i) * |011001> +
(0.125+0.125i)  * |011010> +
(-0.125+0.125i) * |011011> +
(0.125+0.125i)  * |100000> +
(0.125-0.125i)  * |100001> +
(-0.125+0.125i) * |100010> +
```

```
(-0.125-0.125i) * |100011> +
(0.125-0.125i)  * |101000> +
(0.125+0.125i)  * |101001> +
(0.125+0.125i)  * |101010> +
(0.125-0.125i)  * |101011> +
(-0.125-0.125i) * |110000> +
(-0.125+0.125i) * |110001> +
(-0.125+0.125i) * |110010> +
(-0.125-0.125i) * |110011> +
(-0.125+0.125i) * |111000> +
(-0.125-0.125i) * |111001> +
(0.125+0.125i)  * |111010> +
(0.125-0.125i)  * |111011>
Block number 2:
(2.0144e-18+0.125i)  * |000000> +
(-1.6059e-18+0.125i) * |000001> +
(-0.125-2.0144e-18i) * |000010> +
(0.125-1.6059e-18i)  * |000011> +
(0.125+2.0144e-18i)  * |000100> +
(0.125-1.6059e-18i)  * |000101> +
(-2.0144e-18-0.125i) * |000110> +
(-1.6059e-18+0.125i) * |000111> +
(1.6059e-18-0.125i)  * |001000> +
(-2.0144e-18-0.125i) * |001001> +
(0.125-1.6059e-18i)  * |001010> +
(-0.125-2.0144e-18i) * |001011> +
(-0.125+1.6059e-18i) * |001100> +
(-0.125-2.0144e-18i) * |001101> +
(-1.6059e-18+0.125i) * |001110> +
(-2.0144e-18-0.125i) * |001111> +
(-0.125-1.6059e-18i) * |010000> +
(0.125-2.0144e-18i)  * |010001> +
(1.6059e-18+0.125i)  * |010010> +
(-2.0144e-18+0.125i) * |010011> +
(1.6059e-18+0.125i)  * |010100> +
(2.0144e-18-0.125i)  * |010101> +
(-0.125-1.6059e-18i) * |010110> +
(-0.125+2.0144e-18i) * |010111> +
(-0.125+2.0144e-18i) * |011000> +
(0.125+1.6059e-18i)  * |011001> +
(-2.0144e-18+0.125i) * |011010> +
(1.6059e-18+0.125i)  * |011011> +
(-2.0144e-18+0.125i) * |011100> +
(-1.6059e-18-0.125i) * |011101> +
(-0.125+2.0144e-18i) * |011110> +
(-0.125-1.6059e-18i) * |011111> +
(-1.6059e-18-0.125i) * |100000> +
(-2.0144e-18+0.125i) * |100001> +
(0.125+1.6059e-18i)  * |100010> +
(0.125-2.0144e-18i)  * |100011> +
(0.125+1.6059e-18i)  * |100100> +
(-0.125+2.0144e-18i) * |100101> +
(-1.6059e-18-0.125i) * |100110> +
(2.0144e-18-0.125i)  * |100111> +
(2.0144e-18-0.125i)  * |101000> +
(1.6059e-18+0.125i)  * |101001> +
(0.125-2.0144e-18i)  * |101010> +
(0.125+1.6059e-18i)  * |101011> +
(0.125-2.0144e-18i)  * |101100> +
(-0.125-1.6059e-18i) * |101101> +
(2.0144e-18-0.125i)  * |101110> +
(-1.6059e-18-0.125i) * |101111> +
(-0.125-2.0144e-18i) * |110000> +
(-0.125+1.6059e-18i) * |110001> +
```

```
(2.0144e-18+0.125i)  * |110010> +
(1.6059e-18-0.125i)  * |110011> +
(-2.0144e-18-0.125i) * |110100> +
(1.6059e-18-0.125i)  * |110101> +
(0.125+2.0144e-18i)  * |110110> +
(-0.125+1.6059e-18i) * |110111> +
(0.125-1.6059e-18i)  * |111000> +
(0.125+2.0144e-18i)  * |111001> +
(1.6059e-18-0.125i)  * |111010> +
(2.0144e-18+0.125i)  * |111011> +
(-1.6059e-18+0.125i) * |111100> +
(2.0144e-18+0.125i)  * |111101> +
(-0.125+1.6059e-18i) * |111110> +
(0.125+2.0144e-18i)  * |111111>
```

```
decoded = Bob.decode(encrypted_states);
%Show first 2 blocks of decrypted binary
for i=1:2
    X = ['Block number ',num2str(i),':'];
    disp(X)
    disp(decoded{i});
end
```

```
Block number 1:
0110
Block number 2:
1000
```

```
received_message = Bob.binaryToMessage(decoded);
disp(received_message);
```

```
hello
```

Below we have a second simulation with different block sizes, signature size, and size of the clifford gate set. We can still see that Bob is able to successfully decrypt and return the new message

```
Alice = sender(10,3,1,40);
Bob = receiver(10,3,1,40);
codeword = Alice.messageToBinary("goodbye")
```

```
codeword = 19×1 string
"011"
"001"
"110"
"110"
"111"
"101"
"101"
"111"
"011"
"001"
    ⋮
```

```
encrypted_states = Alice.encrypt(codeword);
decoded = Bob.decode(encrypted_states);
received_message = Bob.binaryToMessage(decoded);
disp(received_message);
```

goodbye