QKD Simulation

Example Main Program

```
% 10 random data bits
Data=randi([0 1],1,10);

% Encoding
[S,polarization]=encoder(Data);

% Insecure Quantum channel
R=channel(S);

% Decoding
[D,basisindex]=decoder( R );

% Authenticated channel handshake
indices=
authenticatedChannelHandshake(polarization,basisindex)

% keep only bits measured using same basis
D=D(indices); % D is the sifted key
```

Outline of Encoder

```
% Encode each bit as polarized photon
% rectilinear basis & diagonal basis angles
basis=[[90, 0] ; [45, 135]];
% sent photons
S=[];
% save polarization of each photon
polarization=[];


for i=1:length(Data)
    % randomly select a basis
    p=randi([1 2]);
    polarization=[ polarization p ];
    % corresponding angles
    A=basis(p,:);
    % map data bit to photon
    S=[S A(Data(i)+1) ];
End
```

Example

```
  0    1   0   0   0   0   0   0   0    1  (Data)
  1    2   2   2   1   1   1   2   2    1  (polarization)
 90  135  45  45  90  90  90  45  45    0  (S)
```

Insecure Quantum channel

(no attack simulated)

```
R=S;
```

Decoder outline

```
% R = received photons
% rectilinear basis & diagonal basis angles
basis=[[90, 0] ; [45, 135]];
% randomly select a basis and corresponding angles
basisindex=randi([1 2]);
B=basis(basisindex,:);
% Measure the polarity of photons
D = [];
for i=1:length(R)
    if R(i)==B(1)
        D=[ D, 0 ];
    elseif R(i)==B(2)
        D = [ D, 1 ];
    else
        D = [ D, randi([0 1]) ];
end
```

Example

```
90  135   45   45   90   90   90   45   45    0  (R)


 1    1    0    0    1    1    0    0    0    0  (D)


     2 (basisindex)
```

Authenticated Channel Handshake outline

```
% polarization=Alice's array of polarizations
% basis=Bob's sent basis to Alice
% indices=Alice sent indices of photons matching the basis
indices=find(polarization==basis);
```

Example

```
1    2    2    2    1    1    1    2    2    1 (polarization)
2 (basis)
2     3    4    8    9 (indices)

1    1    0    0    1    1    0    0    0    0 (Decoding)
1    0    0    0    0 (sifted key)
```