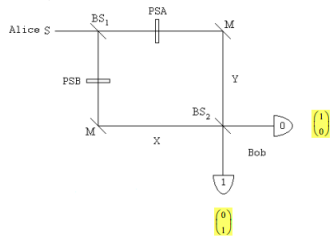
8.92: Quantum Key Distribution Using a Mach-Zehnder Interferometer

Charles H. Bennett proposed the following Mach-Zehnder interferometer for quantum key distribution (*Physical Review Letters* 68, 3121 (1992)).



Alice's source at the left supplies single-photon states, which are split by a symmetric beam splitter BS_1 into a superposition being present in both arms of a Mach-Zehnder interferometer (MZI). Alice (PSA) applies a random 0-, 90-, 180-, or 270-degree phase shift in one arm and Bob (PSB) applies a random 0- or 90-degree phase shift in the other arm. Mirrors direct the photon to a second beam splitter creating two photon paths to each detector and thereby allowing for interference between the paths. After photon detection by Bob, Alice and Bob agree publicly to keep only those results for which their phase shifts differ by 0 or 180 degrees, settings for which the photons behave deterministically at the second beam splitter.

Direction of propagation vectors:

$$x = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad y = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Matrix operators for the interferometer components:

$$\text{Beam splitter:} \quad \text{BS:} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix} \quad \text{Mirror:} \quad \text{M:} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \text{Phase shift:} \begin{pmatrix} e & i\,PSA & 0 \\ 0 & e & i\,PSB \end{pmatrix}$$

Construct a Mach-Zehnder interferometer using these components.

$$\text{MZI(PSA, PSB)} = \text{BS}\,\text{M} \begin{pmatrix} e^{i\,PSA} & 0 \\ 0 & e^{i\,PSB} \end{pmatrix} \text{BS}$$

Probability Detector 0 will fire:

$$\text{Detector\_0(PSA, PSB)} = \left(\left|x^T \text{MZI(PSA, PSB)}\,\text{x}\right|\right)^2$$

Probability Detector 1 will fire:

$$\text{Detector\_1(PSA, PSB)} = \left(\left|y^T \text{MZI(PSA, PSB)}\,\text{x}\right|\right)^2$$

For each of eight possible phase shift settings calculate the probability that detectors |0> and |1> will register the arrival of a photon.

The PSA/PSB settings for which a photon behaves deterministically are highlighted.

		$\text{Detector} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = 0$	$\text{Detector} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = 1$
$PSA = 0deg$	$PSB = 0deg$	$Detector_0(PSA, PSB) = 1$	$Detector_1(PSA, PSB) = 0$
PSA = 0 deg	PSB= 90 deg	Detector_0(PSA, PSB) = 0.5	Detector_1(PSA, PSB) = 0.5
PSA = 90 deg	PSB= 0 deg	Detector_0(PSA, PSB) = 0.5	Detector_1(PSA, PSB) = 0.5
$PSA = 90deg$	$PSB = 90deg$	$Detector_0(PSA, PSB) = 1$	$Detector_1(PSA, PSB) = 0$
$PSA = 180deg$	$PSB = 0deg$	$Detector_0(PSA, PSB) = 0$	$Detector_1(PSA, PSB) = 1$
PSA = 180 deg	PSB= 270 deg	Detector_0(PSA, PSB) = 0.5	Detector_1(PSA, PSB) = 0.5
PSA = 270 deg	PSB= 0 deg	Detector_0(PSA, PSB) = 0.5	Detector_1(PSA, PSB) = 0.5
$PSA = 270deg$	$PSB = 90deg$	$Detector_0(PSA, PSB) = 0$	$Detector_1(PSA, PSB) = 1$

Demonstrate that the detection results at each detector are completely random. In other words, that if someone was monitoring Bob's detectors he or she would see no pattern in the results.

The settings of phase shifters PSA and PSB are changed randomly by Alice and Bob. So given a large number of runs, each pair of settings shown above will occur with probability 1/8 or 12.5%. Overall each detector will register a photon in half the runs.

$$Detector = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \frac{1+\frac{1}{2}+\frac{1}{2}+1+0+\frac{1}{2}+\frac{1}{2}+0}{8} \to \frac{1}{2} \quad Detector = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \frac{0+\frac{1}{2}+\frac{1}{2}+0+1+\frac{1}{2}+\frac{1}{2}+1}{8} \to \frac{1}{2}$$

Demonstrate the use of a secret key to exchange a secure message between a sender and a receiver.

Coding and Decoding a Message

$j = 1..25$

$\text{Key}_j = \text{trunc(rnd(2))}$

$\text{Mes}_j = \text{trunc(rnd(2))}$

$\text{Key}^T = (0 \;\; 0 \;\; 1 \;\; 0 \;\; 1 \;\; 0 \;\; 1 \;\; 0 \;\; 0 \;\; 0 \;\; 1 \;\; 0 \;\; 0)$

$\text{Mes}^T = (1 \;\; 1 \;\; 1 \;\; 0 \;\; 1 \;\; 0 \;\; 1 \;\; 0 \;\; 0 \;\; 1 \;\; 1 \;\; 0 \;\; 1)$

$\text{CMes}_j = \text{Mes}_j \oplus \text{Key}_j$

$\text{CMes}^T = (1 \;\; 1 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 1 \;\; 0 \;\; 0 \;\; 1 \;\; 1 \;\; 0 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 0 \;\; 0 \;\; 1 \;\; 0 \;\; 0)$

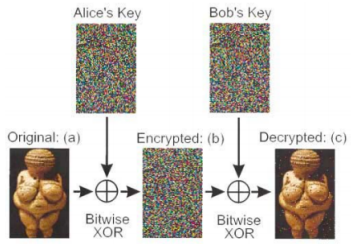
$\text{DMes}_j = \text{CMes}_j \oplus \text{Key}_j$

$\text{Mes}^T = (1 \;\; 1 \;\; 1 \;\; 0 \;\; 1 \;\; 0 \;\; 1 \;\; 0 \;\; 0 \;\; 1 \;\; 1 \;\; 0 \;\; 1 \;\; 0 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 0 \;\; 0 \;\; 1 \;\; 1 \;\; 0 \;\; 0 \;\; 1)$

It is clear by inspection that the message has been accurately decoded. This is confirmed by calculating the difference between the message and the decoded message.

$$(\text{DMes} - \text{Mes})^T = (0 \;\; 0)$$

In 2000 Anton Zeilinger and his research team sent an encrypted photo of the fertility goddess Venus of Willendorf from Alice to Bob, two computers in two buildings about 400 meters apart. The figure summarizing this achievement first appeared in *Physical Review Letters* and later in a review article in *Nature*.



By extending the previous example to two dimensions, it easy to produce a rudimentary simulation of the experiment. Bitwise XOR is nothing more than addition modulo 2. (XOR = CNOT)

The original Venus and the shared key are represented by the following matrices, where the matrix elements are pixels that are either off (0) or on (1).

$$\text{Venus} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \quad \text{Key} = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

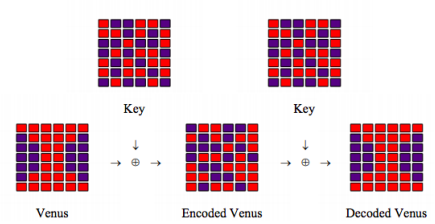
A coded version of Venus is prepared by adding Venus and the Key modulo 2 and sent to Bob.

$$i = 1..7 \quad j = 1..6 \quad \text{CVenus}_{i,j} = \text{Venus}_{i,j} \oplus \text{Key}_{i,j} \quad \text{CVenus} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

Bob adds the key to CVenus modulo 2 and sends the result to his printer.

$$\text{DVenus}_{i,j} = \text{CVenus}_{i,j} \oplus \text{Key}_{i,j} \quad \text{DVenus} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

A graphic summary of the simulation:



Random key production can be implemented as follows:

$$j = 1..20 \quad \text{PSA}_j = \text{trunc}(\text{rnd}(4))\ 90\ \text{deg} \quad \text{PSB}_j = \text{trunc}(\text{rnd}(2))\ 90\ \text{deg}$$

$$\text{Det0}_j = \left[\left| x^T \text{BS}\, \text{M} \begin{pmatrix} e^{i\,PSA_j} & 0 \\ 0 & e^{i\,PSB_j} \end{pmatrix} \text{BS}\, \text{x} \right| \right]^2 \quad \text{Det1}_j = \left[\left| y^T \text{BS}\, \text{M} \begin{pmatrix} e^{i\,PSA_j} & 0 \\ 0 & e^{i\,PSB_j} \end{pmatrix} \text{BS}\, \text{x} \right| \right]^2$$

$\frac{PSA_j}{deg} =$	$\frac{PSB_j}{deg} =$	$\mathrm{Det0}_j =$	$\mathrm{Det1}_j =$
0	0	1	0
0	90	0.5	0.5
180	0	0	1
90	90	1	0
270	90	0	1
0	0	1	0
270	90	0	1
0	90	0.5	0.5
270	0	0.5	0.5
0	90	0.5	0.5
180	90	0.5	0.5
90	90	1	0
180	90	0.5	0.5
180	90	0.5	0.5
0	0	1	0
...	...	...	...