BB84

TSINAME

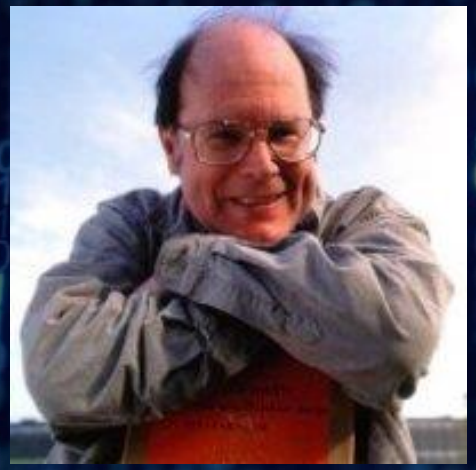
Three Scenarios Implementation

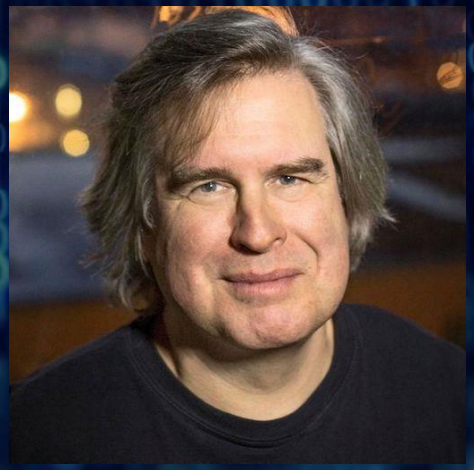
- No Attack
- Measurement – based attack
- Entanglement – based attack

THE BB84 PROTOCOL

Developed by Charles Bennett and Gilles Brassard in 1984, it is the first quantum cryptography protocol in history.

It is a quantum key distribution (QKD) scheme: using the properties of Quantum Mechanics, it allows a safe exchange of the key which will be used for encoding and decoding the message.





OVERVIEW

Alice and Bob have to communicate throughout a **public channel**.

Then they would like to share a **private key**, in order to encrypt their messages.

Let's assume that **the key is a string of N bits**, for example 1101010000110111100.

Then Alice has to send these bits to Bob, possibly avoiding that someone else will get hold of them.

They can do this by using the **BB84 protocol**.

OVERVIEW

- This protocol is based on **4 steps**:

1. **Encoding**
2. **Measurements**
3. **Comparison**
4. **Data analisys**

1. ENCODING

Alice encodes her bits, by preparing a qubit in a certain physical state for every bit.

Before the transmission, Alice and Bob had agreed on the encoding (and decoding) process.

Practically, Alice extracts a string of **N random bits** (they can be 0 or 1 with the **same probability**).

Comparing the i-th bit of the ''key string'' with the i-th bit of the random string, Alice prepares a **qubit** in the following way:

- If the ''key bit'' is **0** and the random bit is **0** $\longrightarrow$ $|0\rangle$
- If the ''key bit'' is **1** and the random bit is **0** $\longrightarrow$ $|1\rangle$
- If the ''key bit'' is **0** and the random bit is **1** $\longrightarrow$ $|+\rangle$
- If the ''key bit'' is **1** and the random bit is **1** $\longrightarrow$ $|-\rangle$

1. ENCODING

Map of the code:

Random string bits:	**0**			**1**		
Encoding:	**0**	$\mapsto$	$\|0\rangle$	**0**	$\mapsto$	$\|+\rangle$
	1	$\mapsto$	$\|1\rangle$	**1**	$\mapsto$	$\|-\rangle$

In fact, denoting with $|\psi\rangle$ the qubit's state:

- If the random bit is **0**, $|\psi\rangle$ is a state vector which belongs to the $\{|\mathbf{0}\rangle, |\mathbf{1}\rangle\}$ basis.
- If the random bit is **1**, $|\psi\rangle$ belongs to the $\{|+\rangle, |-\rangle\}$ basis.

That's why we refer to the ''random string'' as the **basis string**.

1. ENCODING

Relations between vectors of different bases:

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$
$$|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$
$$|0\rangle = \frac{1}{\sqrt{2}}(|+\rangle + |-\rangle)$$
$$|1\rangle = \frac{1}{\sqrt{2}}(|+\rangle - |-\rangle)$$

1. ENCODING

The $\{|\mathbf{0}\rangle, |\mathbf{1}\rangle\}$ vectors are the **eigenstates** of the operator **Z**, and $\{|+\rangle, |-\rangle\}$ are the **eingenstates** of the operator **X**, where Z and X are represented by the matrices:

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

By measuring the physical quantity represented by the operator Z or by the operator X, the qubit's state **collapses** in one of the four possible states of the two bases.

If the qubit's state was an **eigenstate of the measured operator**, the **result of the measurement is certain (probability equal to 1)**.
If not, the qubit's state can only be a eigenstate of the other operator; so **the result is probabilistic (with probability equal to 0,5)**.

TRANSMISSION…

- Quantum channel
- Devices
- Noise
- … many topics, but not covered here!

2. MEASUREMENTS

Bob receives the qubit. Now he has to measure it.

In the same way as Alice, Bob extracts a random string of bits. It is his own **basis string**.

Then he makes a measurement on the i-th qubit, using:

- the operator **Z**, if the i-th bit is equal to **0**;
- the operator **X**, if the i-th bit is equal to **1**.

The possible values which Bob can obtain are **0** or **1**.

Finally, Bob registers the measurement's result (the classical bit **0** or **1**) in another string, which will be called **results string**.

2. MEASUREMENTS

Example:

Alice's bit string:	**1**	*0*	**1**	*0*	**1**	*1*	**0**	**0**
Alice's basis string:	$\underline{0}$	1	$\underline{1}$	0	$\underline{0}$	0	$\underline{1}$	$\underline{0}$
Qubits:	$\|1\rangle$	$\|+\rangle$	$\|-\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|+\rangle$	$\|0\rangle$
Bob's basis string:	$\underline{0}$	0	$\underline{1}$	1	$\underline{0}$	1	$\underline{1}$	$\underline{0}$
Bob's results string:	**1**	*0*	**1**	*1*	**1**	*0*	**0**	**0**

In the table:
When Bob's results are **certain** (same choice of basis), they are printed in **bold**.
Instead, if they are *probabilistic* (different choice of basis), they are in *italics*.
When the same basis has been chosen, the relative bits are <u>underlined</u>.

3. COMPARISON

When all the qubits have been transmitted and measured, Alice and Bob:

1. Make their **basis strings public**;
2. Compare them **bit by bit**;
3. If the compared bits are ***different*** (which means that they have chosen ***two different bases***, so the result is ***probabilistic***), they ***discard*** them and the relative result;
4. If the compared bits are **identical** (so **the chosen bases are identical**, and then the result is **certain**), they **save** them and the relative result.

3. COMPARISON

Example:

Alice's bit string:	**1**	*0*	**1**	*0*	**1**	*1*	**0**	**0**
Alice's basis string:	0	1	1	0	0	0	1	0
Qubits:	$\|1\rangle$	$\|+\rangle$	$\|-\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|+\rangle$	$\|0\rangle$
Bob's basis string:	0	0	1	1	0	1	1	0
Bob's results string:	**1**	*0*	**1**	*1*	**1**	*0*	**0**	**0**
Saved (S) or Discarded (D):	S	D	S	D	S	D	S	S

4. DATA ANALISYS - CRUCIAL PASSAGE

The crucial point of the protocol is this: Alice shares pubblicly the **first half of her encoded bits**, and Bob does the same with the **first half of his results string**.

Then they compare the two half-strings bit by bit, without considering the discarded bits.

	Published				Private			
Alice's bit string:	1	0	1	0				0
Alice's basis string:	0	1	1	0			1	0
Qubits:	$\|1\rangle$	$\|+\rangle$	$\|-\rangle$	$\|0\rangle$			$\|+\rangle$	$\|0\rangle$
Bob's basis string:	0	0	1	1		1	1	0
Bob's results string:	1	0	1	1	1	0	0	0
Saved (S) or Discarded (D):	S	D	S	D	S	D	S	S

3. DATA ANALISYS - EXPECTATIONS

Finally, Alice and Bob do a little data analisys, in order to check that there had not been any interference caused by external agents (hackers).
They consider **only the "saved" bits**. It doesn't matter about the discarded bits!

They define:

- $\boldsymbol{n_{saved}} = total\ number\ of\ \boldsymbol{saved}\ results$
- $\boldsymbol{n_{identical}} = total\ number\ of\ results\ which\ are\ \boldsymbol{identical}$
  $to\ the\ relative\ original\ Alice's\ bits\ (''original''means\ the\ encoded\ Alice's\ bit)$
- $\boldsymbol{n_{different}} = n_{saved} - n_{identical}$
- $\boldsymbol{n_{i,j}}^{\boldsymbol{ide}} = number\ of\ results\ with\ \boldsymbol{value\ j}$
  $which\ have\ been\ obtained\ using\ the\ \boldsymbol{basis\ i}$
  $and\ which\ are\ \boldsymbol{identical}\ to\ the\ relative\ original\ Alice's\ bits$
- $\boldsymbol{n_{i,j}}^{\boldsymbol{diff}} = the\ same\ as\ before, but\ where\ the\ results$
  $are\ \boldsymbol{different}\ from\ the\ relative\ original\ Alice's\ bits$

$\boldsymbol{i, j = 0, 1}$

4. DATA ANALISYS - EXPECTATIONS

From these definitions, it is possibile to compute some interesting probabilities.

- $\boldsymbol{P_{identical}} = \frac{n_{identical}}{n_{saved}} = probability\ of\ obtaining\ a\ result\ wich\ is\ \boldsymbol{identical}$ $to\ the\ relative\ original\ Alice's\ bit$
- $\boldsymbol{P_{different}} = \frac{n_{different}}{n_{saved}} = probability\ of\ obtaining\ a\ result\ wich\ is\ \boldsymbol{different}$ $from\ the\ relative\ original\ Alice's\ bit$
- $\boldsymbol{P_i^{identical}(j)} = \frac{n_{i,j}{}^{ide}}{n_{saved}} = probability\ of\ obtaining\ the\ \boldsymbol{result\ j, identical}\ to\ the$ $relative\ original\ Alice's\ bit, having\ used\ the\ \boldsymbol{basis\ i}$
- $\boldsymbol{P_i^{different}(j)} = \frac{n_{i,j}{}^{diff}}{n_{saved}} = probability\ of\ obtaining\ the\ \boldsymbol{result\ j, different}\ from\ the$ $relative\ original\ Alice's\ bit, having\ used\ the\ \boldsymbol{basis\ i}$

$\boldsymbol{i, j = 0, 1}$

4. DATA ANALISYS - EXPECTATIONS

We assume that there is no noise (ideal situation).

First, we know that, if we measure an operator on his own eigenstates, the result is certain.

When Alice and Bob have chosen the same basis of eigenstates, they are doing exactly this.
This means that Bob will **always** obtain a result which is **identical** to the encoded bit of Alice:

- $\boldsymbol{P_{identical} = 100\ \%}$
- $\boldsymbol{P_{different} = 0\ \%}$

4. DATA ANALISYS - EXPECTATIONS

The same happens when we consider the probabilities of each possible single value of the results.

No result can be different from the relative Alice's encoded bit: each one must be identical to it.

Also, the probability that Alice chooses a certain basis is 50 %. So **she can encode the same bit in two different ways** with **50%** probability each.
The possible outcomes of Bob's results are **only two (0 or 1)**, with **50%** probability.
From another point of view, there are four possible codings, each with 25% probability, which are the four possible qubits.

- $P_i^{identical}(j) = P_{basis\ i} \cdot P_{result\ j} = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4} = 25\ \%$
- $P_i^{different}(j) = 0\ \%$

$i, j = 0, 1$

Example: $P_0^{identical}(1) = probability\ of\ obtaining\ the\ result\ 1\ using\ the\ basis\ 0$
$= probability\ of\ having\ prepared\ the\ qubit\ in\ the\ state\ |1\rangle$

4. DATA ANALISYS - EXPECTATIONS

The table sum up all these probabilities:

(%)	$P_{same\ basis}$	$P_{different\ basis}$	$P_{identical}$	$P_{different}$
Overall:	50	50	100	0
$P_i(j)$	$P_0(0)$	$P_0(1)$	$P_1(0)$	$P_1(1)$
Identical:	25	25	25	25
Different:	0	0	0	0

In the table there are also the probabilities $P_{same\ basis}$ and $P_{different\ basis}$, which are related to the probability that Alice and Bob had chosen the same basis (or not).
They are 50% each.

4. DATA ANALISYS - CONCLUSIONS

If the probabilities are the same as the expectations, Alice and Bob can conclude that the key transmission has been successfully completed.

Now they share a key which is made up by the bits they had measured with the same choice of basis. Referring to the next table, for example, the key is:

100

Alice's bit string:	1	0	1	0	**1**	1	**0**	**0**
Alice's basis string:	0	1	1	0	0	0	1	0
Qubits:	$\|1\rangle$	$\|-\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|+\rangle$	$\|0\rangle$
Bob's basis string:	0	0	1	1	0	1	1	0
Bob's results string:	1	0	1	1	**1**	0	**0**	**0**
Saved (S) or Discarded (D):	S	D	S	D	S	D	S	S

Public: not usable

4. DATA ANALISYS - CONCLUSIONS

This was an example with only a few bits (8 bits).

If Alice extracts **N bits** (N large) at the beginning of the protocol, the key will consist of only **N/4 bits**.

That's because 50% of the N bits will be discarded for a different choice of basis, and, after that, the 50% of the saved bits will be published by Alice and Bob (so they are no longer usable for the key).

All this works fine. But what if someone tries to hack the protocol?

HACKER'S ATTEMPTS

Suppose that a sneaky hacker called Eve would like to steal the key.

Eve can try to do that in several ways. In this project, we analyze only two particular types of hacker attack:

1. The **''measurement-based''** attack
2. The **''Entanglement-based''** attack

All these attacks are performed during the **transmission** of the qubits.

Now we will see how these attacks work, and why they **always lead to failure**, because Alice and Bob **can always detect Eve's presence**.

Notice: Eve can't just ''copy'' the qubits sent by Alice and use them for her purposes, because of the **No-Cloning Theorem**.

1. MEASUREMENT – BASED ATTACK

Eve intercepts each single qubit sent by Alice and makes a measurement on it.

She measures in the same way as Bob: for each qubit,

- She extracts a random bit strings which is her **basis string**;
- She measures with the operator **Z** if the bit is **0**, or with the operator **X** if the bit is **1**;
- She registers her results in her own **results string**.

Then she sends the measured qubit to Bob, in order to avoid being discovered.

When Alice and Bob share their basis strings, Eve compares her own basis string with them. So she knows that, in case the i-th bit of her string is **identical** to the i-th bit of the Alice's and Bob's strings, her measurement's result is **certain**.

1. MEASUREMENT – BASED ATTACK

But Eve's measurements introduce a variation in Bob's results. Let'us see why.

Let us say for convenience that we can make measurements with the operators **A** and **B** (where A and B are Z and X, or viceversa). Suppose that **Alice prepares the qubit in an eingenstate of the operator A**.

Then, four situations may have origin:

1. Eve measures the qubit with the operator **A**, and Bob also measures it with the operator **A**;
2. Eve measures the qubit with **A**, but Bob measures it with the operator **B**, different from A;
3. Eve measures the qubit with the operator **B**, and Bob also measures it with **B**;
4. Eve measures the qubit with the operator **B**, but Bob measures it with **A**.

1. MEASUREMENT – BASED ATTACK

We can **avoid considering cases 2. and 3.**; in fact, when Alice prepares the qubit in an eigenstate of the operator **A**, but Bob measures with the operator **B**, they are performing a **different choice of basis**. Therefore, the corrisponding bits are **discarded**.

Cases 1. and 4. are more interesting.

In case 1., Eve chooses the same basis of Alice and Bob. Then her measurement gives a **certain** result, which is the same result of Bob, identical to Alice's encoded bit.
When Eve measures with A, the state of the qubit, which **already** was in the eigenstate of A, **remains in the eigenstate of A**.
So, Bob ''finds the qubit'' in the same state in which Alice had prepared it.
Eve has achieved her purpose: she had intercepted a bit which will make up the key, and without being discovered, because she has not introduced any variation in Bob's result!

1. MEASUREMENT – BASED ATTACK

But case 4. is the crucial case.

In case 4., Eve measures with the ''wrong operator''. She has performed a choice of basis which is different from Alice's and Bob choice.

Then, if the qubit was prepared in an eigenstate of the operator A, when Eve measures it **with the operator B**, the qubit **collapses in an eigenstate of the operator B**.

So, when Bob measures the qubit **with the operator A**, he obtains a result which is **probabilistic** (not certain). Half the time, Bob gets **0**, but in the other cases **1**. This happens **despite Alice and Bob had performed the same choice of basis**!

So, Alice and Bob **can detect Eve's presence** in this case.

1. MEASUREMENT – BASED ATTACK

Let's see how they can reveal Eve's interaction.

Let us consider only the **"saved bits"** (the discarded bits, corrisponding to the 2. and 3. cases, have been thrown away).
We can only have the cases 1. and 4., each with **50% probability**.

In 50% of times, Eve does not perturb Bob's results (**case 1.**)

In the other 50% of times (**case 4.**), Bob can obtain the result **0** with 50% of probability, and idem **1** with 50% of probability.
This implies that **in 50% of cases** Bob obtains a result which is **different** to the Alice's encoded bit (example: Alice encodes 0 and Bob obtains 1, or Alice encodes 1 and Bob obtains 0).
But **in the other 50% of cases** Bob can obtain a result which is **identical** instead (example: Alice encodes 0 and Bob obtains 0, or Alice encodes 1 and Bob obtains 1). In this situation, Eve has been very lucky!

1. MEASUREMENT – BASED ATTACK

Thus we can calculate the following probabilities:

- $P_{identical} = P_{case\,1} + P_{case\,4} \cdot P_{same\,result} = \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = \frac{3}{4} = 75\%$
- $P_{different} = P_{case\,4} \cdot P_{different\,result} = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4} = 25\%$

Where: $P_{case\,1} = P_{case\,4} = 50\%$
And: $P_{same\,result} = P_{different\,result} = 50\%$

We can also calculate $P_i^{identical}(j)$ and $P_i^{different}(j)$:

- $P_i^{identical}(j) = P_{basis\,i} \cdot P_{result\,j} \cdot P_{identical} = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{3}{4} = \frac{3}{16} = 18,75\%$
- $P_i^{different}(j) = P_{basis\,i} \cdot P_{result\,j} \cdot (P_{case\,4} \cdot P_{different\,result}) = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{4} =$
  $= \frac{1}{16} = 6,25\%$

Where $i, j = 0, 1$ and $P_{basis\,i} = P_{basis\,j} = 50\%$

1. MEASUREMENT – BASED ATTACK

To sum up:

Scenarios	$P_{identical}$	$P_{different}$
No Attack:	100	0
Measurement-based attack:	75	25

	$P_i^{identical}(j)$				$P_i^{different}(j)$			
Scenarios	$P_0(0)$	$P_0(1)$	$P_1(0)$	$P_1(1)$	$P_0(0)$	$P_0(1)$	$P_1(0)$	$P_1(1)$
No attack:	25	25	25	25	0	0	0	0
M-attack:	18,75	18,75	18,75	18,75	6,25	6,25	6,25	6,25

1. MEASUREMENT – BASED ATTACK

Comments about this attack.

Alice and Bob can detect Eve's presence simply by analyzing the probability of obtaining results which are different form Alice's encoded bits.

If this probability is $P_{different} \approx 25\% \neq 0\%$, then they can say that a hacker had tried to attack the transmission.

It is remarkable that, **in any case, Eve fails her purpose**.
In fact, when Eve measures the qubits, she must discard the results which Alice and Bob, on their own, threw away because of a different choice of basis. This happens in 50% of times. But Eve also must discard 50% of her results, because they correspond to a choice of basis which is different from Alice's choice (and then they give only probabilistic results).
So, Eve can only recover the 25% of the key's bits, not the whole key!

2. ENTANGLEMENT – BASED ATTACK

Eve can also try to create an entangled state with Alice's qubits.

Suppose that Eve makes up an entangled state between her own qubit, prepared in the state $|0\rangle$, and Alice's transmitted qubit. This is implemented using a CNOT gate. Eve's purpose now is to **wait for the end of Bob's measurement, and after that to measure her own qubits with the operator Z, in order to obtain the same results as Bob!**

However, Eve can create entangled states only in 2 out of 4 cases. That's because, if Alice prepares the qubit in the state:

- $|0\rangle \longrightarrow$ the system state becomes $|00\rangle \longrightarrow$ CNOT $|00\rangle = |00\rangle \longrightarrow$ **Not entangled**
- $|1\rangle \longrightarrow |01\rangle \longrightarrow$ CNOT $|01\rangle = |11\rangle \longrightarrow$ **Not entangled**
- $|+\rangle \longrightarrow |0+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |01\rangle) \longrightarrow$ **CNOT** $|0+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \longrightarrow$ **Entangled**
- $|-\rangle \longrightarrow |0-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |01\rangle) \longrightarrow$ **CNOT** $|0-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \longrightarrow$ **Entangled**

2. ENTANGLEMENT – BASED ATTACK

Then Eve can create entangled states only in the 50% of cases.

This happens, as said before, only if Alice had prepared the qubit in the state $|+\rangle$ or in the state $|-\rangle$. This corresponds to the case in which Alice chooses the **''basis 1''** (eigenstates of the operator X).
In the other cases, Eve has not interacted with Alice's qubits.

Considering the ''saved bits'' (same choice of basis):

- If Alice chose the **0 basis** (eigenstates of the operator Z), then Bob chose the **0 basis**, and there are **not entangled states** (50% probability);
- If Alice chose the **1 basis**, then Bob chose the **1 basis** and there are **only entangled states** (50% probability).

This means that Bob measures the entangled states **only with the operator X.**

2. ENTANGLEMENT – BASED ATTACK

We can write the entangle state (for example) **CNOT |0 +⟩** in this way:

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(|0\rangle\,|0\rangle + |1\rangle\,|1\rangle) =$$
$$= \frac{1}{\sqrt{2}}[\frac{1}{\sqrt{2}}(|+\rangle + |-\rangle)(\frac{1}{\sqrt{2}}(|+\rangle + |-\rangle) + \frac{1}{\sqrt{2}}(|+\rangle - |-\rangle)(\frac{1}{\sqrt{2}}(|+\rangle - |-\rangle)] =$$
$$= \frac{1}{\sqrt{2}}[\frac{1}{2}(2 \cdot |++\rangle + 2 \cdot |--\rangle)] =$$
$$= \frac{1}{\sqrt{2}}(|++\rangle + |--\rangle)$$

The we can see that, when Bob measures his entangled qubit with the operator **X**, he obtains a **probabilistic result**, in spite of having chosen the same basis as Alice!

Notice that an analogous reasoning can be made for **CNOT |0 –⟩**.

2. ENTANGLEMENT – BASED ATTACK

Let us compute the usual probabilities.

In 50% of cases only, Eve creates entangled states. A Bob's measurement on the entangled qubit produces a **probabilistic result**, which is in **50%** of times **identical** to the encoded Alice's bit, and in the other **50%** of times is **different** from it.

In the other 50% of cases, Eve does not create entangled states. So the situation in this case is the same as in the ''no attack'' scenario, where the **100% of Bob's results are identical to the relative encoded Alice's bits**.

- $P_{identical} = P_{non\,entangled} \cdot 100\% + P_{entangled} \cdot P_{same\,result} = \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot \frac{1}{2} = \frac{3}{4} = 75\%$
- $P_{different} = P_{non\,entangled} \cdot 0\% + P_{entangled} \cdot P_{different\,result} = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4} = 25\%$

Where: $P_{entangled} = P_{non\,entangled} = 50\%$

And: $P_{same\,result} = P_{different\,result} = 50\%$

Notice that $P_{non\,entangled} = P_{basis\,0}$ and $P_{entangled} = P_{basis\,1}$

2. ENTANGLEMENT – BASED ATTACK

We can also calculate $P_i^{identical}(j)$ and $P_i^{different}(j)$.

Now we need to distinguish between the choice of the basis 0 and the choice of the basis 1.

Basis 0 (non entangled states):

- $P_0^{identical}(j) = P_{basis\,0} \cdot P_{result\,j} \cdot P_{identical} = \frac{1}{2} \cdot \frac{1}{2} \cdot 1 = \frac{1}{4} = 25\%$
- $P_0^{different}(j) = P_{basis\,0} \cdot P_{result\,j} \cdot P_{different} = \frac{1}{2} \cdot \frac{1}{2} \cdot 0 = 0\%$

Basis 1 (entangled states):

- $P_1^{identical}(j) = P_{basis\,1} \cdot P_{result\,j} \cdot P_{identical} = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{8} = 12,5\%$
- $P_1^{different}(j) = P_{basis\,1} \cdot P_{result\,j} \cdot P_{different} = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = 12,5\%$

Where $j = 0, 1$

2. ENTANGLEMENT – BASED ATTACK

To sum up:

Scenarios	$P_{identical}$	$P_{different}$
No Attack:	100	0
Measurement-based attack:	75	25
Entanglement-based attack:	75	25

	$P_i^{identical}(j)$				$P_i^{different}(j)$			
Scenarios	$P_0(0)$	$P_0(1)$	$P_1(0)$	$P_1(1)$	$P_0(0)$	$P_0(1)$	$P_1(0)$	$P_1(1)$
No attack:	25	25	25	25	0	0	0	0
M-attack:	18,75	18,75	18,75	18,75	6,25	6,25	6,25	6,25
E-attack:	25	25	12,25	12,25	0	0	12,25	12,25

2. ENTANGLEMENT – BASED ATTACK

Comments:

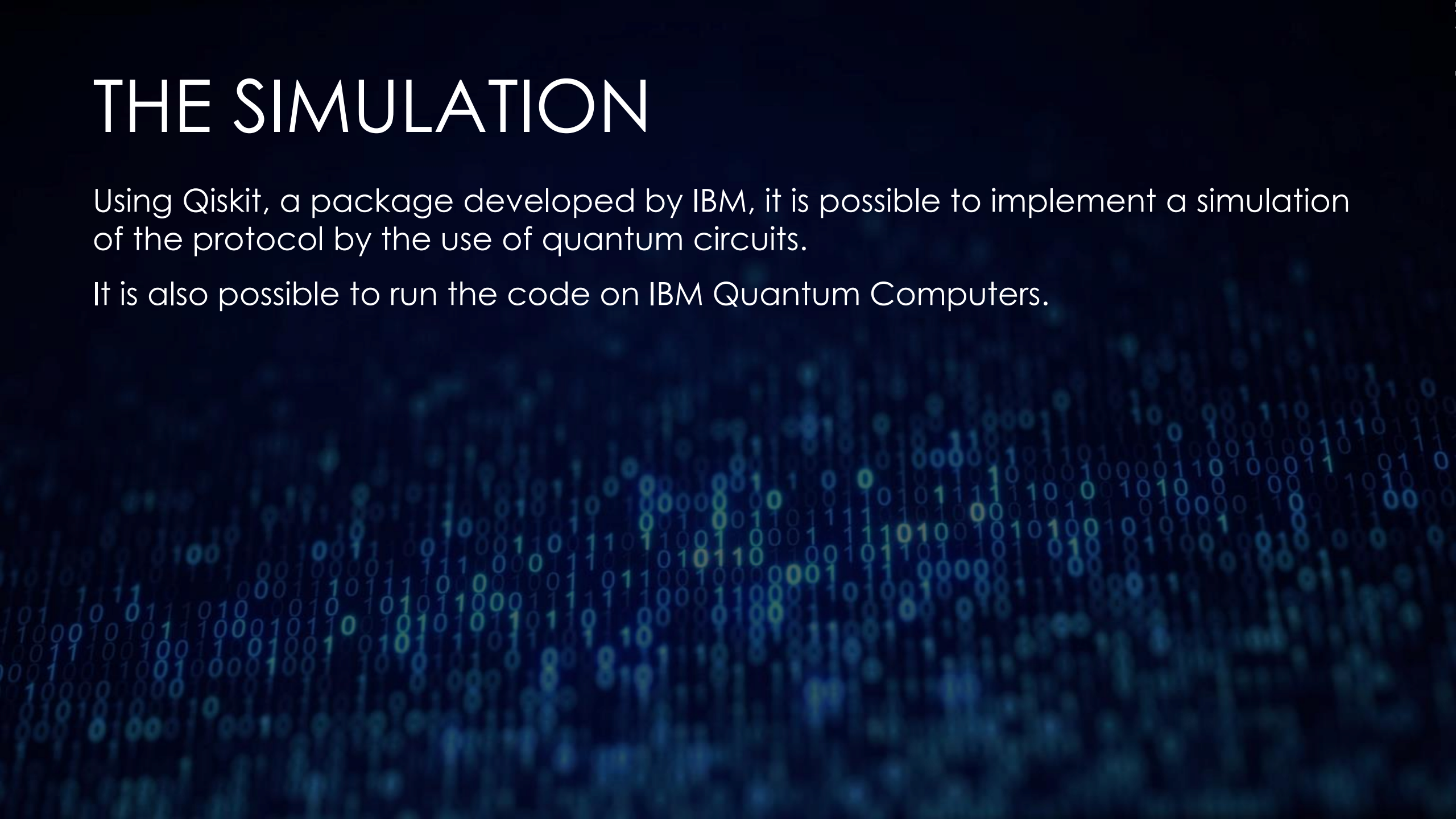
Also in this case, it is possible for Alice and Bob to detect Eve's presence through the analysis of the probability of obtaining results which are different from the original Alice's bits.
Although $P_{different} = 25\%$ is the same in each type of attack, we can distinguish between them comparing $\boldsymbol{P_i}^{\boldsymbol{identical}}(\boldsymbol{j})$ and $\boldsymbol{P_i}^{\boldsymbol{different}}(\boldsymbol{j})$ in the two scenarios of attack (see the above tables).

Performing this type of attack, Eve can recover moreless half of the results of Bob. In spite of that, she introduces a bigger variation in Bob's results, so it is easier for him and Alice to discover her.

THE SIMULATION

Using Qiskit, a package developed by IBM, it is possible to implement a simulation of the protocol by the use of quantum circuits.

It is also possible to run the code on IBM Quantum Computers.



THE CODE

The simulation code is called ‘’bb84_qkd.py’’.

In order to make it cleaner, some functions have been defined before.

THE CODE

X-measurement function:

```
# Implementation of the measurement
# in the {|+>, |->} basis: HZH = X

def x_measure (quantumcircuit, qubit, cbit):
quantumcircuit.h(qubit)
quantumcircuit.measure(qubit, cbit)
quantumcircuit.h(qubit)
return quantumcircuit
```

$$HZH = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix} = X$$

THE CODE

Quantum circuits builder function:

```
# Implementation of a function which builds up
# a single qubit circuit based on Alice's strings

def encoding_circuit_builder (alice_bits, alice_basis, nth_qubit):

i = nth_qubit   # The number n associated to the n-th bit
# which Alice wants to encode

encoding_circuit = QuantumCircuit(1,1)

if alice_bits[i] == 0 and alice_basis[i] == 0:
# Alice chooses {|0>, |1>} basis
pass # Apply I (nothing happens)

if alice_bits[i] == 1 and alice_basis[i] == 0:
# Alice chooses {|0>, |1>} basis
encoding_circuit.x(0) # Apply X-Gate (flip |0> to |1>)

if alice_bits[i] == 0 and alice_basis[i] == 1:
# Alice chooses {|+>, |->} basis
encoding_circuit.h(0) # Apply H-Gate (change |0> to |+>)

if alice_bits[i] == 1 and alice_basis[i] == 1:
# Alice chooses {|+>, |->} basis
encoding_circuit.x(0)
encoding_circuit.h(0) # Apply X-Gate and H-Gate
# (so |0> goes in |->)

encoding_circuit.barrier()

return encoding_circuit
```

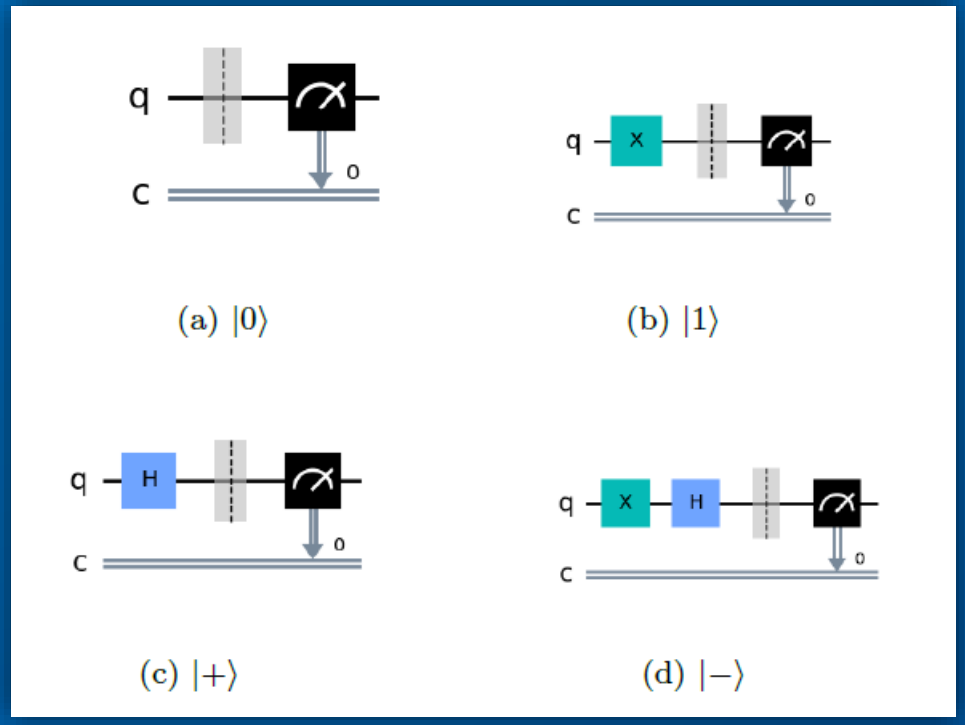
THE CODE

Bob's measurements function:

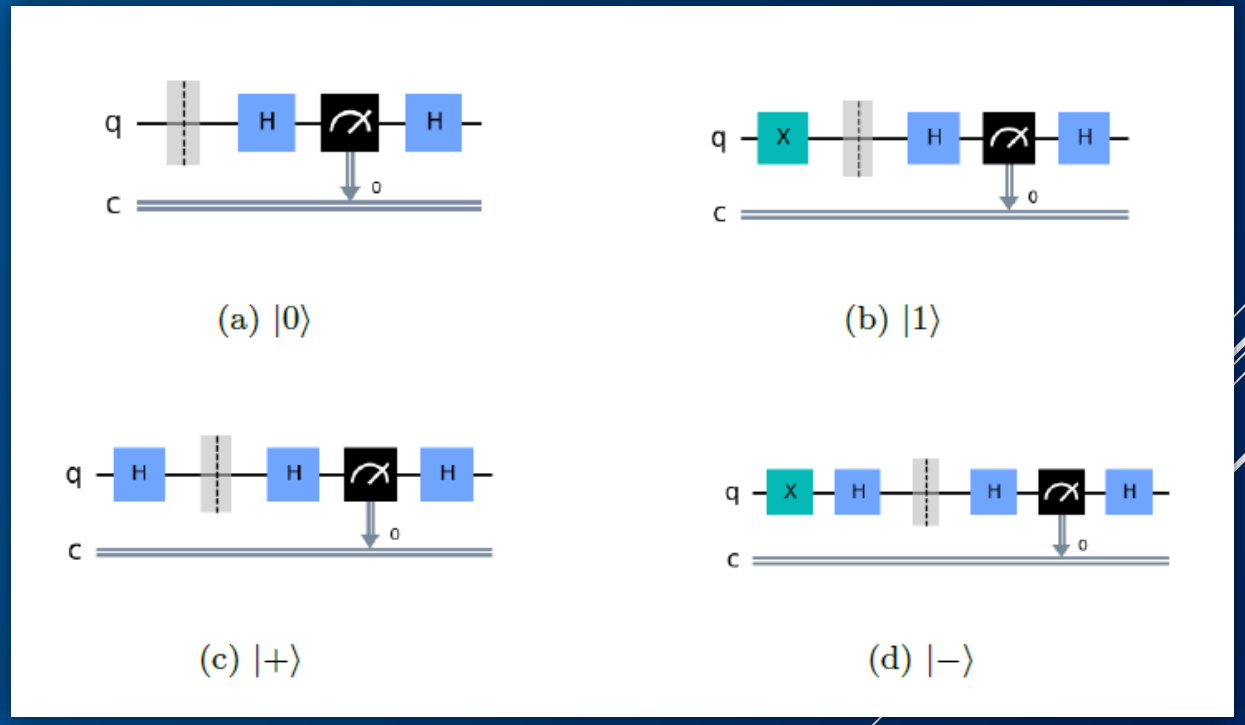
```
# Implementation of the function with which Bob measures Alice's qubit

def circuit_measure (encoding_circuit, bob_basis, bob_measures, nth_qubit):

i = nth_qubit    # The n-th qubit sent by Alice

if bob_basis[i] == 0: # Bob chooses {|0>, |1>} basis
# Measure with the default {|0>, |1>} basis
encoding_circuit.measure(0,0)

# To draw the circuit: encoding_circuit.draw("mpl")

# Now we run the circuit ONLY ONE TIME, and memorize
# the result in the bob_measures list.

backend = Aer.get_backend("qasm_simulator")
job = execute(encoding_circuit, backend, shots=1, memory=True)
result = job.result()
list_of_results = result.get_memory()
bob_measures.append(list_of_results[0])

if bob_basis[i] == 1: # Bob chooses {|+>, |->} basis
# Measure with the {|+>, |->} basis
x_measure(encoding_circuit, 0, 0)

# encoding_circuit.draw("mpl")

# Now we run the circuit ONLY ONE TIME,
# and memorize the result in the bob_measures list.

backend = Aer.get_backend("qasm_simulator")
job = execute(encoding_circuit, backend, shots=1, memory=True)
result = job.result()
list_of_results = result.get_memory()
bob_measures.append(list_of_results[0])
```

THE CODE

The quantum circuit's diagrams:



Measurements with Z



Measurement with X

THE CODE

Measurement-based attack function:

```
def eve_hacking_measure (hacker_activated, encoding_circuit, \
eve_measures, nth_qubit):

if hacker_activated == True:

# Eve measures the n-th qubit sent by Alice.
# After that, Eve sends it to Bob:

i = nth_qubit    # The n-th qubit sent by Alice

if eve_basis[i] == 0: # Eve chooses {|0>, |1>} basis
# Measure with the default {|0>, |1>} basis
encoding_circuit.measure(0,0)

backend = Aer.get_backend("qasm_simulator")
job = execute(encoding_circuit, backend, shots=1, memory=True)
result = job.result()
list_of_results = result.get_memory()
eve_measures.append(list_of_results[0])

if eve_basis[i] == 1: # Eve chooses {|+>, |->} basis
# Measure with the {|+>, |->} basis
x_measure(encoding_circuit, 0, 0)

backend = Aer.get_backend("qasm_simulator")
job = execute(encoding_circuit, backend, shots=1, memory=True)
result = job.result()
list_of_results = result.get_memory()
eve_measures.append(list_of_results[0])

return encoding_circuit

else:
pass
```

THE CODE

Entanglement-based attack function:

```
def eve_hacking_entangle (hacker_activated, encoding_circuit, nth_qubit):

if hacker_activated == True:

# Eve ENTANGLES the n-th qubit sent by Alice with a |0> state qubit.
# After that, Eve sends the entangled qubit to Bob:

eve_q = QuantumRegister(1, "eve_qubit")
encoding_circuit.add_register(eve_q)

encoding_circuit.cx(0, eve_q[0])
encoding_circuit.barrier()

# encoding_circuit.draw("mpl")

return encoding_circuit
```

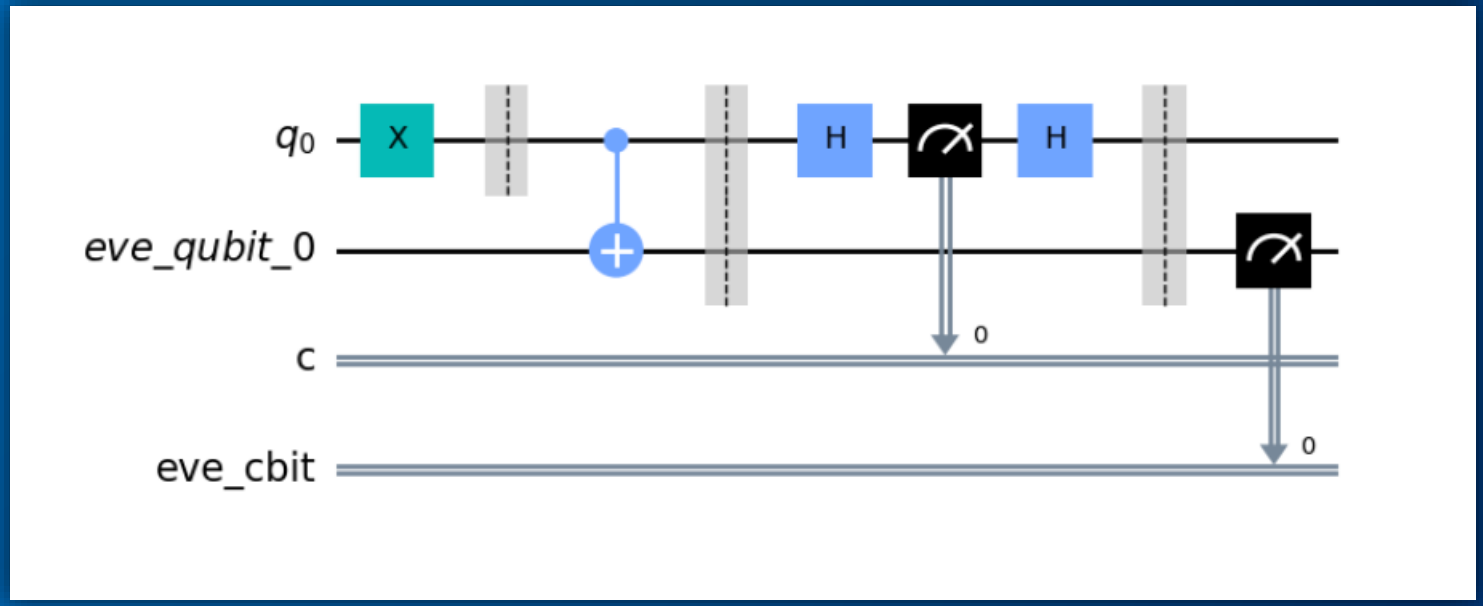
THE CODE

Entanglement-based attack: hacker's measurements function:

```
def eve_entangled_measurement (hacker_activated, encoding_circuit, \
eve_measures, nth_qubit):

if hacker_activated == True:

# Eve measures all her entangled qubits in the {|0>, |1>} basis:

i = nth_qubit    # The number n associated to the n-th qubit

eve_c = ClassicalRegister(1, "eve_cbit")
encoding_circuit.add_register(eve_c)

encoding_circuit.barrier()
encoding_circuit.measure(1,1)

# encoding_circuit.draw("mpl")

backend = Aer.get_backend("qasm_simulator")
job = execute(encoding_circuit, backend, shots=1, memory=True)
result = job.result()
list_of_results = result.get_memory()
eve_measures.append(list_of_results[0])
```

THE CODE

Entangled qubits diagram:



A **CNOT |0 +⟩** state is measured with the operator X by Bob, and Then is measured with the operator Z by Eve.

THE CODE

Main program: extraction of Alice's and Bob's random strings

```
# Number of bits (and then qubits) that Alice is going to use:
number_of_qubits = 10000

# Alice generates n random bits (some of these bits will form the key)

alice_bits = []
for i in range (number_of_qubits):
alice_bits.append(randint(0,1))

print("\nAlice's bits (first 20 bits):\n", alice_bits[0:19])

# Alice randomly chooses the bases in which she is going to measure

alice_basis = []
for i in range (number_of_qubits):
alice_basis.append(randint(0,1))
print("\nAlice's basis (first 20 bits):\n", alice_basis[0:19])

# Bob also randomly chooses the bases in which he is going to measure

bob_basis = []
for i in range (number_of_qubits):
bob_basis.append(randint(0,1))
print("\nBob's basis (first 20 bits):\n", bob_basis[0:19])
```

THE CODE

Main program: choice of the scenario

```
print("\nChoose an option [digit 1, 2 or 3]:")
print("\n1. Transmission without hacker's attack")
print("\n2. Transmission with a measurement-based hacker's attack" \
"\n3. Transmission with an entanglement-based hacker's attack\n")
scelta = input()

if scelta == "1":
hacker_activated1 = False
hacker_activated2 = False
if scelta == "2":
hacker_activated1 = True
hacker_activated2 = False
if scelta == "3":
hacker_activated1 = False
hacker_activated2 = True
if scelta != "1" and scelta != "2" and scelta != "3":
print("\nTry again (digit only 1, 2 or 3)")
```

THE CODE

Main program: extraction of Eve's basis string (only if Scenario 2 has been selected)

```
# Eve randomly chooses the bases in which
# she is going to measure (like Bob)
if hacker_activated1 == True:
eve_basis = []
for i in range (number_of_qubits):
eve_basis.append(randint(0,1))
print("\nEve's basis (first 20 bits):\n", eve_basis[0:19])
```

THE CODE

Main program: the simulation

```
# For each classical bit which Alice wants to encode
# and transmit to Bob, they proceed as it follows:

bob_measures = []
eve_measures = []

for n in range(number_of_qubits):

# Alice codes the n-th bit of her initial string
# as a qubit and sends it to Bob
qubit = encoding_circuit_builder(alice_bits, alice_basis, n)

# Bob measures the qubit with his own basis:
# but what if Eve is hacking the message?
eve_hacking_measure (hacker_activated1, qubit, eve_measures, n)
eve_hacking_entangle (hacker_activated2, qubit, n)
circuit_measure (qubit, bob_basis, bob_measures, n)
eve_entangled_measurement (hacker_activated2, qubit, eve_measures, n)
```

THE CODE

Main program: exporting the datas

```
# Let's see the results of the measurements!

print("\nBob's measurements (first 20 measurements):\n")
print(bob_measures[0:19])

if hacker_activated1 == True or hacker_activated2 == True:
print("\nEve's measurements (first 20 measurements):\n")
print(eve_measures[0:19])

# Now we export the results in a text file:

data_file = open("bb84_data.txt", "w")

for i in range(number_of_qubits):
data_file.write(str(alice_bits[i]))
data_file.write("\t")
data_file.write(str(alice_basis[i]))
data_file.write("\t")
data_file.write(str(bob_basis[i]))
data_file.write("\t")
data_file.write(str(bob_measures[i]))
data_file.write("\n")

data_file.close()

plt.show()
```

DATA ANALYSIS OF THE SIMULATION

In order to perform the data analysis, we use the program ''bb84_data_analysis.py''.

For each scenario, 10 runs of the program ''bb84_qkd.py'' have been launched. Then, for each simulation, a data analysis has been performed.

Finally, we have calculated the mean over all the probabilities of a certain event, for each probability computated in the simulation.

The error on each mean is computed with the standard deviation divided by $\sqrt{10}$, and it is denoted by σ.

DATA ANALYSIS OF THE SIMULATION

Remind - the theoretical expectations are:

Scenarios	$P_{identical}$	$P_{different}$
No Attack:	100	0
Measurement-based attack:	75	25
Entanglement-based attack:	75	25

	$P_i^{identical}(j)$				$P_i^{different}(j)$			
Scenarios	$P_0(0)$	$P_0(1)$	$P_1(0)$	$P_1(1)$	$P_0(0)$	$P_0(1)$	$P_1(0)$	$P_1(1)$
No attack:	25	25	25	25	0	0	0	0
M-attack:	18,75	18,75	18,75	18,75	6,25	6,25	6,25	6,25
E-attack:	25	25	12,25	12,25	0	0	12,25	12,25

DATA ANALYSIS OF THE SIMULATION

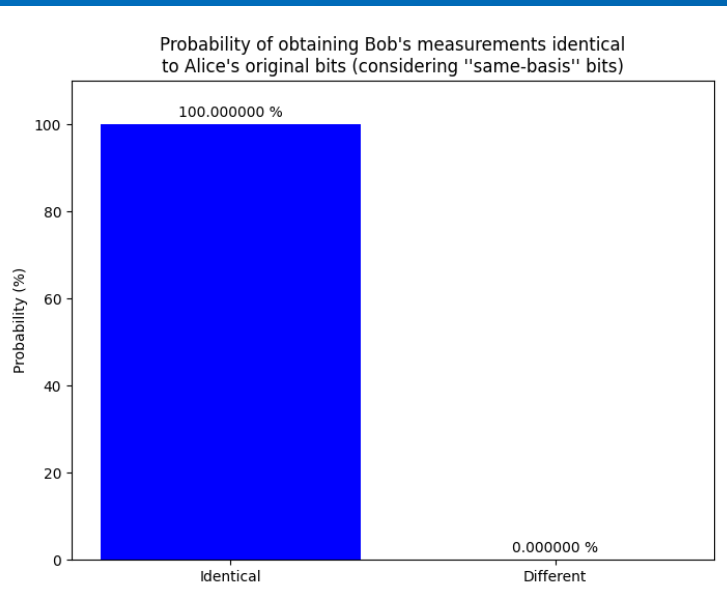
Scenario 1: No Attack

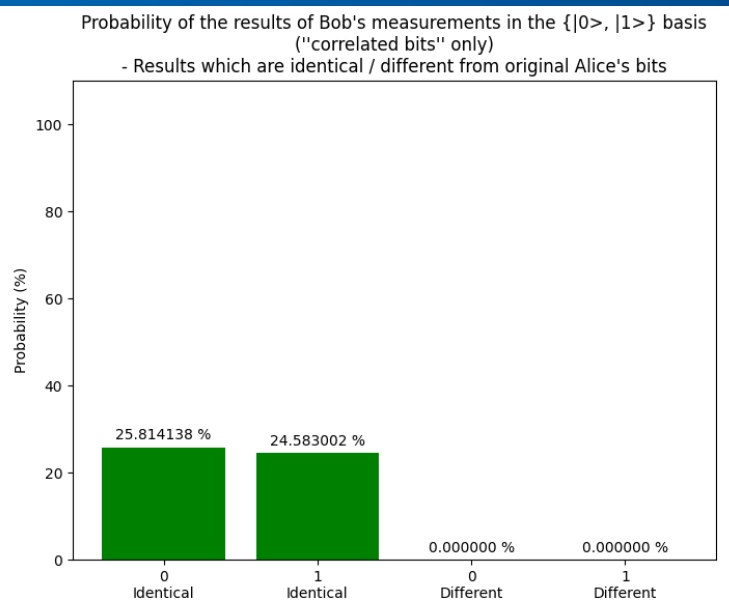
	$P_{same\ basis}$	σ	$P_{different\ basis}$	σ	$P_{identical}$	σ	$P_{different}$	σ
Overall:	50,1	0,1	49,9	0,1	100	0	0	0
$P_i(j)$	$P_0(0)$	σ	$P_0(1)$	σ	$P_1(0)$	σ	$P_1(1)$	σ
Identical:	25,5	0,3	24,8	0,3	24,8	0,2	24,8	0,4
Different:	0	0	0	0	0	0	0	0

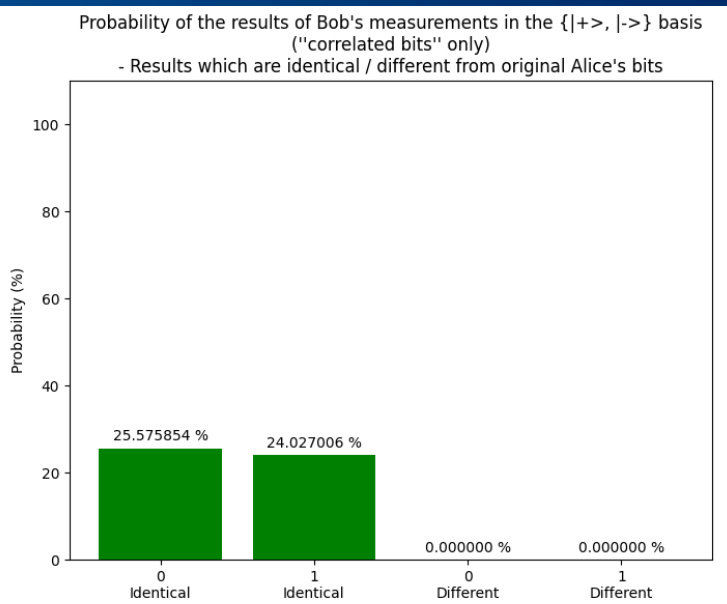
Notice: σ is the error on the probabilities.

DATA ANALYSIS OF THE SIMULATION

Scenario 1: No Attack







(Run number 5)

DATA ANALYSIS OF THE SIMULATION

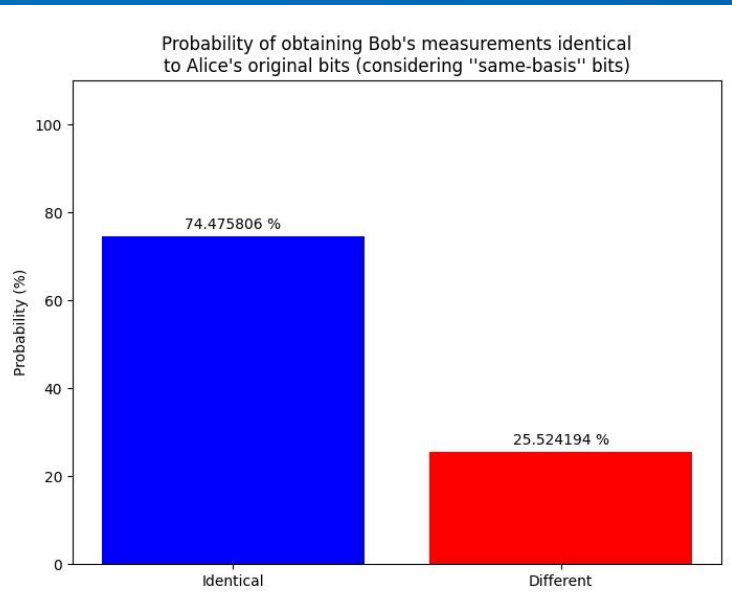
Scenario 2: Measurement-based Attack

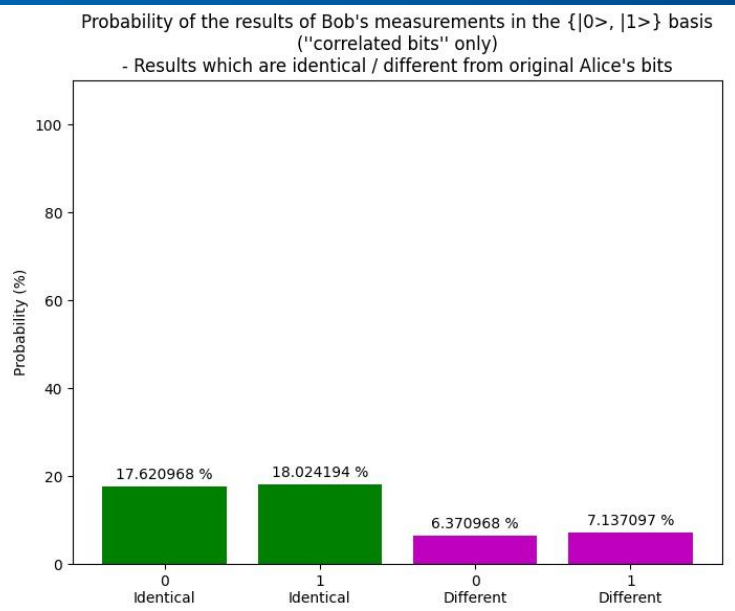
	$P_{same\ basis}$	σ	$P_{different\ basis}$	σ	$P_{identical}$	σ	$P_{different}$	σ
Overall:	49,9	0,1	50,0	0,2	75,2	0,2	24,8	0,2
$P_i(j)$	$P_0(0)$	σ	$P_0(1)$	σ	$P_1(0)$	σ	$P_1(1)$	σ
Identical:	18,8	0,3	19,0	0,3	18,8	0,2	18,7	0,2
Different:	6,4	0,1	6,1	0,2	6,4	0,1	5,9	0,1

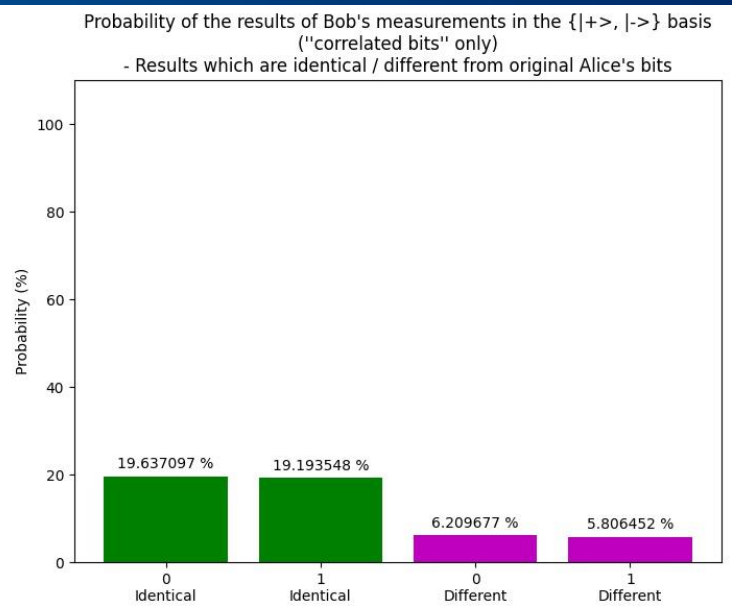
Notice: σ is the error on the probabilities.

DATA ANALYSIS OF THE SIMULATION

Scenario 2: Measurement-based Attack







(Run number 5)

DATA ANALYSIS OF THE SIMULATION

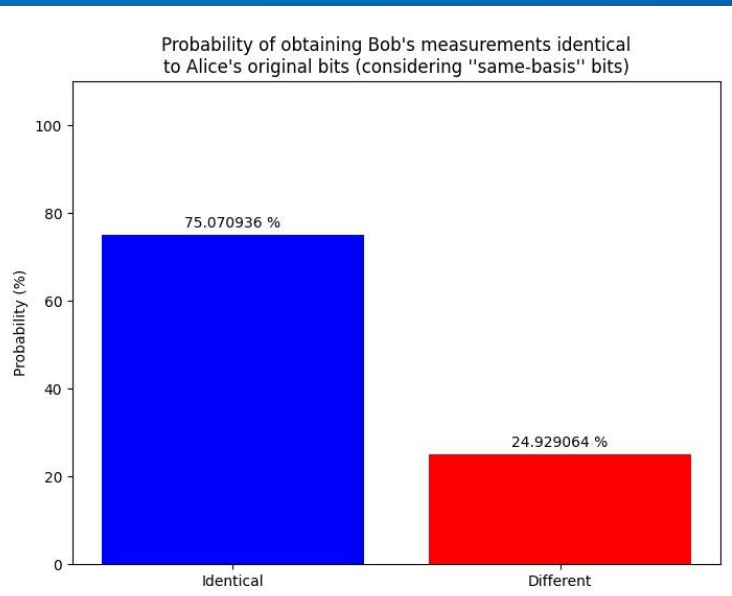
Scenario 3: Entanglement-based Attack

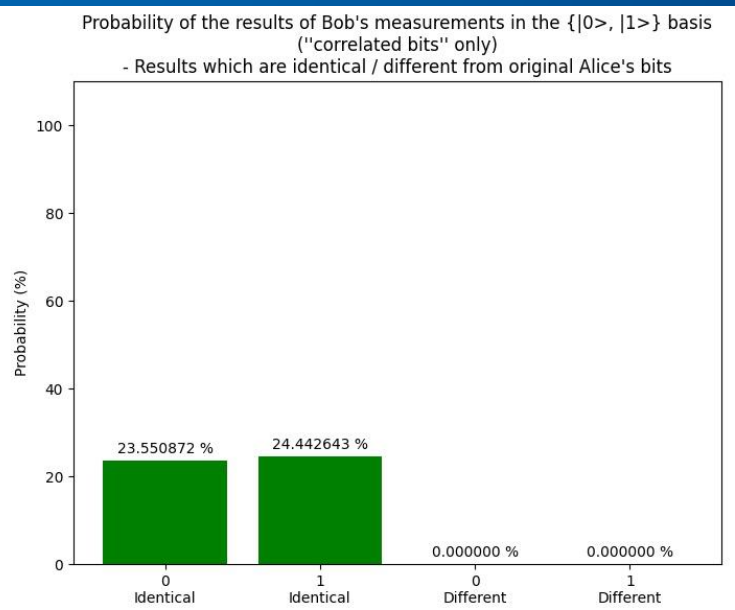
	$P_{same\ basis}$	σ	$P_{different\ basis}$	σ	$P_{identical}$	σ	$P_{different}$	σ
Overall:	49,9	0,1	50,1	0,1	74,9	0,2	25,1	0,2
$P_i(j)$	$P_0(0)$	σ	$P_0(1)$	σ	$P_1(0)$	σ	$P_1(1)$	σ
Identical:	24,8	0,3	24,9	0,3	12,6	0,2	12,6	0,2
Different:	0	0	0	0	12,5	0,1	12,6	0,2

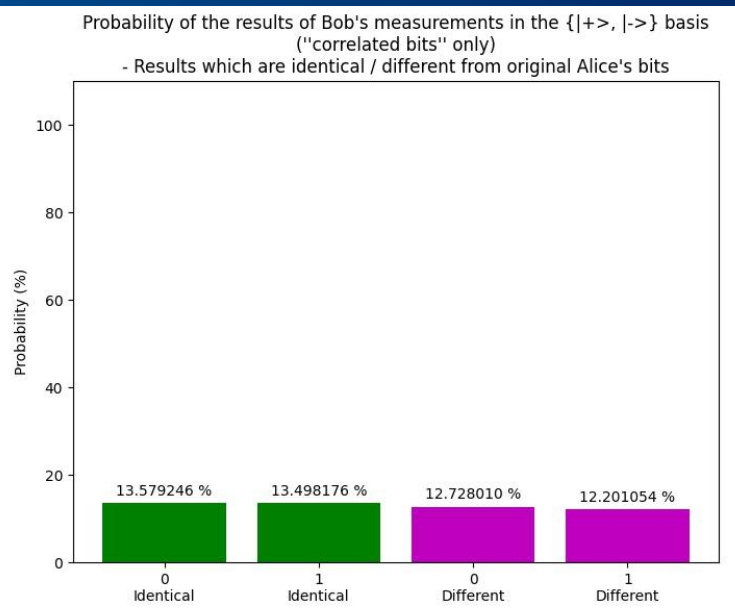
Notice: σ is the error on the probabilities.

DATA ANALYSIS OF THE SIMULATION

Scenario 3: Entanglement-based Attack







(Run number 5)

GOING FURTHER

Next steps:

- Analyzing the noise of a real device;
- Establishing a threshold for the noise;
- Performing the experiment on IBM Quantum Computers.