atelier
protocole BB84 pour l'échange sécurisé des clefs de cryptage (QKD)

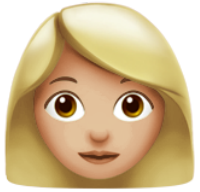
Jean-Michel Torres, IBM Quantum Hub : torresjm@fr.ibm.com

ALICE:  , BOB:  , EVE: 

Alice wants to send a secret message to Bob, she may :

- Protect the communication media
- Hide or encrypt the message,

…to avoid Eve accessing the message



«**BB84** is a Quantum Key distribution process developed by Charles Bennett and Gilles Brassard in 1984.»

HIDING THE TEXT (STEGANOGRAPHY)

History [edit]

The first recorded uses of steganography can be traced back to 440 BC in Greece, when Herodotus mentions two examples in his *Histories*.[4] Histiaeus sent a message to his vassal, Aristagoras, by shaving the head of his most trusted servant, "marking" the message onto his scalp, then sending him on his way once his hair had regrown, with the instruction, "When thou art come to Miletus, bid Aristagoras shave thy head, and look thereon." Additionally, Demaratus sent a warning about a forthcoming attack to Greece by writing it directly on the wooden backing of a wax tablet before applying its beeswax surface. Wax tablets were in common use then as reusable writing surfaces, sometimes used for shorthand.

In his work *Polygraphiae,* Johannes Trithemius developed his so-called "Ave-Maria-Cipher" that can hide information in a Latin praise of God. "*Auctor Sapientissimus Conseruans Angelica Deferat Nobis Charitas Potentissimi Creatoris*" for example contains the concealed word *VICIPEDIA*.[5]

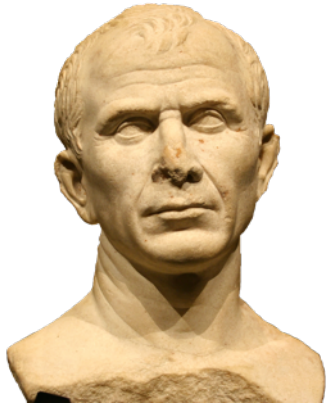
Image of a tree with a steganographically hidden image. The hidden image is revealed by removing all but the two least significant bits of each color component and a subsequent normalization. The hidden image is shown below.

Image of a cat extracted from the tree image above.

👎 You need to communicate the « process », at least once.

source wikipedia (texts and images)

MAKE THE MESSAGE NON READABLE : CRYPTOGRAPHY



a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
t	b	w	g	j	r	o	p	h	y	l	i	z	n	k	q	f	s	v	a	x	d	e	u	m	c

mrsppbowoxd no mocka

qixv ghrrhwhij zthv xnj ijaasj jva akxykxsv wkgjj gj it zjzj ztnhjsj, hi jva tvvjc rtwhij gj rthsj xnj taatfxj vatahvahfxj

Vigénère : to avoid coding a given letter by the same letter at each occurrence, for a message of length N, a length k key is used, and each letter is coded using the n^{th} letter in the key. In the following example, the key is « hello ».

b	o	n	j	o	u	r	t	o	u	t	l	e	m	o	n	d	e
h(8)	e(5)	l(12)	l(12)	o(15)	h(8)	e(5)	l(12)	l(12)	o(15)	h(8)	e(5)	l(12)	l(12)	o(15)	h(8)	e(5)	l(12)
j	t	z	v	d	c	w	p	a	j	b	q	q	y	d	v	i	q

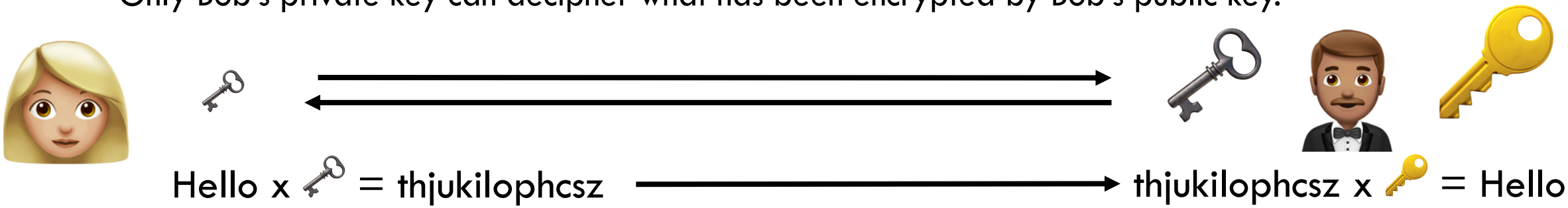
more : https://fr.wikipedia.org/wiki/Chiffre_de_Vigen%C3%A8re.

source wikipedia images

RSA (« RIVEST, SHAMIR, ADELMAN »):

RSA, also called asymetrical key encryption, does not require to communicate the encryption key.

- Alice wants to send a message to Bob.
- She uses Bob's public key to encrypt
- Only Bob's private key can decipher what has been encrypted by Bob's public key.



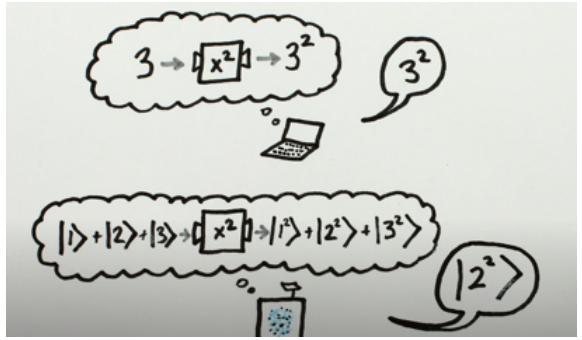
This relies on the fact that is is very difficult to decompose a large integer into it 's prime factors.

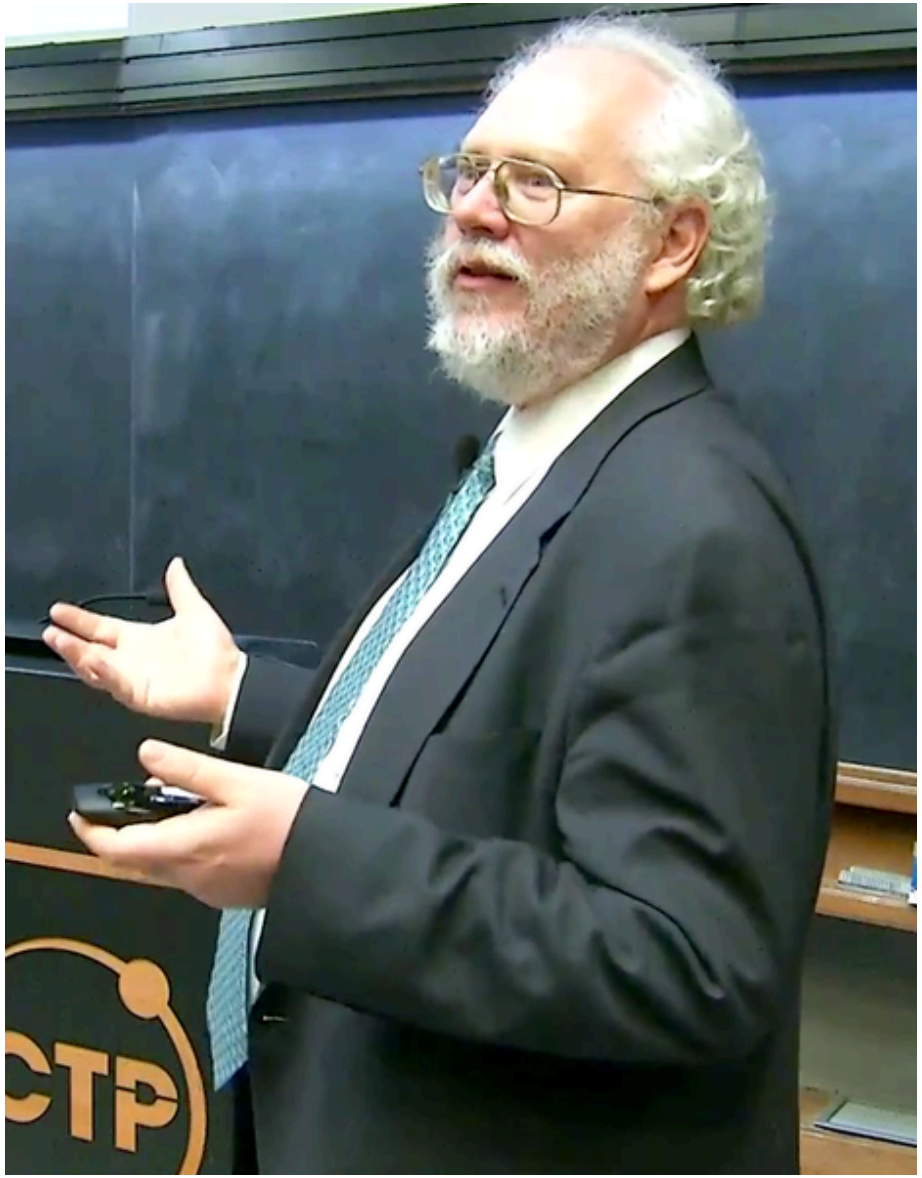
more : https://fr.wikipedia.org/wiki/Chiffrement_RSA

SHOR COMES IN !

In 1994, Professor Peter Shor of MIT proves a quantum algorithm (using superposition and entanglement) can factor large numbers in an exponentially shorter than time on any classical algorithm.

YT Minute Physics :
https://youtu.be/lvTqbM5Dq4Q





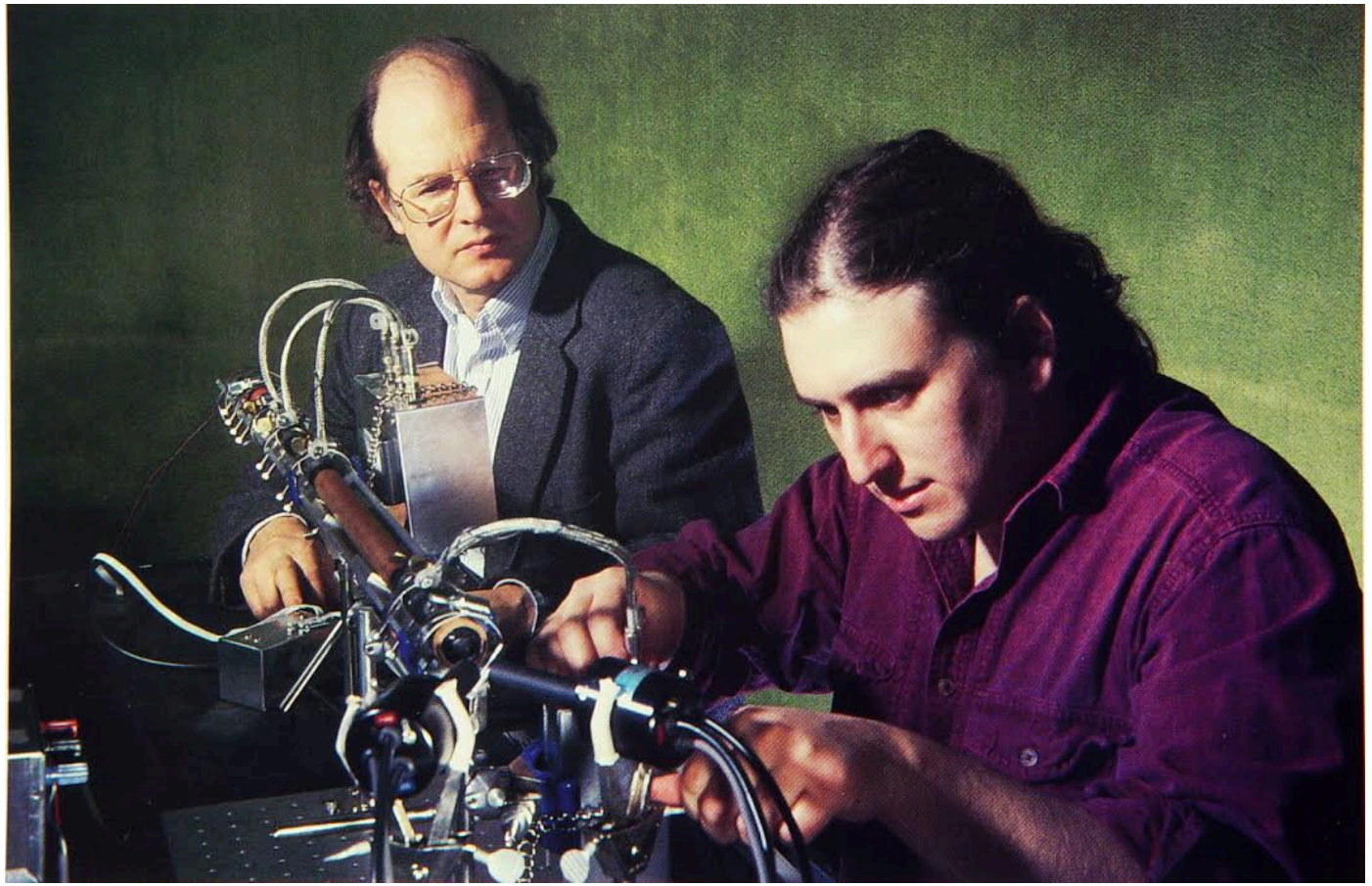
Prof Peter Shor, MIT, credit Wikipedia

BB84

Other technologies are being developed :

Post-Quantum Cryptography.

BB84 protocol allows to know if a communication has been tampered by Eve.

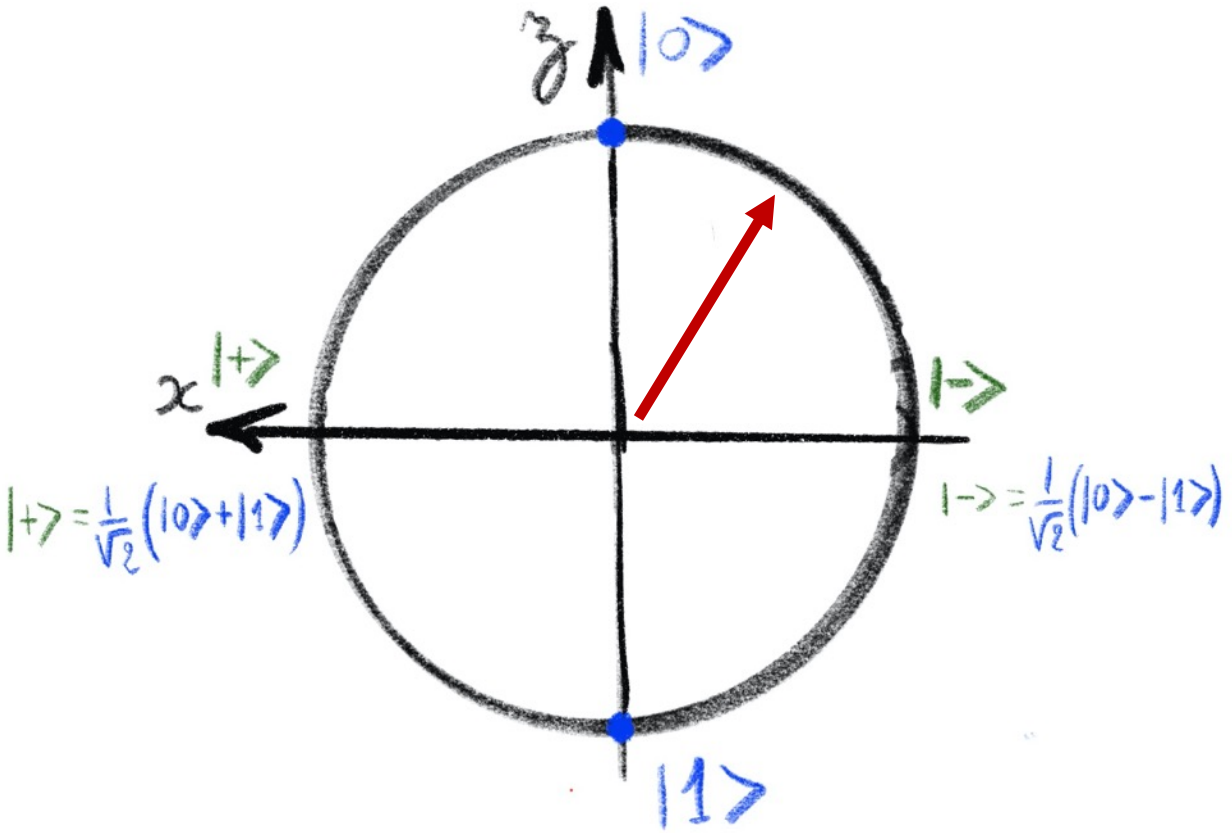


Charles Bennett, John Smolin, credit IBM Research

More : https://fr.wikipedia.org/wiki/Protocole_BB84

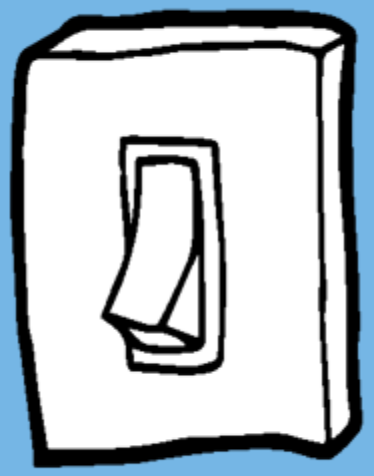
QUANTUM STATE AND MEASUREMENT

- Superposition
- Measurement



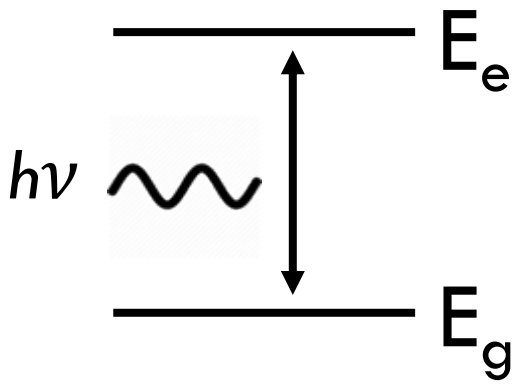
DEFINITION OF A QUANTUM BIT

0



1

« classical bit »



$|e\rangle \sim |1\rangle$

$|g\rangle \sim |0\rangle$

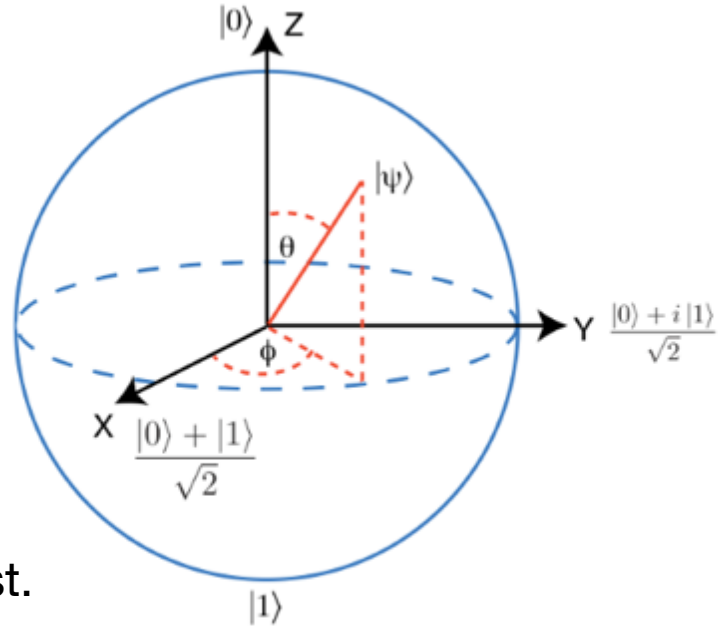
$|\psi\rangle = \alpha|g\rangle + \beta|e\rangle$

$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$

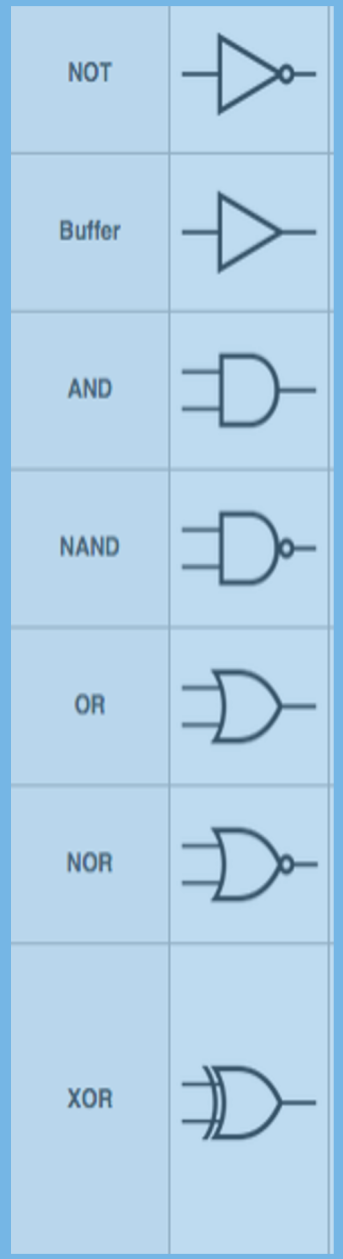
1. For any possible state:
   the measurement can only result in: $|0\rangle$ **or** $|1\rangle$

2. Probability of measuring $|0\rangle$ is $|\alpha|^2$,
   probability of measuring $|1\rangle$ is $|\beta|^2$

3. When the measurement is done, the superposition is lost.



The Bloch sphere

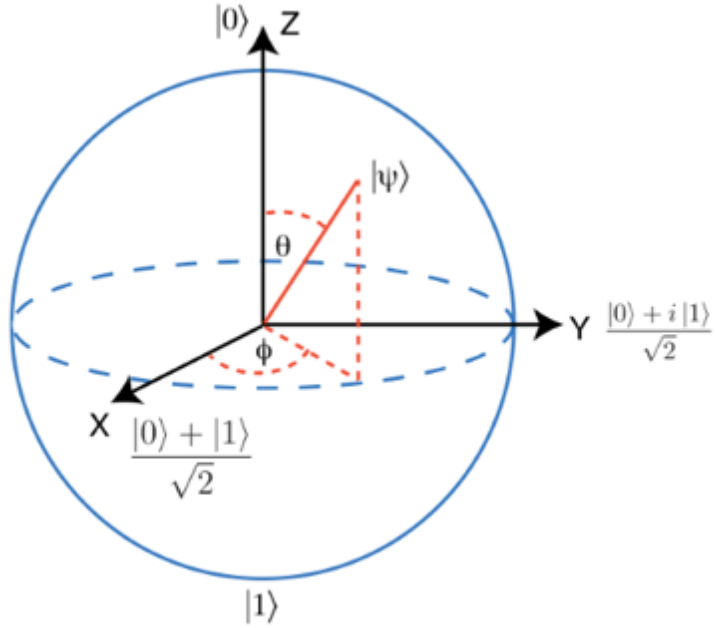


CONTROLLING A QUBIT

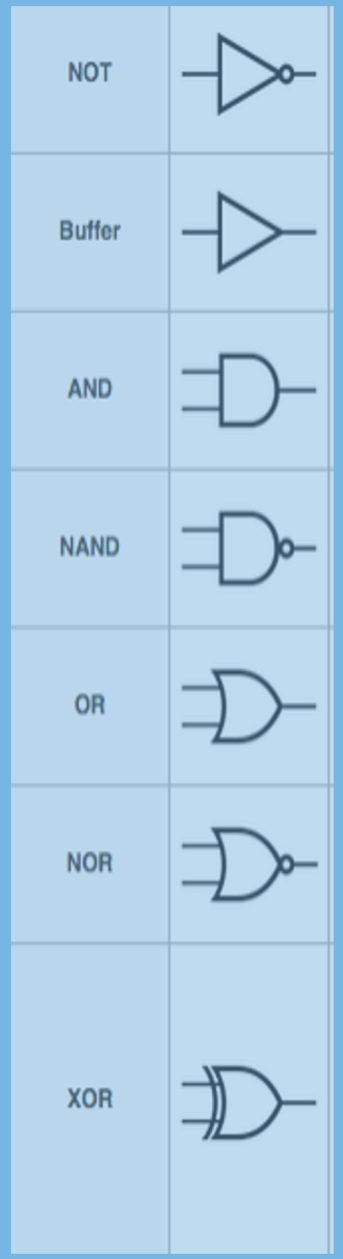
$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \Rightarrow |0\rangle = 1x|0\rangle + 0x|1\rangle$$
$$\Rightarrow |1\rangle = 0x|0\rangle + 1x|1\rangle$$

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}; |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$$

$$X = \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix} \begin{cases} X|0\rangle = |1\rangle \\ X|1\rangle = |0\rangle \end{cases}$$



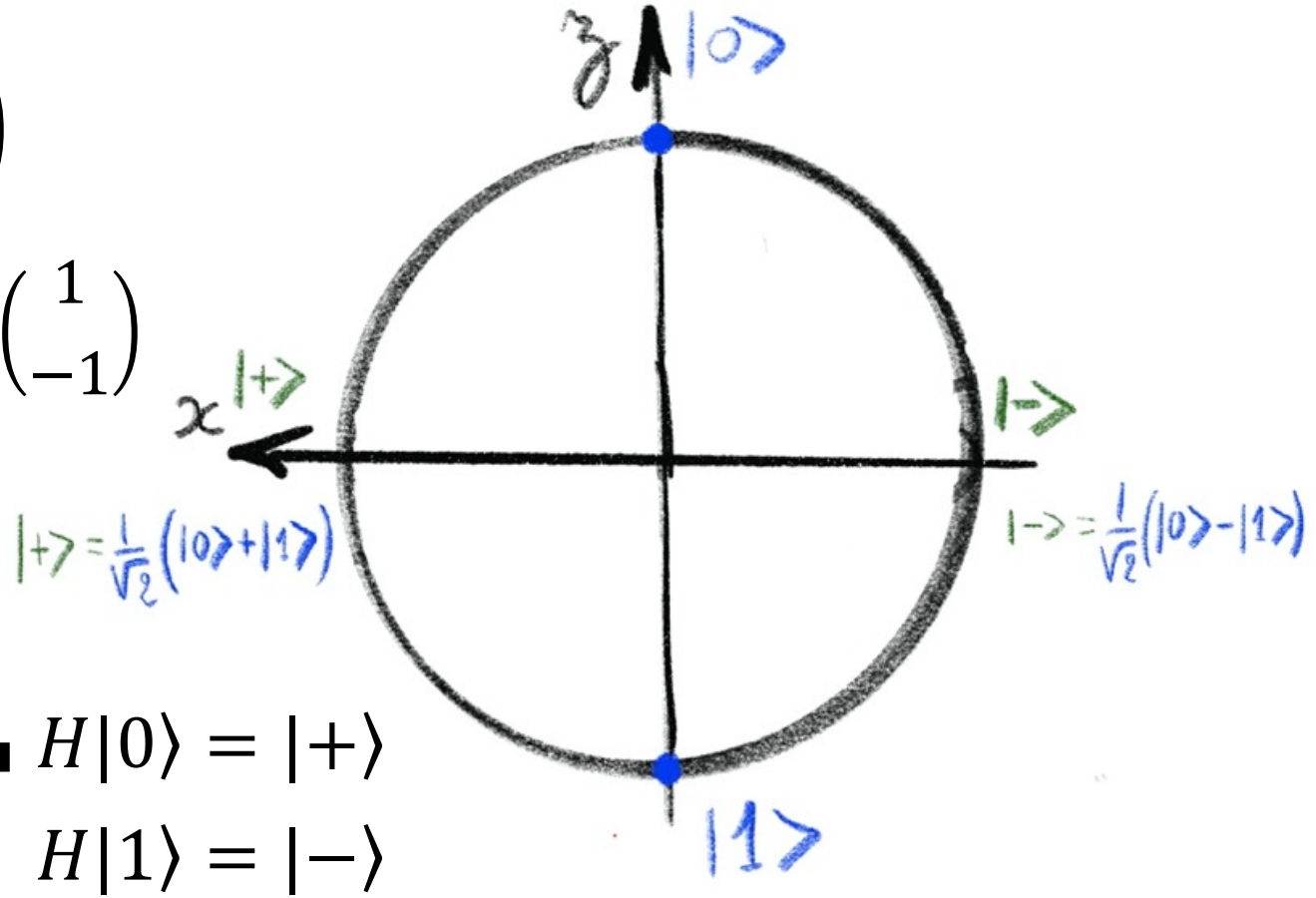
The Bloch sphere



CONTROLLING A QUBIT

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix} ; |1\rangle = \begin{pmatrix}0\\1\end{pmatrix}$$

$$|+\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} ; |-\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$$



$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\1 & -1\end{pmatrix}$$

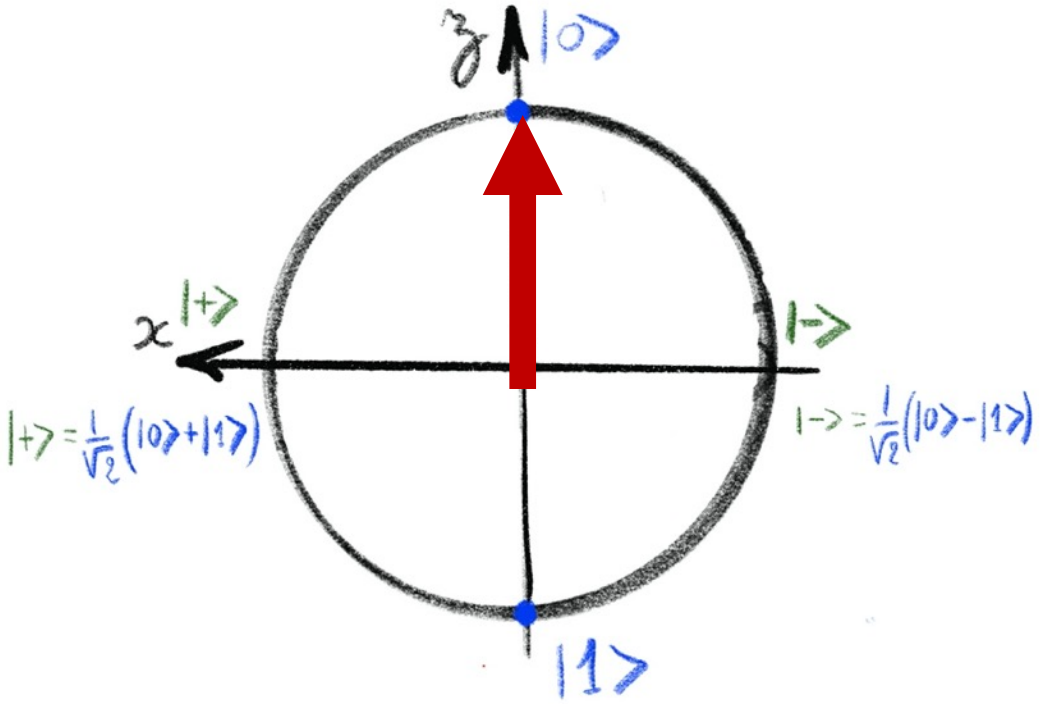
$$\begin{cases} H|0\rangle = |+\rangle \\ H|1\rangle = |-\rangle \\ H|+\rangle = |0\rangle \\ H|-\rangle = |1\rangle \end{cases}$$

$$H^2 = I$$

QUANTUM STATE AND MEASUREMENT

State $|0\rangle$:

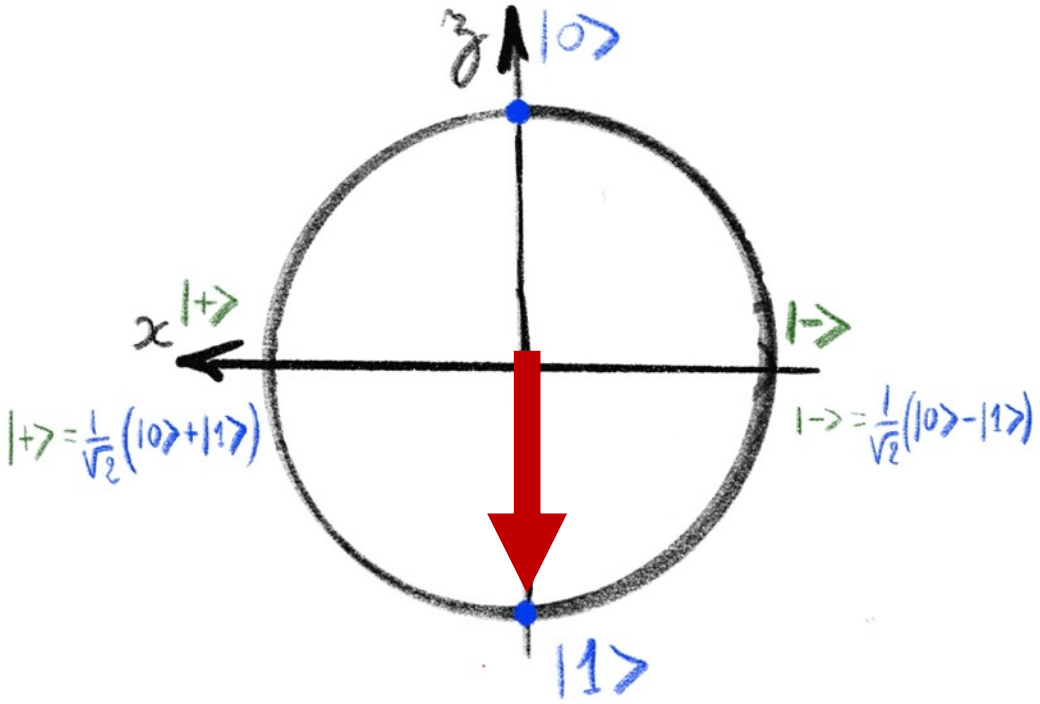
- measurement on z axis : $|0\rangle$
- measurement on x : $|+\rangle \, or \, |-\rangle$ have same probability $(1/2)$, and after measurement state will be $|+\rangle \, or \, |-\rangle$



QUANTUM STATE AND MEASURE

State $|1\rangle$:

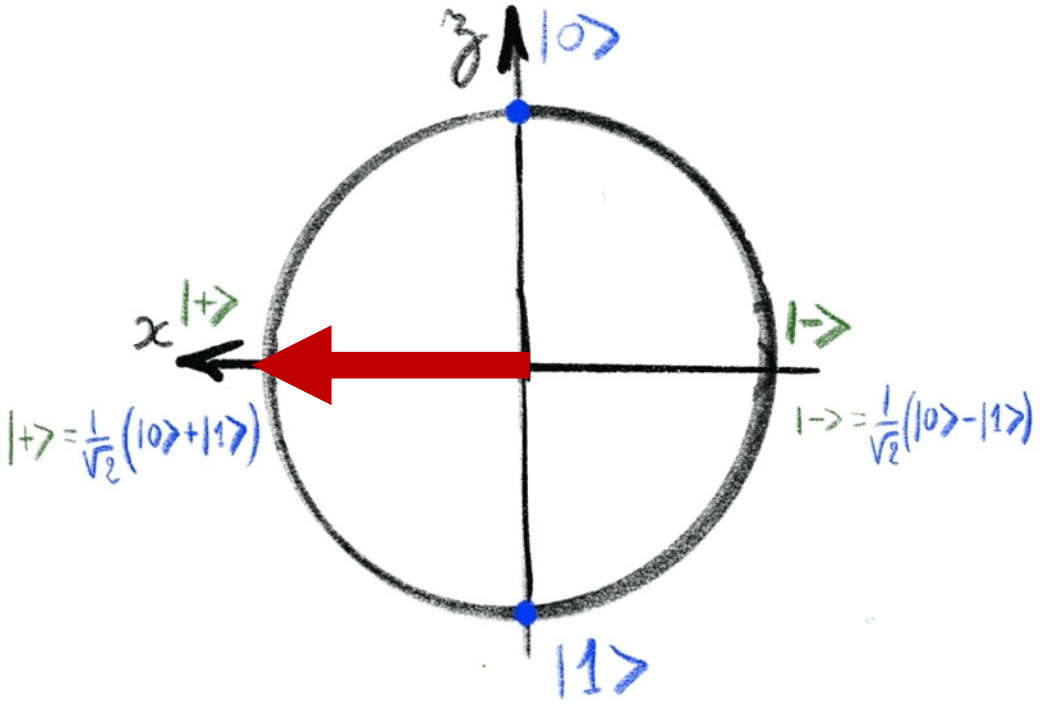
- measurement on z axis : $|1\rangle$
- measurement on x : $|+\rangle \, or \, |-\rangle$ have same probability $(1/2)$, and after measurement state will be $|+\rangle \, or \, |-\rangle$



QUANTUM STATE AND MEASURE

State $|+\rangle$:

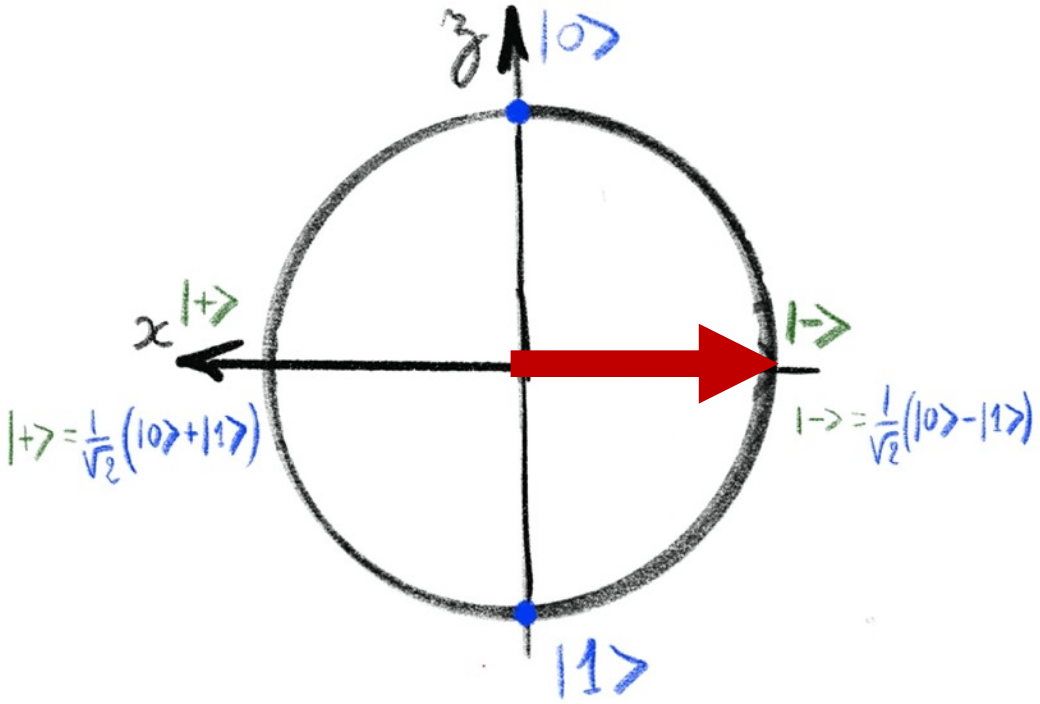
- measurement on x axis : $|+\rangle$
- measurement on z : $|0\rangle\ or|1\rangle$ have same probability $(1/2)$, and after measurement state will be $|0\rangle\ or|1\rangle$



QUANTUM STATE AND MEASURE

State $|-\rangle$:

- measurement on x axis : $|-\rangle$
- measurement on z : $|0\rangle\ or|1\rangle$ have same probability $(1/2)$, and after measurement state will be $|0\rangle\ or|1\rangle$



ALICE KEY : FROM BITS TO STATES

- Alice key is a bitsring (a string of 0 and 1) : 1110110000101…
- Alice base is a string of z and x : zzxzzxxxzxxzxx

❖ Four possible cases (coding convention) :

Alice key	Bit value	0	1	0	1
	Base choice	z	z	x	x
	Qubit state	$\|0\rangle$	$\|1\rangle$	$\|+\rangle$	$\|-\rangle$

BOB MEASUREMENTS CASES (EASY ONES)

Alice key	Bit value	0	0	0	0	1	1	1	1
	Base choice	z	z	x	x	z	z	x	x
	Qubit state	$\|0\rangle$	$\|0\rangle$	$\|+\rangle$	$\|+\rangle$	$\|1\rangle$	$\|1\rangle$	$\|-\rangle$	$\|-\rangle$
Bob	Base choice	z	x	z	x	z	x	z	x
	Measure	$\|0\rangle$			$\|+\rangle$	$\|1\rangle$			$\|-\rangle$
	Key value	0			0	1			1

BOB MEASUREMENTS CASES (CON'T)

Alice key	Bit value	0	0	0	0	1	1	1	1
	Base choice	z	z	x	x	z	z	x	x
	Qubit state	$\|0\rangle$	$\|0\rangle$	$\|+\rangle$	$\|+\rangle$	$\|1\rangle$	$\|1\rangle$	$\|-\rangle$	$\|-\rangle$
Bob	Base choice	z	x	z	x	z	x	z	x
	Measure	$\|0\rangle$	random $\|+\rangle$ or $\|-\rangle$	random $\|0\rangle$ or $\|1\rangle$	$\|+\rangle$	$\|1\rangle$	random $\|+\rangle$ or $\|-\rangle$	random $\|0\rangle$ or $\|1\rangle$	$\|-\rangle$
	Key value	0	random 0 or1	random 0 or1	0	1	random 0 or1	random 0 or1	1

BOB & ALICE EXCHANGE BASES AND REJECT RANDOM RESULTS

Alice key	Bit value	0	0	0	0	1	1	1	1
	Base choice	z	z	x	x	z	z	x	x
	Qubit state	$\|0\rangle$	$\|0\rangle$	$\|+\rangle$	$\|+\rangle$	$\|1\rangle$	$\|1\rangle$	$\|-\rangle$	$\|-\rangle$
Bob	Base choice	z	x	z	x	z	x	z	x
	Measure	$\|0\rangle$	random $\|+\rangle$ or $\|-\rangle$	random $\|0\rangle$ or $\|1\rangle$	$\|+\rangle$	$\|1\rangle$	random $\|+\rangle$ or $\|-\rangle$	random $\|0\rangle$ or $\|1\rangle$	$\|-\rangle$
	Key value	0	random 0 or1	random 0 or1	0	1	random 0 or1	random 0 or1	1
KEEP :		✓			✓	✓			✓

EXAMPLE

Alice has a key. She encodes it on a bases list, Bob receives Alices qubits states, and chooses random basis for measuring each one.
He gets (and stores) values, not knowing if reading is valid. Then, Alice and Bob communicate their basis choices to each other. Alice and Bob can select how to keep the the valid bits only.

Alice	Key value	0 0 0 1 1 0 0 0 0 1 1 0 0 1 1 1 0 1
	Base choice	x z z z x x x z x x x z x x x x z x
Bob	Base choice	z z x z x z x z x x x z z x x z z z
	Keep	n y n y y n y y y y y y n y y n y n

Key : 0 1 1 0 0 0 1 1 0 1 1 0

HOW SAFE IS THIS ?

If Eve was « listening » she has statistically modified one state out of four that would have been selected in Alice and Bob key.
Below we consider only the Alice and Bob valid key bits (after they have selected them).

Alice	key	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	basis	z	z	z	z	x	x	x	x	z	z	z	z	x	x	x	x
Eve	basis	z	z	x	x	z	z	x	x	z	z	x	x	z	z	x	x
	measure	0	0	0	1	0	1	0	0	1	1	0	1	0	1	1	1
Bob	basis	z	z	z	z	x	x	x	x	z	z	z	z	x	x	x	x
	measure	0	0	0	0	0	1	0	1	0	1	0	1	0	0	0	0	1	1	1	1	0	1	0	1	0	1	0	1	1	1	1	1

If Eve was listening, ¼ of Bob's bits are « wrong »
(and Eve has correctly accessed to ¾ of the key)

HOW IS THIS SAFE ?

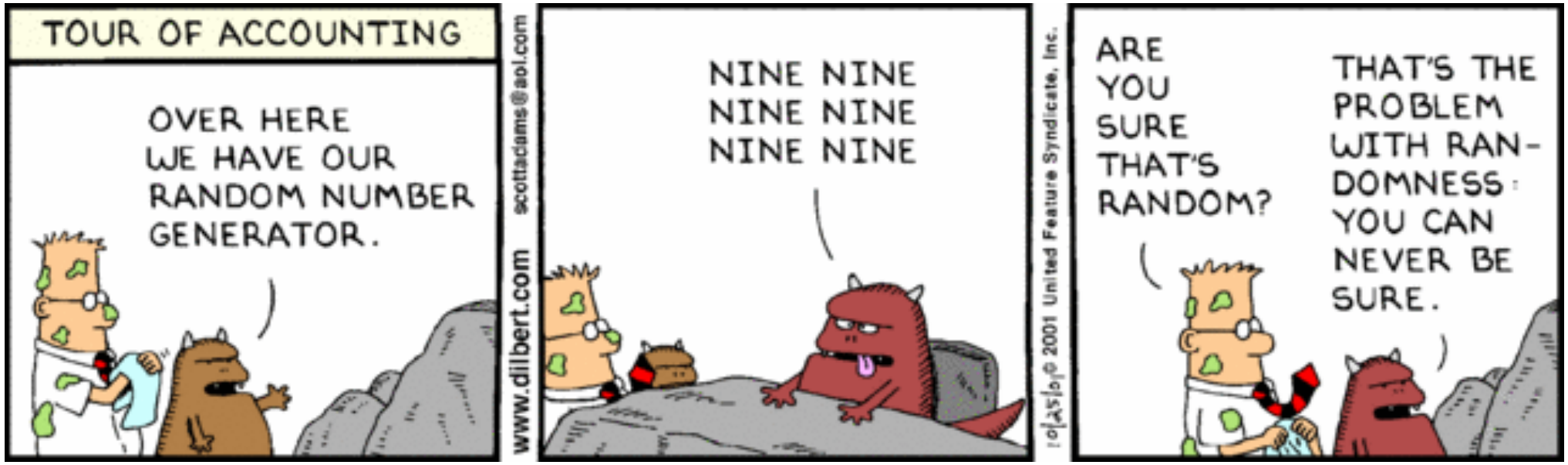
To figure out if this Eve was listening or not, Alice and Bob will « sacrifice » some if their key bits, by exchanging their values.

If they find a discrepancy rate of $\sim 1/4$: the know Eve was listening. Don't use the key !

For example, if they exchange and compare 100 bits, and find no discrepancy, the probability that Eve was listening but was not detected is : $\left(\frac{3}{4}\right)^{100} = 3{,}2\times10^{-13}$ (one out of 3000 billions).

QUANTUM RANDOM NUMBER GENERATION (QRNG)

- BB84 allows secure key distribution (QKD)
- Quantum technology can also generate **true** random numbers (QRNG)



HTTPS://CAMBRIDGEQUANTUM.COM/WP-CONTENT/UPLOADS/2020/09/CQC-IBM-STATEMENT.PDF

Post by Ilyas Khan, CEO of Cambridge Quantum Computing

Cambridge Quantum Computing Launches First Cloud-Based Quantum Random Number Generator Service with Verification

New joint offering with IBM will initially be available to members of the IBM Q Network, delivering certified quantum randomness for the first time

September 17, 2020

Cambridge Quantum Computing (CQC), the global provider of quantum computing software, today launched the world's first cloud-based Quantum Random Number Generation (QRNG) Service with integrated verification for the user, an important stepping stone on the road to Quantum Advantage.

Randomness is an essential and ubiquitous raw material in almost all digital interactions and is used in cybersecurity to encrypt data and communications and perform simulation analysis across many industries, including science, engineering, finance and gaming. The application developed by CQC generates true maximal randomness, or entropy, on an IBM Quantum computer that is device independent and that can be verified and thus certified as truly quantum – and therefore truly random – for the first time.

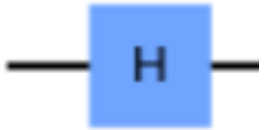
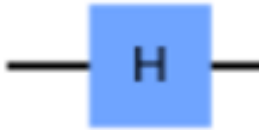
HANDS ON !

How to program a quantum computer with Python and qiskit

```
1  from qiskit import QuantumCircuit, Aer, execute
2
3  backend = Aer.get_backend('qasm_simulator')
4
5  qc = QuantumCircuit(1,1)
6
7  qc.h([0])
8
9  qc.measure([0],[0])
10
11 d = execute(qc,backend,shots=1024).result().get_counts(qc)
12 print(d)
```

```
{'0': 516, '1': 508}
```

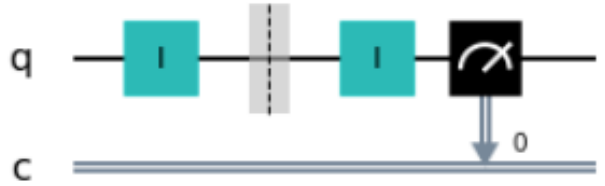
EXAMPLES (GENERATING THE 4 STATES)

$\|0\rangle$	`qc.id([0])`		$\|0\rangle$
$\|0\rangle$	`qc.x([0])`		$\|1\rangle$
$\|0\rangle$	`qc.h([0])`		$\|+\rangle$
$\|1\rangle$	`qc.h([0])`		$\|-\rangle$

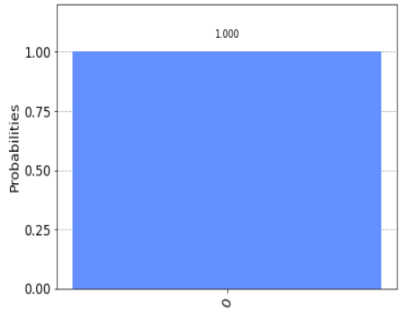
MEASURE ON Z AXIS

|0⟩

```
qc.id([0])
…
qc.id([0])
```

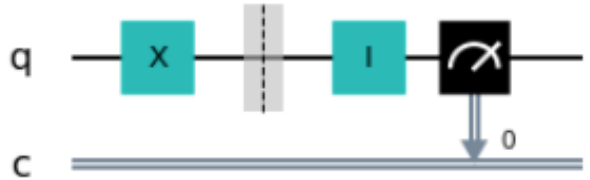


0

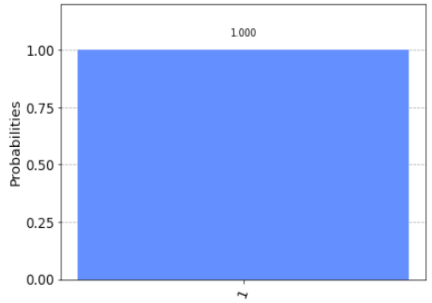


|1⟩

```
qc.x([0])
…
qc.id([0])
```

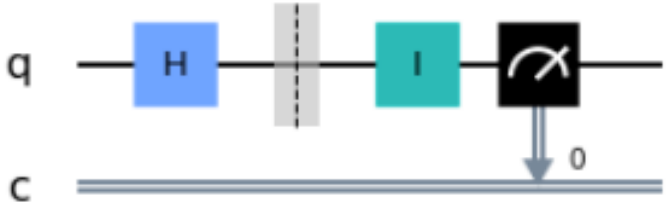


1

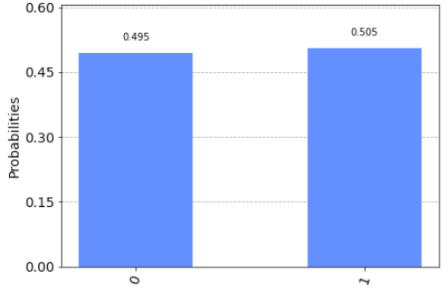


|+⟩

```
qc.h([0])
…
qc.id([0])
```



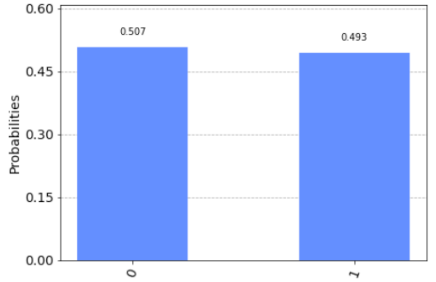
0,1



|-⟩

```
qc.x([0])
qc.h([0])
…
qc.id([0])
```

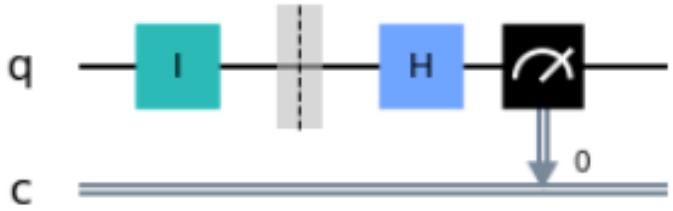
0,1



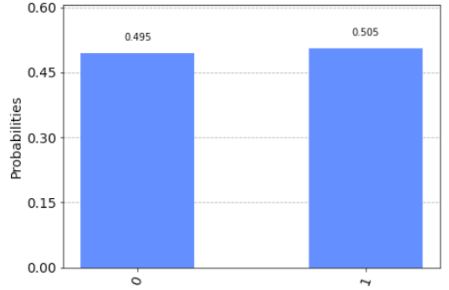
MEASURE ON X AXIS

|0⟩

```
qc.id([0])
…
qc.h([0])
```



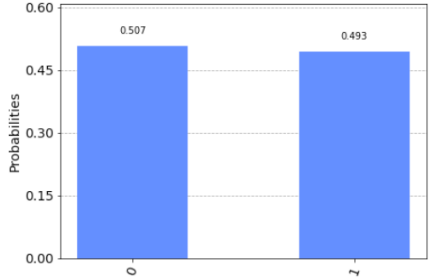
0,1



|1⟩

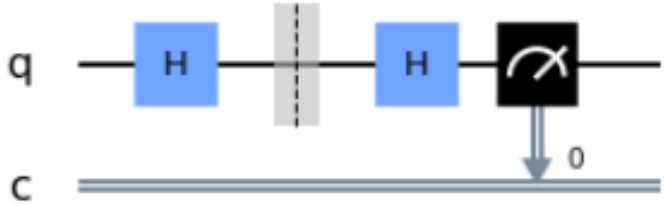
```
qc.x([0])
…
qc.h([0])
```

0,1

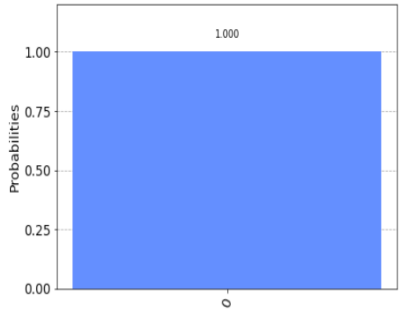


|+⟩

```
qc.h([0])
…
qc.h([0])
```



0



|-⟩

```
qc.x([0])
qc.h([0])
…
qc.h([0])
```

1