

protocole BB84 pour l'échange sécurisé de clefs d'encryption (QKD)

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

Novembre 2021

ALICE: , BOB: , EVE: 

Alice souhaite envoyer un message secret à Bob, elle peut :

- protéger le canal de communication et transmettre en clair
 - cacher le message
 - encrypter the message,
- ...pour éviter que Eve puisse lire le message



«**BB84** est un protocole d'échange de clef basé sur les principes de la mécanique quantique développé par Charles Bennett et Gilles Brassard en 1984.»

CACHER LE TEXTE (STEGANOGRAPHIE)

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]

👉 Il faut communiquer la « recette »

[source wikipedia](#) (texts and images)

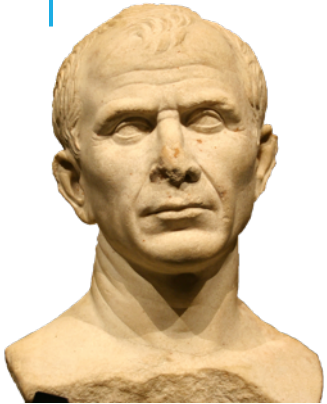


Image of a tree with a steganographically hidden image. The hidden image is revealed by removing all but the two least significant [bits](#) of each [color component](#) and a subsequent [normalization](#). The hidden image is shown below.



Image of a cat extracted from the tree image above.

RENDRE LE MESSAGE ILLISIBLE : CRYPTOGRAPHIE



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

mrspbbowoxd no mocka

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

qixv ghrrhwhij zthv xni ijaasj jva
akxykxsv wkgij gj it zizj ztnhjsj,
hi jva tvvjc rtwij gj rthsj xni
taatfxj vatahvahfxj

Vigénère : pour éviter qu'une lettre soit toujours codée par la même lettre, on utilise un clef de taille k et chaque lettre du message est « décalée » par la $k^{\text{ième}}$ lettre (modulo k) de la clef. Dans cet exemple la clef est « hello » de longueur 5 :

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.



Il faut communiquer la clef

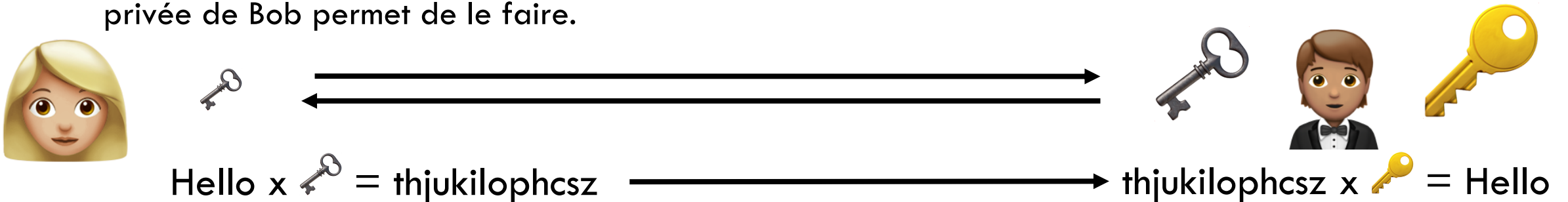


source wikipedia images

RSA (« RIVEST, SHAMIR, ADELMAN »):

RSA, aussi appelé codage à clefs asymétriques, permet de ne pas avoir à communiquer la clef de décryptage préalablement.

- Alice veut envoyer un message à Bob.
- Elle utilise la clef publique de Bob pour **encrypter** le message.
- La clef publique ne permet pas de décrypter le message qu'elle a encrypté, seule la clef privée de Bob permet de le faire.



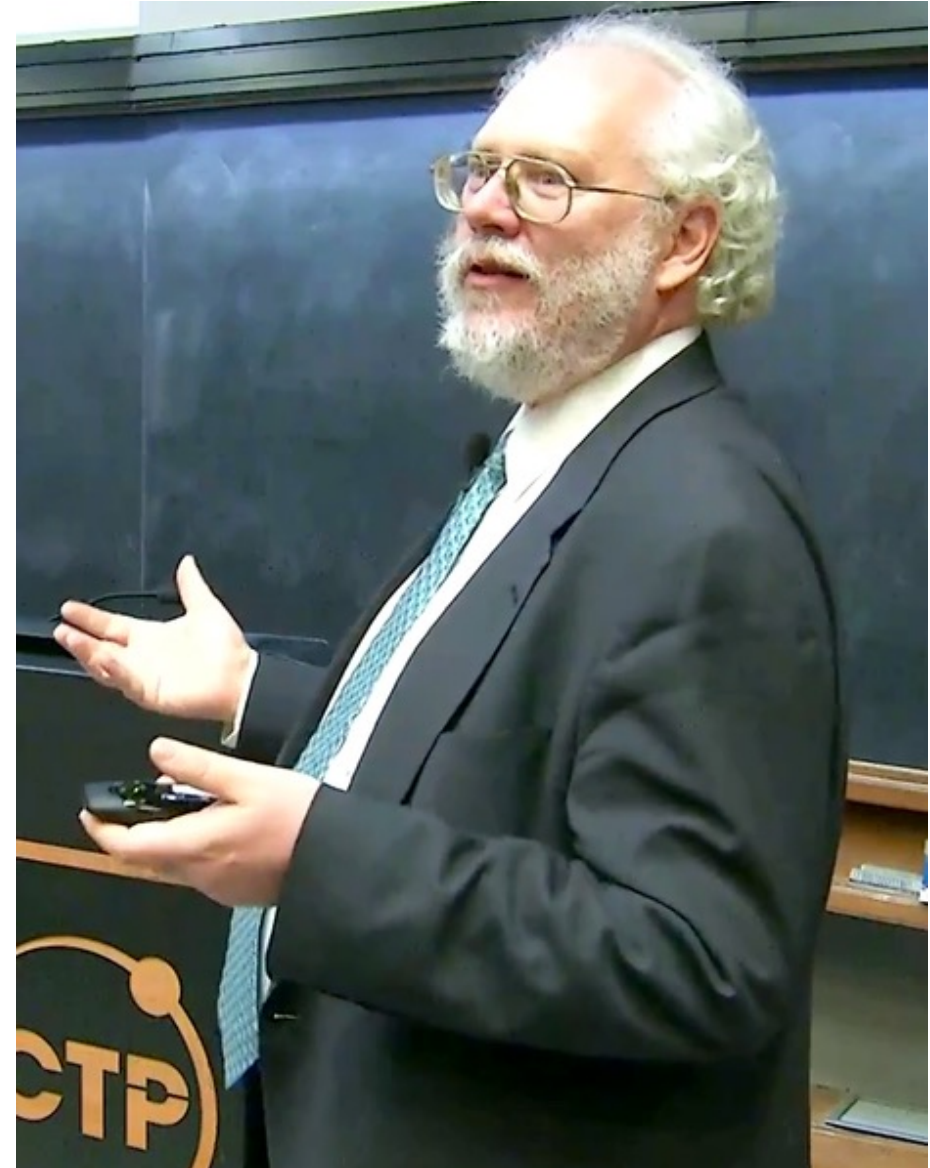
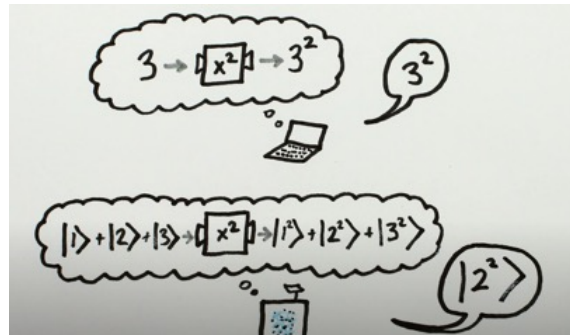
Ceci repose sur le fait qu'il est très difficile de décomposer un grand nombre entier en ses facteurs premiers.

plus de détails sur : https://fr.wikipedia.org/wiki/Chiffrement_RSA

VOICI PETER SHOR !

En 1994, le professeur Peter Shor du M.I.T. apporte la preuve qu'un algorithme quantique (utilisant la superposition et l'intrication quantique) peut trouver les facteurs premiers d'un nombre en un temps exponentiellement plus court qu'avec tout algorithme classique.

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



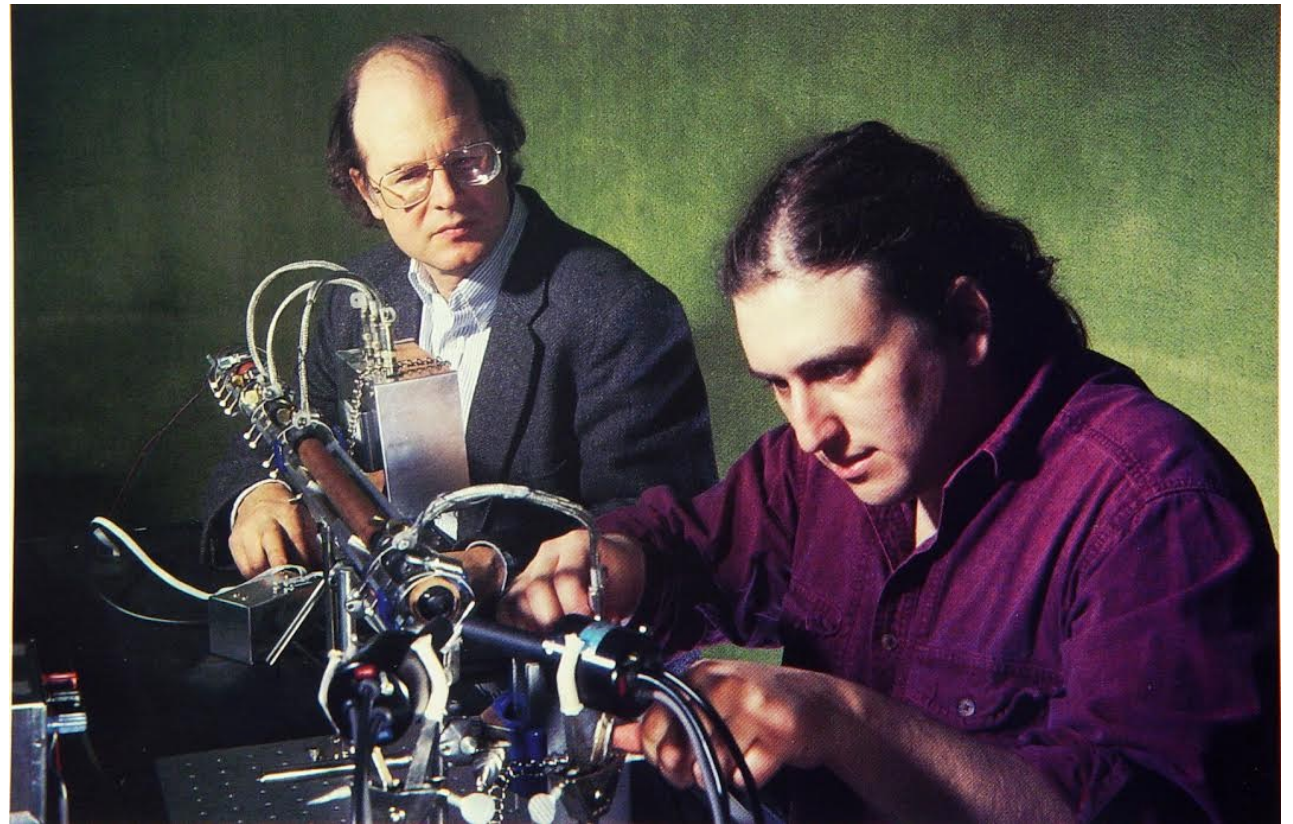
Prof Peter Shor, MIT, credit Wikipedia

BB84

Mais dans ce domaine, d'autres technologies se développent :

La Cryptographie Post Quantique.

le protocole BB84 permet de savoir (de manière très fiable) si une communication a été interceptée ou pas par Eve.

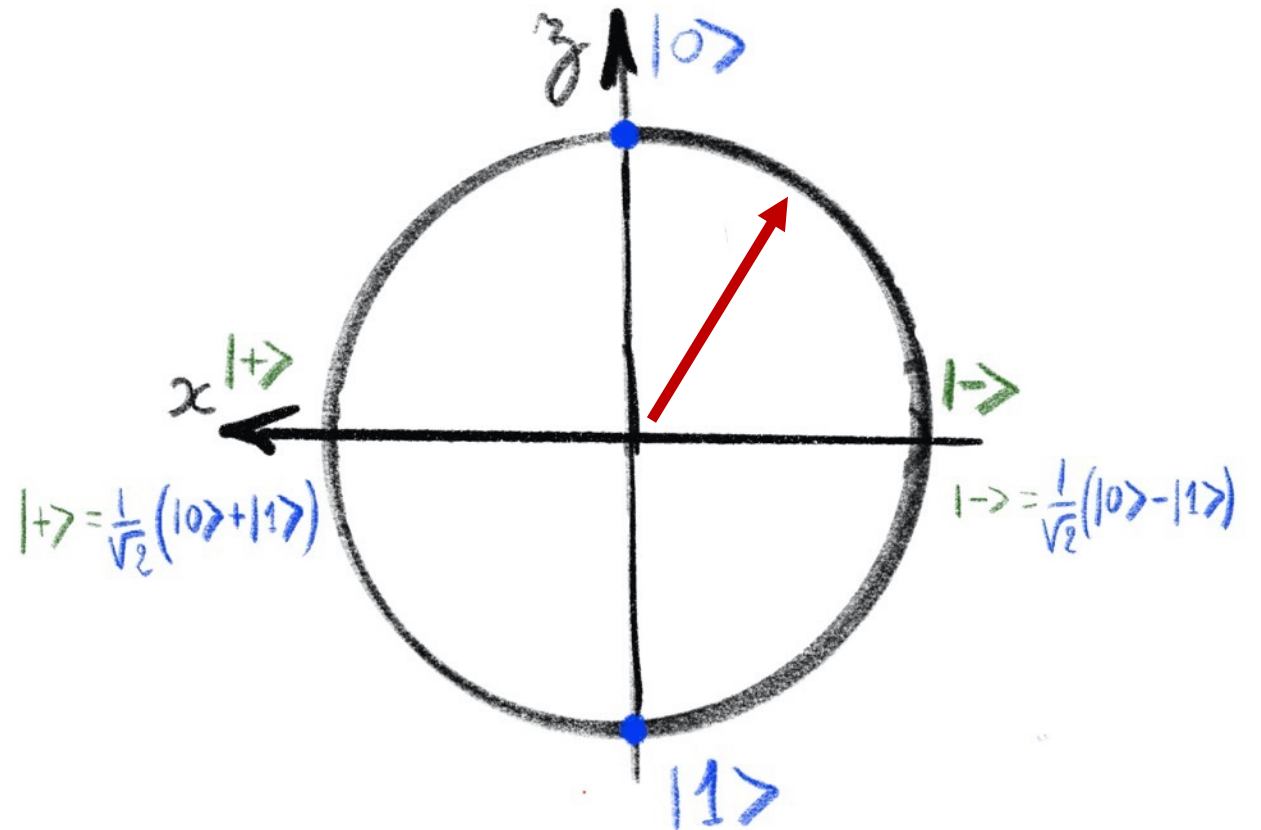


Charles Bennett, John Smolin (Credit IBM Research)

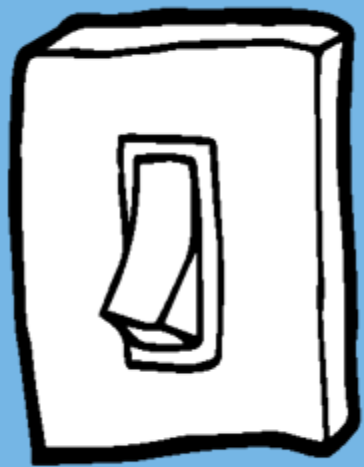
Plus de détails : https://fr.wikipedia.org/wiki/Protocole_BB84

ETAT QUANTIQUE ET MESURE

- Superposition
- Mesure



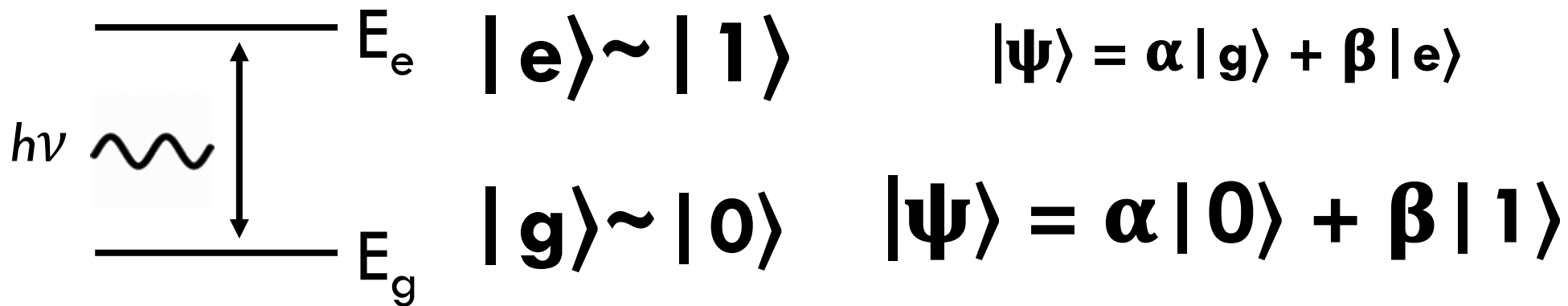
0



1

« classical bit »

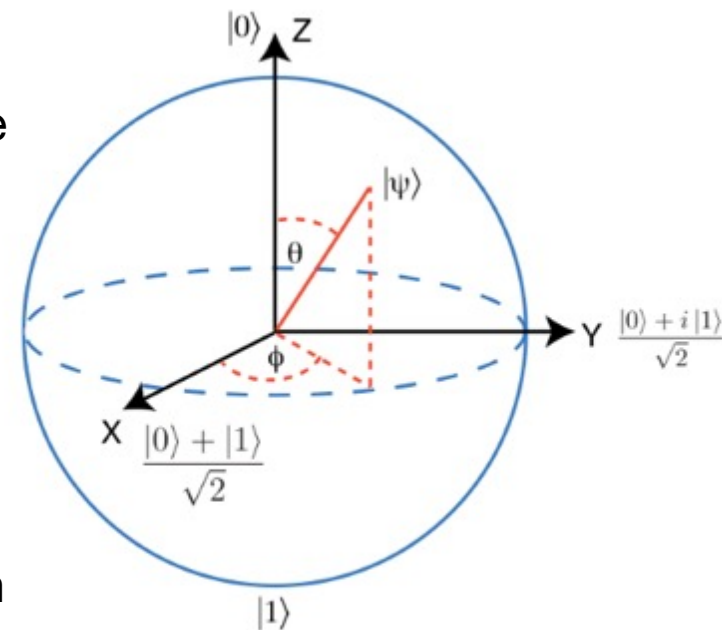
DEFINITION DU QUANTUM BIT










1 Pour un état quelconque, la mesure ne peut donner que **$|0\rangle$ ou $|1\rangle$**

2 La probabilité de mesurer $|0\rangle$ est $|\alpha|^2$,
la probabilité de mesurer $|1\rangle$ est $|\beta|^2$

3 Au moment où la mesure est effectuée, la superposition est perdue.



The Bloch sphere

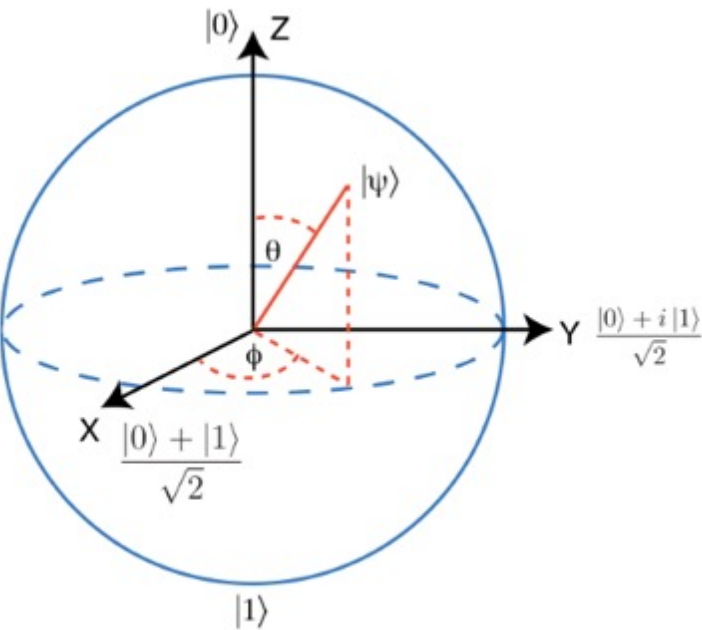
NOT	
Buffer	
AND	
NAND	
OR	
NOR	
XOR	

CONTROLLER UN QUBIT








$$\begin{aligned}
 |\psi\rangle &= \alpha |0\rangle + \beta |1\rangle \Rightarrow |0\rangle = 1x|0\rangle + 0x|1\rangle \\
 &\Rightarrow |1\rangle = 0x|0\rangle + 1x|1\rangle
 \end{aligned}$$

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

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \left\{ \begin{array}{l} X|0\rangle = |1\rangle \\ X|1\rangle = |0\rangle \end{array} \right.$$



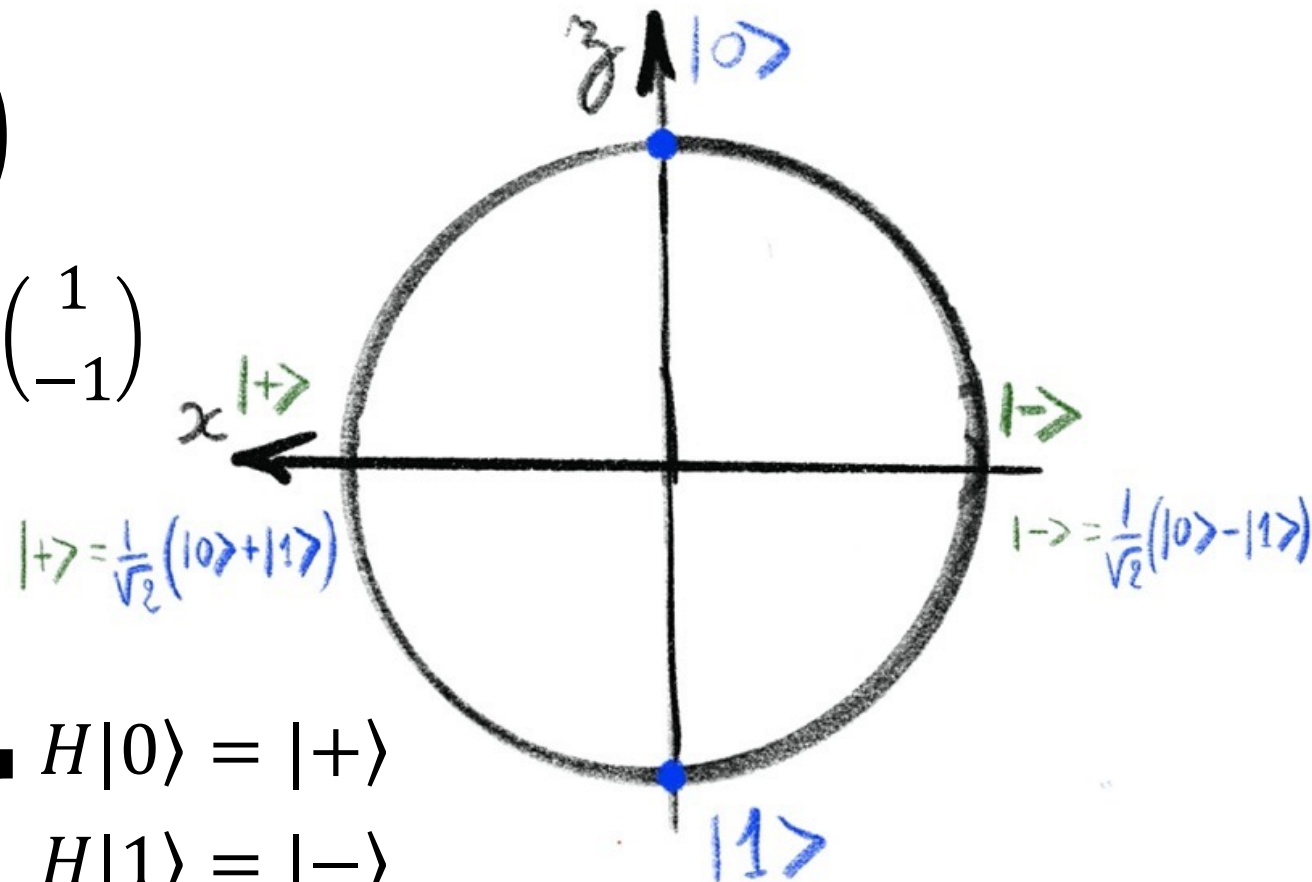
The Bloch sphere

NOT	
Buffer	
AND	
NAND	
OR	
NOR	
XOR	

CONTROLLER UN QUBIT

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

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



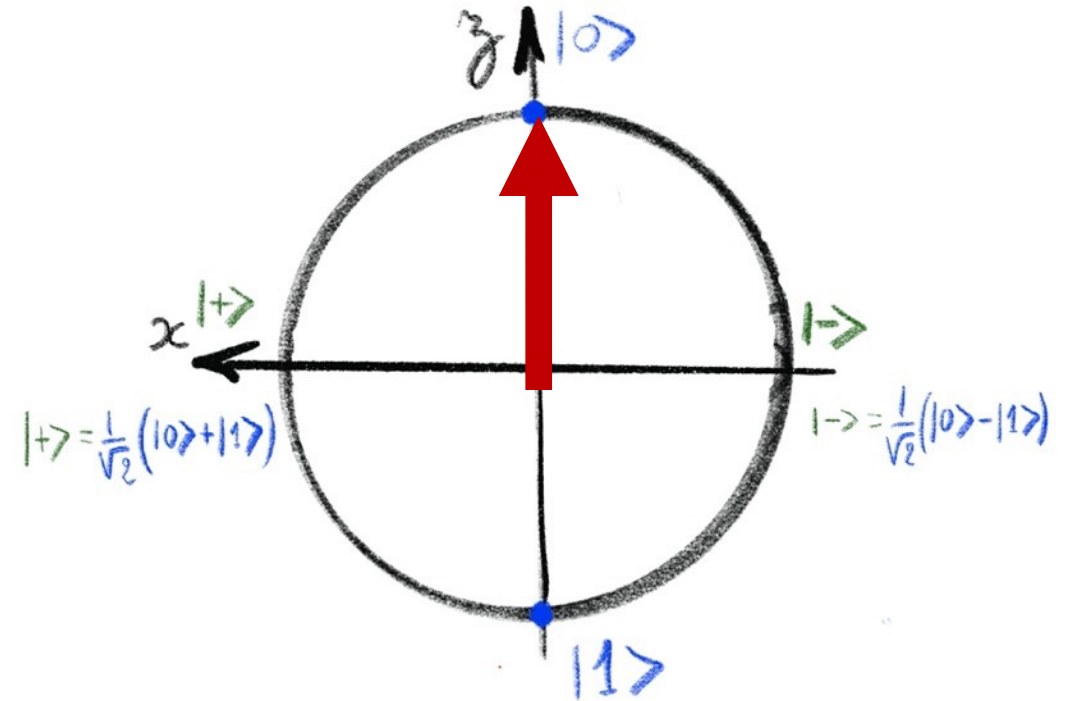
$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \left\{ \begin{array}{l} H|0\rangle = |+\rangle \\ H|1\rangle = |-\rangle \\ H|+\rangle = |0\rangle \\ H|-\rangle = |1\rangle \end{array} \right.$$

$$H^2 = I$$

ETAT QUANTIQUE ET MESURE

Etat $|0\rangle$:

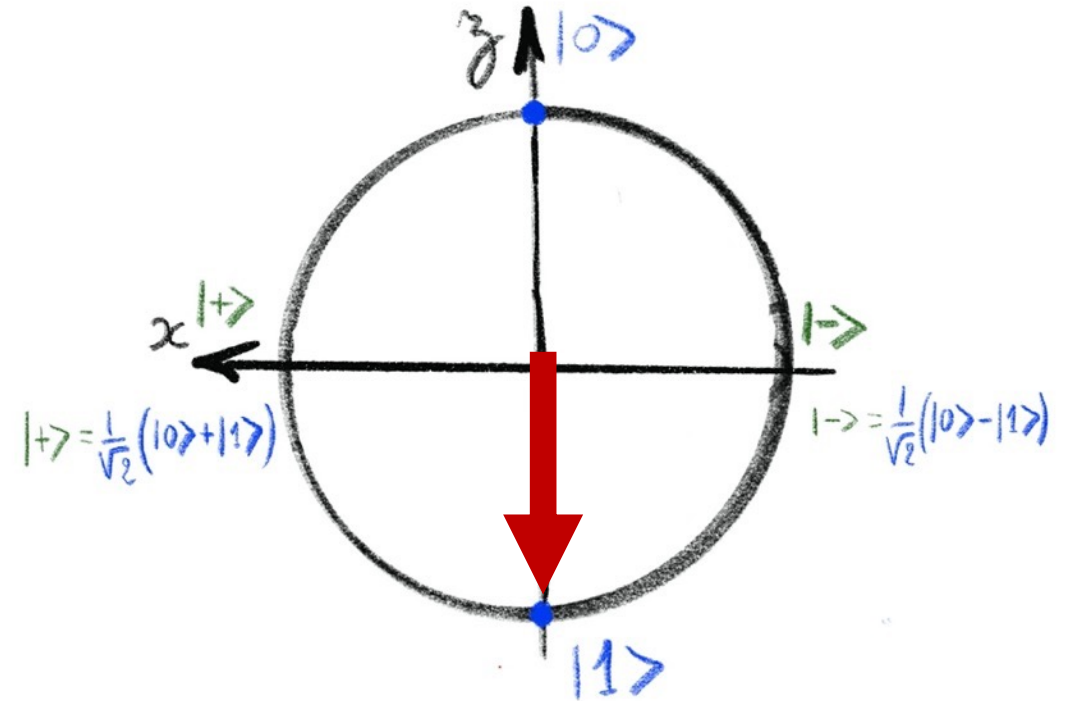
- mesure sur l'axe z : $|0\rangle$
- mesure sur l'axe x : $|+\rangle$ ou $|-\rangle$ ont la même probabilité de mesure ($1/2$), et après la mesure, l'état devient $|+\rangle$ ou $|-\rangle$



ETAT QUANTIQUE ET MESURE

Etat $|1\rangle$:

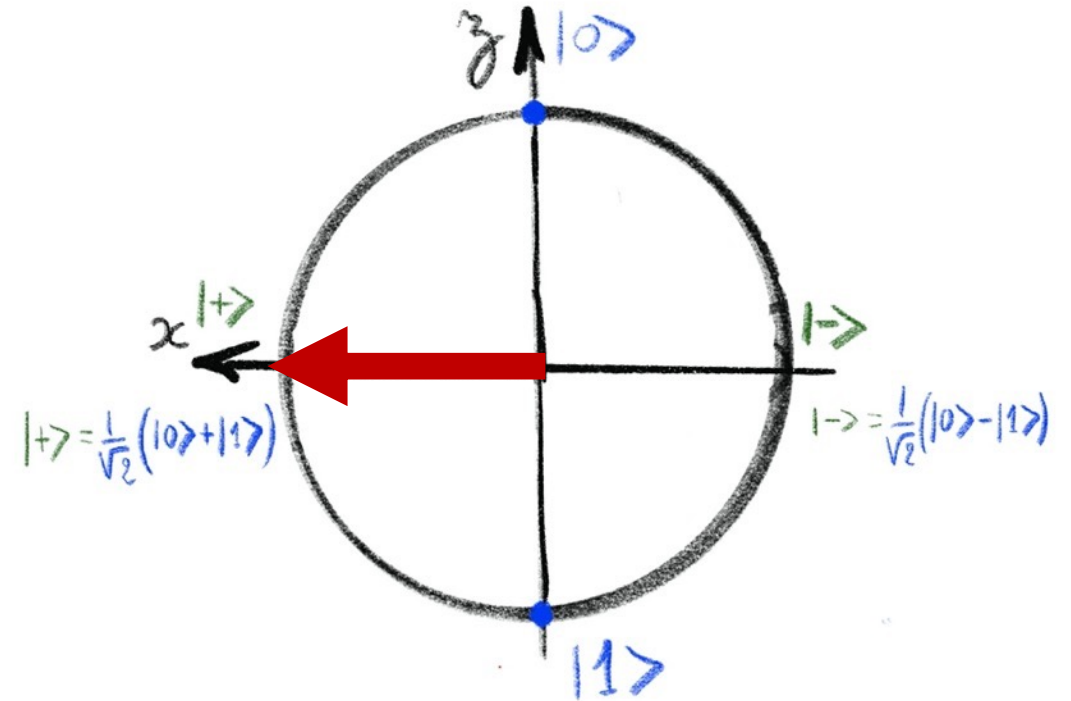
- mesure sur l'axe z : $|1\rangle$
- mesure sur l'axe x : $|+\rangle$ ou $|-\rangle$ ont la même probabilité de mesure ($1/2$), et après la mesure l'état devient $|+\rangle$ ou $|-\rangle$



ETAT QUANTIQUE ET MESURE

Etat $|+\rangle$:

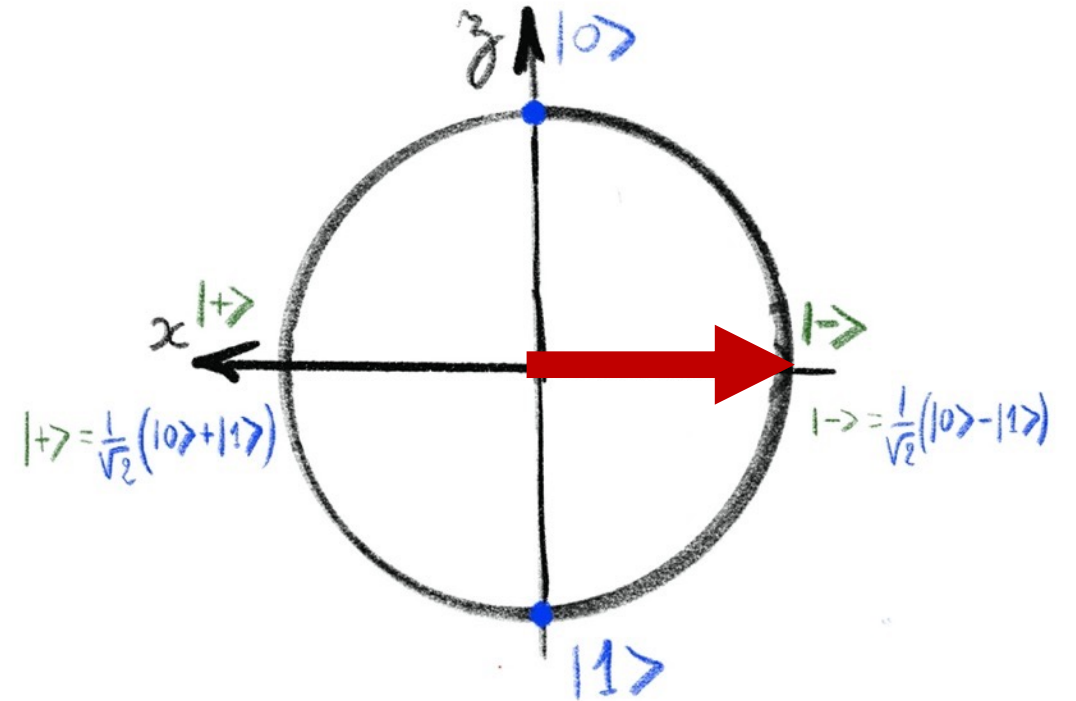
- mesure sur l'axe x : $|+\rangle$
- mesure sur l'axe z : $|0\rangle$ ou $|1\rangle$ on la même probabilité de mesure ($1/2$), et après la mesure, l'état sera $|0\rangle$ ou $|1\rangle$



ETAT QUANTIQUE ET MESURE

Etat $|-\rangle$:

- mesure sur l'axe x : $|-\rangle$
- mesure sur l'axe z : $|0\rangle$ ou $|1\rangle$ ont la même probabilité de mesure ($1/2$), et après la mesure l'état devient $|0\rangle$ ou $|1\rangle$



CODAGE DE LA CLEF D'ALICE : CODAGE A 4 ETATS

- La clef d'Alice est une chaîne de 0 et 1 : 1110110000101 ...
- Les bases d'Alice sont une suite de z et de x : zzxzxzxzxzxzx ...
- ❖ Il y a 4 cas possibles, ils constituent la convention de codage :

Clef d'Alice	Valeur du bit	0	1	0	1
	Choix de base	z	z	x	x
	Etat du qubit	$ 0\rangle$	$ 1\rangle$	$ +\rangle$	$ -\rangle$

MESURES DE BOB SUR DES BASES CHOISIES ALÉATOIREMENT (CAS EVIDENTS)

Alice key	Valeur du bit	0	0	0	0	1	1	1	1
	Choix de base	z	z	x	x	z	z	x	x
	Etat du qubit	$ 0\rangle$	$ 0\rangle$	$ +\rangle$	$ +\rangle$	$ 1\rangle$	$ 1\rangle$	$ -\rangle$	$ -\rangle$
Bob	Choix de base	z	x	z	x	z	x	z	x
	Mesure	$ 0\rangle$			$ +\rangle$	$ 1\rangle$			$ -\rangle$
	Valeur de la clef	0			0	1			1

MESURES DE BOB SUR DES BASES CHOISIES ALÉATOIREMENT (SUITE)

Alice key	Valeur du bit	0	0	0	0	1	1	1	1
	Choix de base	z	z	x	x	z	z	x	x
	Etat du qubit	$ 0\rangle$	$ 0\rangle$	$ +\rangle$	$ +\rangle$	$ 1\rangle$	$ 1\rangle$	$ -\rangle$	$ -\rangle$
Bob	Choix de base	z	x	z	x	z	x	z	x
	Mesure	$ 0\rangle$	$ +\rangle$ ou $ -\rangle$ au hasard	$ 0\rangle$ ou $ 1\rangle$ au hasard	$ +\rangle$	$ 1\rangle$	$ +\rangle$ ou $ -\rangle$ au hasard	$ 0\rangle$ ou $ 1\rangle$ au hasard	$ -\rangle$
	Valeur de la clef	0	0 or 1 au hasard	0 or 1 au hasard	0	1	0 or 1 au hasard	0 or 1 au hasard	1

BOB & ALICE ECHANGENT LEUR BASES ET REJETTENT LES RÉSULTATS OBTENUS PAR BOB AVEC DU HASARD

Alice key	Valeur du bit	0	0	0	0	1	1	1	1
	Choix de base	z	z	x	x	z	z	x	x
	Etat du qubit	$ 0\rangle$	$ 0\rangle$	$ +\rangle$	$ +\rangle$	$ 1\rangle$	$ 1\rangle$	$ -\rangle$	$ -\rangle$
Bob	Choix de base	z	x	z	x	z	x	z	x
	Mesure	$ 0\rangle$	$ +\rangle$ ou $ -\rangle$ au hasard	$ 0\rangle$ ou $ 1\rangle$ au hasard	$ +\rangle$	$ 1\rangle$	$ +\rangle$ ou $ -\rangle$ au hasard	$ 0\rangle$ ou $ 1\rangle$ au hasard	$ -\rangle$
	Valeur de la clef	0	0 or 1 au hasard	0 or 1 au hasard	0	1	0 or 1 au hasard	0 or 1 au hasard	1
	Garder :	✓			✓	✓			✓

EXEMPLE

Alice a une clef, elle l'encode avec sa liste de bases. Bob reçoit les 4 états et les mesure avec sa liste de bases. Il obtient et mémorise les valeurs mesurées. Ensuite Alice et Bob échangent leur listes de bases ce qui leur permet de sélectionner, chacun de son côté les bits « valides » constituant leur clef commune.

Alice	Clef d'Alice	0 0 0 1 1 0 0 0 0 1 1 0 0 1 1 1 0 1
	Choix de bases	x z z z x x x z x x x z x x x x z x
Bob	Choix de bases	z z x z x z x z x x x z z x x z z z
	garder	n y n y y n y y y y y y n y y n y n

clef commune : 0 1 1 0 0 0 1 1 0 1 1 0

QUELLE EST LA SÉCURITÉ DE CE PROTOCOLE ?

Si Eve a « écouté » elle a statistiquement modifié un état sur quatre du flux transmis, parmi les bits qui sont « à garder » pour la clef commune.

Dans le tableau ci-dessous, on ne considère que les cas où les bits ont été retenus pour la clef commune par Alice et Bob.

Alice	clef	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	base	z	z	z	z	x	x	x	x	z	z	z	z	x	x	x	x
Eve	base	z	z	x	x	z	z	x	x	z	z	x	x	z	z	x	x
	mesure	0	0	0	1	0	1	0	0	1	1	0	1	0	1	1	1
Bob	base	z	z	z	z	x	x	x	x	z	z	z	z	x	x	x	x
	mesure	0	0	0	0	0	1	0	1	0	1	0	1	0	0	0	0

Si Eve a écouté alors $\frac{1}{4}$ des bits de Bob sont « faux », ceux en rouge (ceux en bleu sont juste mais par hasard).

Et Eve a recueilli correctement $\frac{3}{4}$ des bits de clef commune

QUELLE EST LA SÉCURITÉ DE CE PROTOCOLE ?

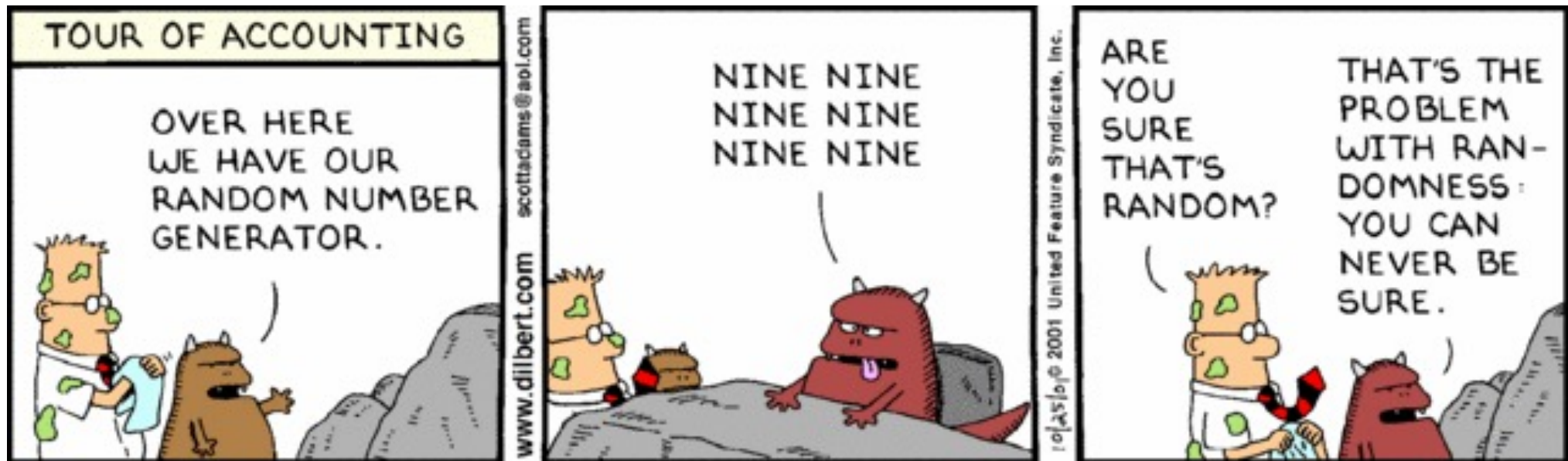
Pour savoir si Eve a écouté ou pas, Alice et Bob vont « sacrifier » une partie des bits de la clef, en échangeant en clair leur valeur.

**Si ils constatent une différence sur environ $\frac{1}{4}$ des valeurs échangées :
ils savent que Eve a écouté : il ne doivent pas utiliser la clef !**

Par exemple, si ils échangent et comparent 100 bits, et ne trouvent pas de différence, la probabilité que Eve ait écouté sans être détectée par ce protocole vaut : $\left(\frac{3}{4}\right)^{100} = 3,2 \times 10^{-13}$ (une chance sur 3000 milliards).

GENERATION DE NOMBRE ALEATOIRE QUANTIQUE (QRNG)

- BB84 permet la distribution sécurisée de key d'encryption (QKD)
- La technologie quantique permet aussi de générer des nombre purement aléatoires (QRNG)



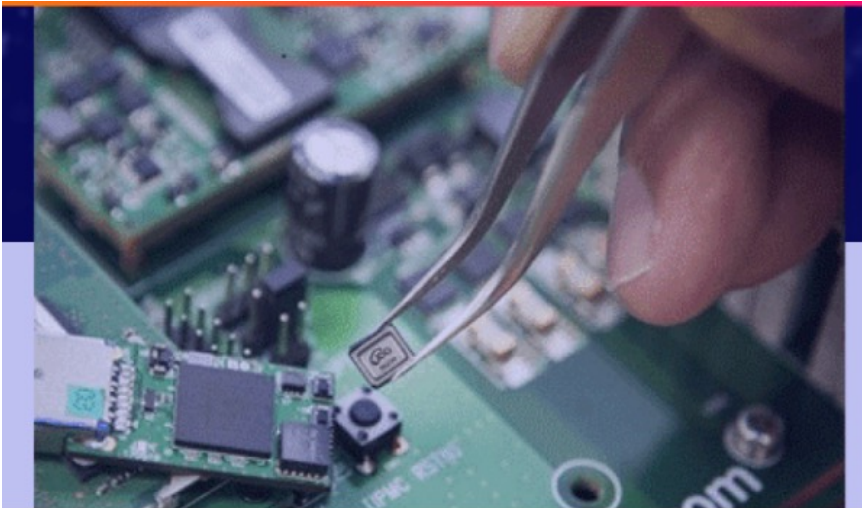
[Discover Q](#)[About QF](#)[Registration](#)[/ ABOUT QUANTUM FLAGSHIP / NEWSROOM](#)[← All News](#)

ID Quantique's quantum entropy source implemented in Samsung's latest 5G smartphone

PUBLISHED ON JUNE 8TH, 2020

QRANGE project partner ID Quantique has announced that its newest Quantum Random Number Generator (QRNG) chip has been integrated in the 'Galaxy A Quantum', a custom edition of the Samsung Galaxy A71 5G smartphone commercialised by SK Telecom to protect its customers' most valuable information.

Generating strong keys from a reliable entropy source is the cornerstone of any security system. Quantum-based technologies rely on quantum random number generators (QRNGs) for securing the communications on telecommunication networks, since these generators have proven to maintain this security at the highest level possible.



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.

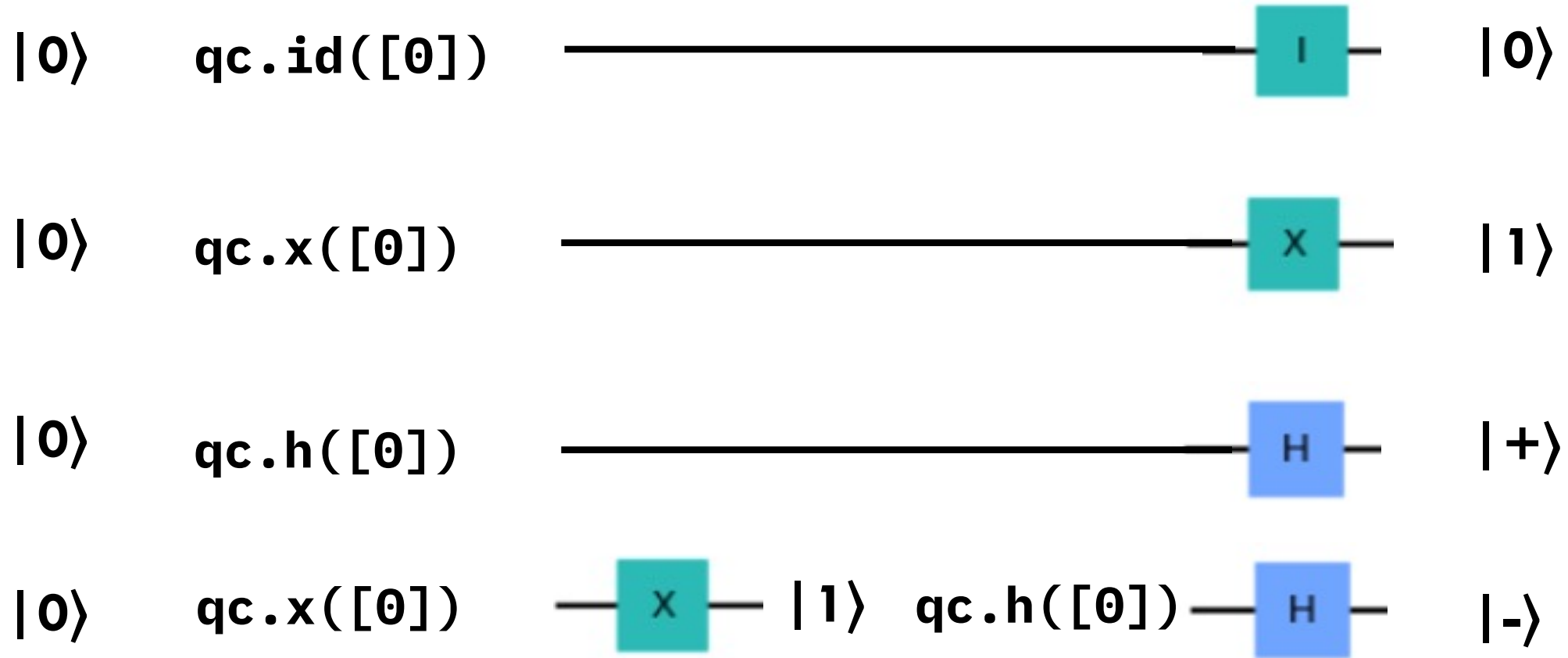
A VOUS DE JOUER !

Programmer un ordinateur quantique à l'aide de Python et 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}
```

EXEMPLES (GENERATION DES 4 ETATS)



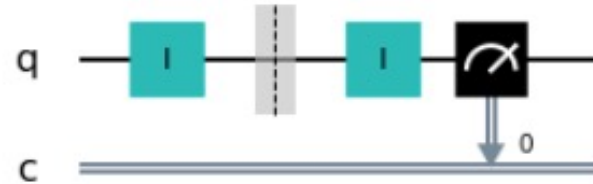
BOB MESURE SUR Z

$|0\rangle$

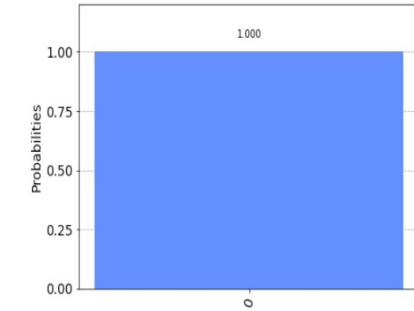
`qc.id([0])`

...

`qc.id([0])`



0

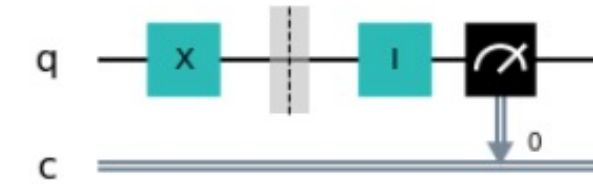


$|1\rangle$

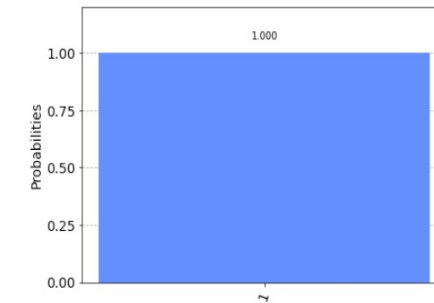
`qc.x([0])`

...

`qc.id([0])`



1

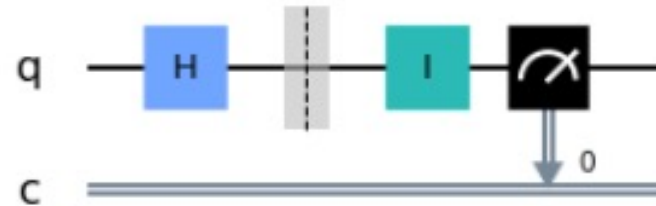


$|+\rangle$

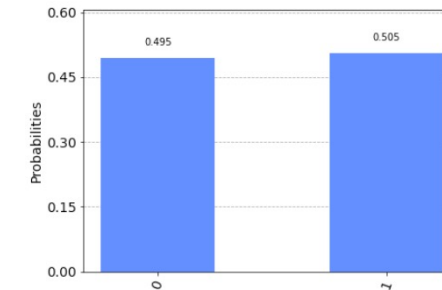
`qc.h([0])`

...

`qc.id([0])`



0,1



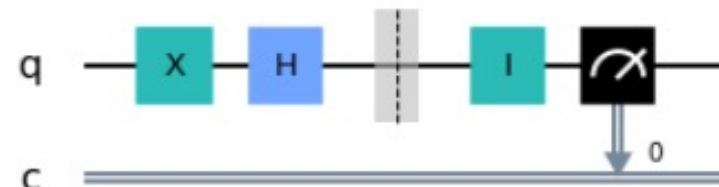
$|-\rangle$

`qc.x([0])`

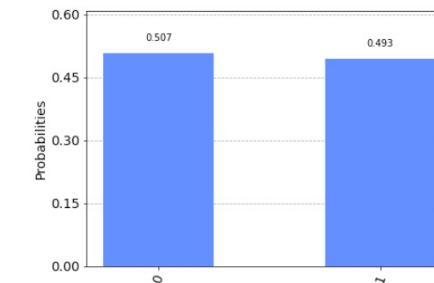
`qc.h([0])`

...

`qc.id([0])`



0,1

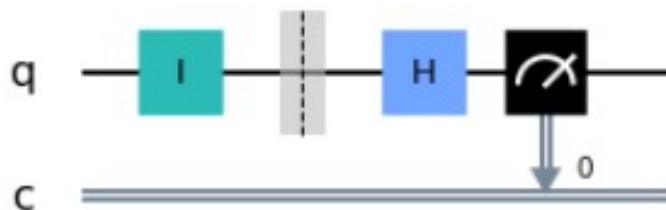


BOB MESURE SUR X

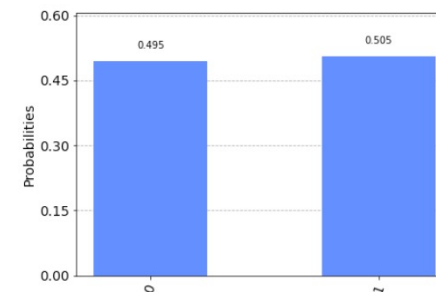
$|0\rangle$

`qc.id([0])`

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



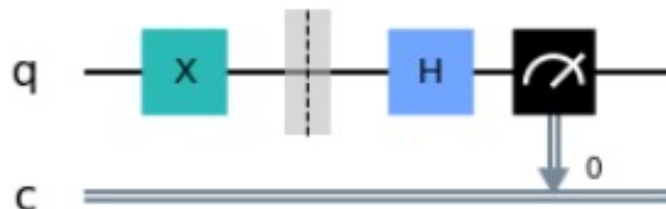
0,1



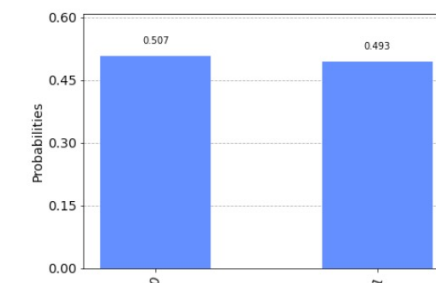
$|1\rangle$

`qc.x([0])`

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



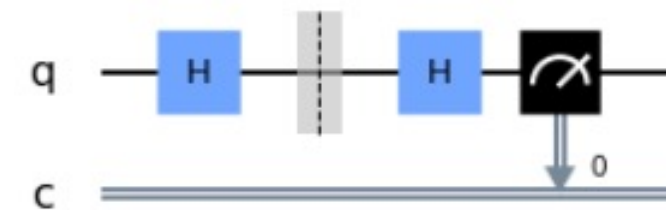
0,1



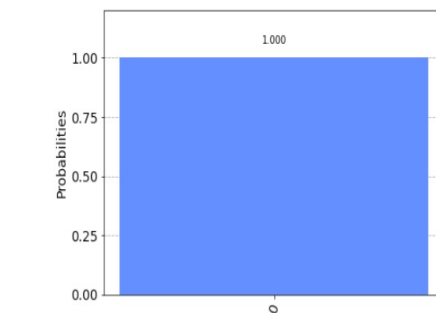
$|+\rangle$

`qc.h([0])`

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



0

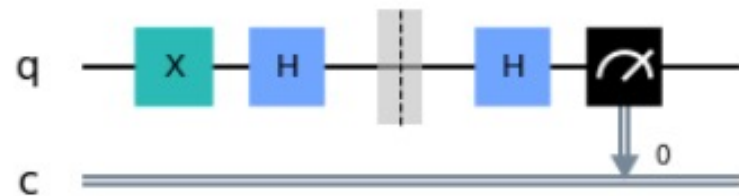


$|-\rangle$

`qc.x([0])`

`qc.h([0])`

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



1

