

NORTHERN  
ARIZONA  
UNIVERSITY®



# INF 638

## Cryptography & Cryptosystems

### Section 6: Quantum Cryptography

**Dr. Bertrand Cambou**

Professor of Practice NAU, Cybersecurity  
School of Informatics, Computing, and Cyber-Systems

[Bertrand.cambou@nau.edu](mailto:Bertrand.cambou@nau.edu)

# INF 638: Cryptography & Cryptosystems

- ❖ 1- Motivation & Definitions
- ❖ 2- Elements of Number theory
- ❖ 3- Early Cryptographic methods
- ❖ 4- Symmetrical Cryptography: DES
- ❖ 5- Symmetrical Cryptography: AES
- ❖ → 6- Quantum Cryptography: Key distribution
- ❖ 7- Elements of Asymmetrical Cryptography
- ❖ 8- Asymmetrical Cryptography: RSA
- ❖ 9- ECC Key Distribution
- ❖ 10- PKI & Digital Signatures
- ❖ 11- Hash Functions
- ❖ 12- Smartcards

## 6- Quantum Cryptography



- ❖ 6-1 History
- ❖ 6-2 Quantum Key Distribution
  - ❖ Protocol BB84 of QKD
  - ❖ Eye dropping of QKD
- ❖ 6-3- Quantum cryptography
  - ❖ Quantum computers
  - ❖ Shor algorithm
  - ❖ Post quantum cryptography

# Two fields for quantum cryptography:

## Key distribution and Crypto-analysis

### ➤ History of Quantum cryptography Key Distribution (QKD)

- Stephen Wiesner (1970's) wrote the paper: “conjugate coding”
- Charles Bennett and Gilles Brassard published in 1984 the basis for:  
“Quantum Key Distribution (QKD)”
- Prototype BB84 of QKD developed in 1991
- Arthur Ekert invented independently a similar protocol in 1991
- The technology has similarity with the one used for optical fiber

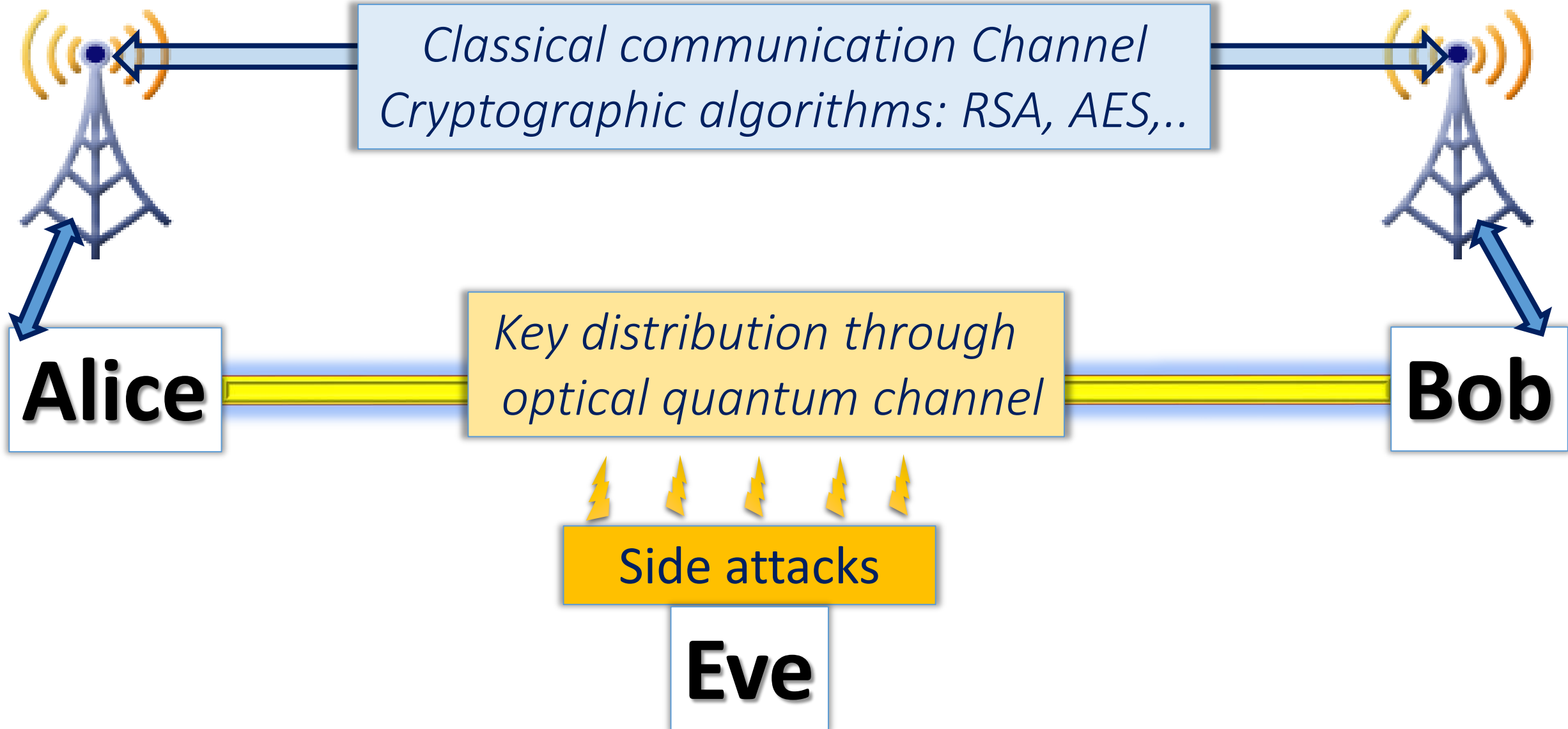
### ➤ Quantum computers for crypto-analysis

- In the late 90's Peter Shor developed an algorithm to break RSA with quantum computers. Can be use to break ECC
- Post quantum computer cryptography as a response to the threat

# Quantum Cryptography

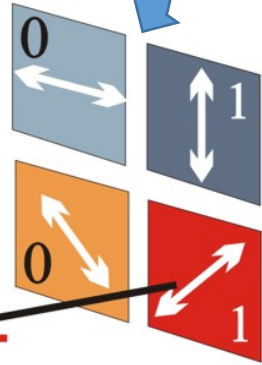
- ❖ 1- History
- ❖ 2- Quantum Key Distribution
  - ❖ Protocol BB84 of QKD
  - ❖ Eye dropping of QKD
- ❖ 4- Quantum cryptography
  - ❖ Quantum computers
  - ❖ Shor algorithm
  - ❖ Post quantum cryptography

# Key distribution with quantum cryptography methods



# BB84 protocol Step-A: Alice transmit a data stream

**Alice  
Random  
Polarizer**



Bit Sequence

**Alice: Random Bases Sequence**



**Data  
Stream:**

Random Polarizer

**+**:  $0^\circ$  -  $90^\circ$

**X**:  $45^\circ$  -  $135^\circ$

Bits: 0 or 1

		Polarizer	
		+	x
Bits	0	0	0
	1	1	1

## BB84 protocol Step A:

Alice transmit using optical communication channel



**A1-** Alice prepare a large stream of bits ( ex 5,000)

**A2-** She prepare random numbers for setting up the polarizer **+** or **X** for each bit

**A3-** Alice transmit the steam with each bit having either a **+** or a **X** polarization

---

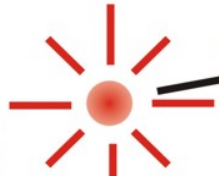
At this point Alice is aware of two streams of data:

- The stream of bits transmitted in the right sequence
- The stream describing the position of her polarizer in the same sequence (**+** or **X** )

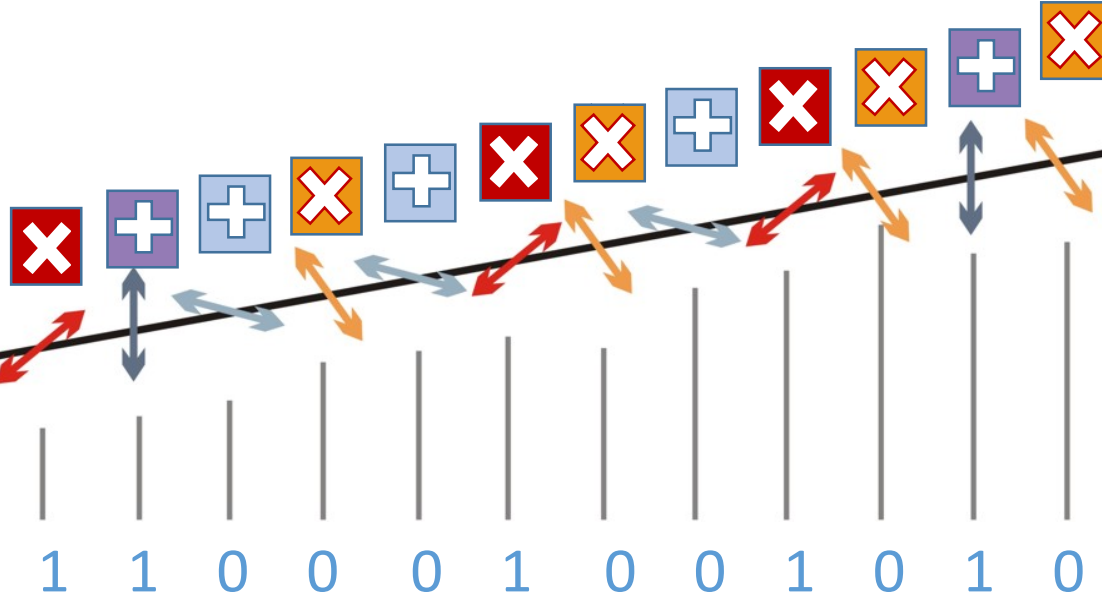
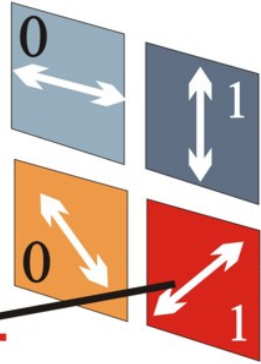


# BB84 protocol Step-B: Bob receive the data stream

Alice



Alice: Bits Sequence



H/V Basis



45° Basis

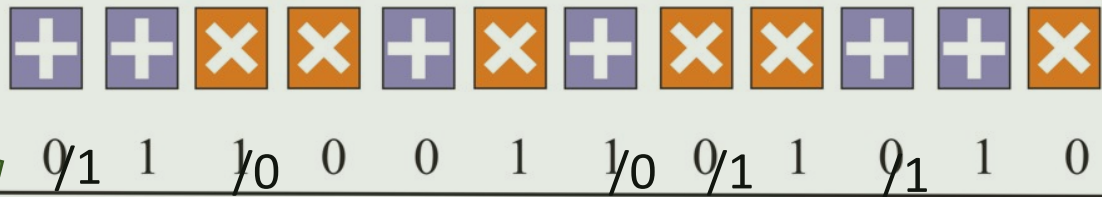


Bob  
Random  
Polarizer

Bases Sequence

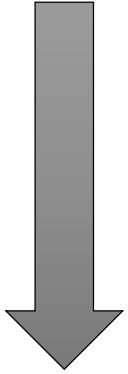
Detection Results

Bob Quantum  
Data Stream



## BB84 protocol Step-B

Bob reception using optical communication channel



---

B4- Bob analyze the data with his own randomly ordered sequence (polarizer **+** or **X**)

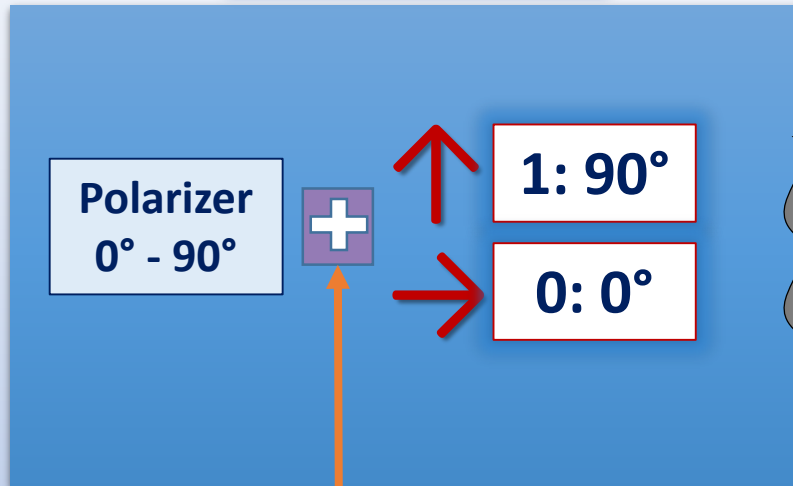
---

At this point Bob has two streams of data:

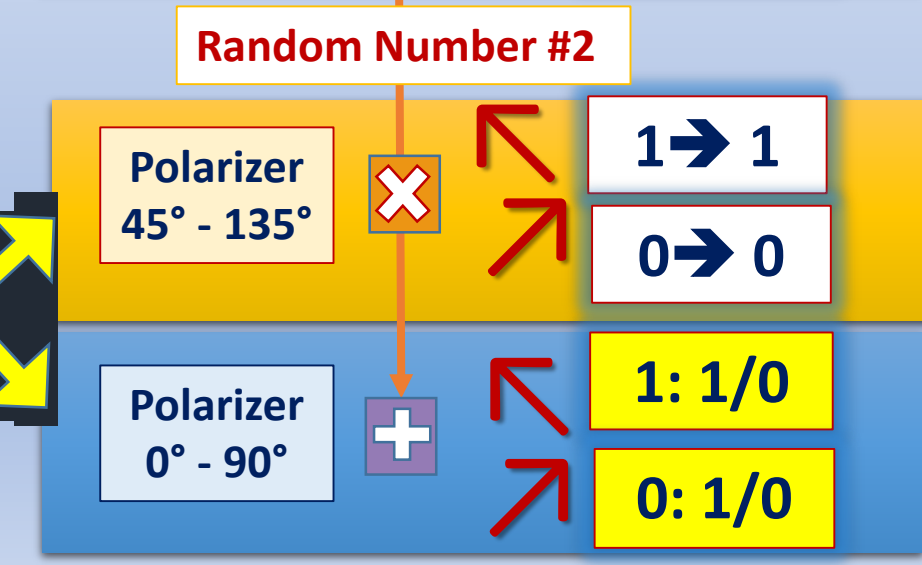
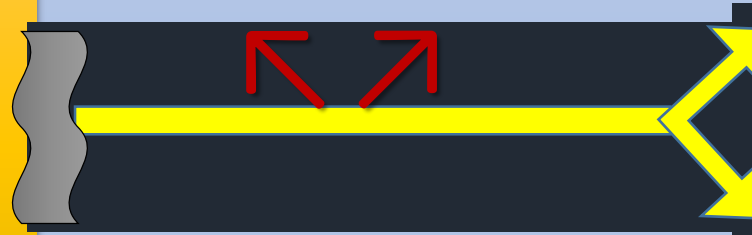
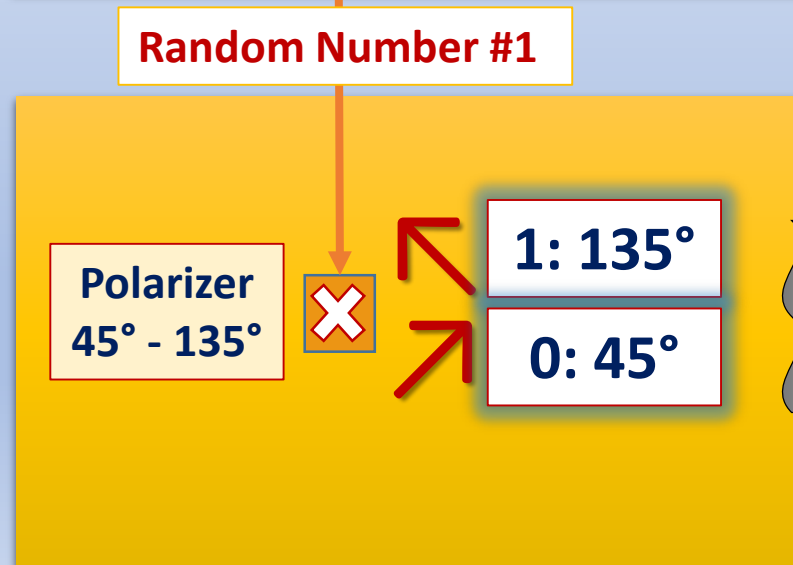
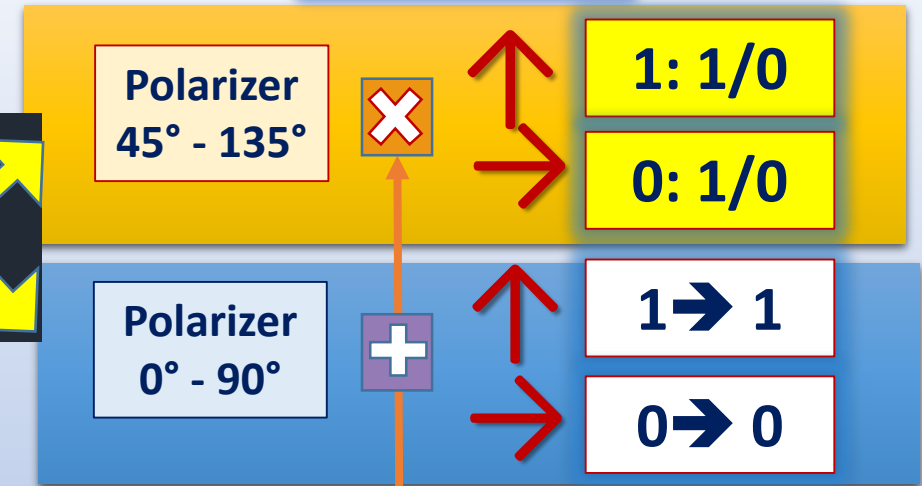
- The data stream of bits passing his polarizer
- The data stream (**+** or **X**) describing the position of his polarizer

# QKD: polarization through random numbers(BB84)







































## Transmit



## Receive



# BB84 protocol Step-C: Analysis of the communication

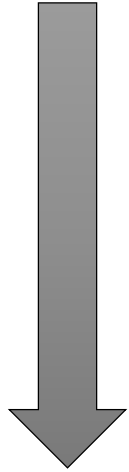
Shared Position		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Transmit	Bases																			
	Bit Sequence	1	1	0	0	0	1	0	0	1	0	1	0	0	1	0	1	1	0	0
		Random use of vertical & horizontal polarization																		
Receive	Bases																			
	Bit Sequence	1/0	1	1/0	0	0	1	1/0	1/0	1	1/0	1	0	1/0	1/0	1/0	1	1	1/0	0
Match (?= 50%)		?	Y	?	Y	Y	Y	?	?	Y	?	Y	Y	?	?	?	Y	Y	?	Y

Stream correctly transmitted: 1, 0, 0, 1, 1, 1, 0, 1, 1, 0

Matching position: 2, 4, 5, 6, 9, 11, 12, 16, 17, 19

## BB84 protocol Step C

Data exchange using regular communication channel



C5- Alice and Bob compare (non secure) the position of their random numbers

C6- They have a common stream of bits based on the matching positions

C7- Compare the error rate – If bad they assume eye dropping

C8- If good Alice and Bob can construct a set of common secret key

➤ Alice knows the positions where Bob has his polarizer on the right orientation

➤ She can pick within these positions the positions giving the secret key

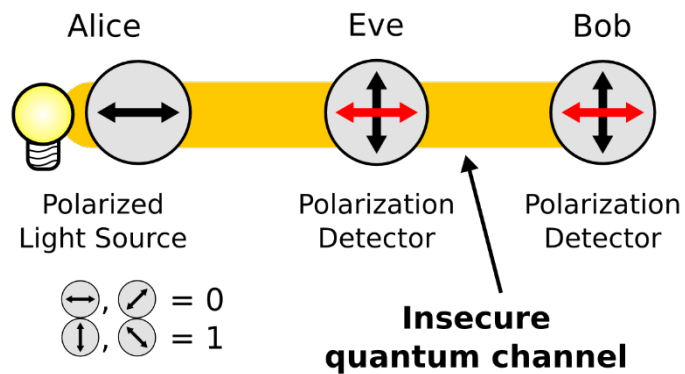
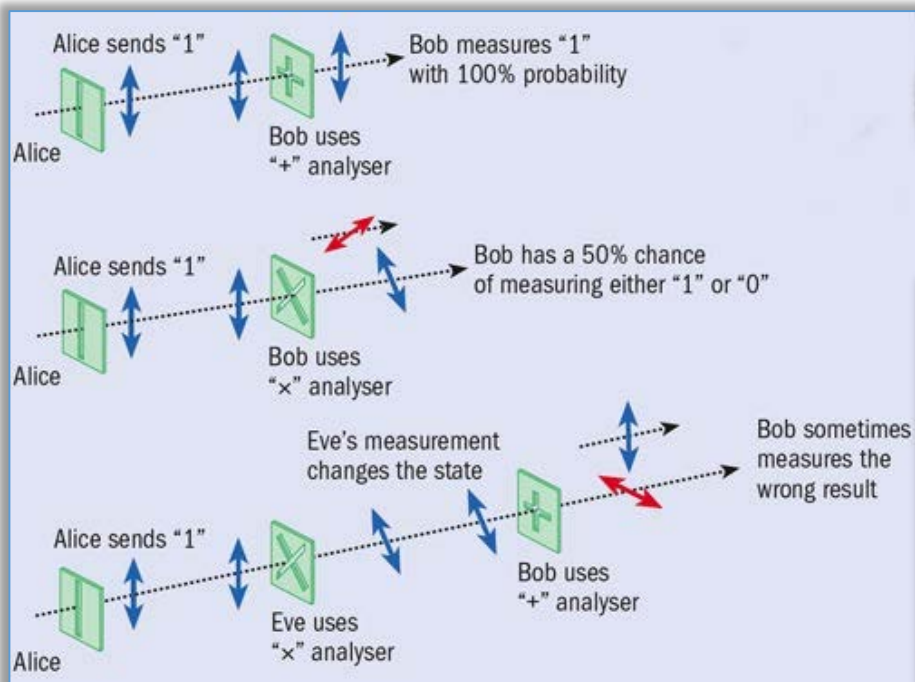
Ex: if the secret key is 1001 Alice gives the position 2, 4, 5, 11

➤ Bob can also communicate with Alice to send the position of a secret key

## Summary: Key distribution with quantum cryptography

- Both Alice and Bob activate their random light polarizer
- Alice send a data stream through the optical link and both polarizer
- Alice & Bob exchange the position of the polarizers during transfer
- Validation – Acceptable error rate to avoid eye dropping
- Key exchange
- Use cryptography for non secured communication

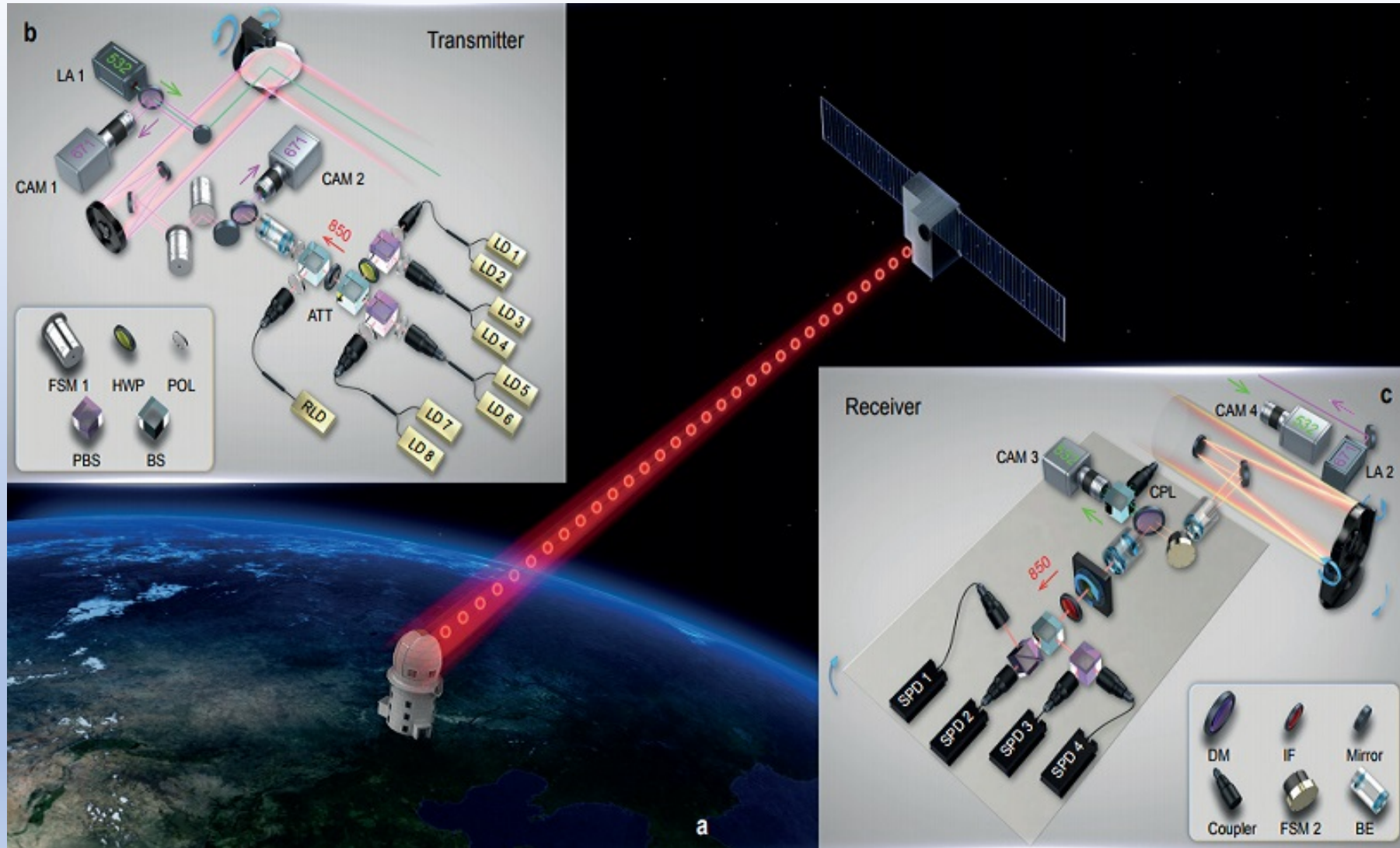
# Eye dropping protection with quantum cryptography



Alice's random bit	0	1	1	0	1	0	0	1
Alice's random bases	+	×	+	×	×	+	×	+
Alice's polarizations	→	↖	↑	↗	↖	→	↗	↑
Eve's random bases	+	+	×	×	+	×	×	+
Polarization Eve measures and sends	→	→	↗	↗	↑	↗	↗	↑
Bob's random bases	+	×	×	×	+	×	+	+
Polarization Bob measures	→	↗	↗	↗	↑	↗	→	↑
Shared secret key	0	0		0				1
Error generated	✓	✗		✓				✓



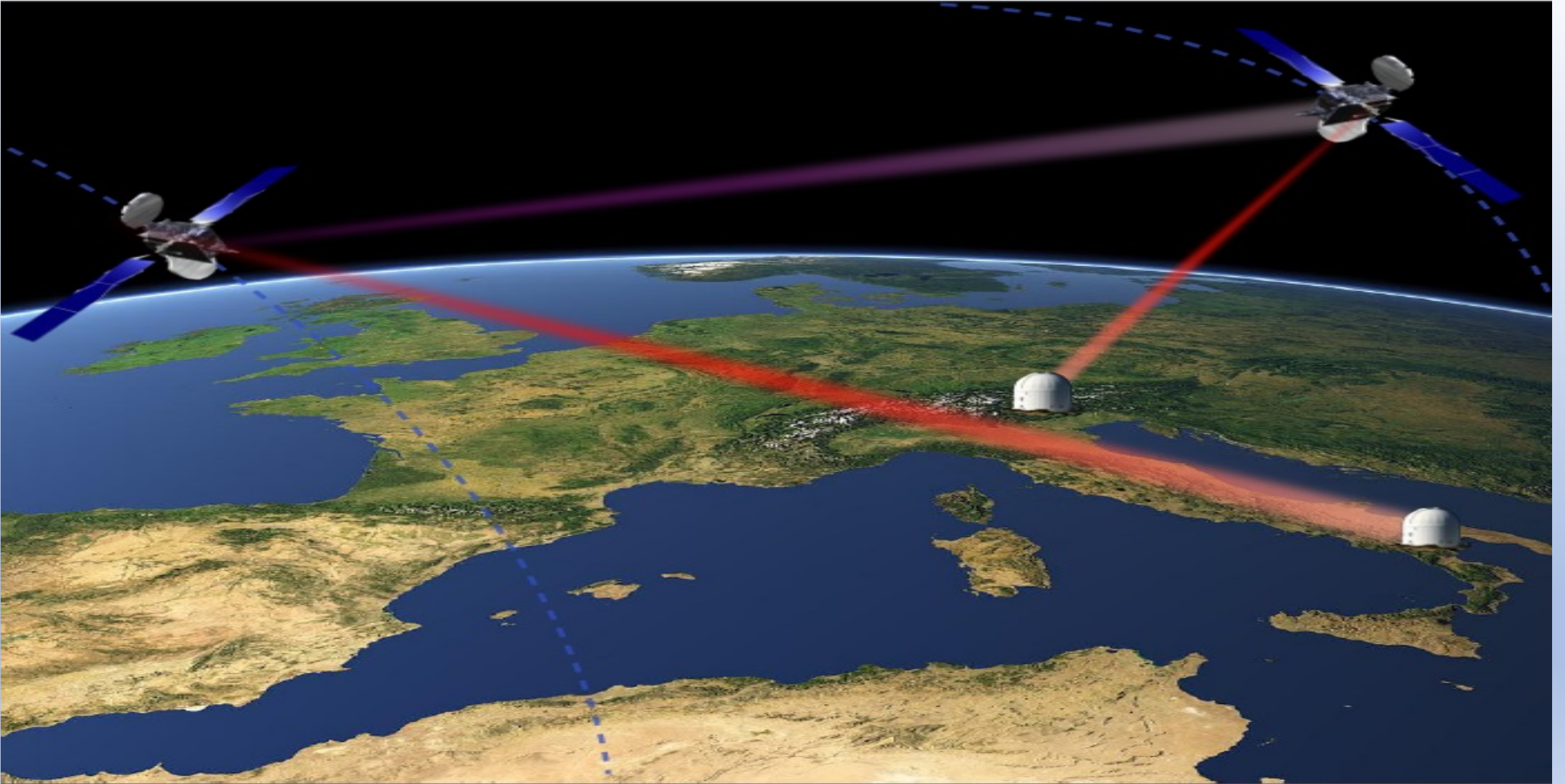
# QKD: example of satellite to ground communication



Satellite to ground quantum key distribution; Shen-Kai Liao and al; *Nature* **549**, 43–47 (07 September 2017)



# QKD: example of ground to ground communication



# Quantum Cryptography

- ❖ 1- History
- ❖ 2- Quantum Key Distribution
  - ❖ Protocol BB84 of QKD
  - ❖ Eye dropping of QKD
- ❖ → 4- Quantum cryptography
  - ❖ Quantum computers
  - ❖ Shor algorithm
  - ❖ Post quantum cryptography

# NSA dreams of quantum computer that can break encryption

Spy agency is apparently not close to a quantum breakthrough, but it's trying.

by [Jon Brodtkin](#) - Jan 2, 2014 3:34pm PST

The National Security Agency is conducting what it calls "basic research" to determine whether it's possible to build a quantum computer that would be useful for breaking encryption. The news isn't surprising—it would be surprising if the NSA *wasn't* researching quantum computing given the [measures it's taken](#) to undermine encryption standards used to [protect Internet communications](#). The NSA's quantum work was described in documents leaked by Edward Snowden and [published today in the Washington Post](#). A three-page [NSA document](#) describes a project to conduct "basic research in quantum physics and architecture/engineering studies to determine if, and how, a cryptologically useful quantum computer can be built."

This is part of a \$79.7 million research program called "Penetrating Hard Targets." A project goal for fiscal 2013 was to "Demonstrate dynamical decoupling and complete quantum control on two semiconductor qubits," the basic building block of a large-scale quantum computer. The NSA description of the program says the agency will "[c]ontinue research of quantum communications technology to support the development of novel Quantum Key Distribution (QKD) attacks and assess the security of new QKD system designs."

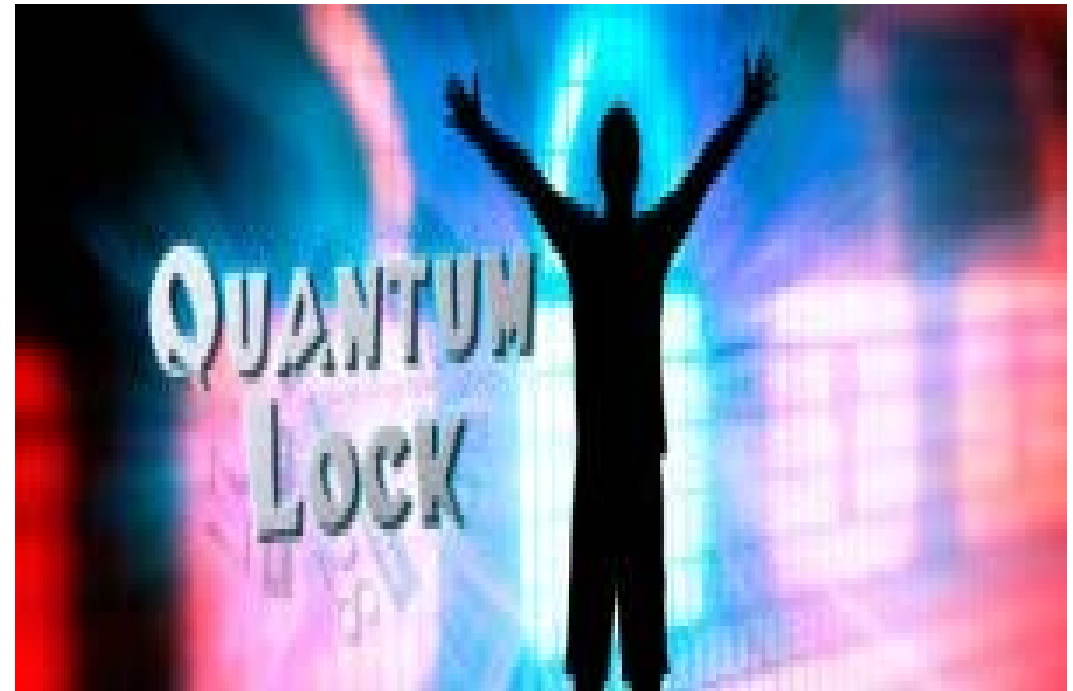
## [QUANTUM CRYPTOGRAPHY: YESTERDAY, TODAY, AND TOMORROW](#)

Does it have a future? Classic cryptology isn't budging, but all depends on QKD. There's nothing specific enough in the NSA document to conclude that the agency is any more advanced in its quantum computing research than the rest of the scientific community. "It seems improbable that the NSA could be that far ahead of the open world without anybody knowing it," MIT Professor of Electrical Engineering and Computer Science Scott Aaronson told the *Post*.

Some of the NSA's experiments are carried out "in large, shielded rooms known as Faraday cages, which are designed to prevent electromagnetic energy from coming in or out," the *Post* wrote.

"Those, according to one brief description, are required 'to keep delicate quantum computing experiments running.'" The article noted that "the NSA appears to regard itself as running neck and neck with quantum computing labs sponsored by the European Union and the Swiss government" but has no hopes of an "immediate breakthrough."

[Another NSA document](#) describing why the agency has classified its research says it's hoping to devise ways to protect US systems against quantum attack. While the NSA wants to "develop cryptanalytic QC [quantum computers] to attack high-grade public key encryption systems," it also seeks "to protect our own systems against adversarial cryptanalytic QC efforts."

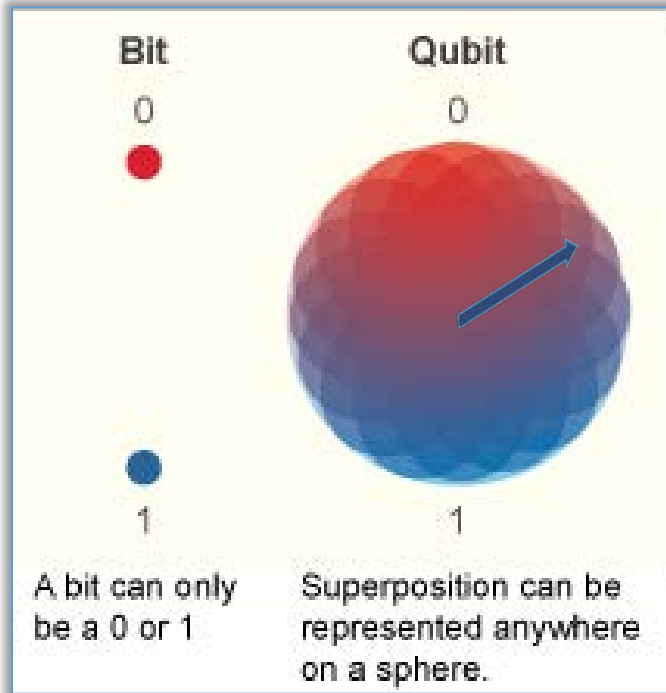


# Why should I care about quantum computers (QC)?

- General QC **are not useful as general-purpose computing** devices.
- QC can solve some **mathematical problems** much faster.
- Large quantum computer could break most **PKI cryptography**:
  - Factoring large numbers (like the ones used by **RSA**)
  - Taking discrete logs over the integers *mod*  $p$  (like regular **DH**)
  - Taking discrete logs over **elliptic curves** (like ECDH and ECDSA).

Classical computers can emulate quantum computers, but only with  
exponential slowdown.

# Basics of quantum computers



**Binary system (Qubit):** two possible positions  $|0\rangle$  or  $|1\rangle$

Probability is:  $|\psi\rangle = C_0 |0\rangle + C_1 |1\rangle$   $C_0^2 + C_1^2 = 1$

One possible event is:  $C_0 = C_1 = 1/\sqrt{2}$

**Projection into two orthogonal directions:**

$$|A\rangle = 1/\sqrt{2} (|0\rangle + |1\rangle)$$

$$C_A = 1/\sqrt{2} (C_0 + C_1)$$

$$|B\rangle = 1/\sqrt{2} (|0\rangle - |1\rangle)$$

$$C_B = 1/\sqrt{2} (C_0 - C_1)$$

$$|\psi\rangle = C_A |A\rangle + C_B |B\rangle$$

$$C_A^2 + C_B^2 = 1$$



# Basics of quantum computers

**Two Qubit:** four possible positions  $|00\rangle$ ,  $|01\rangle$ ,  $|10\rangle$ , or  $|11\rangle$

Probability given by:

$$|\psi\rangle = 1/\sqrt{4} (C_{00}|00\rangle + C_{01}|01\rangle + C_{10}|10\rangle + C_{11}|11\rangle)$$
$$1/4 \sum_{i,j} |C_{ij}|^2 = 1$$

**Three Qubit:** eight possible positions  $|000\rangle$ ,  $|001\rangle$ ,  $|010\rangle$ , ..., or  $|111\rangle$

Probability given by:

$$|\psi\rangle = 1/\sqrt{8} (C_{000}|000\rangle + C_{001}|001\rangle + C_{010}|010\rangle + \dots + C_{111}|111\rangle)$$
$$1/8 \sum_{i,j,k} |C_{ijk}|^2 = 1$$

**Superposition of  $n \in 2^n$  states:**

$$|0000 \dots 00\rangle \rightarrow |0\rangle$$

$$|0000 \dots 01\rangle \rightarrow |1\rangle$$

$$|0000 \dots 10\rangle \rightarrow |2\rangle$$

$$\dots \quad |i\rangle$$

$$|1111 \dots 11\rangle \rightarrow |2^n\rangle$$

$$|\psi\rangle = 1/\sqrt{2^n} \sum_{i=0 \text{ to } 2^n} C_i |i\rangle$$

$$1/2^n \sum_{i=1 \text{ to } 2^n} |C_i|^2 = 1$$

# Basics of quantum computers

*Example superposition of 8  $\in$  256 states:*

$$|00000000\rangle \rightarrow |0\rangle$$

$$|00000001\rangle \rightarrow |1\rangle$$

$$|00000010\rangle \rightarrow |2\rangle$$

$$\dots \quad |i\rangle$$

$$|11111111\rangle \rightarrow |255\rangle$$

Function:  $|\psi_7\rangle = C(|0\rangle + |7\rangle + |14\rangle + \dots + |245\rangle + |255\rangle)$

Add 5:  $|\psi_7\rangle + 5 = C(|5\rangle + |12\rangle + |19\rangle + \dots + |250\rangle + |1\rangle)$

## Boolean logic of quantum computers

### *Example of Not-Gate for one quabit*

$$\text{Unot } |0\rangle = |1\rangle$$

$$\text{Unot } |1\rangle = |0\rangle$$

$$\text{Unot } (C_0 |0\rangle + C_1 |1\rangle) = C_1 |0\rangle + C_0 |1\rangle$$

$$\text{Unot} \begin{pmatrix} C_0 \\ C_1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} C_0 \\ C_1 \end{pmatrix} = \begin{pmatrix} C_1 \\ C_0 \end{pmatrix}$$



# Boolean logic of quantum computers

## *Example of NOT-gate for two quabits*

$$|\psi_1\rangle = C_{0,1}|0\rangle + C_{1,1}|1\rangle$$

$$|\psi_2\rangle = C_{0,2}|0\rangle + C_{1,2}|1\rangle$$

$$\text{Unot } |\psi_1 \psi_2\rangle = |\psi_2 \psi_1\rangle$$

$$\text{Unot} \begin{pmatrix} C_{0,1} \\ C_{1,1} \\ C_{0,2} \\ C_{1,2} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} C_{0,1} \\ C_{1,1} \\ C_{0,2} \\ C_{1,2} \end{pmatrix} = \begin{pmatrix} C_{0,2} \\ C_{1,2} \\ C_{0,1} \\ C_{1,1} \end{pmatrix}$$

# Boolean logic of quantum computers

## *Example of Control NOT for two quabits*

$$U_{\text{cnot}}|00\rangle = |00\rangle$$

$$U_{\text{cnot}}|01\rangle = |01\rangle$$

$$U_{\text{cnot}}|10\rangle = |11\rangle$$

$$U_{\text{cnot}}|11\rangle = |10\rangle$$

$$U_{\text{cnot}} \begin{pmatrix} C_{0,1} \\ C_{1,1} \\ C_{0,2} \\ C_{1,2} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} C_{0,1} \\ C_{1,1} \\ C_{0,2} \\ C_{1,2} \end{pmatrix} = \begin{pmatrix} C_{0,1} \\ C_{1,1} \\ C_{1,2} \\ C_{0,2} \end{pmatrix}$$

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$|01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$|10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$|11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

# Boolean logic of quantum computers

## *Z-gate for One quabit*

$$U_z |0\rangle = |0\rangle$$

$$U_z |1\rangle = -|1\rangle$$

$$|\psi\rangle = C_{0,1} |0\rangle + C_{1,1} |1\rangle$$

$$U_z |\psi\rangle = C_{0,1} |0\rangle - C_{1,1} |1\rangle$$

$$U_z \begin{pmatrix} C_{0,1} \\ C_{1,1} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} C_{0,1} \\ C_{1,1} \end{pmatrix} = \begin{pmatrix} C_{0,1} \\ -C_{1,1} \end{pmatrix}$$

# Boolean logic of quantum computers

## *S-gate for One quabit*

$$U_S |0\rangle = |0\rangle$$

$$U_S |1\rangle = i|1\rangle$$

$$U_S \begin{pmatrix} C_{0,1} \\ C_{1,1} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \begin{pmatrix} C_{0,1} \\ C_{1,1} \end{pmatrix} = \begin{pmatrix} C_{1,1} \\ iC_{0,1} \end{pmatrix}$$

## *T-gate for One quabit*

$$U_T |0\rangle = |0\rangle$$

$$U_T |1\rangle = \exp(i\pi/4) |1\rangle$$

$$U_T \begin{pmatrix} C_{0,1} \\ C_{1,1} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & \exp(i\pi/4) \end{pmatrix} \begin{pmatrix} C_{0,1} \\ C_{1,1} \end{pmatrix} = \begin{pmatrix} C_{1,1} \\ \exp(i\pi/4)C_{0,1} \end{pmatrix}$$

## *H-gate for One quabit*

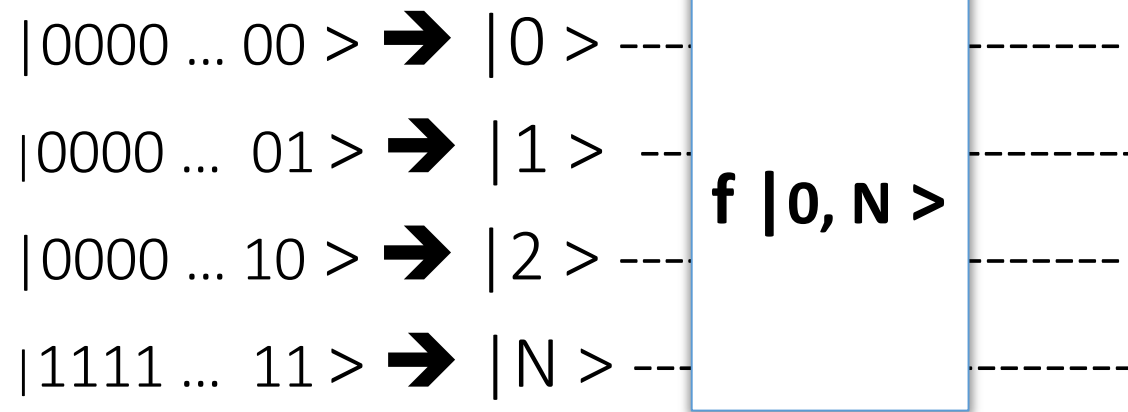
$$U_H |0\rangle = 1/\sqrt{2} (|0\rangle + |1\rangle)$$

$$U_H |1\rangle = 1/\sqrt{2} (|0\rangle - |1\rangle)$$

$$U_H \begin{pmatrix} C_{0,1} \\ C_{1,1} \end{pmatrix} = 1/\sqrt{2} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} C_{0,1} \\ C_{1,1} \end{pmatrix} = 1/\sqrt{2} \begin{pmatrix} C_{0,1} + C_{1,1} \\ C_{0,1} - C_{1,1} \end{pmatrix}$$

# Entanglement of quantum computers

*Quantum Entanglement:*  $f |0, 1\rangle = f[|0\rangle + |1\rangle + |2\rangle + \dots + |N\rangle]$



- ☐ The most probable state can be extracted from a quantum computer in one computing cycle.
- ☐ Complicated mathematical computations can be done in parallel.
- ☐ Serial algorithms do not benefit from the architecture.

# Quantum Cryptography

- ❖ 1- History
- ❖ 2- Quantum Key Distribution
  - ❖ Protocol BB84 of QKD
  - ❖ Eye dropping of QKD
- ❖ 4- Quantum cryptography
  - ❖ Quantum computers
  - ❖ Shor algorithm
  - ❖ Post quantum cryptography

# Peter Shor's Factoring Algorithm

## Objective:

Let us assume that  $N = P \times Q$  with  $P$  and  $Q$  prime

Knowing only  $N$ , can we find  $P$  and  $Q$ ?

➔ if yes, we can break RSA

## Background in number theory

- i) If  $n$  is the length of  $N$  in bits. Total entropy is  $2^n$
- ii) Assuming that we find  $a$  such as  $a^2 \equiv 1 \pmod{N}$   
 $(a - 1)(a + 1) = a^2 - 1 \rightarrow (a - 1)(a + 1) \equiv 0 \pmod{N}$
- iii) The order of  $a$  modulo  $N$  is the smallest integer  $r$  such as:  
 $a^r \equiv 1 \pmod{N}$  with  $N > r > 0$ .
- iv) Assuming that  $r$  is even:  $(a^{r/2} - 1)(a^{r/2} + 1) \equiv 0 \pmod{N}$   
 $a^{r/2} \equiv b \pmod{N}$   
If  $N$  does not divide  $(b - 1)$  or  $(b + 1)$  then:  
 $\rightarrow P = \gcd(b - 1, N)$  and  $Q = \gcd(b + 1, N)$   
[  $P$  &  $Q$  are the only numbers below  $N$  with  $P \cdot Q \equiv 0 \pmod{N}$  ]



# Shor's Factoring Algorithm:

**Step-1:** pick  $a$  an integer  $N > a > 0$

**Step-2:** Find  $r$  with a quantum computer  $a^r \equiv 1 \pmod{N}$   
Try all  $r$ 's lower than  $N$

**Step-3:** If  $r$  even, and  $N$  does not divide  $(a^{r/2} - 1)$  or  $(a^{r/2} + 1)$   
 $\rightarrow P = \gcd(a^{r/2} - 1, N)$  and  $Q = \gcd(a^{r/2} + 1, N)$

**Step-4:** If  $r$  odd or if  $N$  divides  $(a^{r/2} - 1)$  or  $(a^{r/2} + 1)$   
 $\rightarrow$  Go back to **step-1** and pick a different  $a$  to find a different  $r$

# Peter Shor's Factoring Algorithm

Example#1:  $N = 15$      $a=7$      $f(k) = a^k \bmod N$

$k$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
$f(k)$	1	7	4	13	1	7	4	13	1	7	4	13	1	7	4	13	1	7

$$r=4 \Rightarrow a^4 = 7^4 = 2401 \not\equiv 1 \bmod 15$$

$$a^{r/2} = 7^2 = 49 \not\equiv 4 \bmod 15$$

$$\gcd(a^{r/2} + 1, N) = \gcd(5, 15) = 5$$

$$\gcd(a^{r/2} - 1, N) = \gcd(3, 15) = 3$$

Example#2:  $N = 15$      $a=2$      $f(k) = a^k \bmod N$

$k$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
$f(k)$	1	2	4	8	1	2	4	8	1	2	4	8	1	2	4	8	1	2

$$r=4 \Rightarrow a^4 = 2^4 = 16 \not\equiv 1 \bmod 15$$

$$a^{r/2} = 2^2 = 4 \not\equiv 4 \bmod 15$$

$$\gcd(a^{r/2} + 1, N) = \gcd(5, 15) = 5$$

$$\gcd(a^{r/2} - 1, N) = \gcd(3, 15) = 3$$

# Peter Shor's Factoring Algorithm

Example#3:  $N = 77$      $a=10$      $f(k) = a^k \bmod N$

$k$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
$f(k)$	1	10	23	76	67	54	1	10	23	76	67	54	1	10	23	76	67	54

$$r=6 \rightarrow a^6 = 10^6 = 1000^2 = (-1)^2 \equiv 1 \bmod 77$$

$$a^{r/2} = 10^3 \not\equiv \pm 1 \bmod 77$$

$$\gcd(a^{r/2} + 1, N) = \gcd(77, 77) = 77 \rightarrow \text{not good!}$$

$$\gcd(a^{r/2} - 1, N) = \gcd(75, 77) = 1 \rightarrow \text{not good!}$$

(Note that  $77 \times 13 = 1001$ , so  $1000 \equiv -1 \bmod 77$ )

Example#4:  $N = 77$      $a=8$      $f(k) = a^k \bmod N$

$k$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
$f(k)$	1	8	64	50	15	43	36	57	71	29	1	8	64	50	15	43	36	57

$$r=10 \rightarrow a^{10} = 8^{10} = 43 \times 43 = 1849 = 1 + (77 \times 24) \equiv 1 \bmod 77$$

$$a^{r/2} = 8^5 \not\equiv \pm 1 \bmod 77$$

$$\gcd(a^{r/2} + 1, N) = \gcd(44, 77) = 11$$

$$\gcd(a^{r/2} - 1, N) = \gcd(42, 77) = 7$$

# Quantum Cryptography

- ❖ 1- History
- ❖ 2- Quantum Key Distribution
  - ❖ Protocol BB84 of QKD
  - ❖ Eye dropping of QKD
- ❖ 4- Quantum cryptography
  - ❖ Quantum computers
  - ❖ Shor algorithm
  - ❖ Post quantum cryptography

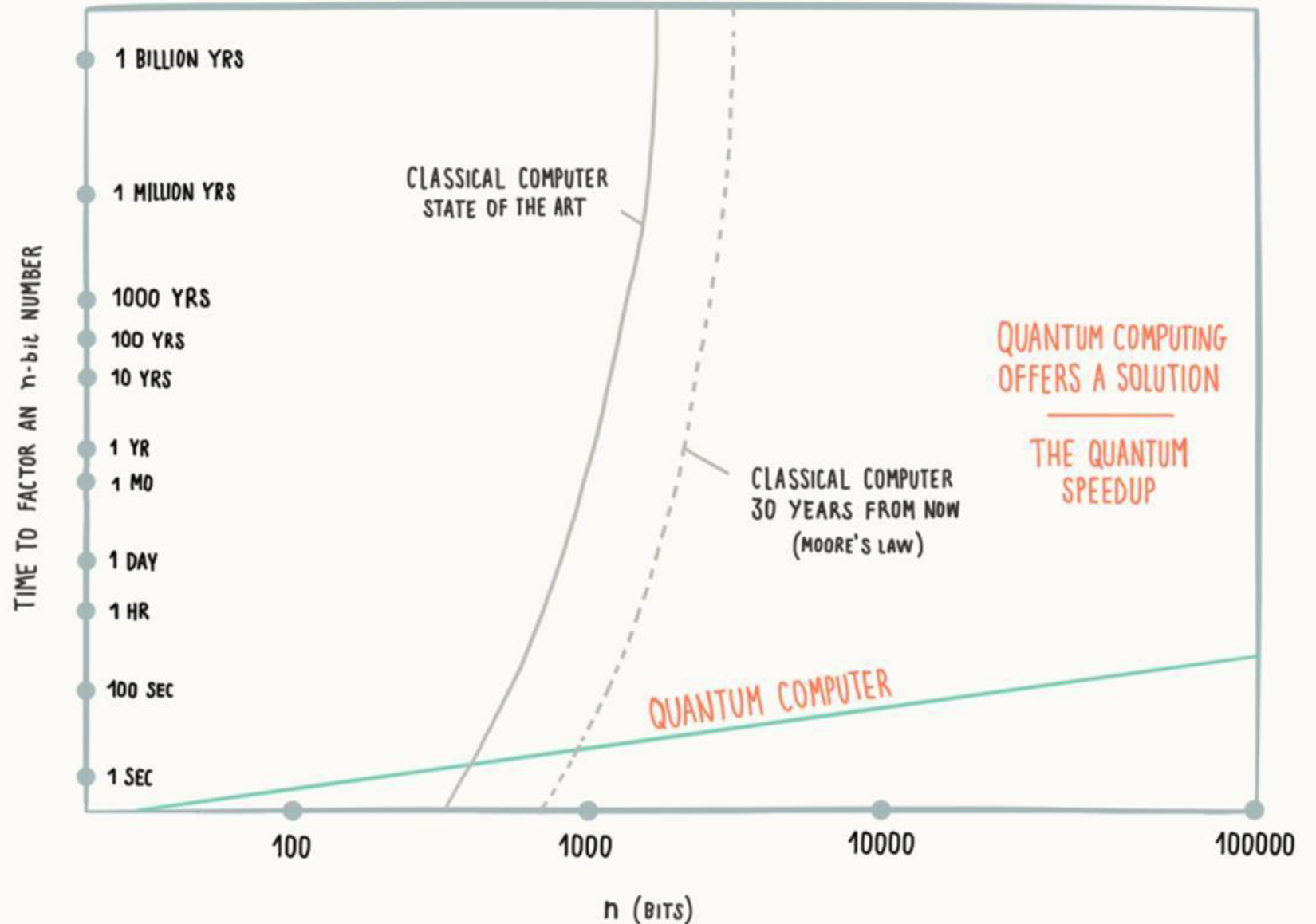
# Efficiency of quantum computers

For quantum computers, Peter Shor's factoring algorithm is *polynomial*

Example: time to factor 1,000 binary bit number

Classical: *>100 years*

Quantum: *1 minute*



# Efficiency of quantum computers

Factoring algorithm (RSA)			EC discrete logarithm (ECC)			classical
$n$	$\approx \#$ qubits	time	$n$	$\approx \#$ qubits	time	time
	$2n$	$4n^3$		$f'(n)$ ( $f(n)$ )	$360n^3$	
512	1024	$0.54 \cdot 10^9$	110	700 (800)	$0.5 \cdot 10^9$	$C$
1024	2048	$4.3 \cdot 10^9$	163	1000 (1200)	$1.6 \cdot 10^9$	$C \cdot 10^8$
2048	4096	$34 \cdot 10^9$	224	1300 (1600)	$4.0 \cdot 10^9$	$C \cdot 10^{17}$
3072	6144	$120 \cdot 10^9$	256	1500 (1800)	$6.0 \cdot 10^9$	$C \cdot 10^{22}$
15360	30720	$1.5 \cdot 10^{13}$	512	2800 (3600)	$50 \cdot 10^9$	$C \cdot 10^{60}$

In this table,  $n$  refers to the modulus size for RSA, and the field size for ECC. Look at the rightmost column, which represents time taken by the classical algorithm, and compare it to the "time" columns, which represent how much a quantum computer would take. As  $n$  increases, the amount of time the quantum computer would take stays in the same ballpark, whereas, for a classical computer, it increases (almost) exponentially. Therefore, increasing  $n$  is an effective strategy for keeping up with ever-faster classical computers, but it is ineffective at increasing the run time for a quantum computer.

Cryptographic Algorithm	Type	Purpose	Impact from large-scale quantum computer
AES-256	Symmetric key	Encryption	Larger key sizes needed
SHA-256, SHA-3		Hash functions	Larger output needed
RSA	Public key	Signatures, key establishment	No longer secure
ECDSA, ECDH (Elliptic Curve Cryptography)	Public key	Signatures, key exchange	No longer secure
DSA (Finite Field Cryptography)	Public key	Signatures, key exchange	No longer secure

**Table 1 - Impact of Quantum Computing on Common Cryptographic Algorithms**

## **Definition of post quantum computer cryptography:**

Cryptography capable to resist attacks by quantum computers

NIST concluded that quantum computers will break known asymmetrical algorithms  
(NIST= National Institute of Standards and Technology)

NIST has started a tender to find a new standard



# Post Quantum Computer Algorithm

## Timeline

- ▶ Fall 2016 – formal Call For Proposals
- ▶ Nov 2017 – Deadline for submissions
- ▶ 3–5 years – Analysis phase
  - NIST will report its findings
- ▶ 2 years later – Draft standards ready
- ▶ Workshops
  - Early 2018 – submitter's presentations
  - One or two during the analysis phase

## Time schedule?

These large quantum computers don't *officially* exist yet.

If you look at the qubits column, you'll see that these attacks require large universal quantum computers. The state of the art in those only has a handful of qubits. In 2011, IBM successfully factored 143 using a 4-qubit quantum computer. Scaling the number of qubits up is troublesome. In that light, larger key sizes may prove effective after all; we simply don't know yet how hard it is to build quantum computers that big.

D-wave, a quantum computing company, has produced computers with 128 and 512 qubits and even >1000 qubits. While there is some discussion if D-waves provide quantum speedup or are even real quantum computers at all; there is no discussion that they are not *universal* quantum computers. Specifically, they only claim to solve one particular problem called quantum annealing. The 1000 qubit D-Wave 2X cannot factor RSA moduli of ~512 bits or solve discrete logs on curves of ~120 bits.

The systems at risk implement asymmetric encryption, signatures, and Diffie-Hellman key exchanges. That's no accident: all post-quantum alternatives are asymmetric algorithms. Post-quantum secure symmetric cryptography is easier: we can just use bigger key sizes, which are still small enough to be practical and result in fast primitives. Quantum computers simply halve the security level, so all we need to do to maintain a 128 bit security level is to use ciphers with 256 bit keys, like Salsa20.

Quantum computers also have an advantage against SIDH, but both are still exponential in the field size. The SIDH scheme in the new paper has 192 bits of security against a classical attacker, but still has 128 bits of security against a quantum attacker. That's in the same ballpark as most symmetric cryptography, and better than the 2048-bit RSA certificates that underpin the security of the Internet.

NORTHERN  
ARIZONA  
UNIVERSITY®



# QUESTIONS ?

**Dr. Bertrand Cambou**

Professor of Practice NAU, Cybersecurity  
School of Informatics, Computing, and Cyber-Systems

[Bertrand.cambou@nau.edu](mailto:Bertrand.cambou@nau.edu)