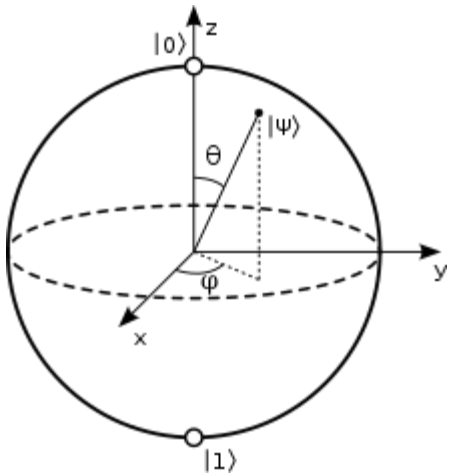
Intro to Quantum Computing

Recap...and Beyond



QC in a Nutshell

QC in a Nutshell

- A new computational model

QC in a Nutshell

- A new computational model

If QM seem so difficult to simulate on a classical model of computation, why not try to create a new model harnessing the power of QM formalism?

QC in a Nutshell

- A new computational model

If QM seem so difficult to simulate on a classical model of computation, why not try to create a new model harnessing the power of QM formalism?

- Instead of bits, we have Qbits.

QC in a Nutshell

- A new computational model

If QM seem so difficult to simulate on a classical model of computation, why not try to create a new model harnessing the power of QM formalism?

- Instead of bits, we have Qbits.
- *Superposition, entanglement, interference* etc.

QC in a Nutshell

- A new computational model

If QM seem so difficult to simulate on a classical model of computation, why not try to create a new model harnessing the power of QM formalism?

- Instead of bits, we have Qbits.
- *Superposition, entanglement, interference* etc.
- These are properties that in certain cases can provide a significant computational speedup.

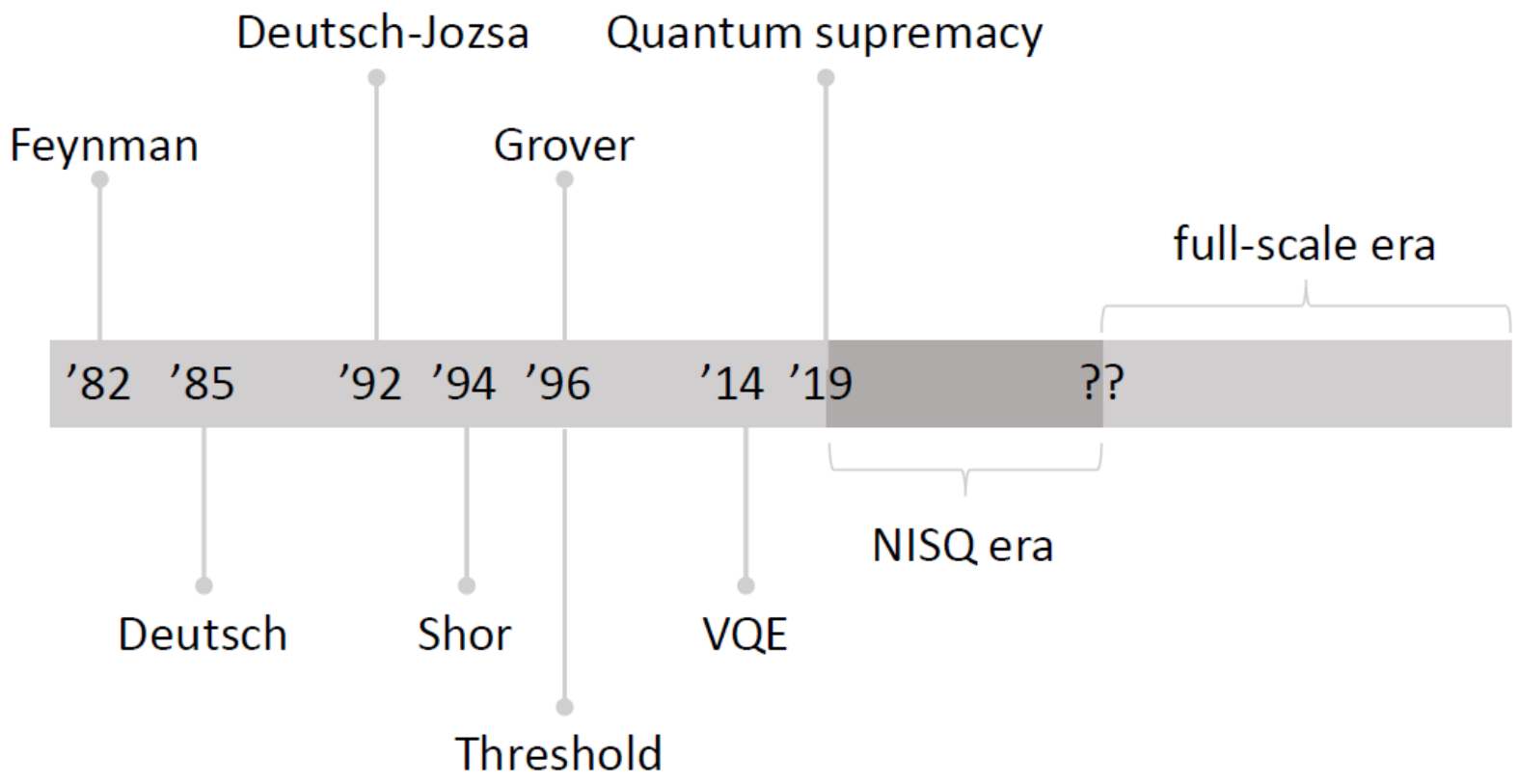
QC in a Nutshell

- A new computational model

If QM seem so difficult to simulate on a classical model of computation, why not try to create a new model harnessing the power of QM formalism?

- Instead of bits, we have Qbits.
- *Superposition, entanglement, interference* etc.
- These are properties that in certain cases can provide a significant computational speedup.
- Quantum algorithm zoo: *quantumalgorithmzoo.org*

Timeline


Deutsch-Jozsa
Quantum supremacy
Feynman
Grover
full-scale era
'82
'85
'92
'94
'96
'14
'19
??
NISQ era
Deutsch
Shor
VQE
Threshold

Timeline

- **VQE**: Variational Quantum Eigensolver

Timeline

- **VQE**: Variational Quantum Eigensolver
- **Quantum Supremacy** demonstrated by Google in 2019 (Boson Sampling).

Timeline

- **VQE**: Variational Quantum Eigensolver
- **Quantum Supremacy** demonstrated by Google in 2019 (Boson Sampling).
- **NISQ**: Noise-Intermediate Scale Quantum (Computer). Term coined by John Preskill (Caltech) https://www.arxiv-vanity.com/papers/1801.00862/

Timeline

- **VQE**: Variational Quantum Eigensolver
- **Quantum Supremacy** demonstrated by Google in 2019 (Boson Sampling).
- **NISQ**: Noise-Intermediate Scale Quantum (Computer). Term coined by John Preskill (Caltech) https://www.arxiv-vanity.com/papers/1801.00862/
- Much of the current research on QM is on finding real-world applications of quantum computing using NISQs.

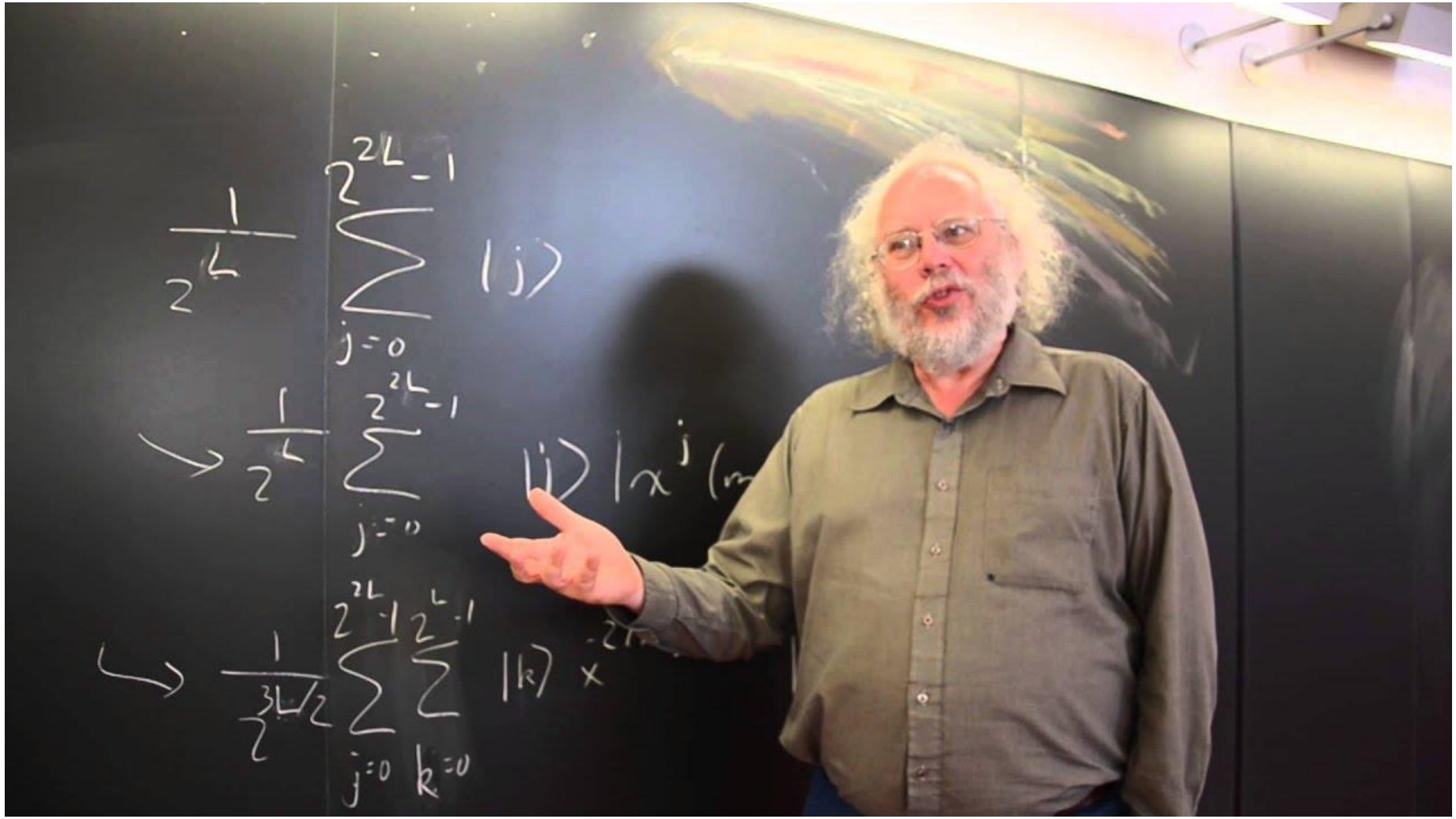
Timeline

- **VQE**: Variational Quantum Eigensolver
- **Quantum Supremacy** demonstrated by Google in 2019 (Boson Sampling).
- **NISQ**: Noise-Intermediate Scale Quantum (Computer). Term coined by John Preskill (Caltech) https://www.arxiv-vanity.com/papers/1801.00862/
- Much of the current research on QM is on finding real-world applications of quantum computing using NISQs.
- Not always possible! Many (most!) QC algorithms *do* require noise-free, full-scale quantum machines!

Some Quantum Algorithms

Algorithm	**Function**	**Speed-up**	**Era**
Shor	factoring	super-polynomial	full-scale
Grover	search	polynomial	full-scale
HHL	linear algebra	super-polynomial	full-scale
QPE	chemistry	super-polynomial	full-scale
VQE	chemistry	heuristic*	NISQ
Annealing	optimisation	heuristic	NISQ
QAOA	optimisation	heuristic	NISQ

Shor's Algorithm



Shor’s Algorithm

- Discovered (invented) in 1994 by Peter Shor (MIT), then at Bell Labs, New Jersey.

Shor’s Algorithm

- Discovered (invented) in 1994 by Peter Shor (MIT), then at Bell Labs, New Jersey.
- The most dramatic result demonstrating the power of QC.

Shor’s Algorithm

- Discovered (invented) in 1994 by Peter Shor (MIT), then at Bell Labs, New Jersey.
- The most dramatic result demonstrating the power of QC.
- Very clever, yet not insanely difficult.

Shor’s Algorithm

- Discovered (invented) in 1994 by Peter Shor (MIT), then at Bell Labs, New Jersey.
- The most dramatic result demonstrating the power of QC.
- Very clever, yet not insanely difficult.
- Concerns the Integer Factorization problem (equivalently the Discrete Logarithm)

Shor's Algorithm

- Discovered (invented) in 1994 by Peter Shor (MIT), then at Bell Labs, New Jersey.
- The most dramatic result demonstrating the power of QC.
- Very clever, yet not insanely difficult.
- Concerns the Integer Factorization problem (equivalently the Discrete Logarithm)
- For which no known classical algorithm can solve it in poly-time.

Shor's Algorithm

- Discovered (invented) in 1994 by Peter Shor (MIT), then at Bell Labs, New Jersey.
- The most dramatic result demonstrating the power of QC.
- Very clever, yet not insanely difficult.
- Concerns the Integer Factorization problem (equivalently the Discrete Logarithm)
- For which no known classical algorithm can solve it in poly-time.
- And probably won't ever be discovered…

Shor’s Algorithm

- It is an algorithm for the following problem:

Shor's Algorithm

- It is an algorithm for the following problem:
- Given $\boldsymbol{N = p \cdot q}$ where both $\boldsymbol{p, q}$ are *prime* numbers, find $\boldsymbol{p, q}$.

Shor's Algorithm

- It is an algorithm for the following problem:
- Given $\boldsymbol{N = p \cdot q}$ where both $\boldsymbol{p, q}$ are *prime* numbers, find $\boldsymbol{p, q}$.
- $\boldsymbol{N = 171 \Rightarrow p = 3, q = 19.}$

Shor's Algorithm

- It is an algorithm for the following problem:
- Given $\boldsymbol{N = p \cdot q}$ where both $\boldsymbol{p, q}$ are *prime* numbers, find $\boldsymbol{p, q}$.
- $\boldsymbol{N = 171 \Rightarrow p = 3, q = 19.}$
- $\boldsymbol{N = 18848997157? \Rightarrow p = 13729, q = 13729933.}$

Shor's Algorithm

- It is an algorithm for the following problem:
- Given $\boldsymbol{N = p \cdot q}$ where both $\boldsymbol{p, q}$ are *prime* numbers, find $\boldsymbol{p, q}$.
- $\boldsymbol{N = 171 \Rightarrow p = 3, q = 19.}$
- $\boldsymbol{N = 18848997157? \Rightarrow p = 13729, q = 13729933.}$
- How can we find $\boldsymbol{p, q}$?

Shor's Algorithm

- It is an algorithm for the following problem:
- Given $\boldsymbol{N = p \cdot q}$ where both $\boldsymbol{p, q}$ are *prime* numbers, find $\boldsymbol{p, q}$.
- $\boldsymbol{N = 171 \Rightarrow p = 3, q = 19.}$
- $\boldsymbol{N = 18848997157? \Rightarrow p = 13729, q = 13729933.}$
- How can we find $\boldsymbol{p, q}$?
- *Trivial solution*: try all divisors up to $\sqrt{\boldsymbol{N}} = \boldsymbol{N^{\frac{1}{2}}}$.

Shor's Algorithm

- It is an algorithm for the following problem:
- Given $\boldsymbol{N = p \cdot q}$ where both $\boldsymbol{p, q}$ are *prime* numbers, find $\boldsymbol{p, q}$.
- $\boldsymbol{N = 171 \Rightarrow p = 3, q = 19.}$
- $\boldsymbol{N = 18848997157? \Rightarrow p = 13729, q = 13729933.}$
- How can we find $\boldsymbol{p, q}$?
- *Trivial solution*: try all divisors up to $\sqrt{\boldsymbol{N}} = \boldsymbol{N^{\frac{1}{2}}}$.
- *Exponential* running time!!

Shor's Algorithm

- It is an algorithm for the following problem:
- Given $\boldsymbol{N = p \cdot q}$ where both $\boldsymbol{p, q}$ are *prime* numbers, find $\boldsymbol{p, q}$.
- $\boldsymbol{N = 171 \Rightarrow p = 3, q = 19.}$
- $\boldsymbol{N = 18848997157? \Rightarrow p = 13729, q = 13729933.}$
- How can we find $\boldsymbol{p, q}$?
- *Trivial solution*: try all divisors up to $\sqrt{\boldsymbol{N}} = \boldsymbol{N^{\frac{1}{2}}}$.
- *Exponential* running time!!
- *Do you see why?*

Shor's Algorithm

- It is an algorithm for the following problem:
- Given $\boldsymbol{N = p \cdot q}$ where both $\boldsymbol{p, q}$ are *prime* numbers, find $\boldsymbol{p, q}$.
- $\boldsymbol{N = 171 \Rightarrow p = 3, q = 19.}$
- $\boldsymbol{N = 18848997157? \Rightarrow p = 13729, q = 13729933.}$
- How can we find $\boldsymbol{p, q}$?
- *Trivial solution*: try all divisors up to $\sqrt{\boldsymbol{N}} = \boldsymbol{N^{\frac{1}{2}}}$.
- *Exponential* running time!!
- *The running time is exponential in n=log(N), which is the number of digits needed to specify N*
- Best classical algorithm $\boldsymbol{O((N)^{\frac{1}{3}})}$

Shor’s Algorithm

- Shor’s algorithm solves the Integer Factorization problem on a Quantum Computer in time $\boldsymbol{O((\log N)^3)}$ time.

Shor's Algorithm

- Shor's algorithm solves the Integer Factorization problem on a Quantum Computer in time $\boldsymbol{O((\log N)^3)}$ time.
- Dramatic speedup!!!

Shor's Algorithm

- Shor's algorithm solves the Integer Factorization problem on a Quantum Computer in time $\boldsymbol{O((\log N)^3)}$ time.
- Dramatic speedup!!!
- Based on *Quantum Fourier Transform / Quantum Phase Estimation*

Fourier Transform

Fourier Transform

- In the early 1800s French mathematician *Joseph Fourier* discovered the Fourier *transform*.

Fourier Transform

- In the early 1800s French mathematician *Joseph Fourier* discovered the Fourier *transform*.
- He didn't name it like that though…(ugh)

Fourier Transform

- In the early 1800s French mathematician *Joseph Fourier* discovered the Fourier *transform*.
- He didn't name it like that though…(ugh)
- It allows *frequency components* of signals to be extracted

Fourier Transform

- In the early 1800s French mathematician *Joseph Fourier* discovered the Fourier *transform*.
- He didn't name it like that though…(ugh)
- It allows *frequency components* of signals to be extracted
- Still at the heart of modern-day signal/image processing.

Fourier Transform

- In the early 1800s French mathematician ***Joseph Fourier*** discovered the Fourier ***transform***.
- He didn't name it like that though…(ugh)
- It allows ***frequency components*** of signals to be extracted
- Still at the heart of modern-day signal/image processing.
- For input signal x with N components:

$$y_k = \frac{1}{\sqrt{N}} \sum_j x_j e^{i\frac{2\pi jk}{N}}$$

Fourier Transform

- In the early 1800s French mathematician ***Joseph Fourier*** discovered the Fourier ***transform***.
- He didn't name it like that though…(ugh)
- It allows ***frequency components*** of signals to be extracted
- Still at the heart of modern-day signal/image processing.
- For input signal x with N components:

$$y_k = \frac{1}{\sqrt{N}} \sum_j x_j e^{i\frac{2\pi jk}{N}}$$

- *We go from time domain to frequency domain.*

Quantum Fourier Transform

- Invented by Coppersmith (1994).

Quantum Fourier Transform

- Invented by Coppersmith (1994).
- Polished and utilized by Shor (1994).

Quantum Fourier Transform

- Invented by Coppersmith (1994).
- Polished and utilized by Shor (1994).
- It is applied on the vector of amplitudes of some composite quantum state of $\boldsymbol{n}$ Qbits

Quantum Fourier Transform

- Invented by Coppersmith (1994).
- Polished and utilized by Shor (1994).
- It is applied on the vector of amplitudes of some composite quantum state of $\boldsymbol{n}$ Qbits
- It can transform $\boldsymbol{2^n}$ amplitudes using less than $\boldsymbol{n^2}$ Hadamard and Control gates.

Quantum Fourier Transform

- Invented by Coppersmith (1994).
- Polished and utilized by Shor (1994).
- It is applied on the vector of amplitudes of some composite quantum state of $\boldsymbol{n}$ Qbits
- It can transform $\boldsymbol{2^n}$ amplitudes using less than $\boldsymbol{n^2}$ Hadamard and Control gates.
- Classical Fourier Transform would need $\boldsymbol{O(n2^n)}$ such operations.

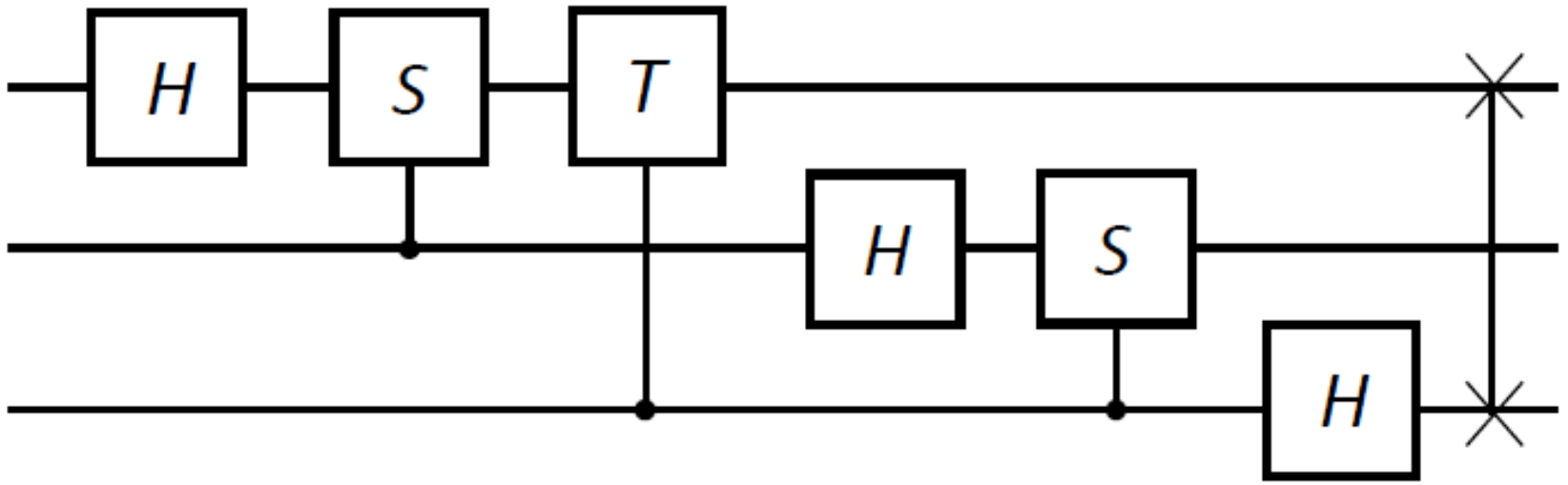
Quantum Fourier Transform on 3 QBits

Quantum Fourier Transform on 3 QBits

- We prepare three Qbits $|q_1 q_2 q_3\rangle$

Quantum Fourier Transform on 3 QBits

- We prepare three Qbits $|q_1 q_2 q_3\rangle$



- *S,T* are phase change matrices (map $|\mathbf{0}\rangle \rightarrow |\mathbf{0}\rangle$ and $|\mathbf{1}\rangle \rightarrow e^{i\phi}|\mathbf{1}\rangle$).
- The last gate performs a SWAP.

Quantum Fourier Transform

- *But what this has to do with Integer Factorization*?

Quantum Fourier Transform

- *But what this has to do with Integer Factorization*?
- For (non-trivial) number theoretic reasons, we need to find the smallest $\boldsymbol{r}$ such that $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x} + \boldsymbol{r})$ for the following $\boldsymbol{f}$:

Quantum Fourier Transform

- *But what this has to do with Integer Factorization*?
- For (non-trivial) number theoretic reasons, we need to find the smallest $\boldsymbol{r}$ such that $\boldsymbol{f(x) = f(x + r)}$ for the following $\boldsymbol{f}$:

$$\boldsymbol{f(x) = a^x \bmod N}$$

Quantum Fourier Transform

- *But what this has to do with Integer Factorization*?
- For (non-trivial) number theoretic reasons, we need to find the smallest $\boldsymbol{r}$ such that $\boldsymbol{f(x) = f(x + r)}$ for the following $\boldsymbol{f}$:

$$\boldsymbol{f(x) = a^x \bmod N}$$

- a is *some* initial guess of the factors of $\boldsymbol{N}$.

Quantum Fourier Transform

- *But what this has to do with Integer Factorization*?
- For (non-trivial) number theoretic reasons, we need to find the smallest $\boldsymbol{r}$ such that $\boldsymbol{f(x) = f(x + r)}$ for the following $\boldsymbol{f}$:

$$\boldsymbol{f(x) = a^x \bmod N}$$

- a is *some* initial guess of the factors of $\boldsymbol{N}$.
- If we know the period $\boldsymbol{r}$ above, then $\boldsymbol{a^{r/2} + 1}$ or $\boldsymbol{a^{r/2} - 1}$ are (much) better guesses of a factor of $\boldsymbol{N}$!

Quantum Fourier Transform

- *But what this has to do with Integer Factorization*?
- For (non-trivial) number theoretic reasons, we need to find the smallest $\boldsymbol{r}$ such that $\boldsymbol{f(x) = f(x + r)}$ for the following $\boldsymbol{f}$:

$$\boldsymbol{f(x) = a^x \bmod N}$$

- a is *some* initial guess of the factors of $\boldsymbol{N}$.
- If we know the period $\boldsymbol{r}$ above, then $\boldsymbol{a^{r/2} + 1}$ or $\boldsymbol{a^{r/2} - 1}$ are (much) better guesses of a factor of $\boldsymbol{N}$!
- Indeed, with probability $> \frac{1}{3}$ for ***any guess*** a as a factor of $\boldsymbol{N}$, one of $\boldsymbol{a^{r/2} + 1}$ or $\boldsymbol{a^{r/2} - 1}$ is a prime factor of $\boldsymbol{N}$!

Quantum Fourier Transform

- *But what this has to do with Integer Factorization*?
- For (non-trivial) number theoretic reasons, we need to find the smallest $\boldsymbol{r}$ such that $\boldsymbol{f(x) = f(x + r)}$ for the following $\boldsymbol{f}$:

$$\boldsymbol{f(x) = a^x \bmod N}$$

- a is ***some*** initial guess of the factors of $\boldsymbol{N}$.
- If we know the period $\boldsymbol{r}$ above, then $\boldsymbol{a^{r/2} + 1}$ or $\boldsymbol{a^{r/2} - 1}$ are (much) better guesses of a factor of $\boldsymbol{N}$!
- Indeed, with probability $> \frac{1}{3}$ for ***any guess*** a as a factor of $\boldsymbol{N}$, one of $\boldsymbol{a^{r/2} + 1}$ or $\boldsymbol{a^{r/2} - 1}$ is a prime factor of $\boldsymbol{N}$!
- We can find $\boldsymbol{r}$ ***efficiently*** by *Quantum Fourier Transform*!

But why we care about factoring?

- Finding the prime factors of $\boldsymbol{N = p \cdot q}$ seems cute…

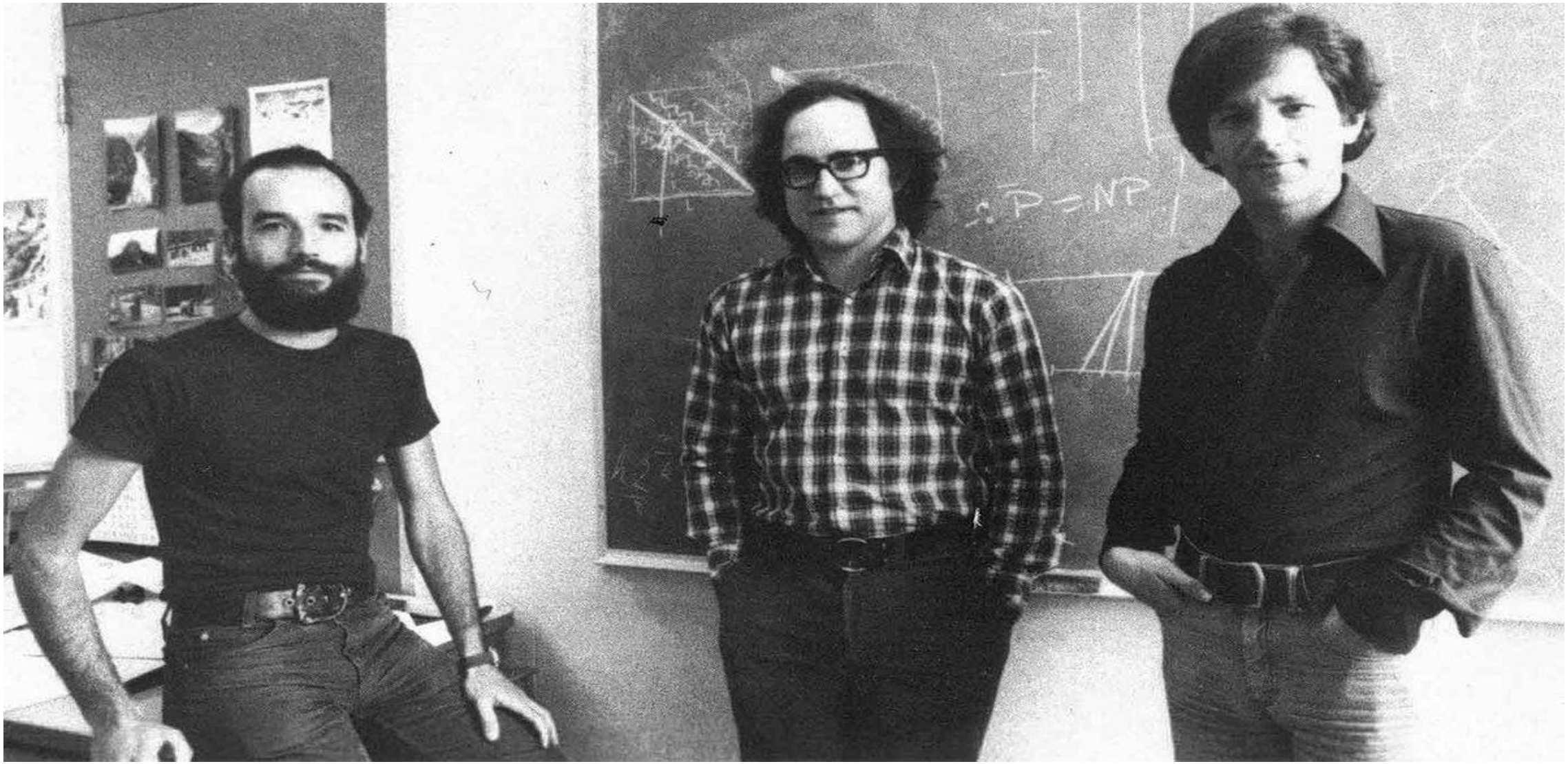
But why we care about factoring?

- Finding the prime factors of $\boldsymbol{N = p \cdot q}$ seems cute…
- But, *does it actually matter in practice*?

But why we care about factoring?

- Finding the prime factors of $\boldsymbol{N = p \cdot q}$ seems cute…
- But, *does it actually matter in practice*?
- Or it is just one more toy problem only mathematicians care about?

But why we care about factoring?



The RSA cryptosystem

The RSA cryptosystem

- Discovered in 1977 by *Rivest, Shamir, Adleman*

The RSA cryptosystem

- Discovered in 1977 by *Rivest, Shamir, Adleman*
- It is based on the difficulty of *factoring large numbers*.

The RSA cryptosystem

- Discovered in 1977 by *Rivest, Shamir, Adleman*
- It is based on the difficulty of *factoring large numbers*.
- It is a *public* key krypto-system where the public key is $\boldsymbol{N = p \cdot q}$.

The RSA cryptosystem

- Discovered in 1977 by *Rivest, Shamir, Adleman*
- It is based on the difficulty of *factoring large numbers*.
- It is a *public* key krypto-system where the public key is $\boldsymbol{N = p \cdot q}$.
- Everyone has access to $\boldsymbol{N}$.

The RSA cryptosystem

- Discovered in 1977 by *Rivest, Shamir, Adleman*
- It is based on the difficulty of *factoring large numbers*.
- It is a *public* key krypto-system where the public key is $\boldsymbol{N = p \cdot q}$.
- Everyone has access to $\boldsymbol{N}$.
- Only the person who initiates the protocol knows $\boldsymbol{p}, \boldsymbol{q}$.

The RSA cryptosystem

- Discovered in 1977 by *Rivest, Shamir, Adleman*
- It is based on the difficulty of *factoring large numbers*.
- It is a *public* key krypto-system where the public key is $\boldsymbol{N = p \cdot q}$.
- Everyone has access to $\boldsymbol{N}$.
- Only the person who initiates the protocol knows $\boldsymbol{p}, \boldsymbol{q}$.
- Everyone can encrypt messages with $\boldsymbol{N}$.

The RSA cryptosystem

- Discovered in 1977 by *Rivest, Shamir, Adleman*
- It is based on the difficulty of *factoring large numbers*.
- It is a *public* key krypto-system where the public key is $\boldsymbol{N = p \cdot q}$.
- Everyone has access to $\boldsymbol{N}$.
- Only the person who initiates the protocol knows $\boldsymbol{p}, \boldsymbol{q}$.
- Everyone can encrypt messages with $\boldsymbol{N}$.
- Nobody can decrypt the encrypted messages **unless** they know $\boldsymbol{p}, \boldsymbol{q}$!

The RSA cryptosystem

- Apparently, the British Intelligence (in particular *Clifford Cocks*) was aware of such a system years before RSA was published.

The RSA cryptosystem

- Apparently, the British Intelligence (in particular *Clifford Cocks*) was aware of such a system years before RSA was published.
- But it was classified **TOP SECRET**.

The RSA cryptosystem

- Apparently, the British Intelligence (in particular *Clifford Cocks*) was aware of such a system years before RSA was published.
- But it was classified **TOP SECRET**.
- It was declassified in 1997.

The RSA cryptosystem

- Apparently, the British Intelligence (in particular *Clifford Cocks*) was aware of such a system years before RSA was published.
- But it was classified **TOP SECRET**.
- It was declassified in 1997.
- Among with other hilarious stuff

The RSA cryptosystem

- Apparently, the British Intelligence (in particular *Clifford Cocks*) was aware of such a system years before RSA was published.
- But it was classified **TOP SECRET**.
- It was declassified in 1997.
- Among with other hilarious stuff
- And so R-S-A took all the glory.

The RSA cryptosystem

- But wait…

The RSA cryptosystem

- But wait…
- *Does it mean now that the RSA Cryptosystem is in danger?*

The RSA cryptosystem

- But wait…
- *Does it mean now that the RSA Cryptosystem is in danger?*
- Not really…

The RSA cryptosystem

- But wait…
- *Does it mean now that the RSA Cryptosystem is in danger?*
- Not really…
- Current Quantum machines can factor 21=7 times 3

The RSA cryptosystem

- But wait…
- *Does it mean now that the RSA Cryptosystem is in danger?*
- Not really…
- Current Quantum machines can factor 21=7 times 3 (with high probability).

The RSA cryptosystem

- But wait…
- *Does it mean now that the RSA Cryptosystem is in danger?*
- Not really…
- Current Quantum machines can factor 21=7 times 3 (with high probability).
- So we are safe

The RSA cryptosystem

- But wait…
- *Does it mean now that the RSA Cryptosystem is in danger?*
- Not really…
- Current Quantum machines can factor 21=7 times 3 (with high probability).
- So we are safe (for now).

The RSA cryptosystem

- But wait…
- *Does it mean now that the RSA Cryptosystem is in danger?*
- Not really…
- Current Quantum machines can factor 21=7 times 3 (with high probability).
- So we are safe (for now).
- But still, we should start thinking about Quantum-resistance Crypto protocols.

Quantum Key Distribution

Quantum Key Distribution

- It can be used to securely generate a **key** shared between Alice and Bob.

Quantum Key Distribution

- It can be used to securely generate a **key** shared between Alice and Bob.
- *Without having to personally meet and exchange information.*

Quantum Key Distribution

- It can be used to securely generate a **key** shared between Alice and Bob.
- *Without having to personally meet and exchange information.*
- In order to do that need

Quantum Key Distribution

- It can be used to securely generate a **key** shared between Alice and Bob.
- *Without having to personally meet and exchange information.*
- In order to do that need
- (i) a classical channel of communication (to send bits), and

Quantum Key Distribution

- It can be used to securely generate a **key** shared between Alice and Bob.
- *Without having to personally meet and exchange information.*
- In order to do that need
- (i) a classical channel of communication (to send bits), and
- (ii) a quantum channel of communication (to send Qbits).

Quantum Key Distribution

- It can be used to securely generate a **key** shared between Alice and Bob.
- *Without having to personally meet and exchange information.*
- In order to do that need
- (i) a classical channel of communication (to send bits), and
- (ii) a quantum channel of communication (to send Qbits).
- Alice and Bob have access and can measure Qbits in states $|\mathbf{0}\rangle, |\mathbf{1}\rangle, |+\rangle, |-\rangle$

BB84 Quantum Protocol for QKD

- Alice has some binary string eg **0100111**.

BB84 Quantum Protocol for QKD

- Alice has some binary string eg **0100111**.
- She encodes each bit {**0,1**} as either, $\{|\mathbf{0}\rangle,|\mathbf{1}\rangle\}$ or $\{|+\rangle,|-\rangle\}$ completely at random.

BB84 Quantum Protocol for QKD

- Alice has some binary string eg **0100111**.
- She encodes each bit {**0,1**} as either, $\{|\mathbf{0}\rangle,|\mathbf{1}\rangle\}$ or $\{|+\rangle,|-\rangle\}$ completely at random. She send then to Bob the resulting bit.

BB84 Quantum Protocol for QKD

- Alice has some binary string eg **0100111**.
- She encodes each bit $\{\mathbf{0},\mathbf{1}\}$ as either, $\{|\mathbf{0}\rangle,|\mathbf{1}\rangle\}$ or $\{|+\rangle,|-\rangle\}$ completely at random. She send then to Bob the resulting bit.
- Bob receives through the Quantum channel Alice's Qbit and makes a measurement either in the $\{|\mathbf{0}\rangle,|\mathbf{1}\rangle\}$ or $\{|+\rangle,|-\rangle\}$ basis.

BB84 Quantum Protocol for QKD

- Alice has some binary string eg **0100111**.
- She encodes each bit $\{\mathbf{0},\mathbf{1}\}$ as either, $\{|\mathbf{0}\rangle,|\mathbf{1}\rangle\}$ or $\{|+\rangle,|-\rangle\}$ completely at random. She send then to Bob the resulting bit.
- Bob receives through the Quantum channel Alice's Qbit and makes a measurement either in the $\{|\mathbf{0}\rangle,|\mathbf{1}\rangle\}$ or $\{|+\rangle,|-\rangle\}$ basis.
- Bob announces to the public which basis he made his measurement.

BB84 Quantum Protocol for QKD

- Alice has some binary string eg **0100111**.
- She encodes each bit $\{\mathbf{0},\mathbf{1}\}$ as either, $\{|\mathbf{0}\rangle,|\mathbf{1}\rangle\}$ or $\{|+\rangle,|-\rangle\}$ completely at random. She send then to Bob the resulting bit.
- Bob receives through the Quantum channel Alice's Qbit and makes a measurement either in the $\{|\mathbf{0}\rangle,|\mathbf{1}\rangle\}$ or $\{|+\rangle,|-\rangle\}$ basis.
- Bob announces to the public which basis he made his measurement.
- Alice replies through the classical public channel whether Bob was correct or not.

BB84 Quantum Protocol for QKD

- Alice has some binary string eg **0100111**.
- She encodes each bit $\{\mathbf{0},\mathbf{1}\}$ as either, $\{|\mathbf{0}\rangle,|\mathbf{1}\rangle\}$ or $\{|+\rangle,|-\rangle\}$ completely at random. She send then to Bob the resulting bit.
- Bob receives through the Quantum channel Alice's Qbit and makes a measurement either in the $\{|\mathbf{0}\rangle,|\mathbf{1}\rangle\}$ or $\{|+\rangle,|-\rangle\}$ basis.
- Bob announces to the public which basis he made his measurement.
- Alice replies through the classical public channel whether Bob was correct or not.
- If Bob was correct, they keep the Qbit and append it the current key.

BB84 Quantum Protocol for QKD

- Alice has some binary string eg **0100111**.
- She encodes each bit $\{\mathbf{0},\mathbf{1}\}$ as either, $\{|\mathbf{0}\rangle,|\mathbf{1}\rangle\}$ or $\{|+\rangle,|-\rangle\}$ completely at random. She send then to Bob the resulting bit.
- Bob receives through the Quantum channel Alice's Qbit and makes a measurement either in the $\{|\mathbf{0}\rangle,|\mathbf{1}\rangle\}$ or $\{|+\rangle,|-\rangle\}$ basis.
- Bob announces to the public which basis he made his measurement.
- Alice replies through the classical public channel whether Bob was correct or not.
- If Bob was correct, they keep the Qbit and append it the current key.
- Otherwise, they discard the bit.

BB84 Quantum Protocol for QKD

- On average, Bob will be right half of the time and wrong half of the time in his guess.

BB84 Quantum Protocol for QKD

- On average, Bob will be right half of the time and wrong half of the time in his guess.
- *They test half of their obtained bits for consistency, at random*.

BB84 Quantum Protocol for QKD

- On average, Bob will be right half of the time and wrong half of the time in his guess.
- *They test half of their obtained bits for consistency, at random*.
- So, on average, Alice and Bob will discard 3/4 of their bits.

BB84 Quantum Protocol for QKD

- On average, Bob will be right half of the time and wrong half of the time in his guess.
- *They test half of their obtained bits for consistency, at random*.
- So, on average, Alice and Bob will discard 3/4 of their bits.
- At the end, they will end up with a shared key to encrypt their communication.

BB84 Quantum Protocol for QKD

- On average, Bob will be right half of the time and wrong half of the time in his guess.
- *They test half of their obtained bits for consistency, at random*.
- So, on average, Alice and Bob will discard 3/4 of their bits.
- At the end, they will end up with a shared key to encrypt their communication.
- And also, anyone trying to listen to their communication channel will lead to detectable changes!

BB84 Quantum Protocol for QKD

- On average, Bob will be right half of the time and wrong half of the time in his guess.
- *They test half of their obtained bits for consistency, at random*.
- So, on average, Alice and Bob will discard 3/4 of their bits.
- At the end, they will end up with a shared key to encrypt their communication.
- And also, anyone trying to listen to their communication channel will lead to detectable changes!
- So, this results to a secure key distribution!

BB84 Quantum Protocol for QKD: Example

Alice's Bit	Alice basis	QBit	Bob's Basis	Bob's Bit

BB84 Quantum Protocol for QKD: Example

Alice's Bit	Alice basis	QBit	Bob's Basis	Bob's Bit
0				

BB84 Quantum Protocol for QKD: Example

Alice's Bit	Alice basis	QBit	Bob's Basis	Bob's Bit
0	$\|0\rangle, \|1\rangle$			

BB84 Quantum Protocol for QKD: Example

Alice's Bit	Alice basis	QBit	Bob's Basis	Bob's Bit
0	$\|0\rangle, \|1\rangle$	$\|0\rangle$		

BB84 Quantum Protocol for QKD: Example

Alice's Bit	Alice basis	QBit	Bob's Basis	Bob's Bit
0	$\|0\rangle, \|1\rangle$	$\|0\rangle$	$\|0\rangle, \|1\rangle$	

BB84 Quantum Protocol for QKD: Example

Alice's Bit	Alice basis	QBit	Bob's Basis	Bob's Bit
0	$\|0\rangle, \|1\rangle$	$\|0\rangle$	$\|0\rangle, \|1\rangle$	0

BB84 Quantum Protocol for QKD: Example

Alice's Bit	Alice basis	QBit	Bob's Basis	Bob's Bit
0	$\|0\rangle, \|1\rangle$	$\|0\rangle$	$\|0\rangle, \|1\rangle$	0
1	$\|+\rangle, \|-\rangle$	$\|-\rangle$	$\|+\rangle, \|-\rangle$	1

BB84 Quantum Protocol for QKD: Example

Alice's Bit	Alice basis	QBit	Bob's Basis	Bob's Bit
0	$\|0\rangle, \|1\rangle$	$\|0\rangle$	$\|0\rangle, \|1\rangle$	0
1	$\|+\rangle, \|-\rangle$	$\|-\rangle$	$\|+\rangle, \|-\rangle$	1
1	$\|0\rangle, \|1\rangle$	$\|1\rangle$	$\|+\rangle, \|-\rangle$	$\mathbf{0}\ \boldsymbol{or}\ \mathbf{1}$

BB84 Quantum Protocol for QKD: Example

Alice's Bit	Alice basis	QBit	Bob's Basis	Bob's Bit
0	$\|0\rangle, \|1\rangle$	$\|0\rangle$	$\|0\rangle, \|1\rangle$	0
1	$\|+\rangle, \|-\rangle$	$\|-\rangle$	$\|+\rangle, \|-\rangle$	1
1	$\|0\rangle, \|1\rangle$	$\|1\rangle$	$\|+\rangle, \|-\rangle$	**0 *or* 1**
0	$\|0\rangle, \|1\rangle$	$\|0\rangle$	$\|0\rangle, \|1\rangle$	0
1	$\|0\rangle, \|1\rangle$	$\|1\rangle$	$\|0\rangle, \|1\rangle$	1
0	$\|+\rangle, \|-\rangle$	$\|+\rangle$	$\|+\rangle, \|-\rangle$	0
1	$\|0\rangle, \|1\rangle$	$\|1\rangle$	$\|+\rangle, \|-\rangle$	**0 *or* 1**
1	$\|+\rangle, \|-\rangle$	$\|-\rangle$	$\|0\rangle, \|1\rangle$	**0 *or* 1**

BB84 Quantum Protocol for QKD: Example

- *Why this is perfectly secure?*

BB84 Quantum Protocol for QKD: Example

- *Why this is perfectly secure?*
- Let's assume that Charlie hears on the quantum channel Alice and Bob share.

BB84 Quantum Protocol for QKD: Example

- *Why this is perfectly secure?*
- Let's assume that Charlie hears on the quantum channel Alice and Bob share.
- What Charlie can do is *intercept*, *measure* and *retransmit* whatever he measured.

BB84 Quantum Protocol for QKD: Example

- *Why this is perfectly secure?*
- Let's assume that Charlie hears on the quantum channel Alice and Bob share.
- What Charlie can do is *intercept*, *measure* and *retransmit* whatever he measured.
- You can see why it fails…

BB84 Quantum Protocol for QKD: Example

- *Why this is perfectly secure?*
- Let's assume that Charlie hears on the quantum channel Alice and Bob share.
- What Charlie can do is *intercept*, *measure* and *retransmit* whatever he measured.
- You can see why it fails…
- He doesn't know which basis Alice has transmitted her Qbit!

BB84 Quantum Protocol for QKD: Example

- *Why this is perfectly secure?*
- Let's assume that Charlie hears on the quantum channel Alice and Bob share.
- What Charlie can do is *intercept*, *measure* and *retransmit* whatever he measured.
- You can see why it fails…
- He doesn't know which basis Alice has transmitted her Qbit!
- With 50% he makes the wrong measurement, and he sends to Bob the wrong Qbit.

BB84 Quantum Protocol for QKD: Example

- *Why this is perfectly secure?*
- Let's assume that Charlie hears on the quantum channel Alice and Bob share.
- What Charlie can do is *intercept*, *measure* and *retransmit* whatever he measured.
- You can see why it fails…
- He doesn't know which basis Alice has transmitted her Qbit!
- With 50% he makes the wrong measurement, and he sends to Bob the wrong Qbit.
- Let's see how this works out…

BB84 Quantum Protocol for QKD: Example

Alice's Bit	Alice basis	QBit	Charlie's Basis	Charlie's Bit	QBit	Bob's Basis	Bob's Bit

BB84 Quantum Protocol for QKD: Example

Alice's Bit	Alice basis	QBit	Charlie's Basis	Charlie's Bit	QBit	Bob's Basis	Bob's Bit
0	$\|0\rangle, \|1\rangle$	$\|0\rangle$					

BB84 Quantum Protocol for QKD: Example

Alice's Bit	Alice basis	QBit	Charlie's Basis	Charlie's Bit	QBit	Bob's Basis	Bob's Bit
0	$\|0\rangle, \|1\rangle$	$\|0\rangle$	$\|0\rangle, \|1\rangle$				

BB84 Quantum Protocol for QKD: Example

Alice's Bit	Alice basis	QBit	Charlie's Basis	Charlie's Bit	QBit	Bob's Basis	Bob's Bit
0	$\|0\rangle, \|1\rangle$	$\|0\rangle$	$\|0\rangle, \|1\rangle$	0			

BB84 Quantum Protocol for QKD: Example

Alice's Bit	Alice basis	QBit	Charlie's Basis	Charlie's Bit	QBit	Bob's Basis	Bob's Bit
0	$\|0\rangle, \|1\rangle$	$\|0\rangle$	$\|0\rangle, \|1\rangle$	0	$\|0\rangle$		

BB84 Quantum Protocol for QKD: Example

Alice's Bit	Alice basis	QBit	Charlie's Basis	Charlie's Bit	QBit	Bob's Basis	Bob's Bit
0	$\|0\rangle, \|1\rangle$	$\|0\rangle$	$\|0\rangle, \|1\rangle$	0	$\|0\rangle$	$\|0\rangle, \|1\rangle$	

BB84 Quantum Protocol for QKD: Example

Alice's Bit	Alice basis	QBit	Charlie's Basis	Charlie's Bit	QBit	Bob's Basis	Bob's Bit
0	$\|0\rangle, \|1\rangle$	$\|0\rangle$	$\|0\rangle, \|1\rangle$	0	$\|0\rangle$	$\|0\rangle, \|1\rangle$	0

BB84 Quantum Protocol for QKD: Example

Alice's Bit	Alice basis	QBit	Charlie's Basis	Charlie's Bit	QBit	Bob's Basis	Bob's Bit
0	$\|0\rangle, \|1\rangle$	$\|0\rangle$	$\|0\rangle, \|1\rangle$	0	$\|0\rangle$	$\|0\rangle, \|1\rangle$	0
1	$\|+\rangle, \|-\rangle$	$\|-\rangle$					

BB84 Quantum Protocol for QKD: Example

Alice's Bit	Alice basis	QBit	Charlie's Basis	Charlie's Bit	QBit	Bob's Basis	Bob's Bit
0	$\|0\rangle, \|1\rangle$	$\|0\rangle$	$\|0\rangle, \|1\rangle$	0	$\|0\rangle$	$\|0\rangle, \|1\rangle$	0
1	$\|+\rangle, \|-\rangle$	$\|-\rangle$	$\|0\rangle, \|1\rangle$	0 or 1			

BB84 Quantum Protocol for QKD: Example

Alice's Bit	Alice basis	QBit	Charlie's Basis	Charlie's Bit	QBit	Bob's Basis	Bob's Bit
0	$\|0\rangle, \|1\rangle$	$\|0\rangle$	$\|0\rangle, \|1\rangle$	0	$\|0\rangle$	$\|0\rangle, \|1\rangle$	0
1	$\|+\rangle, \|-\rangle$	$\|-\rangle$	$\|0\rangle, \|1\rangle$	0 or 1	$\{\|0\rangle, \|1\rangle\}$		

BB84 Quantum Protocol for QKD: Example

Alice's Bit	Alice basis	QBit	Charlie's Basis	Charlie's Bit	QBit	Bob's Basis	Bob's Bit
0	$\|0\rangle, \|1\rangle$	$\|0\rangle$	$\|0\rangle, \|1\rangle$	0	$\|0\rangle$	$\|0\rangle, \|1\rangle$	0
1	$\|+\rangle, \|-\rangle$	$\|-\rangle$	$\|0\rangle, \|1\rangle$	0 or 1	$\{\|0\rangle, \|1\rangle\}$	$\|+\rangle, \|-\rangle$	

BB84 Quantum Protocol for QKD: Example

Alice's Bit	Alice basis	QBit	Charlie's Basis	Charlie's Bit	QBit	Bob's Basis	Bob's Bit
0	$\|0\rangle, \|1\rangle$	$\|0\rangle$	$\|0\rangle, \|1\rangle$	0	$\|0\rangle$	$\|0\rangle, \|1\rangle$	0
1	$\|+\rangle, \|-\rangle$	$\|-\rangle$	$\|0\rangle, \|1\rangle$	0 or 1	$\{\|0\rangle, \|1\rangle\}$	$\|+\rangle, \|-\rangle$	$0\ or\ 1$

BB84 Quantum Protocol for QKD: Example

Alice's Bit	Alice basis	QBit	Charlie's Basis	Charlie's Bit	QBit	Bob's Basis	Bob's Bit
0	$\|0\rangle, \|1\rangle$	$\|0\rangle$	$\|0\rangle, \|1\rangle$	0	$\|0\rangle$	$\|0\rangle, \|1\rangle$	0
1	$\|+\rangle, \|-\rangle$	$\|-\rangle$	$\|0\rangle, \|1\rangle$	0 or 1	$\{\|0\rangle, \|1\rangle\}$	$\|+\rangle, \|-\rangle$	$0\ or\ 1$
1	$\|0\rangle, \|1\rangle$	$\|1\rangle$	$\|+\rangle, \|-\rangle$	0 or 1	$\{\|+\rangle, \|-\rangle\}$	$\|+\rangle, \|-\rangle$	$0\ or\ 1$
0	$\|0\rangle, \|1\rangle$	$\|0\rangle$	$\|+\rangle, \|-\rangle$	0 or 1	$\{\|+\rangle, \|-\rangle\}$	$\|0\rangle, \|1\rangle$	$0\ or\ 1$
1	$\|0\rangle, \|1\rangle$	$\|1\rangle$	$\|0\rangle, \|1\rangle$	1	$\|1\rangle$	$\|0\rangle, \|1\rangle$	1
0	$\|+\rangle, \|-\rangle$	$\|+\rangle$	$\|+\rangle, \|-\rangle$	0	$\|+\rangle$	$\|+\rangle, \|-\rangle$	1
1	$\|0\rangle, \|1\rangle$	$\|1\rangle$	$\|0\rangle, \|1\rangle$	1	$\|1\rangle$	$\|+\rangle, \|-\rangle$	$0\ or\ 1$
1	$\|+\rangle, \|-\rangle$	$\|-\rangle$	$\|0\rangle, \|1\rangle$	0 or 1	$\{\|0\rangle, \|1\rangle\}$	$\|0\rangle, \|1\rangle$	$0\ or\ 1$

BB84 Quantum Protocol for QKD: Example

Alice's Bit	Alice basis	QBit	Charlie's Basis	Charlie's Bit	QBit	Bob's Basis	Bob's Bit
0	$\|0\rangle, \|1\rangle$	$\|0\rangle$	$\|0\rangle, \|1\rangle$	0	$\|0\rangle$	$\|0\rangle, \|1\rangle$	0
1	$\|+\rangle, \|-\rangle$	$\|-\rangle$	$\|0\rangle, \|1\rangle$	0 or 1	$\{\|0\rangle, \|1\rangle\}$	$\|+\rangle, \|-\rangle$	$0\ or\ 1$
1	$\|0\rangle, \|1\rangle$	$\|1\rangle$	$\|+\rangle, \|-\rangle$	0 or 1	$\{\|+\rangle, \|-\rangle\}$	$\|+\rangle, \|-\rangle$	$0\ or\ 1$
0	$\|0\rangle, \|1\rangle$	$\|0\rangle$	$\|+\rangle, \|-\rangle$	0 or 1	$\{\|+\rangle, \|-\rangle\}$	$\|0\rangle, \|1\rangle$	$0\ or\ 1$
1	$\|0\rangle, \|1\rangle$	$\|1\rangle$	$\|0\rangle, \|1\rangle$	1	$\|1\rangle$	$\|0\rangle, \|1\rangle$	1
0	$\|+\rangle, \|-\rangle$	$\|+\rangle$	$\|+\rangle, \|-\rangle$	0	$\|+\rangle$	$\|+\rangle, \|-\rangle$	0
1	$\|0\rangle, \|1\rangle$	$\|1\rangle$	$\|0\rangle, \|1\rangle$	1	$\|1\rangle$	$\|+\rangle, \|-\rangle$	$0\ or\ 1$
1	$\|+\rangle, \|-\rangle$	$\|-\rangle$	$\|0\rangle, \|1\rangle$	0 or 1	$\{\|0\rangle, \|1\rangle\}$	$\|0\rangle, \|1\rangle$	$0\ or\ 1$

And that was it…

- Thank you all!
- Hope you enjoyed and you learned something interesting ☺
- Best of luck on your exam(s) and everything else ☺