Lecture

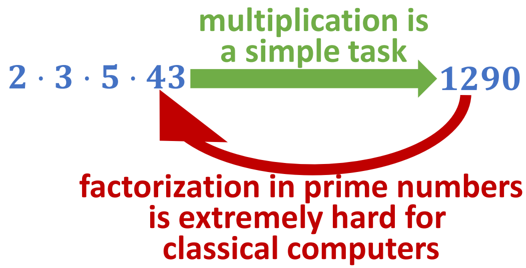
Quantum Computing

(CS5070)

Quantum Cryptography: Shor, Quantum Key Distribution

Professor Dr. rer. nat. habil. Sven Groppe
https://www.ifis.uni-luebeck.de/~groppe

Shor's Algorithm[1]



- factoring integers in polynomial time
  - Depth of quantum circuit[2] to factor integer N: $O((log\ N)^2 (log\ log\ N)(log\ log\ log\ N))$
  - superpolynomial speedup, i.e., almost exponentially faster than the most efficient known classical factoring algorithm (general number field sieve): $O(e^{1.9(log\ N)^{\frac{1}{3}}(log\ log\ N)^{\frac{2}{3}}})$
- Important for cryptography $\longrightarrow$ Post-Quantum Cryptography
- Most quantum algorithms with superpolynomial speedup like Shor's algorithm are based on quantum Fourier transforms (quantum analogue of inverse discrete Fourier transform)

[1] [Shor, 1994] [2] [Beckman et al., 1996]

Shor's Algorithm - Idea

i:	1	2	3	4	5	6	7	8	9	10	11	12	13	...
2^i:	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192	...
2^i **mod** 15:	2	4	8	**1**	2	4	8	**1**	2	4	8	**1**	2	...
2^i **mod** 21:	2	4	8	16	11	**1**	2	4	8	16	11	**1**	2	...

Shor's Algorithm - Idea

i:	1	2	3	4	5	6	7	8	9	10	11	12	13	...
2^i:	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192	...
2^i **mod** 15:	2	4	8	**1**	2	4	8	**1**	2	4	8	**1**	2	...
2^i **mod** 21:	2	4	8	16	11	**1**	2	4	8	16	11	**1**	2	...

- **Observations:**
  - The given mod-sequences are periodic!
  - Each period ends with 1!

Shor's Algorithm - Idea

i:	1	2	3	4	5	6	7	8	9	10	11	12	13	...
2^i:	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192	...
2^i **mod** 15:	2	4	8	**1**	2	4	8	**1**	2	4	8	**1**	2	...
2^i **mod** 21:	2	4	8	16	11	**1**	2	4	8	16	11	**1**	2	...

- **Observations:**
  - The given mod-sequences are periodic!
  - Each period ends with 1!

- In general:

$$a^1, a^2, \cdots, a^r = 1, a^1, a^2, \ldots \pmod N$$

order of $a =$ the *smallest* positive r such that $a^r = 1 \pmod N$

Shor's Algorithm - Number Theory

- Euler's Theorem: $\forall a \in \mathbb{Z}_N^* \text{ with } gcd(a,N)=1 : a^{\varphi(N)} = 1 \bmod N$, where Euler's phi function: $\varphi(N) = |\{a \in \mathbb{N} | 1 \leq a \leq N \wedge gcd(a,N)=1\}|$ and greatest common divisor $gcd(a,b) = \begin{cases} b & \text{if } a \bmod b = 0 \\ gcd(b, a \bmod b) & \text{otherwise} \end{cases}$
- Suppose $N = p^k \cdot m$ with p prime and $k, m \in \mathbb{N}_{\geq 1} : gcd(m,p)=1$
  $\Rightarrow \varphi(N) = \varphi(p^k) \cdot \varphi(m) = (p-1) \cdot p^{k-1} \cdot \varphi(m)$ (rules for Euler's Phi)

Shor's Algorithm - Number Theory

- Euler's Theorem: $\forall a \in \mathbb{Z}_N^* \text{ with } gcd(a, N) = 1 : a^{\varphi(N)} = 1 \bmod N$,
  where Euler's phi function: $\varphi(N) = |\{a \in \mathbb{N} | 1 \leq a \leq N \wedge gcd(a, N) = 1\}|$
  and greatest common divisor $gcd(a, b) = \begin{cases} b & \text{if } a \bmod b = 0 \\ gcd(b, a \bmod b) & \text{otherwise} \end{cases}$
- Suppose $N = p^k \cdot m$ with p prime and $k, m \in \mathbb{N}_{\geq 1} : gcd(m, p) = 1$
  $\Rightarrow \varphi(N) = \varphi(p^k) \cdot \varphi(m) = (p-1) \cdot p^{k-1} \cdot \varphi(m)$ (rules for Euler's Phi)
- Fact: $\boxed{r \text{ must divide } \varphi(N) = (p-1) \cdot p^{k-1} \cdot \varphi(m)}$
  Proof:
  $\varphi(N) = s \cdot r + t$, where $s, t \in \mathbb{N} \text{ with } 0 \leq t < r$
  $1 \overset{Euler}{=} a^{\varphi(N)} = a^{s \cdot r + t} = a^{s \cdot r} \cdot a^t = (a^r)^s \cdot a^t = 1^s \cdot a^t \pmod N$
  $\Rightarrow t = 0$ (since r is the *smallest*) $\Rightarrow \varphi(N) = (p-1) \cdot p^{k-1} \cdot \varphi(m) = s \cdot r$ $\square$

Shor's Algorithm - Number Theory

- Euler's Theorem: $\forall a \in \mathbb{Z}_N^*$ with $gcd(a, N) = 1 : a^{\varphi(N)} = 1 \bmod N$,
  where Euler's phi function: $\varphi(N) = |\{a \in \mathbb{N} | 1 \leq a \leq N \wedge gcd(a, N) = 1\}|$
  and greatest common divisor $gcd(a,b) = \begin{cases} b & \text{if } a \bmod b = 0 \\ gcd(b, a \bmod b) & \text{otherwise} \end{cases}$
- Suppose $N = p^k \cdot m$ with p prime and $k, m \in \mathbb{N}_{\geq 1} : gcd(m, p) = 1$
  $\Rightarrow \varphi(N) = \varphi(p^k) \cdot \varphi(m) = (p-1) \cdot p^{k-1} \cdot \varphi(m)$ (rules for Euler's Phi)
- Fact: $\boxed{r \text{ must divide } \varphi(N) = (p-1) \cdot p^{k-1} \cdot \varphi(m)}$
  Proof:
  $\varphi(N) = s \cdot r + t$, where $s, t \in \mathbb{N}$ with $0 \leq t < r$
  $1 \overset{Euler}{=} a^{\varphi(N)} = a^{s \cdot r + t} = a^{s \cdot r} \cdot a^t = (a^r)^s \cdot a^t = 1^s \cdot a^t \pmod N$
  $\Rightarrow t = 0$ (since r is the *smallest*) $\Rightarrow \varphi(N) = (p-1) \cdot p^{k-1} \cdot \varphi(m) = s \cdot r$ $\square$

Conclusions: Learn $r \Rightarrow$ We learn a factor of $(p-1) \cdot p^{k-1} \cdot \varphi(m)$
Repeat with a different $a \Rightarrow$ Learn another factor of $(p-1) \cdot p^{k-1} \cdot \varphi(m)$ (with high prob.)
Eventually we learn full $(p-1) \cdot p^{k-1} \cdot \varphi(m) \Rightarrow$ Can find p

Shor's Algorithm - Number Theory

- Suppose: r is even
  - Then: $0 = \underbrace{a^r}_{=1} - 1 = (a^{\frac{r}{2}})^2 - 1 = \underbrace{(a^{\frac{r}{2}} + 1) \cdot (a^{\frac{r}{2}} - 1)}_{\text{remember: } x^2-1=(x-1)\cdot(x+1)} \pmod N$

  $\Rightarrow N$ divides $(a^{\frac{r}{2}} + 1) \cdot (a^{\frac{r}{2}} - 1)$

Shor's Algorithm - Number Theory

- Suppose: r is even
  - Then: $0 = \underbrace{a^r}_{=1} - 1 = (a^{\frac{r}{2}})^2 - 1 = \underbrace{(a^{\frac{r}{2}} + 1) \cdot (a^{\frac{r}{2}} - 1)}_{\text{remember: } x^2-1=(x-1)\cdot(x+1)} \pmod N$

    $\Rightarrow N$ divides $(a^{\frac{r}{2}} + 1) \cdot (a^{\frac{r}{2}} - 1)$
- Additionally suppose: $a^{\frac{r}{2}} \neq \pm 1 \pmod N$
  - Then: N does *neither* divide $(a^{\frac{r}{2}} + 1)$ *nor* $(a^{\frac{r}{2}} - 1)$

    $\Rightarrow p$ divides $(a^{\frac{r}{2}} + 1)$ or divides $(a^{\frac{r}{2}} - 1)$

Shor's Algorithm - Number Theory

- Suppose: r is even
  - Then: $0 = \underbrace{a^r}_{=1} - 1 = (a^{\frac{r}{2}})^2 - 1 = \underbrace{(a^{\frac{r}{2}} + 1) \cdot (a^{\frac{r}{2}} - 1)}_{\text{remember: } x^2-1=(x-1)\cdot(x+1)} \pmod N$

    $\Rightarrow N$ divides $(a^{\frac{r}{2}} + 1) \cdot (a^{\frac{r}{2}} - 1)$
- Additionally suppose: $a^{\frac{r}{2}} \neq \pm 1 \pmod N$
  - Then: N does *neither* divide $(a^{\frac{r}{2}} + 1)$ *nor* $(a^{\frac{r}{2}} - 1)$

    $\Rightarrow p$ divides $(a^{\frac{r}{2}} + 1)$ or divides $(a^{\frac{r}{2}} - 1)$
- Then: $gcd(a^{\frac{r}{2}} + 1, N) = p \vee gcd(a^{\frac{r}{2}} - 1, N) = p$

Shor's Algorithm - Number Theory

- Suppose: r is even
  - Then: $0 = \underbrace{a^r}_{=1} - 1 = (a^{\frac{r}{2}})^2 - 1 = \underbrace{(a^{\frac{r}{2}} + 1) \cdot (a^{\frac{r}{2}} - 1)}_{\text{remember: } x^2-1=(x-1)\cdot(x+1)} \pmod N$

    $\Rightarrow N$ divides $(a^{\frac{r}{2}} + 1) \cdot (a^{\frac{r}{2}} - 1)$
- Additionally suppose: $a^{\frac{r}{2}} \neq \pm 1 \pmod N$
  - Then: N does *neither* divide $(a^{\frac{r}{2}} + 1)$ *nor* $(a^{\frac{r}{2}} - 1)$

    $\Rightarrow p$ divides $(a^{\frac{r}{2}} + 1)$ or divides $(a^{\frac{r}{2}} - 1)$
- Then: $gcd(a^{\frac{r}{2}} + 1, N) = p \vee gcd(a^{\frac{r}{2}} - 1, N) = p$
- How likely is r even and $a^{\frac{r}{2}} \neq \pm 1$?
  - Results in number theory show probability $\geq \frac{1}{2}$

Shor's Algorithm - Pseudo Code

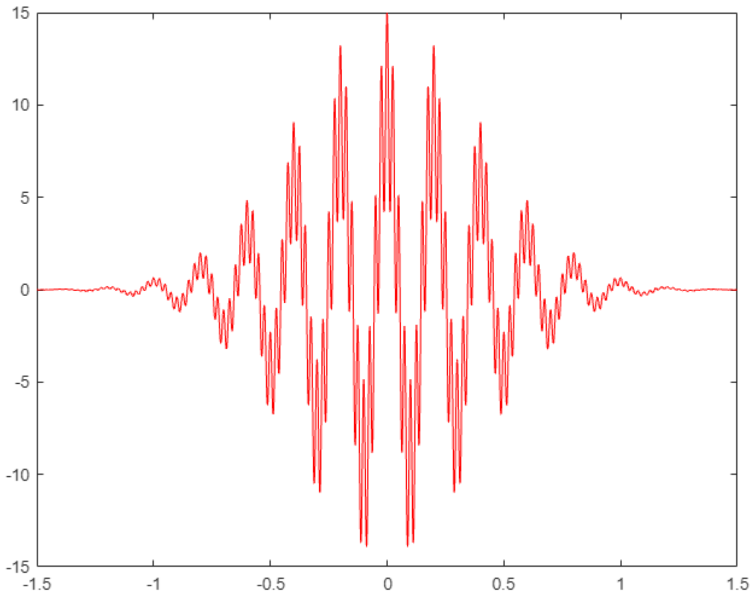
```
Algorithm Shor(N:Integer)

while(true){
  a = random(1, N - 1)
  b = gcd(a, N)
  if(b > 1){
    return b // this is already a non-trivial factor of N!
  }
  r = order(N, a) // magic done by quantum computing! → Quantum Fourier transform
  if(r is even){
    x = a^(r/2) (mod N)
    if(x != -1){ // x!=1 because r is smallest!
      return (gcd(x + 1, N), gcd(x - 1, N)) // determine two non-trivial factors!
    }
  }
}
```

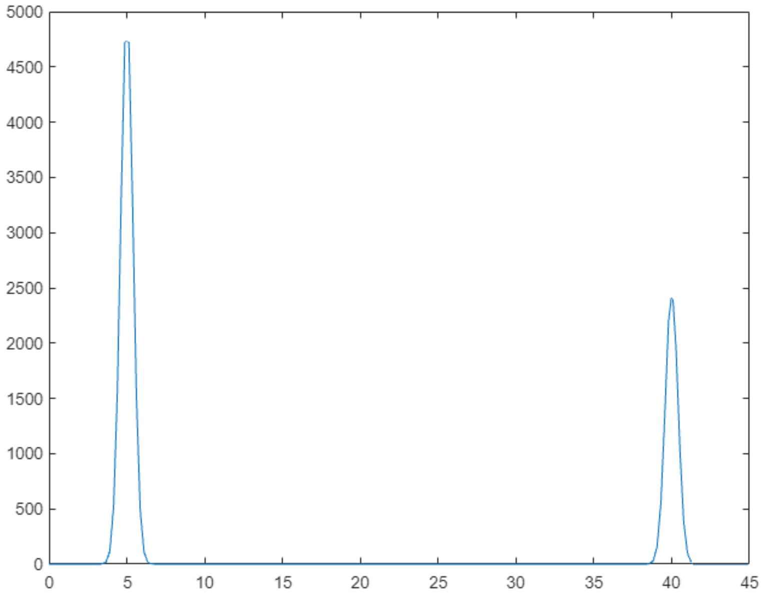
- Hybrid algorithm, where quantum computing is used to find r
  - r can be very large $\Rightarrow$ Classical approach too slow
- Remark: Pure classical algorithm with finding r on classical computer by Miller [M'76]

Fourier Transform for Determination of Frequency

$$f(t) = (10 \cdot cos(2 \cdot \pi \cdot 5 \cdot t) + 5 \cdot cos(2 \cdot \pi \cdot 40 \cdot t)) \cdot e^{-\pi \cdot t^2}$$

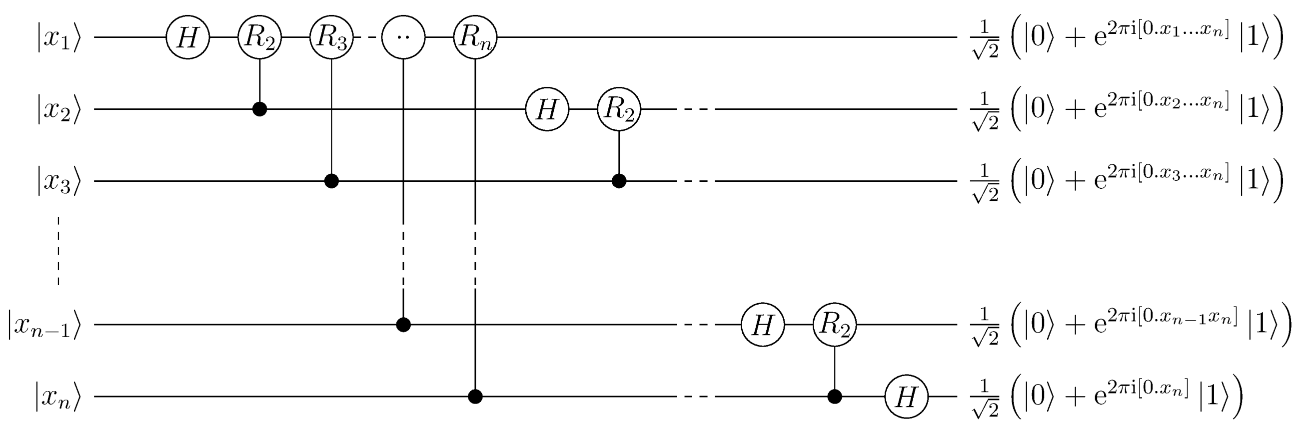


Positive part of the absolute value of the Fourier transform:



(Quantum) Fourier Transform

- The classical Fourier transform acts on a vector $(x_0, x_1, \ldots, x_{N-1}) \in \mathbb{C}^N$ and maps it to the vector $(y_0, y_1, \ldots, y_{N-1}) \in \mathbb{C}^N$ according to the formula:
$y_k = \frac{1}{\sqrt{N}} \cdot \sum_{n=0}^{N-1} x_n \cdot \omega_N^{-kn}, k = 0, 1, 2, \ldots, N-1$, where $\omega_N = e^{\frac{2\pi i}{N}}$ and ω_N^n is an N-th root of unity
- The quantum Fourier transform acts on a quantum state $|x\rangle = \sum_{i=0}^{N-1} x_i \cdot |i\rangle$ and maps it to a quantum state $\sum_{i=0}^{N-1} y_i \cdot |i\rangle$ according to the formula:
$y_k = \frac{1}{\sqrt{N}} \cdot \sum_{n=0}^{N-1} x_n \cdot \omega_N^{nk}, k = 0, 1, 2, \ldots, N-1$
- The inverse quantum Fourier transform acts similarly but with
$x_n = \frac{1}{\sqrt{N}} \cdot \sum_{k=0}^{N-1} y_k \cdot \omega_N^{-nk}, n = 0, 1, 2, \ldots, N-1$
- Quantum circuit of quantum Fourier transform:



Consequences of Shor’s algorithm

- Factoring is solvable in quantum polynomial time
  - Totally breaks RSA
    - To factor 2048 bit RSA integers:
      - 8 hours using 20 million noisy qubits [GE19]
      - less than a week using less than 1 million noisy qubits [G25]

Consequences of Shor’s algorithm

- Factoring is solvable in quantum polynomial time
  - Totally breaks RSA
    - To factor 2048 bit RSA integers:
      - 8 hours using 20 million noisy qubits [GE19]
      - less than a week using less than 1 million noisy qubits [G25]
- Modified Shor solves discrete logarithm problem
  - Totally breaks discrete log-based crypto
  - Including elliptic curve cryptography

Consequences of Shor's algorithm

- Factoring is solvable in quantum polynomial time
  - Totally breaks RSA
    - To factor 2048 bit RSA integers:
      - 8 hours using 20 million noisy qubits [GE19]
      - less than a week using less than 1 million noisy qubits [G25]
- Modified Shor solves discrete logarithm problem
  - Totally breaks discrete log-based crypto
  - Including elliptic curve cryptography
- Is public-key crypto dead?

Consequences of Shor's algorithm

- Factoring is solvable in quantum polynomial time
  - Totally breaks RSA
    - To factor 2048 bit RSA integers:
      - 8 hours using 20 million noisy qubits [GE19]
      - less than a week using less than 1 million noisy qubits [G25]
- Modified Shor solves discrete logarithm problem
  - Totally breaks discrete log-based crypto
  - Including elliptic curve cryptography
- Is public-key crypto dead?
- $\longrightarrow$ **Post-quantum** cryptography
  - Classical algorithms believed to withstand quantum attacks
    - National Institute of Standards and Technology (NIST) Post-Quantum Cryptography Standardization
      - program and competition by NIST to update their standards to include post-quantum cryptography

More Details on NIST PQC Standardization

- December **2016** - November **2017**: public call with 82 submissions
  - 69 candidates met both the submission requirements and the minimum acceptability criteria
  - posted online for public review and comments
- January **2019**: NIST selected 26 algorithms
- July **2020**: NIST selected 7 finalists and 8 alternates
- June **2021**: more thorough analysis of the theoretical and empirical evidence used to justify the security, as well as performance benchmarking using optimized implementations on a variety of soft- and hardware platforms
- August **2023**: NIST selected 4 algorithms for standardization: CRYSTALS–KYBER, along with three digital signature schemes: CRYSTALS–Dilithium, FALCON, and SPHINCS+, which are the basis for 3 public drafts of Federal Information Processing Standards (FIPS)
  - FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard
  - FIPS 204: Module-Lattice-Based Digital Signature Standard
  - FIPS 205: Stateless Hash-Based Digital Signature Standard

Other aspects of Cryptography and Quantum Computers

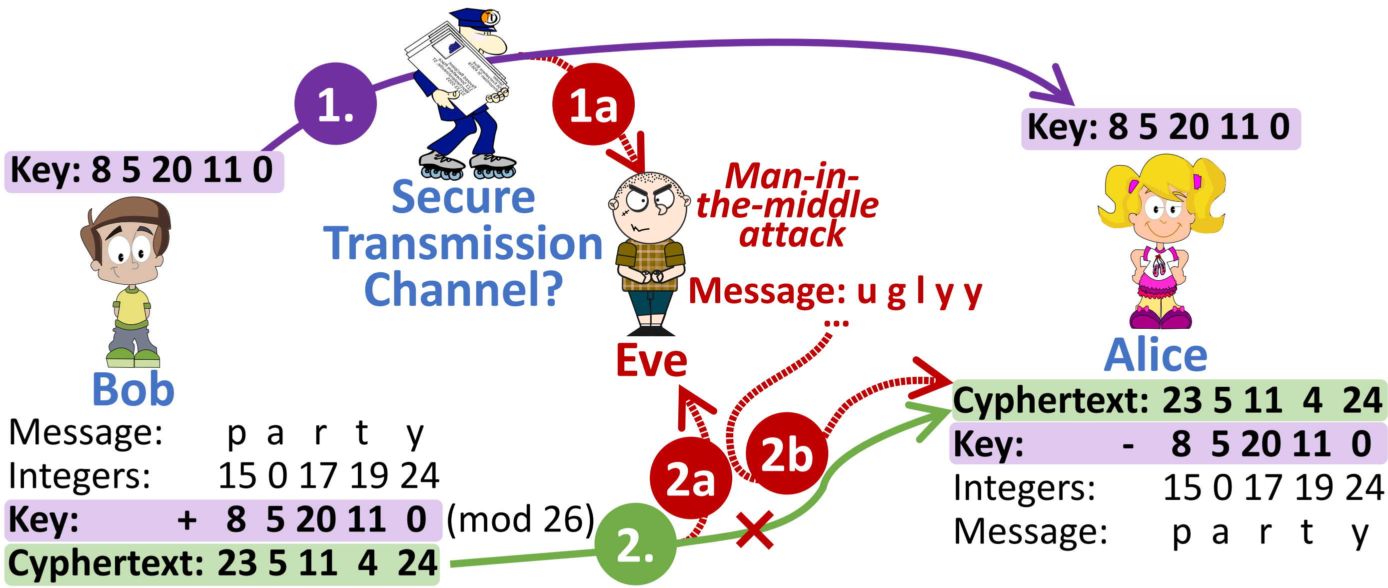
- Symmetric cryptography
  - Grover's algorithm
    - solves $O(2^n)$ problems in $O(2^{\frac{n}{2}})$ quantum steps
  - Solution
    - double key-lengths, e.g., $128 \rightarrow 256$

Other aspects of Cryptography and Quantum Computers

- Symmetric cryptography
  - Grover's algorithm
    - solves $O(2^n)$ problems in $O(2^{\frac{n}{2}})$ quantum steps
  - Solution
    - double key-lengths, e.g., $128 \rightarrow 256$

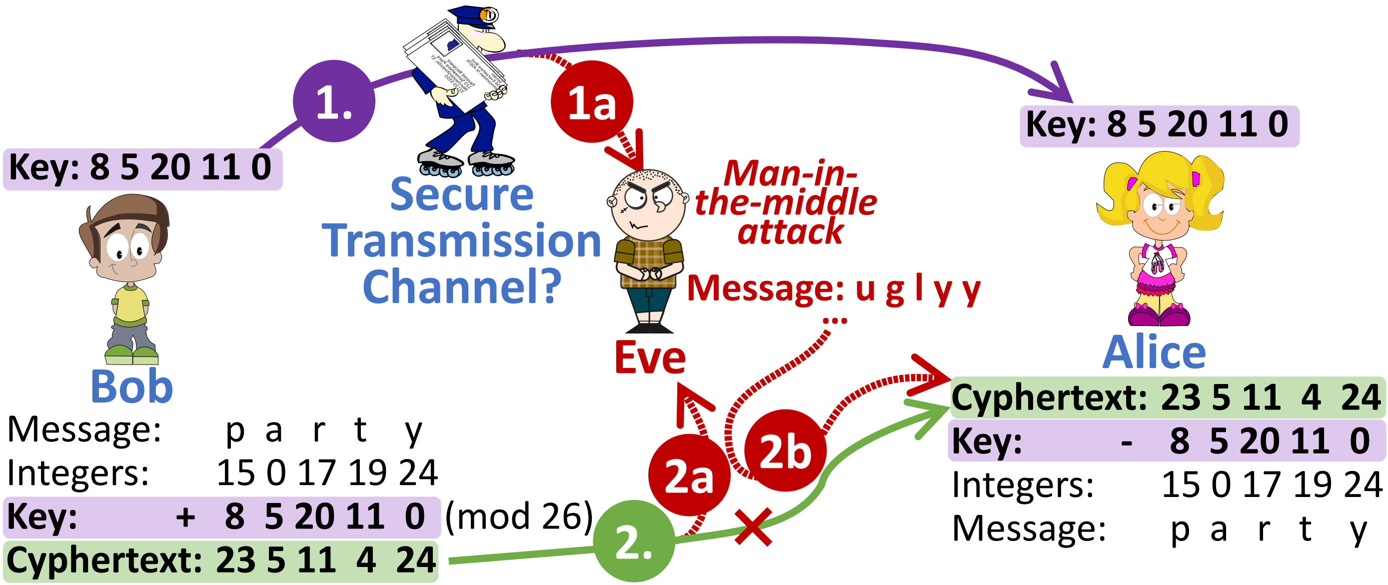
- The other way round: Quantum cryptography
  - Use quantum mechanics to build cryptography
  - Example: Quantum key distribution (on following slides)

One-time pad 1/2



- **information-theoretically secure**, i.e., provably uncrackable
  - under the precondition that the key cannot be stolen
  - even with infinite computing power, an adversary would not be able to gain any type of information about the plaintext by studying the ciphertext alone
  - message length can be obscured by adding additional superfluous characters

One-time pad 2/2



- **Drawback:** key must be at least as long as the message and must be transferred through a secure communication channel
  - Why not just sending the message through the secure communication channel?
  - Few scenarios like personally delivering keys for seldom communication via public channels in the future
  - $\Rightarrow$ one-time pad is not widely used in classical cryptography

Quantum Key Distribution

- **Goals**
  - Sending the key over possibly insecure channel
  - Alice and Bob will definitely recognize stealing the key/eavesdropping
    - Being warned they don't send messages
    - Try again later or via another channel

  $\Rightarrow$ Man-in-the-middle attack is not possible!
- **Means**
  - Quantum mechanics

Quantum Key Distribution

- **Goals**
  - Sending the key over possibly insecure channel
  - Alice and Bob will definitely recognize stealing the key/eavesdropping
    - Being warned they don't send messages
    - Try again later or via another channel

  $\Rightarrow$ Man-in-the-middle attack is not possible!
- **Means**
  - Quantum mechanics
- Several protocols
  - BB84 (our focus!) [BB'84]
  - E91 [E'91]
  - ...

BB84 Quantum Key Distribution - Step 1

- Alice chooses
  - a **random** sequence I of m bits (0 or 1)
  - a **random** sequence A of m bases ($\mathbb{S}$ or $\mathbb{H}$)
    - $\mathbb{S}$: standard basis $(|0\rangle, |1\rangle)$
    - $\mathbb{H}$: Hadamard basis $(|+\rangle, |-\rangle) = (\frac{|0\rangle+|1\rangle}{\sqrt{2}}, \frac{|0\rangle-|1\rangle}{\sqrt{2}})$
- $\forall i \in \{0, \ldots, m-1\}$:

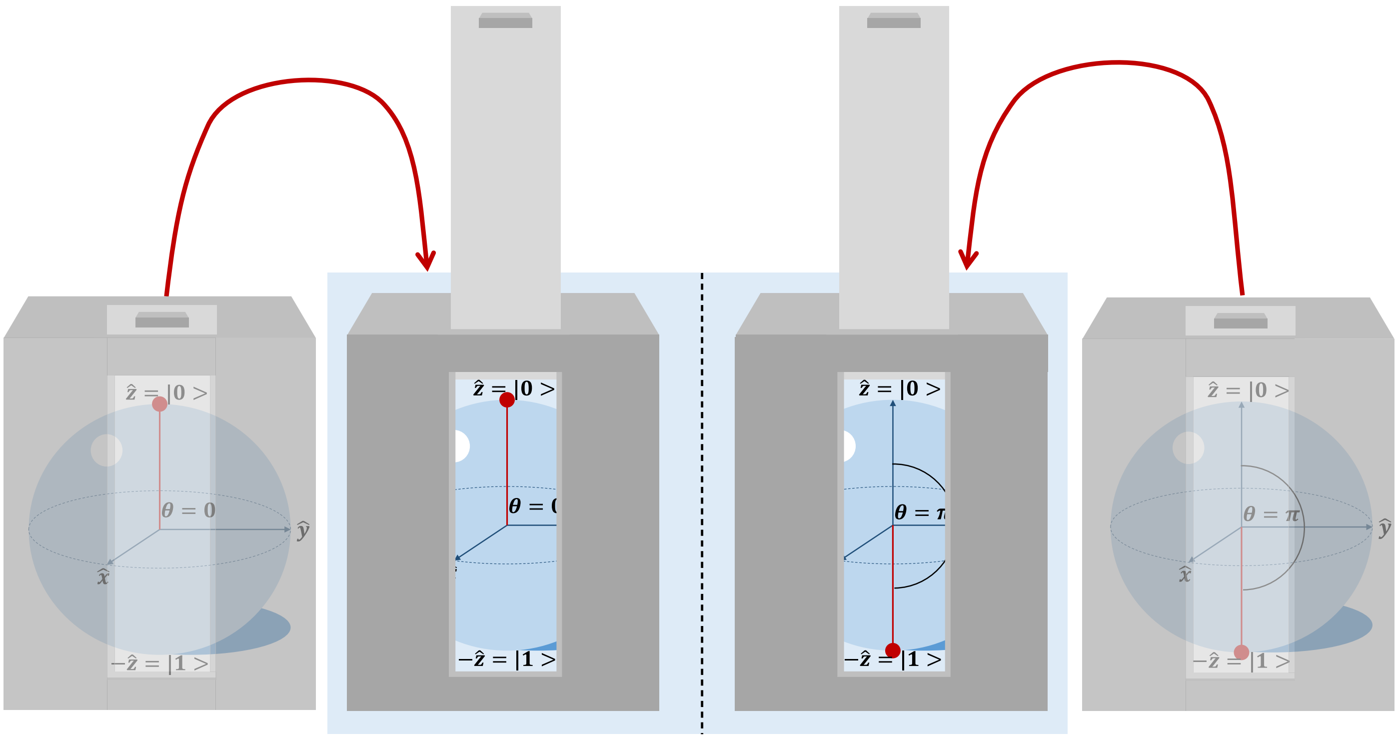
  Alice encodes the i-th bit $I[i]$ as qubit in the i-th basis $A[i]$
- Example:

Alice randomly chooses I and A to generate Q														
Bits I	0	0	1	0	1	1	0	0	1	0	1	1	0	1
Bases A	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{S}$
Qubits Q	$H\|0\rangle$	$H\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$H\|1\rangle$	$H\|0\rangle$	$\|0\rangle$	$H\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$H\|1\rangle$	$H\|0\rangle$	$\|1\rangle$

- Alice sends qubits Q to Bob

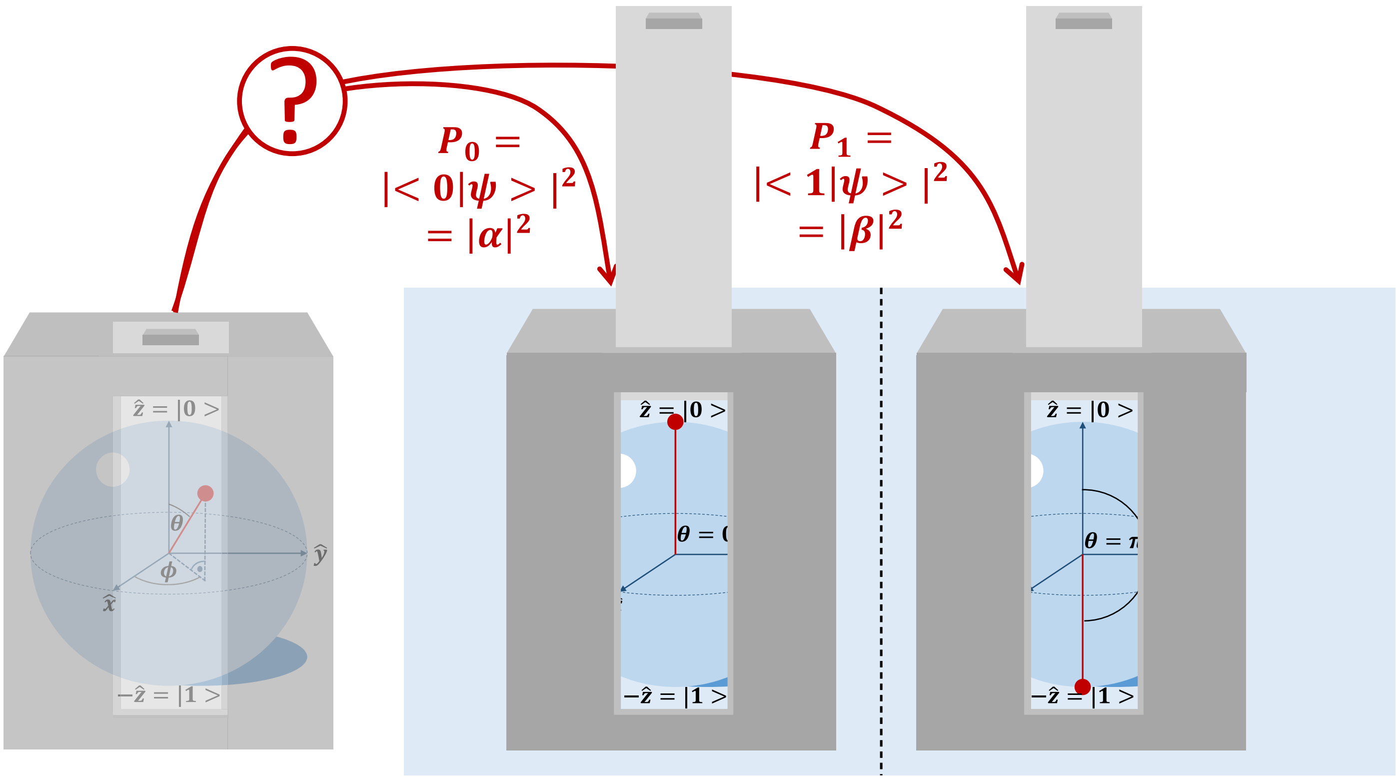
Quantum Measurement/Observation 1/2

- The state is not destroyed by a measurement/observation in quantum mechanical systems for state $|0\rangle$ and $|1\rangle$:



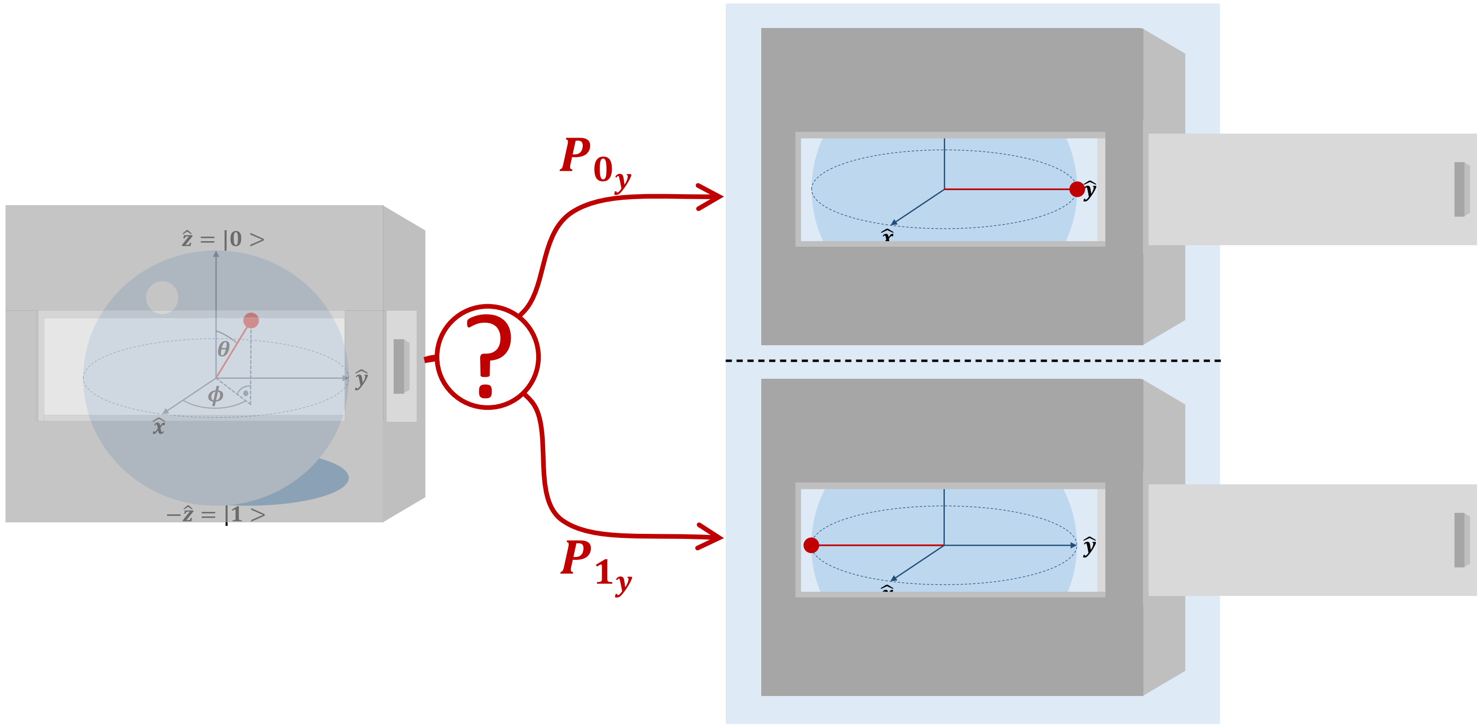
Quantum Measurement/Observation 2/2

- During observation a superposition state collapses to $|0\rangle$ or $|1\rangle$ according to corresponding probabilities:



Measurement/Observation along other axis (here y-axis)

- However, observation typically according to z-axis



BB84 Quantum Key Distribution - Step 2

- Bob
  - receives qubits Q from Alice (but no other information in this step)
  - chooses a **random** sequence B of m bases ($\mathbb{S}$ or $\mathbb{H}$)
  - measures the i-th qubit with the i-th basis $B[i]$ and gets the i-th bit $J[i]$
    - Case $A[i] = B[i]$: $I[i] = J[i]$
    - Case $A[i] \neq B[i]$: $J[i]$ randomly collapses to 0 or 1 (example: marked as **?**)
- Example:

Alice randomly chooses I and A to generate Q														
Bits I	0	0	1	0	1	1	0	0	1	0	1	1	0	1
Bases A	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{S}$
Qubits Q	$H\|0\rangle$	$H\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$H\|1\rangle$	$H\|0\rangle$	$\|0\rangle$	$H\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$H\|1\rangle$	$H\|0\rangle$	$\|1\rangle$
B receives Q, randomly chooses B and measures Q with bases B to determine J (? = 0 or 1, each with prob. $\frac{1}{2}$)														
Qubits Q	$H\|0\rangle$	$H\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$H\|1\rangle$	$H\|0\rangle$	$\|0\rangle$	$H\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$H\|1\rangle$	$H\|0\rangle$	$\|1\rangle$
Bases B	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{H}$
Bits J	**?**	0	**?**	0	1	**?**	**?**	**?**	1	0	1	1	0	**?**

BB84 Quantum Key Distribution - Step 3

- Alice and Bob
  - publicly compare their sequence of bases to find out which bits they supposedly share

Bases A (from Alice)	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{S}$
Bases B (from Bob)	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{H}$
Bit to be used?	✗	✓	✗	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓	✗

BB84 Quantum Key Distribution - Step 4

- Alice and Bob
  - compare some of the bits (to be used) to detect an eavesdropper/man-in-the-middle, and
  - use the rest of the bits as key in one-time-pad approach

Bits I (Alice)	0	0	1	0	1	1	0	0	1	0	1	1	0	1
Bit to be used?	✗	✓	✗	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓	✗
Bits to publicly compare	✗	0	✗		1	✗	✗	✗				1		✗
Bits to use as key (secret!)	✗		✗	0		✗	✗	✗	1	0	1		0	✗

- (Qu)Bits to be sent?

BB84 Quantum Key Distribution - Step 4

- Alice and Bob
  - compare some of the bits (to be used) to detect an eavesdropper/man-in-the-middle, and
  - use the rest of the bits as key in one-time-pad approach

Bits I (Alice)	0	0	1	0	1	1	0	0	1	0	1	1	0	1
Bit to be used?	✗	✓	✗	✓	✓	✗	✗	✗	✓	✓	✓	✓	✓	✗
Bits to publicly compare	✗	0	✗		1	✗	✗	✗				1		✗
Bits to use as key (secret!)	✗		✗	0		✗	✗	✗	1	0	1		0	✗

- (Qu)Bits to be sent?
  - About half of the bases are chosen differently from Alice and Bob, key length = message length l bits (one-time pad!)

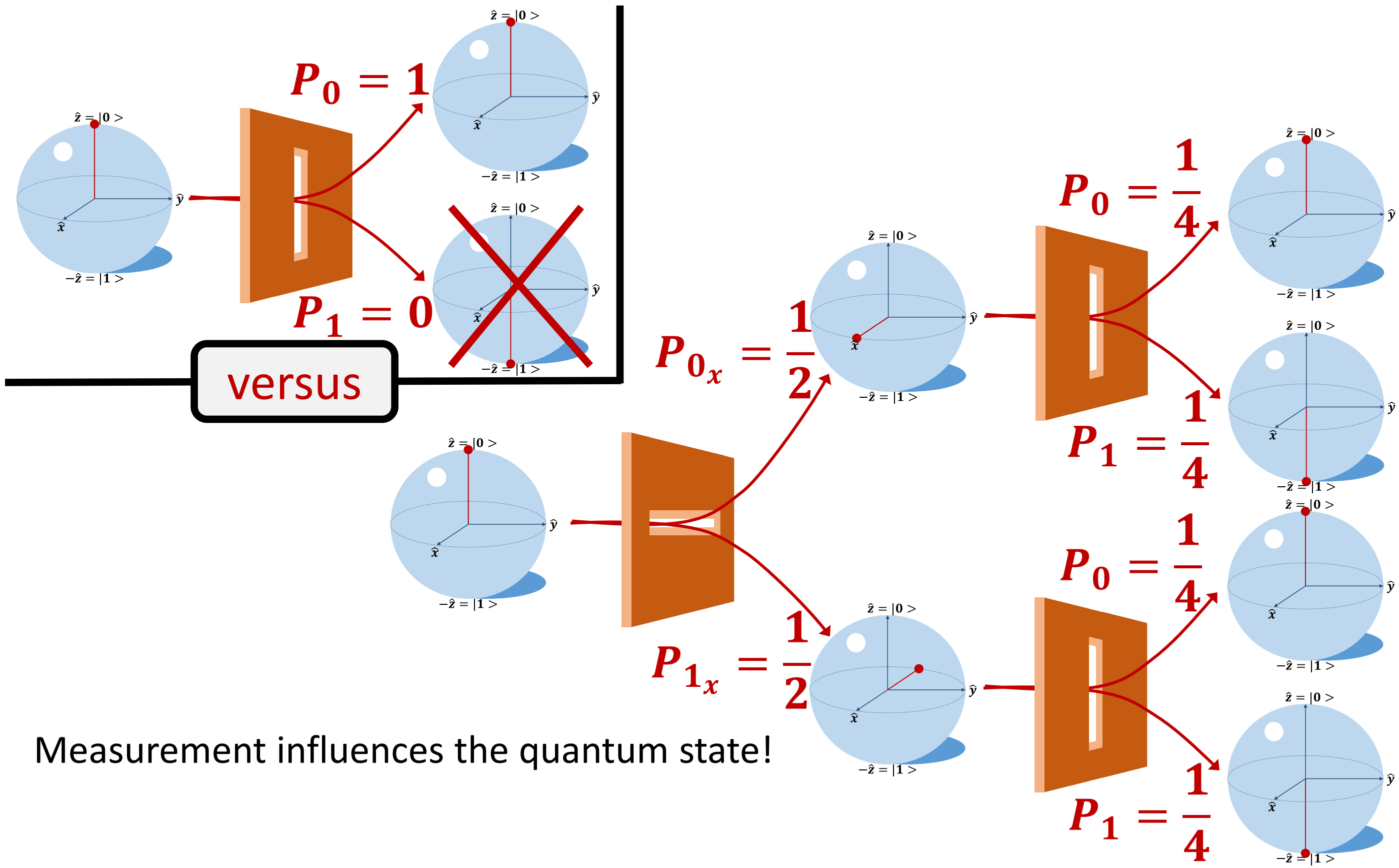
    $\Rightarrow$ **Qubits:** $\approx 2 \cdot (l + k)$ qubits for Q,

    Bits: $l + k$ bits for comparing bases publicly (each of Alice and Bob),

    k bits for detection of eavesdropping (each of Alice and Bob),

    l bits for message

Phenomenon for detection of eavesdropping



Measurement influences the quantum state!

BB84 Quantum Key Distribution - Step 1.5+

- What happens in case of eavesdropping?
- Example:

Eve receives Q and measures it with bases E collapsing quantum states of Q to Q'														
Qubits Q	$H\|0\rangle$	$H\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$H\|1\rangle$	$H\|0\rangle$	$\|0\rangle$	$H\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$H\|1\rangle$	$H\|0\rangle$	$\|1\rangle$
Bases E	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{H}$
Q'	$\|0\rangle$ or $\|1\rangle$	$H\|0\rangle$	$\|1\rangle$	$H\|0\rangle$ or $H\|1\rangle$	$H\|0\rangle$ or $H\|1\rangle$	$\|0\rangle$ or $\|1\rangle$	$H\|0\rangle$	$H\|0\rangle$ or $H\|1\rangle$	$\|0\rangle$ or $\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$ or $\|1\rangle$	$H\|0\rangle$	$H\|0\rangle$ or $H\|1\rangle$
Bob receives Q' instead of Q, measures with bases B to receive J (? = 0 or 1, each with prob. $\frac{1}{2}$)														
Bases B	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{H}$
Bits J	?	0	?	?	?	?	?	?	?	0	1	?	0	?
publicly compare Bob↔Alice	✗	$0 \updownarrow 0$	✗		? $\updownarrow 1$	✗	✗	✗				? $\updownarrow 1$		✗

- **With which probability is eavesdropping detected here?**

BB84 Quantum Key Distribution - Step 1.5+

Eve receives Q and measures it with bases E collapsing quantum states of Q to Q'														
Qubits Q	$H\|0\rangle$	$H\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$H\|1\rangle$	$H\|0\rangle$	$\|0\rangle$	$H\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$H\|1\rangle$	$H\|0\rangle$	$\|1\rangle$
Bases E	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{H}$
Q'	$\|0\rangle$ or $\|1\rangle$	$H\|0\rangle$	$\|1\rangle$	$H\|0\rangle$ or $H\|1\rangle$	$H\|0\rangle$ or $H\|1\rangle$	$\|0\rangle$ or $\|1\rangle$	$H\|0\rangle$	$H\|0\rangle$ or $H\|1\rangle$	$\|0\rangle$ or $\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$ or $\|1\rangle$	$H\|0\rangle$	$H\|0\rangle$ or $H\|1\rangle$
Bob receives Q' instead of Q, measures with bases B to receive J (? = 0 or 1, each with prob. $\frac{1}{2}$)														
Bases B	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{S}$	$\mathbb{S}$	$\mathbb{H}$	$\mathbb{H}$	$\mathbb{H}$
Bits J	?	0	?	?	?	?	?	?	?	0	1	?	0	?
publicly compare Bob↔Alice	✗	$0 \updownarrow 0$	✗		$? \updownarrow 1$	✗	✗	✗				$? \updownarrow 1$		✗

- Here: eavesdropping is detected with probability $1 - \frac{1}{2} \cdot \frac{1}{2} = 75\%$
  - In general: $\approx 1 - \left(\frac{1}{2}\right)^{\frac{k}{2}}$ with k number of bits to compare, assuming Eve chooses $\frac{k}{2}$ bases different from Alice/Bob, such that for each of these $\frac{k}{2}$ bits with a probability of $\frac{1}{2}$ the 'wrong' bit is measured
- Increase #bits to be compared to detect eavesdropping with higher probability

BB84 Quantum Key Distribution - Remarks

- Here assumption:
  - quantum transmission is perfect
- In a real-life setting:
  - use error-correction methods on top of the quantum key distribution protocol

BB84 Quantum Key Distribution - Remarks

- Here assumption:
  - quantum transmission is perfect
- In a real-life setting:
  - use error-correction methods on top of the quantum key distribution protocol
- The International Organization for Standardization (ISO) releases standards for Quantum Key Distribution (QKD) security requirements
  - ISO/IEC 23837-1:2023 "Information security — Security requirements, test and evaluation methods for quantum key distribution — Part 1: Requirements" includes conventional network components, quantum optical components and the entire implementation of QKD protocols

BB84 Quantum Key Distribution - Remarks

- Here assumption:
  - quantum transmission is perfect
- In a real-life setting:
  - use error-correction methods on top of the quantum key distribution protocol
- The International Organization for Standardization (ISO) releases standards for Quantum Key Distribution (QKD) security requirements
  - ISO/IEC 23837-1:2023 "Information security — Security requirements, test and evaluation methods for quantum key distribution — Part 1: Requirements" includes conventional network components, quantum optical components and the entire implementation of QKD protocols
- already research prototypes and commercial products for QKD
  - QKD for secure video conferences demonstrated by Fraunhofer [MW21]
  - Toshiba QKD can be deployed on to an optical fibre network

Summary & Conclusions

- Shor's algorithm
  - Consequences for cryptography $\longrightarrow$ post-quantum cryptography
- One-time pad
  - uncrackable if eavesdropping on the key can be ruled out
- Quantum Key Distribution - BB84
  - Protocol
  - Overhead: Number of (qu)bits to be sent
  - Probability for detection of eavesdropping