Lecture

Quantum Computing

(CS5070)

Quantum Cryptography: Shor, Quantum Key Distribution

Professor Dr. rer. nat. habil. Sven Groppe

https://www.ifis.uni-luebeck.de/index.php?id=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 → **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

\[2 \cdot 3 \cdot 5 \cdot 43 = 1290\]
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 | 1 | 1 | 2 | 4 | 8 | 16 | 11 | 1 | 2 | ...

adapted from: H. Jacobsen, TEK4500 - Introduction to Cryptography, Lecture 12, University of Oslo, 2020 📝 [P'94] 📝 [P'96] 📝
Shor's Algorithm - Idea

- 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 | 11 | 1 | 2 | 4 | 8 | 16 | 11 | 1 | 2 | ...

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

- **In general:**
  $$a^1, a^2, \ldots, a^r = 1, a^1, a^2, \ldots \pmod{N}$$
  
  order of $a$ = the smallest positive $r$ such that $a^r = 1 \pmod{N}$
Shor's Algorithm - Number Theory

- **Euler's Theorem:** \( \forall a \in \mathbb{Z}_N^* \) with \( \gcd(a, N) = 1 \) : \( a^{\varphi(N)} = 1 \mod N \), 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 \( \gcd(a, b) = \begin{cases} b & \text{if } a \mod b = 0 \\ \gcd(b, a \mod b) & \text{otherwise} \end{cases} \)

- 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:** \( \forall a \in \mathbb{Z}_N^* \text{ with } gcd(a, N) = 1 : a^\varphi(N) = 1 \mod N \), where Euler's phi function: \( \varphi(N) = |\{a \in \mathbb{N} | 1 \leq a \leq N \land gcd(a, N) = 1\}| \)
  and greatest common divisor \( gcd(a, b) = \begin{cases} b & \text{if } a \mod b = 0 \\ gcd(b, a \mod b) & \text{otherwise} \end{cases} \)

- 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:** \( r \) must divide \( \varphi(N) = (p - 1) \cdot p^{k-1} \cdot \varphi(m) \)

**Proof:**
\[
\varphi(N) = s \cdot r + t, \text{ where } s, t \in \mathbb{N} \text{ with } 0 \leq t < r \\
1 \overset{\text{Euler}}{=} a^{\varphi(N)} = a^{s \cdot r + t} = a^{s \cdot r} \cdot a^t = (a^r)^s \cdot a^t = 1^s \cdot a^t \mod N \\
\Rightarrow t = 0 \text{ (since } r \text{ is the smallest)} \Rightarrow \varphi(N) = (p - 1) \cdot p^{k-1} \cdot \varphi(m) = s \cdot r \]
Shor's Algorithm - Number Theory

- **Euler's Theorem:** \( \forall a \in \mathbb{Z}_N^* \) with \( \gcd(a, N) = 1 : a^{\varphi(N)} = 1 \mod N \)

where Euler's phi function: \( \varphi(N) = |\{a \in \mathbb{N} | 1 \leq a \leq N \land \gcd(a, N) = 1\}| \)

and greatest common divisor \( \gcd(a, b) = \begin{cases} b & \text{if } a \mod b = 0 \\ \gcd(b, a \mod b) & \text{otherwise} \end{cases} \)

- 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:** \( r \) must divide \( \varphi(N) = (p - 1) \cdot p^{k-1} \cdot \varphi(m) \)

**Proof:**

\[ \varphi(N) = s \cdot r + t, \text{ where } s, t \in \mathbb{N} \text{ with } 0 \leq t < r \]

\[ 1 \overset{Euler}{=} a^{\varphi(N)} = a^{s \cdot r + t} = a^{s \cdot r} \cdot a^t = (a^r)^s \cdot a^t = 1^s \cdot a^t \mod N \]

\[ \Rightarrow t = 0 \text{ (since } r \text{ is the smallest)} \Rightarrow \varphi(N) = (p - 1) \cdot p^{k-1} \cdot \varphi(m) = s \cdot r \]

**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**: \( 0 = a^r \equiv 1 \pmod{N} \)

\[
0 = a^r - 1 = (a^{\frac{r}{2}})^2 - 1 = (a^{\frac{r}{2}} + 1) \cdot (a^{\frac{r}{2}} - 1) \quad \text{(mod } N)\]

remember: \( x^2 - 1 = (x - 1) \cdot (x + 1) \)

\[
\Rightarrow N \text{ divides } (a^{\frac{r}{2}} + 1) \cdot (a^{\frac{r}{2}} - 1)
\]
Shor's Algorithm - Number Theory

- **Suppose:** \( r \) is even
  - Then: \( 0 = a^r - 1 = (a^{\frac{r}{2}})^2 - 1 = (a^{\frac{r}{2}} + 1) \cdot (a^{\frac{r}{2}} - 1) \pmod{N} \)
    - remember: \( x^2 - 1 = (x - 1) \cdot (x + 1) \)
  \[ \Rightarrow N \text{ divides } (a^{\frac{r}{2}} + 1) \cdot (a^{\frac{r}{2}} - 1) \]

- **Additionally suppose:** \( a^{\frac{r}{2}} \neq \pm 1 \pmod{N} \)
  - Then: \( N \) does not divide \( (a^{\frac{r}{2}} + 1) \) nor \( (a^{\frac{r}{2}} - 1) \)
  \[ \Rightarrow p \text{ divides } (a^{\frac{r}{2}} + 1) \text{ or divides } (a^{\frac{r}{2}} - 1) \]
Shor's Algorithm - Number Theory

- **Suppose:** \( r \) is even
  
  - Then: \( 0 = a^r - 1 = (a^{\frac{r}{2}})^2 - 1 = (a^{\frac{r}{2}} + 1) \cdot (a^{\frac{r}{2}} - 1) \) (mod \( N \))  
  
  remember: \( x^2 - 1 = (x - 1) \cdot (x + 1) \)  
  
  \( \Rightarrow \) \( N \) divides \( (a^{\frac{r}{2}} + 1) \cdot (a^{\frac{r}{2}} - 1) \)

- **Additionally suppose:** \( \frac{r}{2} \neq \pm 1 \) (mod \( N \))
  
  - 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:** \( gcd(a^{\frac{r}{2}} + 1, N) = p \lor gcd(a^{\frac{r}{2}} - 1, N) = p \)
Shor's Algorithm - Number Theory

- **Suppose:** \( r \) is even
  - Then: \( 0 = \left( a^\frac{r}{2} \right)^2 - 1 = (a^\frac{r}{2} + 1) \cdot (a^\frac{r}{2} - 1) \pmod{N} \)
    
    \( \Rightarrow \ N \) 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:** \( \gcd(a^\frac{r}{2} + 1, N) = p \lor \gcd(a^\frac{r}{2} - 1, N) = p \)

- **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

```plaintext
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

\[ f(t) = (10 \cdot \cos(2 \cdot \pi \cdot 5 \cdot t) + 5 \cdot \cos(2 \cdot \pi \cdot 40 \cdot t)) \cdot e^{-\pi t^2} \]
(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:
  \[
y_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n \cdot \omega_N^{-kn}, \quad k = 0, 1, 2, \ldots, N - 1,
\]
  where \( \omega_N = e^{\frac{2\pi i}{N}} \) and \( \omega_N^n \) 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:
  \[
y_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n \cdot \omega_N^{-nk}, \quad k = 0, 1, 2, \ldots, N - 1
\]

- The inverse quantum Fourier transform acts similarly but with
  \[
x_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} y_k \cdot \omega_N^{-nk}, \quad n = 0, 1, 2, \ldots, N - 1
\]

- Quantum circuit of quantum Fourier transform:
Consequences of Shor’s algorithm

- **Factoring is solvable in quantum polynomial time**
  - Totally breaks RSA
Consequences of Shor’s algorithm

- **Factoring is solvable in quantum polynomial time**
  - Totally breaks RSA
- **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
- **Modified Shor solves discrete logarithm problem**
  - Totally breaks discrete log-based crypto
  - Including elliptic curve cryptography
- Is public-key crypto dead?

adapted from: H. Jacobsen, TEK4500 - Introduction to Cryptography, Lecture 12, University of Oslo, 2020 [NIST'17]
Consequences of Shor’s algorithm

- **Factoring is solvable in quantum polynomial time**
  - Totally breaks RSA

- **Modified Shor solves discrete logarithm problem**
  - Totally breaks discrete log-based crypto
  - Including elliptic curve cryptography

- **Is public-key crypto dead?**

- **→ Post-quantum cryptography**
  - Classical algorithms believed to withstand quantum attacks
    - NIST Post-Quantum Cryptography Standardization
      - program and competition by NIST to update their standards to include post-quantum cryptography
      - already third round with top candidates based on lattice, code-based, hash-based, multivariate, supersingular elliptic curve isogeny and zero-knowledge proof cryptography
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 $O(2^{n/2})$ 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

**Key:** 8 5 20 11 0

**Message:** party
**Integers:** 15 0 17 19 24
**Key:** + 8 5 20 11 0 (mod 26)

**Cyphertext:** 23 5 11 4 24

**Key:** 8 5 20 11 0

**Message:** utility
**Cyphertext:** 23 5 11 4 24

**Key:** - 8 5 20 11 0

**Integers:** 15 0 17 19 24

**Man-in-the-middle attack**
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
- ⇒ 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
  ⇒ 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]
  - ...


Quantum Computing
Quantum Cryptography: Shor, Quantum Key Distribution

Institut für Informationssysteme | Prof. Dr. habil. S. Groppe
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 (\( S \) or \( H \))
    - \( S \): standard basis \( (|0\rangle, |1\rangle) \)
    - \( H \): Hadamard basis \( (|+\rangle, |−\rangle) = \left( \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \)
- \( \forall i \in \{0, \ldots, m − 1\} \):
  - Alice encodes the \( i\)-th bit \( I[i] \) as qubit in the \( i\)-th basis \( A[i] \)
- Example:

<table>
<thead>
<tr>
<th>Bits</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bases</td>
<td>( H )</td>
<td>( H )</td>
<td>( S )</td>
<td>( S )</td>
<td>( S )</td>
<td>( H )</td>
<td>( H )</td>
<td>( S )</td>
<td>( H )</td>
<td>( S )</td>
<td>( S )</td>
<td>( H )</td>
<td>( H )</td>
<td>( S )</td>
</tr>
<tr>
<td>Qubits</td>
<td>( H</td>
<td>0\rangle )</td>
<td>( H</td>
<td>0\rangle )</td>
<td>(</td>
<td>1\rangle )</td>
<td>(</td>
<td>0\rangle )</td>
<td>(</td>
<td>1\rangle )</td>
<td>( H</td>
<td>1\rangle )</td>
<td>( H</td>
<td>0\rangle )</td>
</tr>
</tbody>
</table>

- 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 $|1\rangle$: 

![Diagram showing quantum states and measurements](image-url)
Quantum Measurement/Observation 2/2

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

$$P_0 = \left|\langle 0|\psi \rangle\right|^2 = |\alpha|^2$$
$$P_1 = \left|\langle 1|\psi \rangle\right|^2 = |\beta|^2$$
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 ($S$ or $H$)
  - measures the $i$-th qubit with the $i$-th basis $B[i]$ and gets the $i$-th bit $J[i]$
    - Case $A[i] \neq B[i]: J[i]$ randomly collapses to 0 or 1 (example: marked as ?)

- Example:

<table>
<thead>
<tr>
<th>Bits $I$</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bases $A$</td>
<td>$H$</td>
<td>$H$</td>
<td>$S$</td>
<td>$S$</td>
<td>$S$</td>
<td>$H$</td>
<td>$H$</td>
<td>$S$</td>
<td>$H$</td>
<td>$S$</td>
<td>$S$</td>
<td>$H$</td>
<td>$H$</td>
<td>$S$</td>
</tr>
<tr>
<td>Qubits $Q$</td>
<td>$H</td>
<td>0\rangle$</td>
<td>$H</td>
<td>0\rangle$</td>
<td>$</td>
<td>1\rangle$</td>
<td>$</td>
<td>0\rangle$</td>
<td>$</td>
<td>1\rangle$</td>
<td>$H</td>
<td>1\rangle$</td>
<td>$H</td>
<td>0\rangle$</td>
</tr>
</tbody>
</table>

Bob 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 $B$ | $S$ | $H$ | $H$ | $S$ | $S$ | $S$ | $S$ | $H$ | $H$ | $S$ | $S$ | $H$ | $H$ | $H$ |
| 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

<table>
<thead>
<tr>
<th>Bases $A$ (from Alice)</th>
<th>H</th>
<th>H</th>
<th>S</th>
<th>S</th>
<th>S</th>
<th>H</th>
<th>H</th>
<th>S</th>
<th>H</th>
<th>S</th>
<th>S</th>
<th>H</th>
<th>H</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bases $B$ (from Bob)</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>S</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>Bit to be used?</td>
<td>X</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Bits $I$ (Alice)</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit to be used?</td>
<td>X</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>Bits to publicly compare</td>
<td>X</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>1</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bits to use as key (secret!)</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- (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

<table>
<thead>
<tr>
<th>Bits $I$ (Alice)</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit to be used?</td>
<td>X</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Bits to publicly compare</td>
<td>X</td>
<td>0</td>
<td>X</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>1</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bits to use as key (secret!)</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

- **(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:**

| Qubits $Q$ | $H|0\rangle$ | $H|0\rangle$ | $|1\rangle$ | $|0\rangle$ | $|1\rangle$ | $H|1\rangle$ | $H|0\rangle$ | $|0\rangle$ | $|1\rangle$ | $H|1\rangle$ | $H|0\rangle$ | $|1\rangle$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bases $E$ | $S$ | $H$ | $S$ | $H$ | $H$ | $S$ | $H$ | $H$ | $S$ | $S$ | $S$ | $H$ |
| $Q'$ | $|0\rangle$ or $|1\rangle$ | $H|0\rangle$ or $H|1\rangle$ | $H|0\rangle$ or $H|1\rangle$ | $|0\rangle$ or $|1\rangle$ | $H|0\rangle$ or $H|1\rangle$ | $|0\rangle$ or $|1\rangle$ | $|0\rangle$ or $|1\rangle$ | $H|0\rangle$ or $H|1\rangle$ |
| Bob receives $Q'$ instead of $Q$, measures with bases $B$ to receive $J$ ($? = 0$ or $1$, each with prob. $\frac{1}{2}$) | $S$ | $H$ | $H$ | $S$ | $S$ | $S$ | $H$ | $H$ | $S$ | $S$ | $H$ | $H$ |
| Bases $B$ | $?\rightarrow 0$ | $?\rightarrow 0$ | $?\rightarrow 0$ | $?\rightarrow 0$ | $?\rightarrow 0$ | $?\rightarrow 0$ | $?\rightarrow 0$ | $?\rightarrow 0$ | $?\rightarrow 0$ |
| Bits $J$ | $X$ | $0$ | $1$ | $X$ | $X$ | $X$ | $?\rightarrow 1$ | $?\rightarrow 1$ | $X$ |
| publicly compare Bob↔Alice | $X$ | $X$ | $X$ | $X$ | $X$ | $X$ | $X$ | $X$ | $X$ |

- **With which probability is eavesdropping detected here?**
**BB84 Quantum Key Distribution - Step 1.5+**

<table>
<thead>
<tr>
<th>Qubits ( Q )</th>
<th>( \text{Bases} \ E )</th>
<th>( \text{Bases} \ B )</th>
<th>( \text{Bits} \ J )</th>
<th>( \text{publicly compare} )</th>
<th>( \text{Bob} \leftrightarrow \text{Alice} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{H}</td>
<td>0 \rangle )</td>
<td>( \text{H}</td>
<td>0 \rangle )</td>
<td>( \text{S} )</td>
<td>( ? )</td>
</tr>
<tr>
<td>( \text{H}</td>
<td>0 \rangle )</td>
<td>( \text{H}</td>
<td>1 \rangle )</td>
<td>( \text{H} )</td>
<td>( 0 )</td>
</tr>
<tr>
<td>( \text{H}</td>
<td>0 \rangle )</td>
<td>( \text{H}</td>
<td>1 \rangle )</td>
<td>( \text{S} )</td>
<td>( ? )</td>
</tr>
<tr>
<td>( \text{H}</td>
<td>0 \rangle )</td>
<td>( \text{H}</td>
<td>1 \rangle )</td>
<td>( \text{H} )</td>
<td>( 1 )</td>
</tr>
<tr>
<td>( \text{H}</td>
<td>0 \rangle )</td>
<td>( \text{H}</td>
<td>1 \rangle )</td>
<td>( \text{H} )</td>
<td>( ? )</td>
</tr>
<tr>
<td>( \text{H}</td>
<td>0 \rangle )</td>
<td>( \text{H}</td>
<td>1 \rangle )</td>
<td>( \text{H} )</td>
<td>( 0 )</td>
</tr>
<tr>
<td>( \text{H}</td>
<td>0 \rangle )</td>
<td>( \text{H}</td>
<td>1 \rangle )</td>
<td>( \text{H} )</td>
<td>( ? )</td>
</tr>
<tr>
<td>( \text{H}</td>
<td>0 \rangle )</td>
<td>( \text{H}</td>
<td>1 \rangle )</td>
<td>( \text{H} )</td>
<td>( 0 )</td>
</tr>
</tbody>
</table>

- **Here:** eavesdropping is detected with probability \( 1 - \frac{1}{2} \cdot \frac{1}{2} = 75\% \)

- In general: \( \approx 1 - \left( \frac{1}{2} \right)^k \) 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
Summary & Conclusions

- **Shor's algorithm**
  - Consequences for cryptography → post-quantum cryptography

- **One-time pad**
  - Un.crackable 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