

Hardwarenahe Programmierung

Prof. Dr. rer. nat. Peter Gerwinski

2. Januar 2020

Hardwarenahe Programmierung

<https://gitlab.cvh-server.de/pgerwinski/hp>

1 Einführung

2 Einführung in C

3 Bibliotheken

4 Hardwarenahe Programmierung

...

4.6 Byte-Reihenfolge – Endianness

4.7 Binärdarstellung negativer Zahlen

4.8 Speicherausrichtung – Alignment

5 Algorithmen

5.1 Differentialgleichungen

5.2 Rekursion

5.3 Aufwandsabschätzungen

5. $\frac{1+i}{\sqrt{2}}$ Quantencomputer

...

...

5.3 Aufwandsabschätzungen – Komplexitätsanalyse

Wann ist ein Programm „schnell“?

Faustregel:

Schachtelung der Schleifen zählen

k Schleifen ineinander $\rightarrow \mathcal{O}(n^k)$

Wie schnell ist RSA-Verschlüsselung?

$c = m^e \% N$ („%“ = „modulo“)

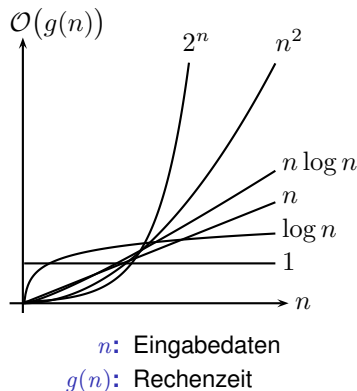
```
int c = 1;
for (int i = 0; i < e; i++)
    c = (c * m) % N;
```

- $\mathcal{O}(e)$ Iterationen
- mit Trick: $\mathcal{O}(\log e)$ Iterationen ($\log e$ = Anzahl der Ziffern von e)

Jede Iteration enthält eine Multiplikation und eine Division.

Aufwand dafür: $\mathcal{O}(\log e)$

\rightarrow Gesamtaufwand: $\mathcal{O}((\log e)^2)$



5.3 Aufwandsabschätzungen – Komplexitätsanalyse

Wann ist ein Programm „schnell“?

Faustregel:

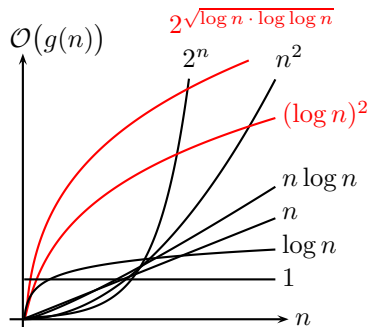
Schachtelung der Schleifen zählen

k Schleifen ineinander $\rightarrow \mathcal{O}(n^k)$

Wie schnell ist RSA?

(n = typische beteiligte Zahl, z. B. e, p, q)

- Ver- und Entschlüsselung (Exponentiation):
 $\mathcal{O}((\log n)^2)$
- Schlüsselerzeugung (Berechnung von d):
 $\mathcal{O}((\log n)^2)$
- Verschlüsselung brechen (Primfaktorzerlegung):
 $\mathcal{O}(2^{\sqrt{\log n \cdot \log \log n}})$



n : Eingabedaten

$g(n)$: Rechenzeit

Die Sicherheit von RSA beruht darauf, daß das Brechen der Verschlüsselung aufwendiger ist als $\mathcal{O}((\log n)^k)$ (für beliebiges k).

5.3 Aufwandsabschätzungen – Komplexitätsanalyse

Wann ist ein Programm „schnell“?

Faustregel:

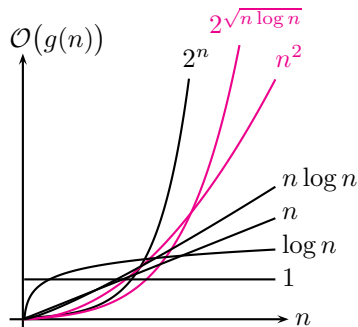
Schachtelung der Schleifen zählen

k Schleifen ineinander $\rightarrow \mathcal{O}(n^k)$

Wie schnell ist RSA?

(n = typische beteiligte Zahl, z. B. e, p, q)

- Ver- und Entschlüsselung (Exponentiation):
 $\mathcal{O}((\log n)^2)$
- Schlüsselerzeugung (Berechnung von d):
 $\mathcal{O}((\log n)^2)$
- Verschlüsselung brechen (Primfaktorzerlegung):
 $\mathcal{O}(2^{\sqrt{\log n \cdot \log \log n}})$



n : Eingabedaten

$g(n)$: Rechenzeit

Die Sicherheit von RSA beruht darauf, daß das Brechen der Verschlüsselung aufwendiger ist als $\mathcal{O}((\log n)^k)$ (für beliebiges k).

5. $\frac{1+i}{\sqrt{2}}$ Quantencomputer

Mit Hilfe eines Quantencomputers ist es möglich, RSA mit dem Aufwand $\mathcal{O}((\log n)^3)$ zu brechen.

Hierfür ist ein Quantencomputer mit mindestens $\log n$ Qubits erforderlich. ($\log n$ ist die Länge des Schlüssels in Bits, derzeit typischerweise 2048 bis 4096.)

Dezember 2001:

IBM präsentiert einen funktionierenden Quantencomputer mit 7 Qubits.

September 2019:

Google präsentiert einen funktionierenden Quantencomputer mit 53 Qubits.

Die Sicherheit von RSA beruht darauf, daß das Brechen der Verschlüsselung aufwendiger ist als $\mathcal{O}((\log n)^k)$ (für beliebiges k).

5. $\frac{1+i}{\sqrt{2}} \cdot i$ Einführung in die Quantenmechanik

Klassische Mechanik (Physik):

Zustand eines Teilchens (Massenpunkt):

Masse m , Ort \vec{x} , Impuls \vec{p} (oder: Geschwindigkeit \vec{v})

5. $\frac{1+i}{\sqrt{2}} \cdot i$ Einführung in die Quantenmechanik

Klassische Mechanik (Physik):

Zustand eines Teilchens (Massenpunkt):

Masse m , Ort \vec{x} , Impuls \vec{p} (oder: Geschwindigkeit \vec{v})

Quantenmechanik:

Zustand eines Teilchens:

Masse m , komplexwertige Wellenfunktion $\psi(\vec{x})$

- Ort: $|\psi(\vec{x})|^2 =$ Wahrscheinlichkeit, das Teilchen am Ort \vec{x} zu messen
- Impuls: Wellenstruktur in der komplexen Phase

5. $\frac{1+i}{\sqrt{2}} \cdot i$ Einführung in die Quantenmechanik

Klassische Mechanik (Physik):

Zustand eines Teilchens (Massenpunkt):

Masse m , Ort \vec{x} , Impuls \vec{p} (oder: Geschwindigkeit \vec{v})

Quantenmechanik:

Zustand eines Teilchens:

Masse m , komplexwertige Wellenfunktion $\psi(\vec{x})$

- Ort: $|\psi(\vec{x})|^2 =$ Wahrscheinlichkeit, das Teilchen am Ort \vec{x} zu messen
- Impuls: Wellenstruktur in der komplexen Phase
- Normierung: $\int_{\mathbb{R}^3} |\psi(\vec{x})|^2 d^3\vec{x} = 1$ („Irgendwo muß das Teilchen ja sein ...“)

5. $\frac{1+i}{\sqrt{2}} \cdot i$ Einführung in die Quantenmechanik

Quantenmechanik:

Zustand eines Teilchens:

Masse m , komplexwertige Wellenfunktion $\psi(\vec{x})$

- Ort: $|\psi(\vec{x})|^2 =$ Wahrscheinlichkeit, das Teilchen am Ort \vec{x} zu messen
- Impuls: Wellenstruktur in der komplexen Phase
- Normierung: $\int_{\mathbb{R}^3} |\psi(\vec{x})|^2 d^3\vec{x} = 1$ („Irgendwo muß das Teilchen ja sein ...“)

Einfachster Fall: Es gibt überhaupt nur 2 Orte.

$$\psi(\vec{x}) = \begin{pmatrix} \psi_1 \\ \psi_0 \end{pmatrix} \quad |\psi_0|^2 + |\psi_1|^2 = 1$$

Schreibweise: Basisvektoren $|0\rangle$ und $|1\rangle$ – „Basiszustände“

$$\psi(\vec{x}) = \psi_0 |0\rangle + \psi_1 |1\rangle$$

5. $\frac{1+i}{\sqrt{2}} \cdot 2i$ Qubits

- Ein Bit kann die (klassischen) Zustände 0 und 1 annehmen.

→ 2 Zustände, beschrieben durch eine ganze Zahl, die 0 oder 1 sein darf



oder



- Ein Qubit kann die quantenmechanischen Basiszustände $|0\rangle$ und $|1\rangle$ annehmen („1 Teilchen, 2 Orte“).

→ unendlich viele Zustände, beschrieben durch zwei komplexe Zahlen ψ_0 und ψ_1 mit $|\psi_0|^2 + |\psi_1|^2 = 1$

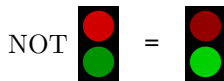
$$\begin{array}{|c|} \hline \text{red circle} \\ \hline \text{green circle} \\ \hline \end{array} = \sqrt{\frac{1}{3}} \begin{array}{|c|} \hline \\ \hline \text{green circle} \\ \hline \end{array} + \sqrt{\frac{2}{3}} \begin{array}{|c|} \hline \text{red circle} \\ \hline \\ \hline \end{array}$$

- Messung: Das Qubit muß sich für $|0\rangle$ oder $|1\rangle$ entscheiden. Die Wahrscheinlichkeit beträgt $|\psi_0|^2$ bzw. $|\psi_1|^2$.

5. $\frac{1+i}{\sqrt{2}}$. $2i$ Qubits

- Eine NOT-Operation auf einem Qubit vertauscht $|0\rangle$ und $|1\rangle$.

$$\text{NOT} \begin{pmatrix} \psi_1 \\ \psi_0 \end{pmatrix} = \begin{pmatrix} \psi_0 \\ \psi_1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \psi_1 \\ \psi_0 \end{pmatrix}$$



5. $\frac{1+i}{\sqrt{2}} \cdot 2i$ Qubits

- Eine NOT-Operation auf einem Qubit vertauscht $|0\rangle$ und $|1\rangle$.

$$\text{NOT} \begin{pmatrix} \psi_1 \\ \psi_0 \end{pmatrix} = \begin{pmatrix} \psi_0 \\ \psi_1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \psi_1 \\ \psi_0 \end{pmatrix}$$

$$\text{NOT} \begin{array}{|c|} \hline \text{red} \\ \hline \text{green} \\ \hline \end{array} = \begin{array}{|c|} \hline \text{red} \\ \hline \text{green} \\ \hline \end{array}$$

- Eine $\sqrt{\text{NOT}}$ -Operation auf einem Qubit vertauscht $|0\rangle$ und $|1\rangle$ halb.

$$\sqrt{\text{NOT}} = \frac{1}{\sqrt{2i}} \begin{pmatrix} i & 1 \\ 1 & i \end{pmatrix}, \quad \text{weil} \quad \begin{pmatrix} i & 1 \\ 1 & i \end{pmatrix} \begin{pmatrix} i & 1 \\ 1 & i \end{pmatrix} = \begin{pmatrix} 0 & 2i \\ 2i & 0 \end{pmatrix}$$

$$\sqrt{\text{NOT}} \begin{array}{|c|} \hline \text{red} \\ \hline \text{green} \\ \hline \end{array} = \begin{array}{|c|} \hline \text{red} \\ \hline \text{green} \\ \hline \end{array} \quad \sqrt{\text{NOT}} \begin{array}{|c|} \hline \text{red} \\ \hline \text{green} \\ \hline \end{array} = \begin{array}{|c|} \hline \text{red} \\ \hline \text{green} \\ \hline \end{array}$$

(Die Information, was wohin wandert, steckt in der komplexen Phase.)

5. $\frac{1+i}{\sqrt{2}}$.3i Quantenverschränkung

2 Qubits: $|0_0\rangle, |1_0\rangle$ und $|0_1\rangle, |1_1\rangle$

Wellenfunktion: $\psi_{00} |0_0\rangle + \psi_{10} |1_0\rangle + \psi_{01} |0_1\rangle + \psi_{11} |1_1\rangle$

5. $\frac{1+i}{\sqrt{2}}$.3i Quantenverschränkung

2 Qubits: $|0_0\rangle, |1_0\rangle$ und $|0_1\rangle, |1_1\rangle$

Wellenfunktion: ~~$\psi_{00} |0_0\rangle + \psi_{10} |1_0\rangle + \psi_{01} |0_1\rangle + \psi_{11} |1_1\rangle$~~

→ $\psi_{00} |00\rangle + \psi_{01} |01\rangle + \psi_{10} |10\rangle + \psi_{11} |11\rangle$

Das hinzugekommene Qubit (Nr. 1)

kann die Basiszustände des ersten Qubits (Nr. 0) mitbenutzen.



können sich vermischen, z. B. zu



5. $\frac{1+i}{\sqrt{2}} \cdot 3i$ Quantenverschränkung

2 Qubits: $|0_0\rangle, |1_0\rangle$ und $|0_1\rangle, |1_1\rangle$

Wellenfunktion: ~~$\psi_{00} |0_0\rangle + \psi_{10} |1_0\rangle + \psi_{01} |0_1\rangle + \psi_{11} |1_1\rangle$~~

→ $\psi_{00} |00\rangle + \psi_{01} |01\rangle + \psi_{10} |10\rangle + \psi_{11} |11\rangle$

Das hinzugekommene Qubit (Nr. 1)

kann die Basiszustände des ersten Qubits (Nr. 0) mitbenutzen.

→ Mit jedem hinzukommenden Qubit verdoppelt sich die Anzahl der Basiszustände.

Beispiel: Ein „Quanten-Byte“ (8 Qubits) hat 256 Basiszustände:

$$|00000000\rangle, |00000001\rangle, |00000010\rangle, \dots |11111111\rangle$$

Der Gesamtzustand eines Quanten-Bytes entspricht

$2^8 = 256$ komplexen Zahlen: $\psi_{00000000}$ bis $\psi_{11111111}$

5. $\frac{1+i}{\sqrt{2}}$.3i Quantenverschränkung

2 Qubits: $|0_0\rangle, |1_0\rangle$ und $|0_1\rangle, |1_1\rangle$

Wellenfunktion: ~~$\psi_{00} |0_0\rangle + \psi_{10} |1_0\rangle + \psi_{01} |0_1\rangle + \psi_{11} |1_1\rangle$~~

→ $\psi_{00} |00\rangle + \psi_{01} |01\rangle + \psi_{10} |10\rangle + \psi_{11} |11\rangle$

Das hinzugekommene Qubit (Nr. 1)

kann die Basiszustände des ersten Qubits (Nr. 0) mitbenutzen.

→ Mit jedem hinzukommenden Qubit verdoppelt sich die Anzahl der Basiszustände.

Beispiel: Ein „Quanten-Byte“ (8 Qubits) hat 256 Basiszustände:

$$|00000000\rangle, |00000001\rangle, |00000010\rangle, \dots |11111111\rangle$$

Der Gesamtzustand eines Quanten-Bytes entspricht

$2^8 = 256$ komplexen Zahlen: $\psi_{00000000}$ bis $\psi_{11111111}$

Der Gesamtzustand eines 64-Bit-Quanten-Registers entspricht

$2^{64} = 18\,446\,744\,073\,709\,551\,616$ komplexen Zahlen.

→ Es ist schon aus Speicherplatzgründen nicht möglich, auf einem klassischen Computer effizient mit Qubits zu rechnen.

5. $\frac{1+i}{\sqrt{2}}$.3i Quantenverschränkung

- CNOT („Controlled NOT“) – eine Art **if**-Anweisung auf 2 Qubits

if (q0)

q1 = ~q1;


$$\text{CNOT} \begin{pmatrix} \psi_{00} \\ \psi_{01} \\ \psi_{10} \\ \psi_{11} \end{pmatrix} = \begin{pmatrix} \psi_{00} \\ \psi_{01} \\ \psi_{11} \\ \psi_{10} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \psi_{00} \\ \psi_{01} \\ \psi_{10} \\ \psi_{11} \end{pmatrix}$$

5. $\frac{1+i}{\sqrt{2}}$.3i Quantenverschränkung

- CNOT („Controlled NOT“) – eine Art **if**-Anweisung auf 2 Qubits

if (q0)

q1 = ~q1;


$$\text{CNOT} \begin{pmatrix} \psi_{00} \\ \psi_{01} \\ \psi_{10} \\ \psi_{11} \end{pmatrix} = \begin{pmatrix} \psi_{00} \\ \psi_{01} \\ \psi_{11} \\ \psi_{10} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \psi_{00} \\ \psi_{01} \\ \psi_{10} \\ \psi_{11} \end{pmatrix}$$

else-Zweig

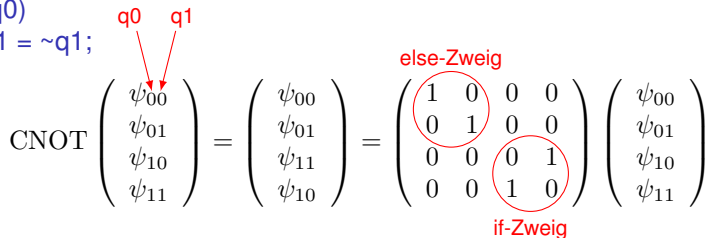
if-Zweig

5. $\frac{1+i}{\sqrt{2}}$.3i Quantenverschränkung

- CNOT („Controlled NOT“) – eine Art **if**-Anweisung auf 2 Qubits

if (q0)

q1 = ~q1;



The diagram shows the CNOT gate matrix as a product of two matrices. Red arrows labeled 'q0' and 'q1' point to the first and second rows of the first matrix, respectively. The second matrix is annotated with 'else-Zweig' above the top-left 2x2 block and 'if-Zweig' below the bottom-right 2x2 block, both circled in red.

$$\text{CNOT} \begin{pmatrix} \psi_{00} \\ \psi_{01} \\ \psi_{10} \\ \psi_{11} \end{pmatrix} = \begin{pmatrix} \psi_{00} \\ \psi_{01} \\ \psi_{11} \\ \psi_{10} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \psi_{00} \\ \psi_{01} \\ \psi_{10} \\ \psi_{11} \end{pmatrix}$$

Dies funktioniert auch dann, wenn die „**if**-Bedingung“ q0 weder $|0\rangle$ („false“) noch $|1\rangle$ („true“) ist, sondern eine Mischung.

Der **if**-Zweig und der **else**-Zweig werden dann gleichzeitig ausgeführt. („Mischung“ heißt also nicht „vielleicht“, sondern „beides“).

5. $\frac{1+i}{\sqrt{2}}$.3i Quantenverschränkung

- CNOT („Controlled NOT“) – eine Art **if**-Anweisung auf 2 Qubits

if (q0)

q1 = ~q1;

$$\text{CNOT} \begin{pmatrix} \psi_{00} \\ \psi_{01} \\ \psi_{10} \\ \psi_{11} \end{pmatrix} = \begin{pmatrix} \psi_{00} \\ \psi_{01} \\ \psi_{11} \\ \psi_{10} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \psi_{00} \\ \psi_{01} \\ \psi_{10} \\ \psi_{11} \end{pmatrix}$$

else-Zweig

if-Zweig

Dies funktioniert auch dann, wenn die „**if**-Bedingung“ q0 weder $|0\rangle$ („false“) noch $|1\rangle$ („true“) ist, sondern eine Mischung.

Der **if**-Zweig und der **else**-Zweig werden dann gleichzeitig ausgeführt. („Mischung“ heißt also nicht „vielleicht“, sondern „beides“).

→ Damit können wir nun alles berechnen, was wir wollen.
(CNOT zusammen mit den 1-Bit-Operationen ist „universell“.)

5. $\frac{1+i}{\sqrt{2}}$.4i Der Shor-Algorithmus

- Ziel: Zerlegung einer Zahl $n = p \cdot q$ in ihre Faktoren
- Herangehensweise:
 1. Wähle x mit $1 < x < n$.
 2. Suche r mit $x^r \% n = 1$.
 3. Wenn r ungerade oder wenn $x^{\frac{r}{2}} \% n = n - 1$, verwirf dieses x . Neuer Versuch.
 4. Ansonsten ist der größte gemeinsame Teiler von $x^{\frac{r}{2}}$ und n ein Faktor von n . :-)

5. $\frac{1+i}{\sqrt{2}}$.4i Der Shor-Algorithmus

- Ziel: Zerlegung einer Zahl $n = p \cdot q$ in ihre Faktoren
- Herangehensweise:
 1. Wähle x mit $1 < x < n$.
 2. Suche r mit $x^r \% n = 1$. \longleftarrow **mit Quantencomputer**
 3. Wenn r ungerade oder wenn $x^{\frac{r}{2}} \% n = n - 1$, verwirf dieses x .
Neuer Versuch.
 4. Ansonsten ist der größte gemeinsame Teiler von $x^{\frac{r}{2}}$ und n
ein Faktor von n . \therefore)

5. $\frac{1+i}{\sqrt{2}}$.4i Der Shor-Algorithmus

- Ziel: Zerlegung einer Zahl $n = p \cdot q$ in ihre Faktoren
- Herangehensweise:
 1. Wähle x mit $1 < x < n$.
 2. Suche r mit $x^r \% n = 1$. \longleftarrow **mit Quantencomputer**
 3. Wenn r ungerade oder wenn $x^{\frac{r}{2}} \% n = n - 1$, verwirf dieses x . Neuer Versuch.
 4. Ansonsten ist der größte gemeinsame Teiler von $x^{\frac{r}{2}}$ und n ein Faktor von n . \therefore)
- Algorithmus, um r zu suchen:
 1. Sei q die nötige Anzahl von Bits, um n^2 speichern zu können.
 2. Lade ein q -Qubit-Register mit dem Wert:

$$A := \underbrace{\frac{|0\rangle + |1\rangle}{\sqrt{2}}, \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \dots, \frac{|0\rangle + |1\rangle}{\sqrt{2}}}_{q \text{ Qubits}}$$

5. $\frac{1+i}{\sqrt{2}}$.4i Der Shor-Algorithmus

- Ziel: Zerlegung einer Zahl $n = p \cdot q$ in ihre Faktoren
- Herangehensweise:
 2. Suche r mit $x^r \% n = 1$. \longleftarrow **mit Quantencomputer**
- Algorithmus, um r zu suchen:
 1. Sei q die nötige Anzahl von Bits, um n^2 speichern zu können.
 2. Lade ein q -Qubit-Register mit dem Wert:

$$A := \underbrace{\frac{|0\rangle + |1\rangle}{\sqrt{2}}, \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \dots, \frac{|0\rangle + |1\rangle}{\sqrt{2}}}_{q \text{ Qubits}}$$

3. Berechne in einem zweiten q -Qubit-Register den Wert $x^A \% n$.
Damit berechnen wir gewissermaßen gleichzeitig $x^a \% n$
für alle Werte, die a annehmen kann (0 bis $2^q - 1$).
Die beiden Register sind nun verschränkt.

5. $\frac{1+i}{\sqrt{2}}$.4i Der Shor-Algorithmus

- Ziel: Zerlegung einer Zahl $n = p \cdot q$ in ihre Faktoren
- Herangehensweise:
 2. Suche r mit $x^r \% n = 1$. \longleftarrow **mit Quantencomputer**
- Algorithmus, um r zu suchen:
 2. Lade ein q -Qubit-Register mit dem Wert:

$$A := \underbrace{\frac{|0\rangle + |1\rangle}{\sqrt{2}}, \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \dots, \frac{|0\rangle + |1\rangle}{\sqrt{2}}}_{q \text{ Qubits}}$$

3. Berechne in einem zweiten q -Qubit-Register den Wert $x^A \% n$.
Damit berechnen wir gewissermaßen gleichzeitig $x^a \% n$
für alle Werte, die a annehmen kann (0 bis $2^q - 1$).
Die beiden Register sind nun verschränkt.
4. Quanten-Fouriertransformation auf dem ersten Register.
Wegen der Verschränkung beeinflusst dies auch das zweite Register.

5. $\frac{1+i}{\sqrt{2}}$.4i Der Shor-Algorithmus

- Ziel: Zerlegung einer Zahl $n = p \cdot q$ in ihre Faktoren
- Herangehensweise:
 2. Suche r mit $x^r \% n = 1$. \longleftarrow **mit Quantencomputer**
- Algorithmus, um r zu suchen:
 3. Berechne in einem zweiten q -Qubit-Register den Wert $x^A \% n$.
Damit berechnen wir gewissermaßen gleichzeitig $x^a \% n$
für alle Werte, die a annehmen kann (0 bis $2^q - 1$).
Die beiden Register sind nun verschränkt.
 4. Quanten-Fouriertransformation auf dem ersten Register.
Wegen der Verschränkung beeinflusst dies auch das zweite Register.
 5. Messung.
Alle Qubits müssen sich für $|0\rangle$ oder $|1\rangle$ entscheiden.
Den Meßwert des zweiten Registers nennen wir r .
Es gilt mit hoher Wahrscheinlichkeit: $x^r \% n = 1$.
 6. Fertig. :-)

5. $\frac{1+i}{\sqrt{2}}$ Quantencomputer

5. $\frac{1+i}{\sqrt{2}}$.5i Fazit

- Ein funktionsfähiger Quantencomputer mit mindestens 2048 Qubits würde aktuelle Verschlüsselungsverfahren unwirksam machen.
 - Betroffen:
 - vertrauliche Kommunikation (z. B. Online-Banking)
 - Fernzugriff auf Rechner, Schutz persönlicher Daten
 - digitale Rechtebeschränkung („Kopierschutz“)
 - digitale Währungen
 - ...
- Chaos

5. $\frac{1+i}{\sqrt{2}}$ Quantencomputer

5. $\frac{1+i}{\sqrt{2}}$.5i Fazit

- Ein funktionsfähiger Quantencomputer mit mindestens 2048 Qubits würde aktuelle Verschlüsselungsverfahren unwirksam machen.
→ Chaos
- Chaos haben wir bereits.
 - Die Allermeisten sind mit persönlichen Daten sehr freigiebig. (z. B. Datenverarbeitung und -speicherung in der Cloud)
 - sozialer und politischer Druck → Sich entziehen wird schwieriger. (z. B. Gesundheitskarte, bargeldlose Geschäfte)
 - Daten bei Firmen und Behörden sind leicht angreifbar.
 - Politische Maßnahmen bewirken oft ihr Gegenteil. (z. B. DSGVO → mehr Bürokratie → weniger Anbieter)
- Gegenmaßnahme: Aufklärung
Bewußtsein für Datenschutz fördern

5. $\frac{1+i}{\sqrt{2}}$ Quantencomputer

5. $\frac{1+i}{\sqrt{2}}$.5i Fazit

- Ein funktionsfähiger Quantencomputer mit mindestens 2048 Qubits würde aktuelle Verschlüsselungsverfahren unwirksam machen.
→ Chaos
- Chaos haben wir bereits.
→ Aufklärung, Bewußtsein für Datenschutz fördern
- demnächst in „Eingebettete Systeme“: Verschlüsselung
Termin noch wählbar

5. $\frac{1+i}{\sqrt{2}}$ Quantencomputer

5. $\frac{1+i}{\sqrt{2}}$.5i Fazit

- Ein funktionsfähiger Quantencomputer mit mindestens 2048 Qubits würde aktuelle Verschlüsselungsverfahren unwirksam machen.
→ Chaos
- Chaos haben wir bereits.
→ Aufklärung, Bewußtsein für Datenschutz fördern
- demnächst in „Eingebettete Systeme“: Verschlüsselung Termin noch wählbar

Vielen Dank für Ihre Aufmerksamkeit!