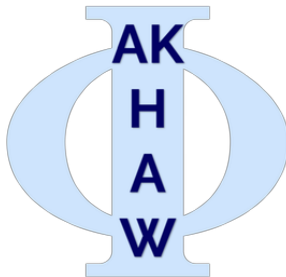


QUANTENCOMPUTING - WAS IST DAS?

Quantentechnologie – von der Lehre zur Anwendung
September 2025

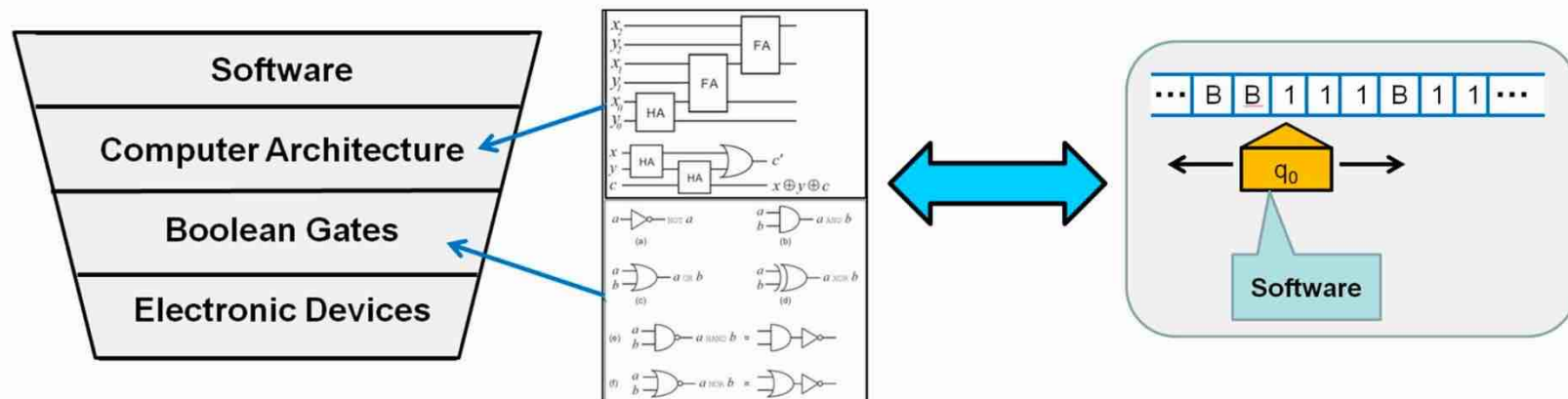


Jörg Hettel
Hochschule Kaiserslautern



Turing Maschine als theoretisches Model für „Berechenbarkeit“.

- Beruht auf der **klassischen Mechanik**
 - Klassisches Bit als Informationsträger
- Typische Fragestellungen
 - Welche Probleme können berechnet werden (Berechenbarkeit)?
 - Wie lange dauert eine Berechnung (Komplexitätstheorie) ?



Beispiel: Suche in einer unstrukturierten Menge



Zwei zentrale Fragestellungen:

- Gibt es einen Algorithmus, der das „schwarze Schaaf“ findet?
 - Ist das Problem algorithmisch lösbar?
- Falls ja, wie effizient ist der Algorithmus?
 - Wie verhält sich die Laufzeit, wenn man die Menge z.B. verdoppelt?



Quanten Turing Maschine als Rechenmodell.

- Qubit als Informationsträger

classical bit	quantum bit (qubit)
a bit has always a unique value	before reading a qubit it have in general no defined state
a bit can only be 0 or 1	a qubit can be in a superposition between 0 and 1 (a 2-dim value)
a bit can exactly be copied and as often as desired	a qubit can generally not be copied
a bit can be read as often as desired without changing it	reading a qubit corresponds to a projection into one of the eigenstates of the measurement operator
reading or writing a bit has no effect on other bits in the register	qubits in a register can be entangled. Reading or writing a qubit influences entangled qubits

Beispiel: Suche in einer unstrukturierten Menge



Komplexität (Laufzeit in Abhängigkeit der Anzahl)

- Klassischer Algorithmus: $O(n)$
- Quanten-Algorithmus: $O(\sqrt{n})$

Beispiel: Faktorisierung

N (hier eine Zahl mit 330 Bit Länge) ist das Produkt von zwei Primzahlen.

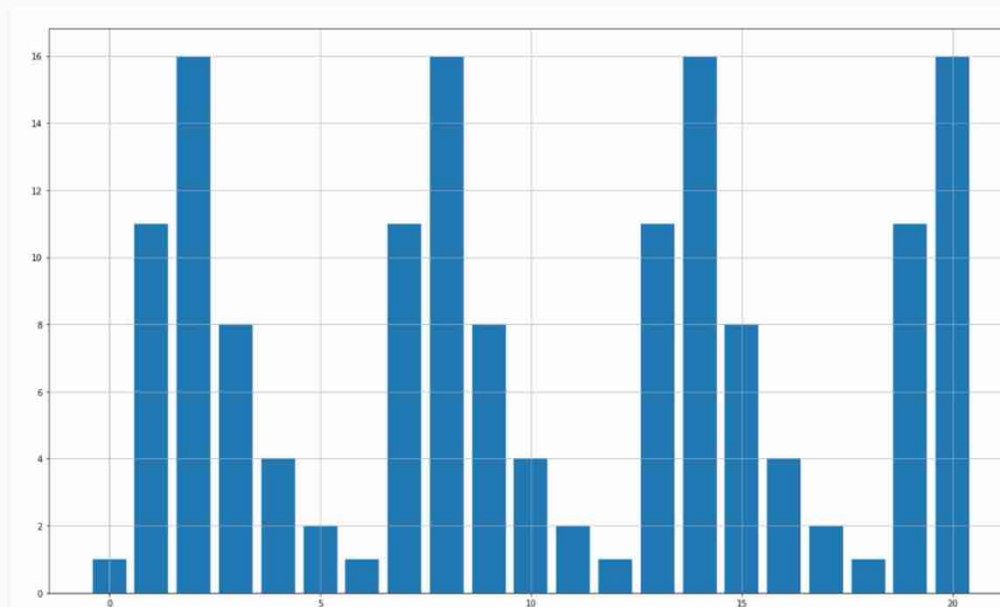
Wie lauten die beiden Primzahlen?

$N = 15226050279225333605356183781326374297180681149613$
 $80688657908494580122963258952897654000350692006139$

Beispiel: Faktorisierung

Ein Quantencomputer kann die Primfaktoren sehr effizient bestimmen.

$$f(x) = 11^x \pmod{3 \cdot 7}$$



Wenn die Periode bekannt ist,
kann man das Modul faktorisieren

$$\begin{aligned} 11^6 \pmod{21} &= 1 \\ 11^6 - 1 \pmod{21} &= 0 \\ (11^3 - 1)(11^3 + 1) \pmod{21} &= 0 \end{aligned}$$

Entweder ist $(11^3 - 1)$ oder
 $(11^3 + 1)$ ein Teiler von 21

$$\begin{aligned} (11^3 - 1) \pmod{21} &= 7 \\ (11^3 + 1) \pmod{21} &= 9 \end{aligned}$$

Beispiel: Faktorisierung

Primfaktoren von $N = p \cdot q$

$N =$ 15226050279225333605356183781326374297180681149613
80688657908494580122963258952897654000350692006139

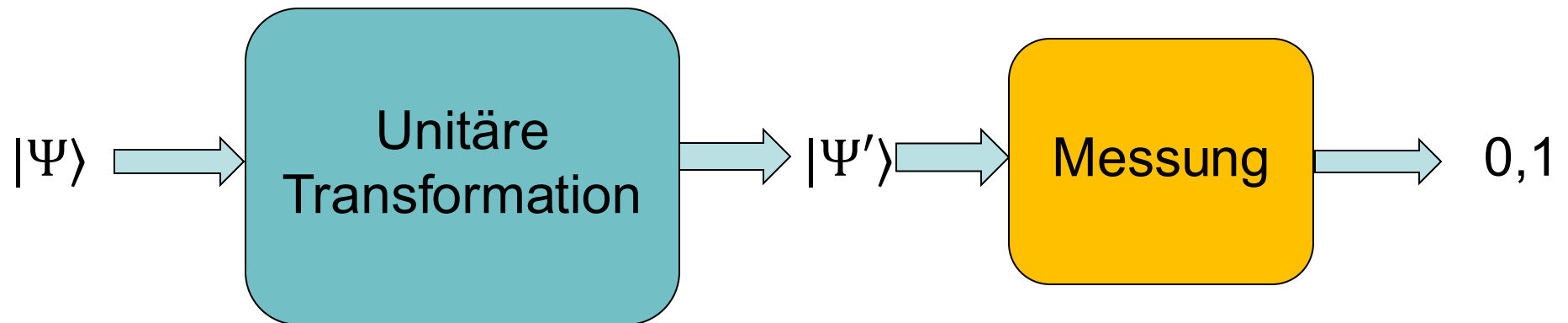
lauten

$p =$ 37975227936943673922808872755445627854565536638199
 $q =$ 40094690950920881030683735292761468389214899724061



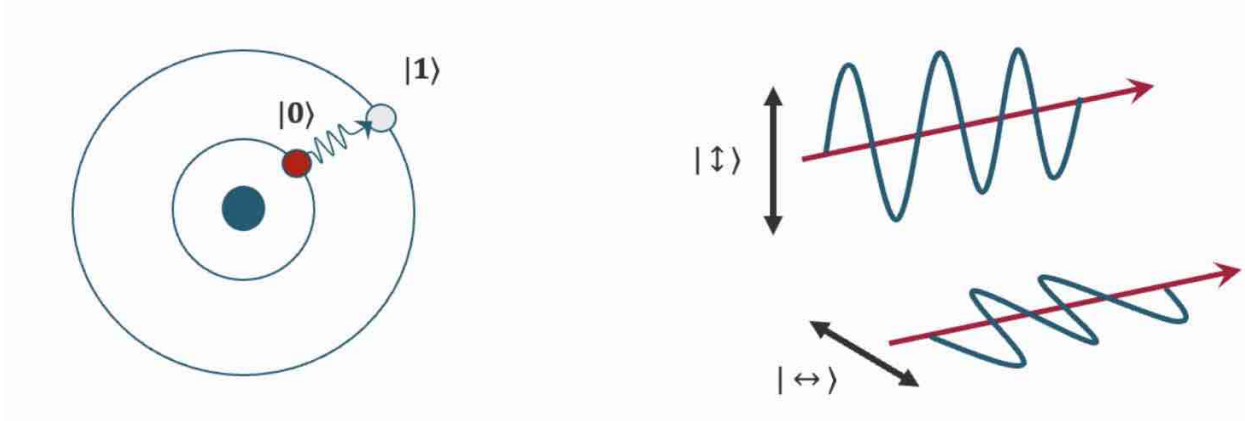
WIE FUNKTIONIERT DAS RECHNEN MIT QUBITS?

Kontrollierte Manipulation eines Quantensystems



Qubit als Informationsträger

Qubit ist ein Quantensystem mit zwei dedizierten Zuständen

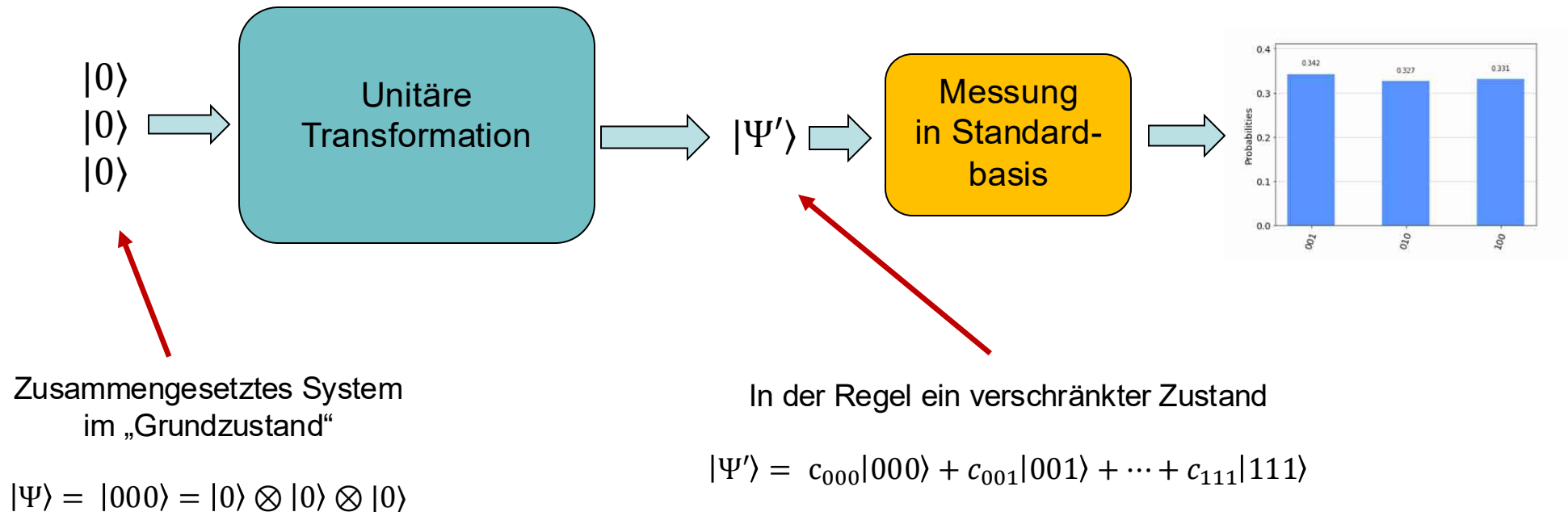


$$|\Psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha |0\rangle + \beta |1\rangle$$

$$\alpha, \beta \in \mathbb{C} \text{ and } |\alpha|^2 + |\beta|^2 = 1.$$

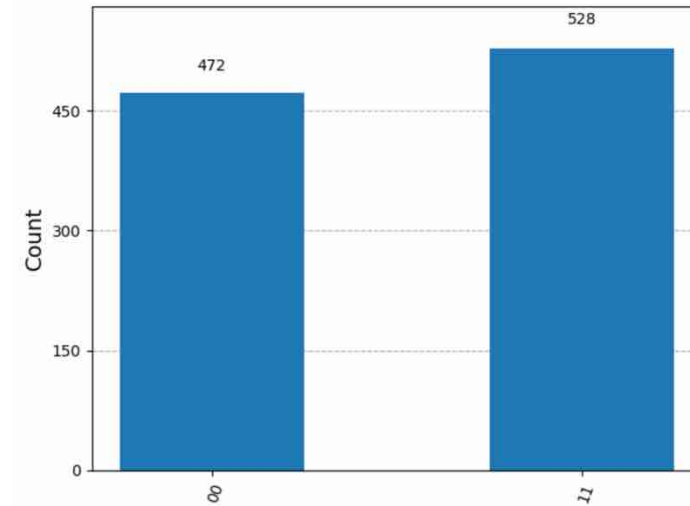
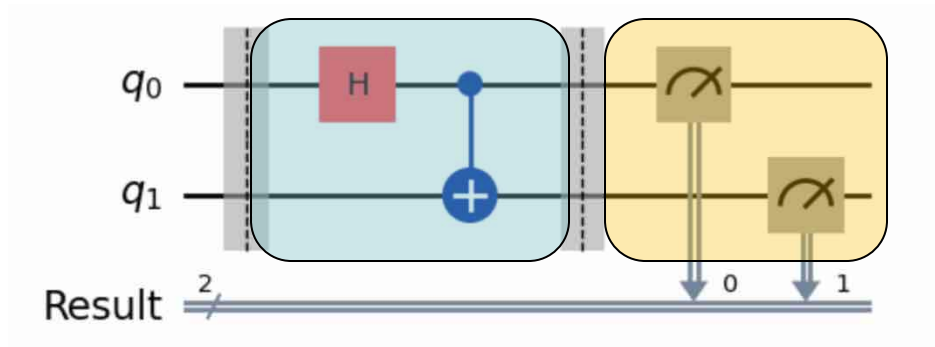
Grundprinzip

Kontrollierte Manipulation eines Quantensystems bestehend aus Qubits



Beispiel: Erzeugung eines EPR-Zustands

Simulation und Messung von 1000 Durchläufen



```

from qiskit.circuit import QuantumRegister, ClassicalRegister, QuantumCircuit

qr = QuantumRegister(2,"q")
cr = ClassicalRegister(2, "Result")

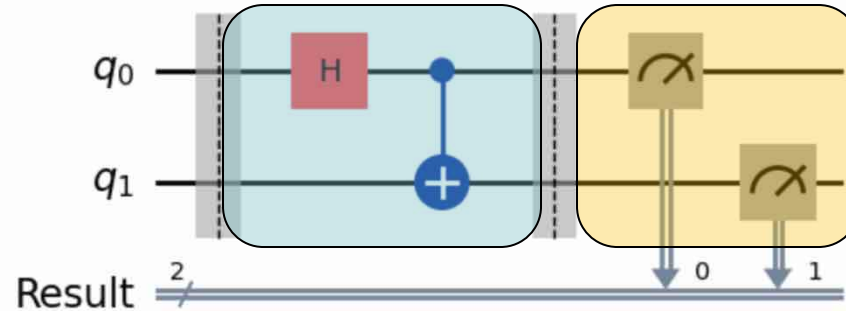
qc = QuantumCircuit(qr,cr)

qc.barrier()
qc.h(0)
qc.cx(0,1)
qc.barrier()

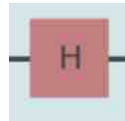
qc.measure(qr,cr)
qc.draw("mpl")

```

Wirkung der Gatter auf ein Quantensystem



Wirkung von



$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Notationsreihenfolge:

$$|q_1 q_0\rangle$$

Wirkung von



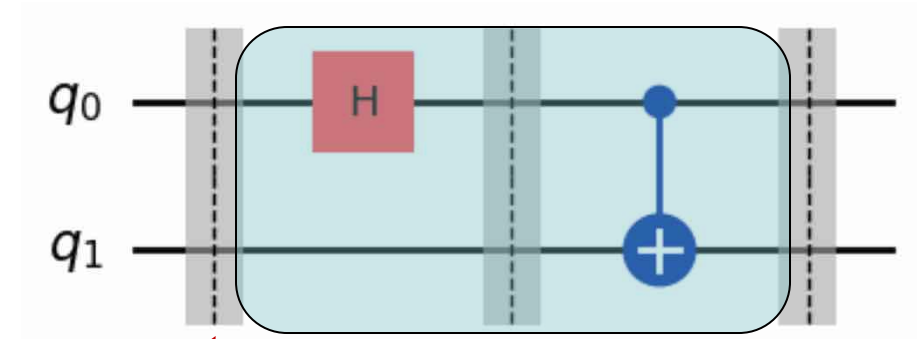
$$|0\rangle|0\rangle \mapsto |0\rangle|0\rangle$$

$$|0\rangle|1\rangle \mapsto |1\rangle|1\rangle$$

$$|1\rangle|0\rangle \mapsto |1\rangle|0\rangle$$

$$|1\rangle|1\rangle \mapsto |0\rangle|1\rangle$$

Analyse der unitären Transformation



```
qr = QuantumRegister(2, "q")
qc = QuantumCircuit(qr)

state_init = Statevector(qc)











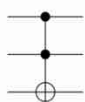
qc.barrier()
qc.h(0)
qc.barrier()
state_inter = Statevector(qc)
qc.cx(0, 1)
qc.barrier()

state_final = Statevector(qc)
qc.draw("mpl")
```

$$|00\rangle$$

$$\frac{\sqrt{2}}{2}|00\rangle + \frac{\sqrt{2}}{2}|01\rangle$$

$$\frac{\sqrt{2}}{2}|00\rangle + \frac{\sqrt{2}}{2}|11\rangle$$

Operator	Gate(s)	Matrix
Pauli-X (X)	 	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)		$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Phase (S, P)		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$ (T)		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
Controlled Not (CNOT, CX)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Controlled Z (CZ)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
SWAP		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Toffoli (CCNOT, CCX, TOFF)		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

Common quantum logic gates by name (including abbreviation), circuit form(s) and the corresponding unitary matrices

https://en.wikipedia.org/wiki/Quantum_logic_gate

Circuit Library

`qiskit.circuit.library`

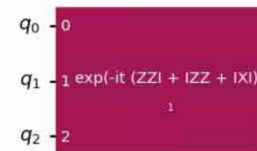
The circuit library is a collection of valuable circuits and building blocks. We call these valuable for different reasons. For instance, they can be used as building blocks for algorithms, serve as benchmarks, or they are circuits conjectured to be difficult to simulate classically.

Elements in the circuit library are either `QuantumCircuit`s or `Instruction`s, allowing them to be easily investigated or plugged into other circuits. This enables fast prototyping and circuit design at higher levels of abstraction.

For example:

```

1 from qiskit.circuit import QuantumCircuit
2 from qiskit.circuit.library import PauliEvolutionGate
3 from qiskit.quantum_info import SparsePauliOp
4
5 hamiltonian = SparsePauliOp(["ZZI", "IZZ", "IXI"], coeffs=[1, 1, -1])
6 gate = PauliEvolutionGate(hamiltonian)
7
8 circuit = QuantumCircuit(hamiltonian.num_qubits)
9 circuit.append(gate, circuit.qubits)
10
11 circuit.draw("mpl")
    
```



Komplexeres Beispiel: Faktorisierung von 15

```
# Order finding problem for N = 15 with a = 2
N = 15
a = 2

# Number of qubits
num_target = floor(log(N - 1, 2)) + 1 # for modular exponentiation operators
num_control = 2 * num_target # for enough precision of estimation

# List of M_b operators in order
k_list = range(num_control)
b_list = [a2kmodN(2, k, 15) for k in k_list]

# Initialize the circuit
control = QuantumRegister(num_control, name="C")
target = QuantumRegister(num_target, name="T")
output = ClassicalRegister(num_control, name="out")
circuit = QuantumCircuit(control, target, output)

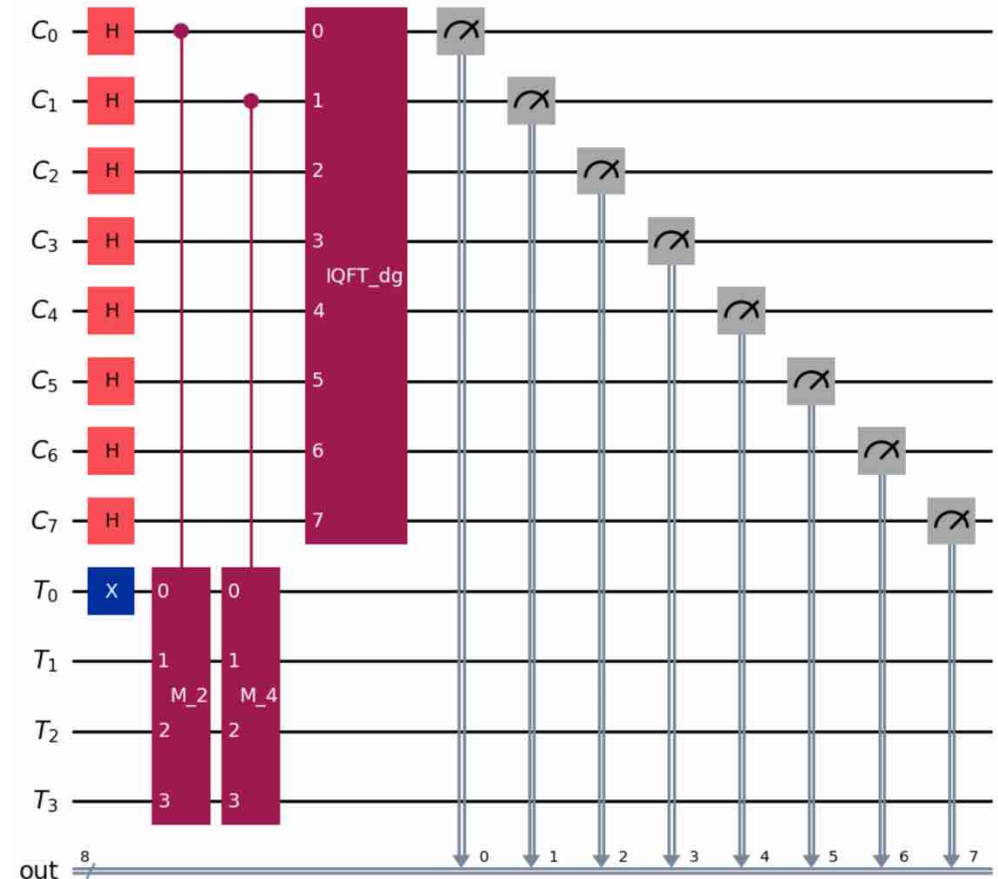
# Initialize the target register to the state |1>
circuit.x(num_control)

# Add the Hadamard gates and controlled versions of the
# multiplication gates
for k, qubit in enumerate(control):
    circuit.h(k)
    b = b_list[k]
    if b == 2:
        circuit.compose(
            M2mod15().control(), qubits=[qubit] + list(target), inplace=True
        )
    elif b == 4:
        circuit.compose(
            M4mod15().control(), qubits=[qubit] + list(target), inplace=True
        )
    else:
        continue # M1 is the identity operator

# Apply the inverse QFT to the control register
circuit.compose(QFT(num_control, inverse=True), qubits=control, inplace=True)

# Measure the control register
circuit.measure(control, output)

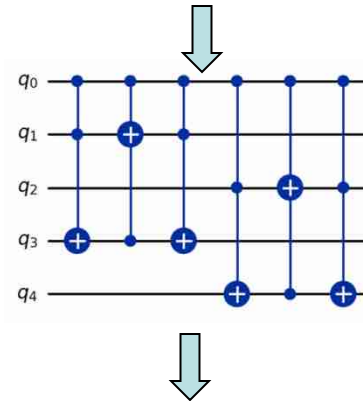
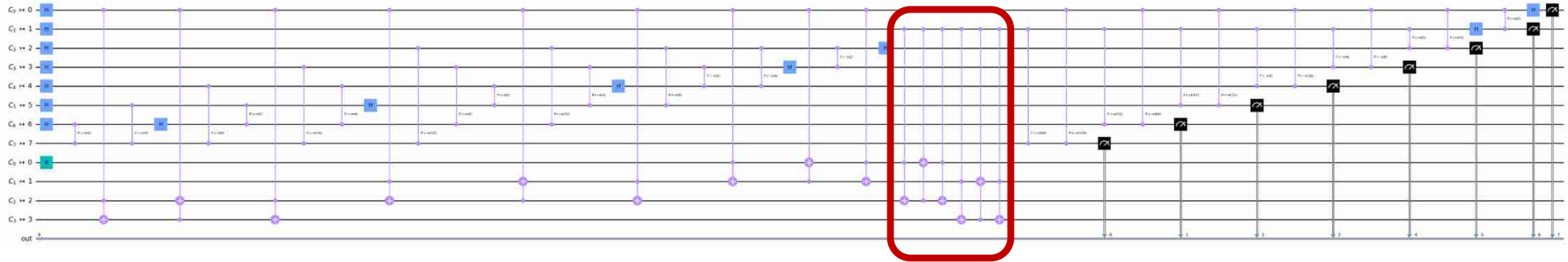
circuit.draw("mpl", fold=-1)
```



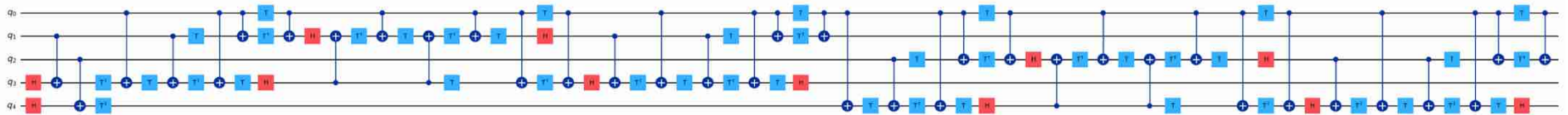
<https://quantum.cloud.ibm.com/docs/en/tutorials/shors-algorithm>

„Hardware-Realisierung“

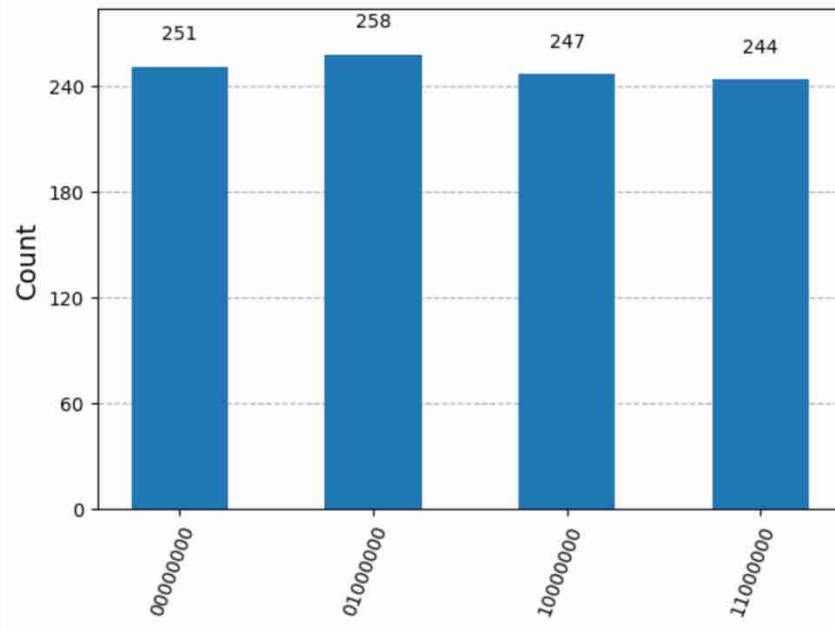
Operator counts: OrderedDict({'cp': 28, 'ccx': 15, 'h': 14, 'measure': 8, 'x': 1})



Operator counts: OrderedDict({'cx': 36, 't': 24, 'tdg': 18, 'h': 12})



Ergebnis der Simulation (1000 Durchläufe)



	Register Output	Phase	Guess for r
0	00000000(bin) = 0(dec)	0/256 = 0.00	1
1	01000000(bin) = 64(dec)	64/256 = 0.25	4
2	10000000(bin) = 128(dec)	128/256 = 0.50	2
3	11000000(bin) = 192(dec)	192/256 = 0.75	4

Phase: theta = 0.25

Order of 2 modulo 15 estimated as: r = 4

*** Non-trivial factor found: 3 ***

<https://quantum.cloud.ibm.com/docs/en/tutorials/shors-algorithm>

Neben dem hier gezeigten „Gatter-Modell“ gibt es noch

Einweg-Quantencomputer

- Hier wird zuerst ein universeller (also vom Problem unabhängiger) verschränkter Quantenzustand generiert (beispielsweise ein Clusterzustand) und die eigentliche Rechnung wird durch gezielte Messungen an diesem Zustand durchgeführt wird.

Adiabatische Quantencomputer

- Die Idee ist, dass man ein leicht zu präparierendes System in ein System überführt, an dessen Grundzustand man interessiert ist, und dessen Zustand dann gemessen. Wenn der Übergang langsam genug erfolgt ist, hat man so die Lösung des Problems.

Erwin Schrödinger (1887-1961)

"We never experiment with just one electron or atom or (small) molecule. In thought-experiments we sometimes assume that we do; this invariably entails ridiculous consequences"

Zitat aus: Schrödinger, E. *Brit. J. Phil. Sci* **3**, 233–247 (1952).

DeepL-Übersetzung:

"Wir experimentieren nie nur mit einem Elektron, einem Atom oder einem (kleinen) Molekül. In Gedankenexperimenten gehen wir manchmal davon aus, dass wir das tun, was immer lächerliche Konsequenzen nach sich zieht"



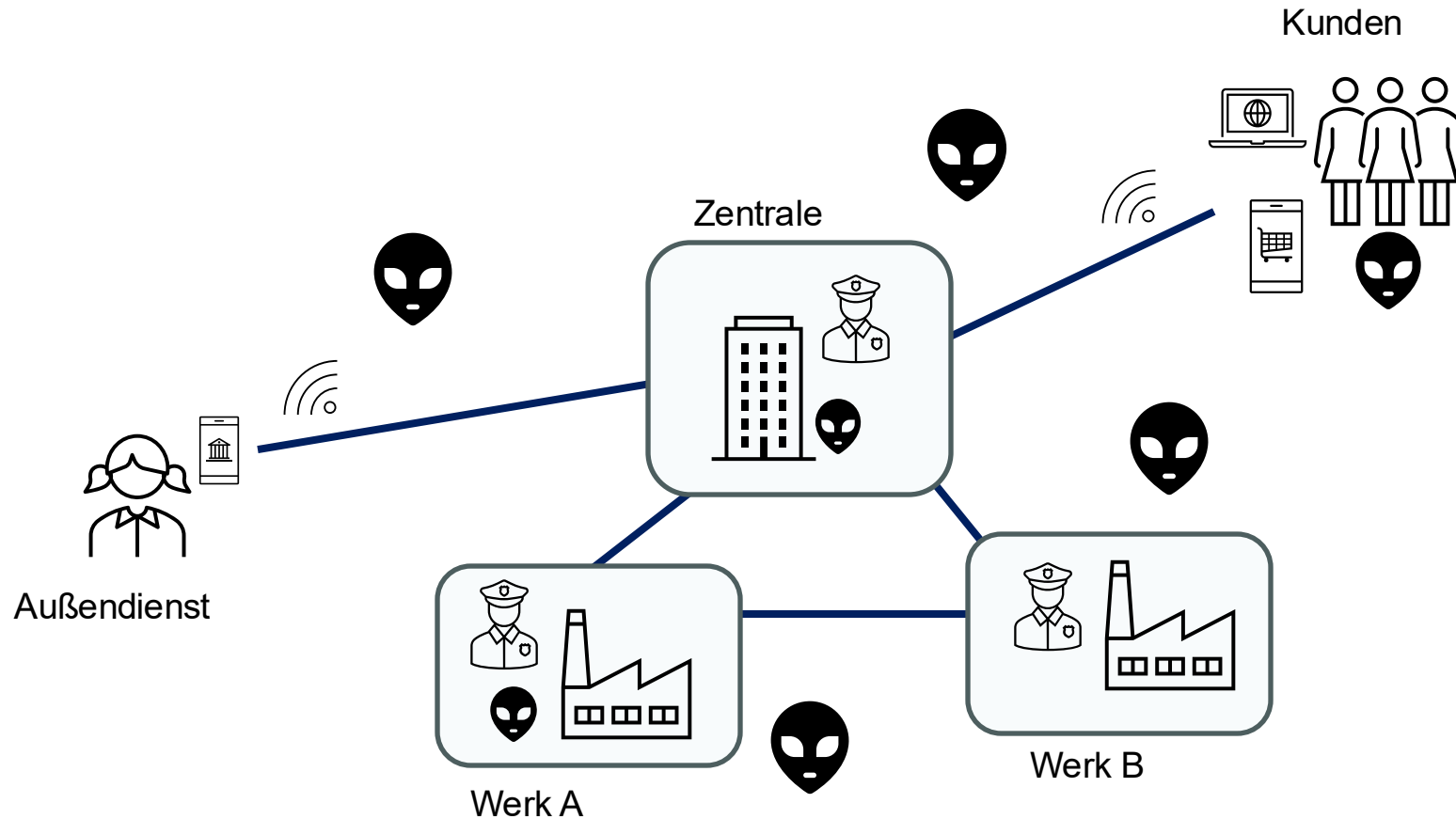
Abbildung aus Wikipedia



QUANTENKOMMUNIKATION UND -KRYPTOGRAPHIE

QUANTENBEDROHUNG

Unser Unternehmen und Bedrohungen

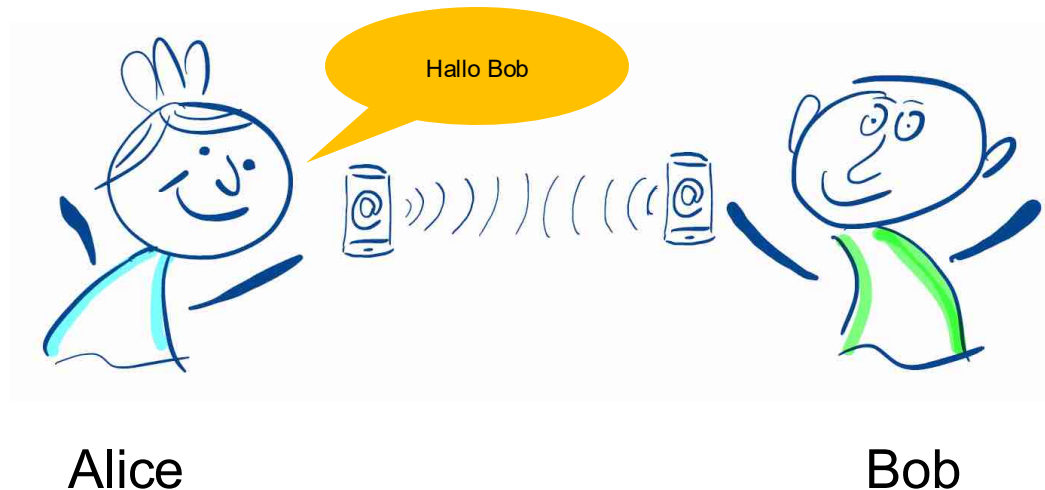


Ziele von Kryptographie

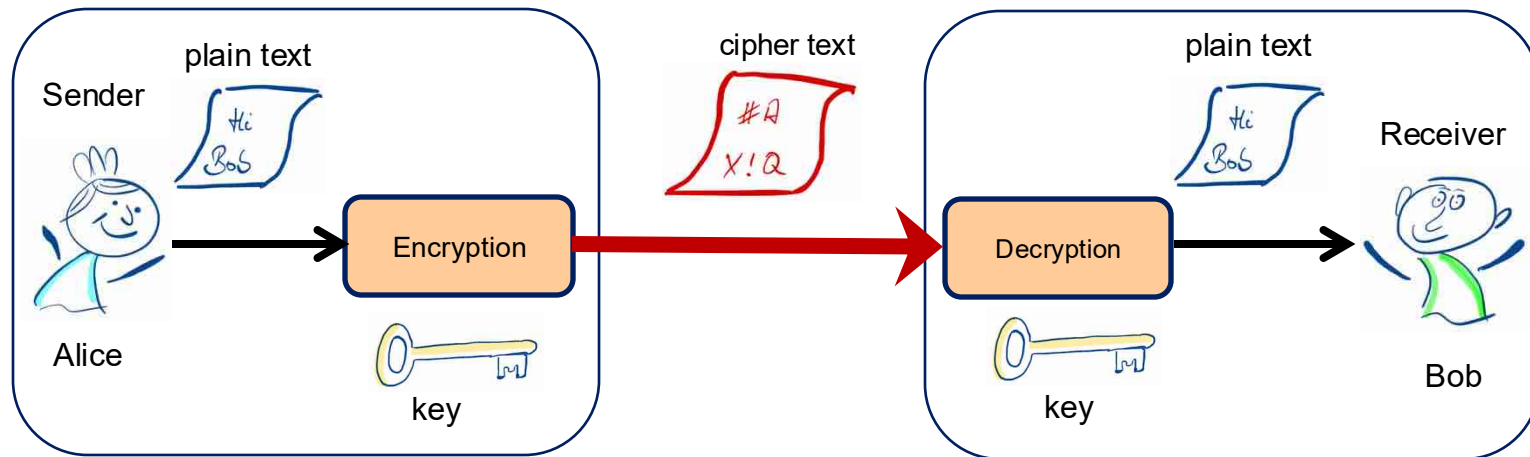
Datenschutz (Vertraulichkeit)

Datenauthenzizität (die Daten stammen vom Absender)

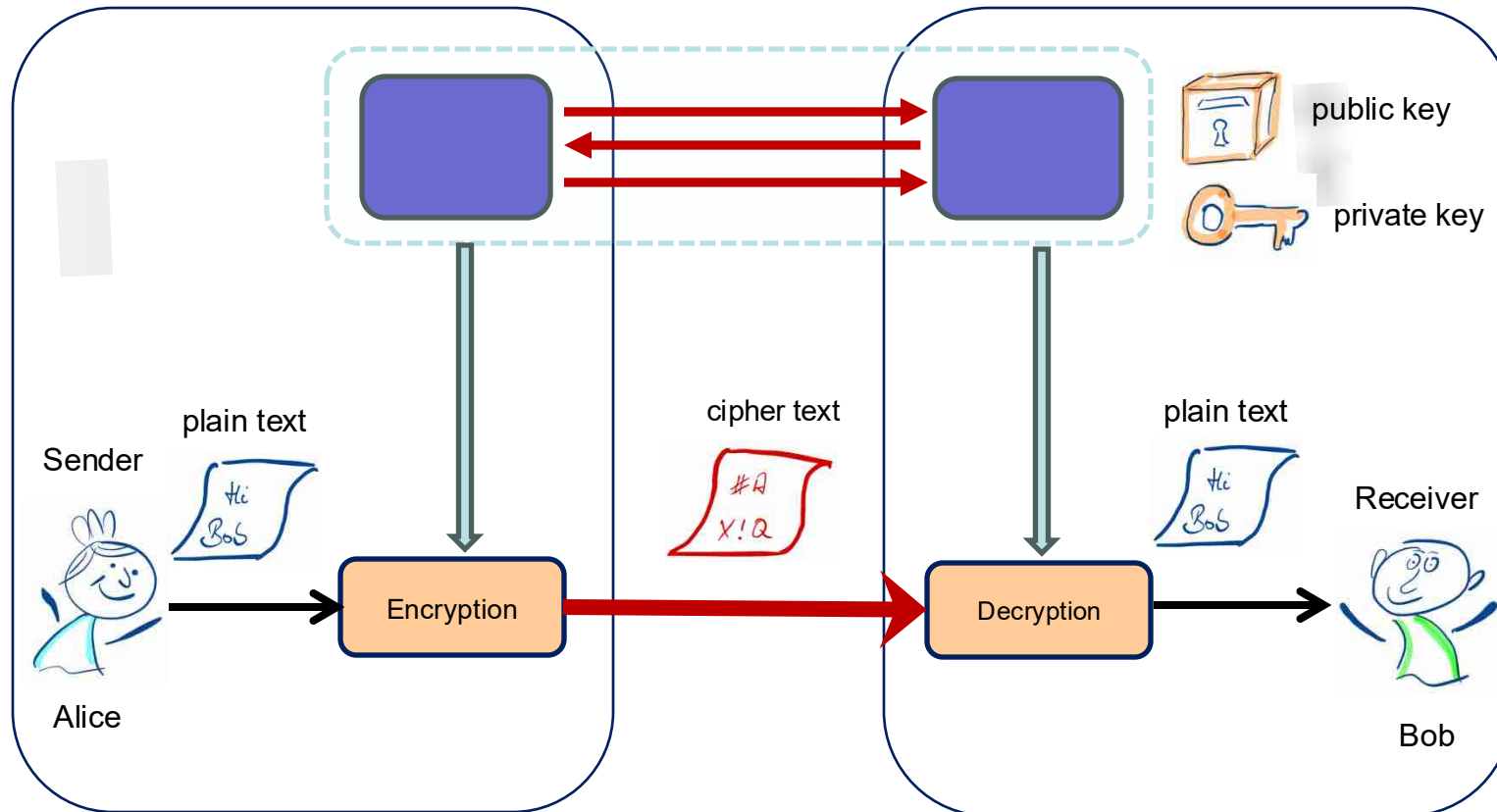
Datenintegrität (die Daten wurden unterwegs nicht verändert)



Kryptographisches Standardschema

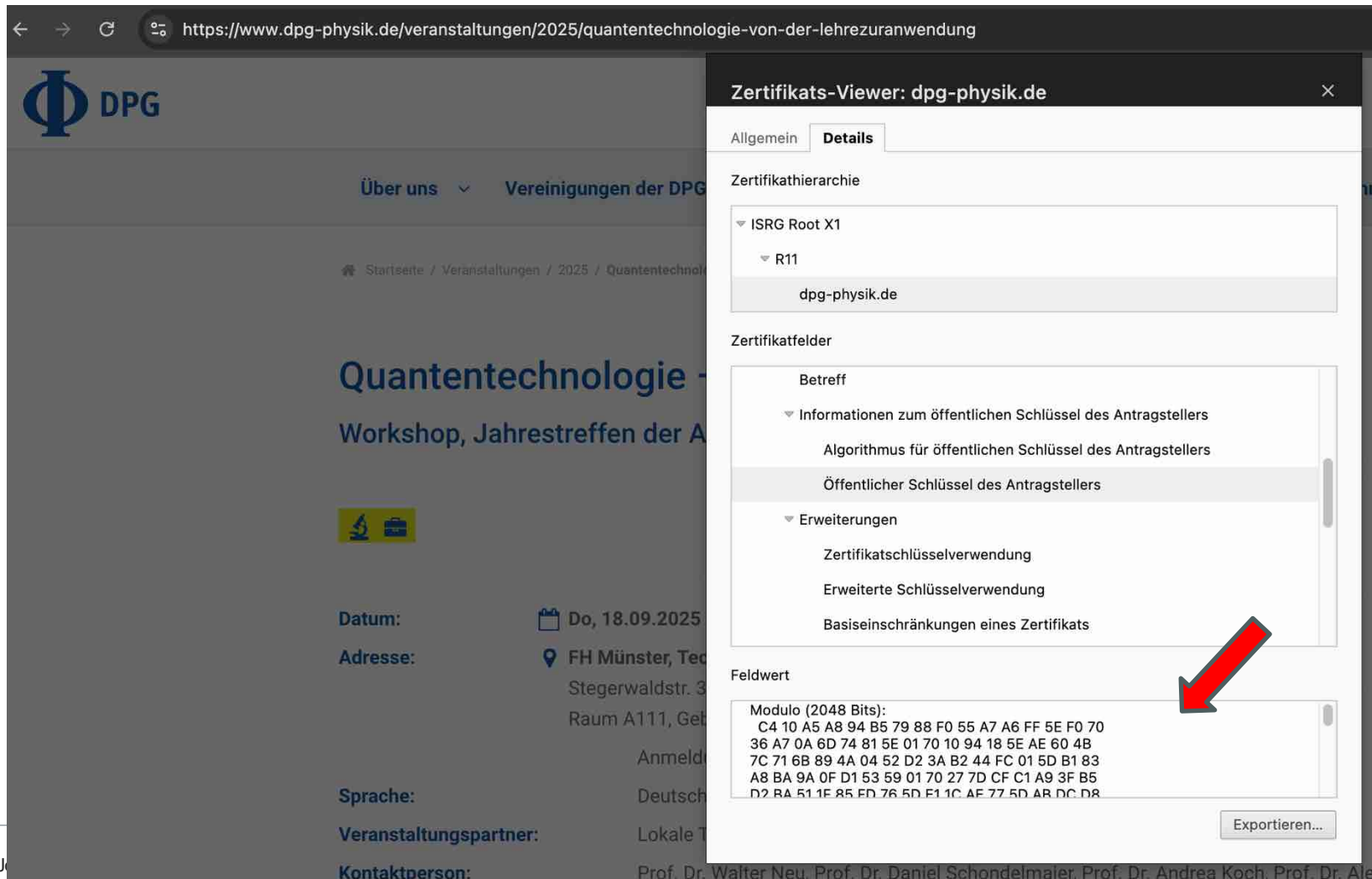


Public-Key Verfahren



Public-Key-Verfahren für Schlüsseltausch

<https://www.dpg-physik.de/>





← → ↻ 🔍 <https://www.dpg-physik.de/veranstaltungen/2025/quantentechnologie-von-der-lehzuranwendung>


DPG


Über uns ▾ Vereinigungen der DPG

Startseite / Veranstaltungen / 2025 / Quantentechnologie

Quantentechnologie - Workshop, Jahrestreffen der A

Datum:  Do, 18.09.2025

Adresse:  FH Münster, Tec Stegerwaldstr. 3 Raum A111, Geb

Sprache: Deutsch

Veranstaltungspartner: Lokale T

Kontaktperson: Prof. Dr. Walter Neu, Prof. Dr. Daniel Schondelmaier, Prof. Dr. Andrea Koch, Prof. Dr. Ale

Zertifikats-Viewer: dpg-physik.de

Allgemein **Details**

Zertifikathierarchie

- ISRG Root X1
 - R1
 - dpg-physik.de

Zertifikatsfelder

Betreff

- Informationen zum öffentlichen Schlüssel des Antragstellers
 - Algorithmus für öffentlichen Schlüssel des Antragstellers
 - Öffentlicher Schlüssel des Antragstellers**
- Erweiterungen
 - Zertifikatschlüsselverwendung
 - Erweiterte Schlüsselverwendung
 - Basiseinschränkungen eines Zertifikats

Feldwert

Modulo (2048 Bits):
C4 10 A5 A8 94 B5 79 88 F0 55 A7 A6 FF 5E F0 70
36 A7 0A 6D 74 81 5E 01 70 10 94 18 5E AE 60 4B
7C 71 6B 89 4A 04 52 D2 3A B2 44 FC 01 5D B1 83
A8 BA 9A 0F D1 53 59 01 70 27 7D CF C1 A9 3F B5
D2 BA 51 1E 85 FD 76 5D E1 1C AF 77 5D AB DC D8

Exportieren...

Öffentlicher Schlüssel



```
00 04 AE A3 DE 0D 07 D6 05 D2 E0 BE 3E EA 97 0C  
36 A1 90 3C 26 1D BA 0E 28 3C 40 44 D9 A1 73 51  
AD 40 01 D0 4C 2D D8 85 47 51 34 82 EC 3D 27 E9  
D2 0D B6 A0 92 7A FD 8E E6 39 0A 5F FA 99 B1 4D  
49 58
```

Mit dem öffentlichen Schlüssel verschlüsselter Sitzungsschlüssel



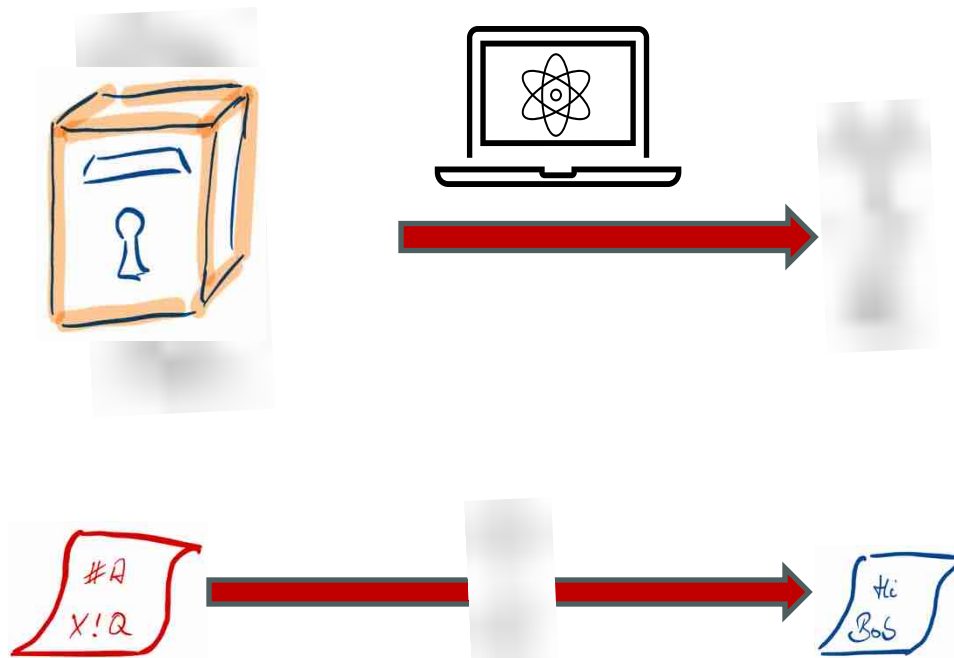
Verschlüsselte Nachricht



Bedrohung durch Quantencomputer

Bei gängigen Public-Key-Verfahren (wie RSA, Elgamal, Diffie-Helman usw.) können Quantencomputer den privaten Schlüssel sehr schnell aus dem öffentlichen Schlüssel berechnen.

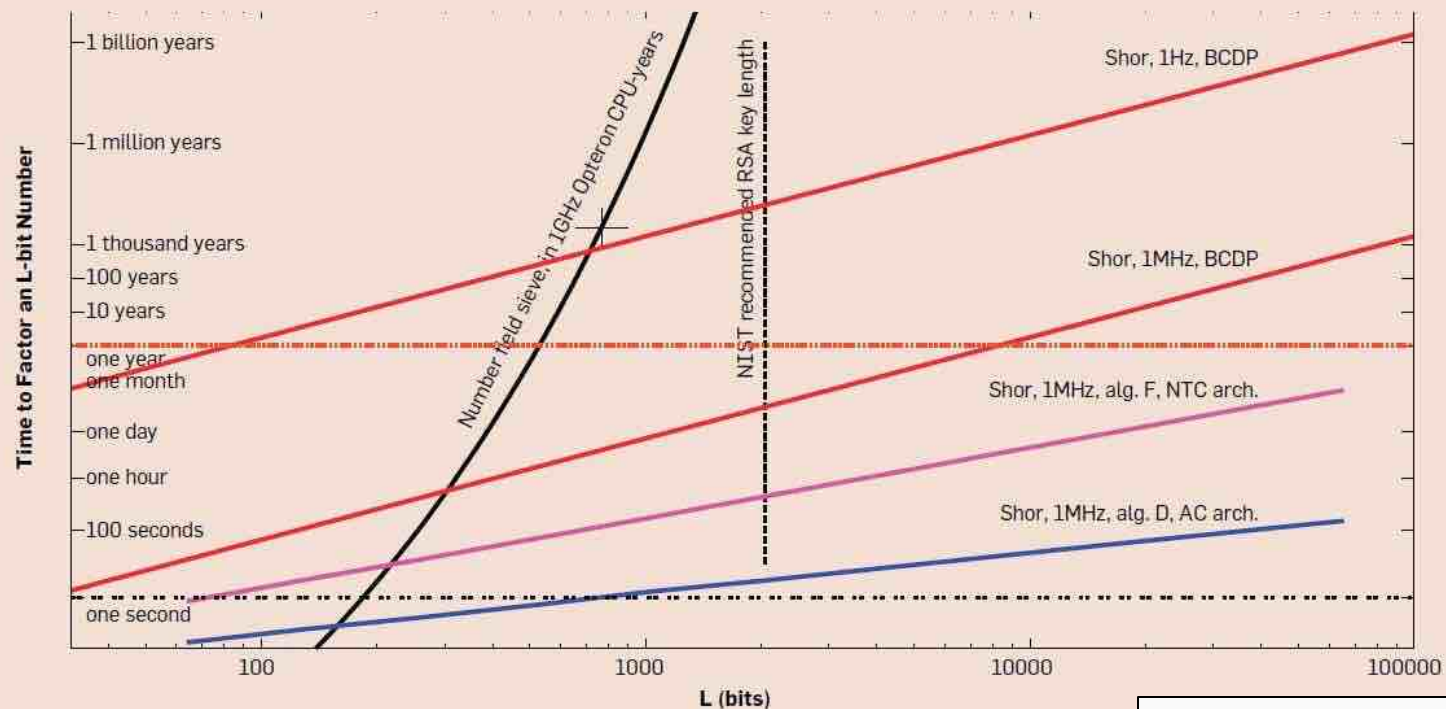
Die Verschlüsselung mit öffentlichen Schlüsseln ist nur rechnerisch sicher!
Sie basiert auf einem mathematischen Problem, von dem man annimmt, dass es keinen effizienten Algorithmus gibt, um es zu lösen.



Zeit für die Schlüsselberechnung

The horizontal axis is the length of the number to be factored. The steep curve is NFS, with the marked point at $L = 768$ requiring 3,300 CPU-years. The vertical line at $L = 2048$ is NIST's 2007 recommendation for RSA key length for data intended to remain secure until 2030. The other lines are various combinations of quantum computer logical clock speed for a three-qubit operation known as a Toffoli gate (1Hz and 1MHz), method of implementing the arithmetic portion of Shor's algorithm (BCDP, D, and F), and quantum computer architecture (NTC and AC, with the primary difference being whether or not long-distance operations are supported). The assumed capacity of a machine in this graph is $2L^2$ logical qubits. This figure illustrates the difficulty of making pronouncements about the speed of quantum computers.

Laufzeit des Shor-Algorithm im Vergleich



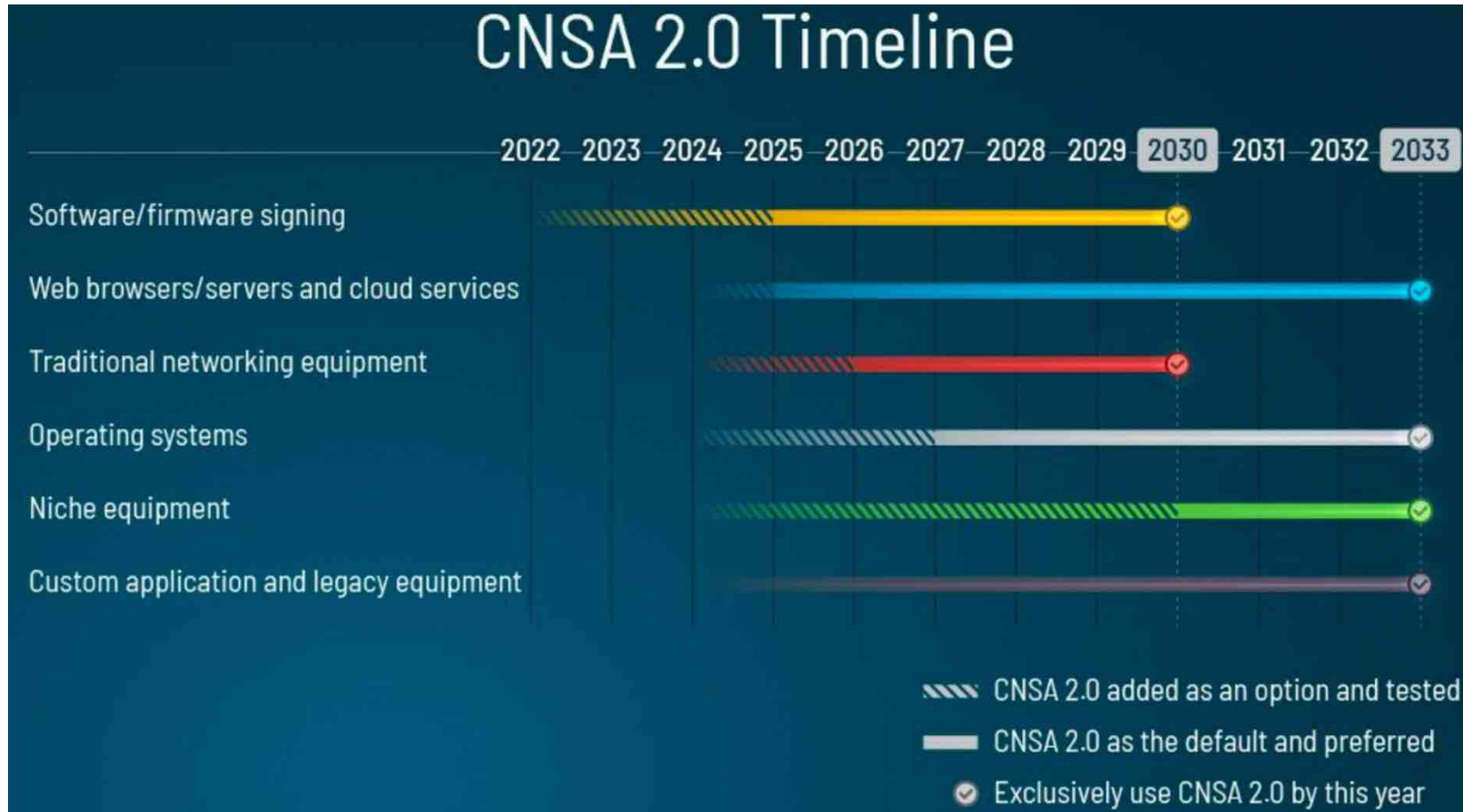
R. Van Meter and C. Horsman.
A blueprint for building a quantum computer.
Commun. ACM, 56(10):84–93, Oct. 2013.

Cryptographic Algorithm	Type	Purpose	Impact from large-scale quantum computer
AES	Symmetric key	Encryption	Larger key sizes needed
SHA-2, SHA-3	-----	Hash functions	Larger output needed
RSA	Public key	Signatures, key establishment	No longer secure
ECDSA, ECDH (Elliptic Curve Cryptography)	Public key	Signatures, key exchange	No longer secure
DSA (Finite Field Cryptography)	Public key	Signatures, key exchange	No longer secure

NIST: Report on Post-Quantum Cryptography
Quelle: <https://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8105.pdf>

1. Suche nach anderen schwierigen mathematischen Problemen, die von einem Quantencomputer nicht effizient gelöst werden können
 - Post-Quantum-Kryptographie
Algorithmen sind immer noch nur rechnerisch sicher!
Es ist nur eine Vermutung, dass es keinen effizienten Lösungsalgorithmus gibt!
2. Den Quantencomputer mit Quantenphysik austricksen
 - Quantenschlüssel-Verteilung
Mathematische Beweise für „perfekte“ Sicherheit existieren

Zeitplan der NSA für die Umstellung der Software von Drittanbietern auf Post-Quantum-Kryptographie.

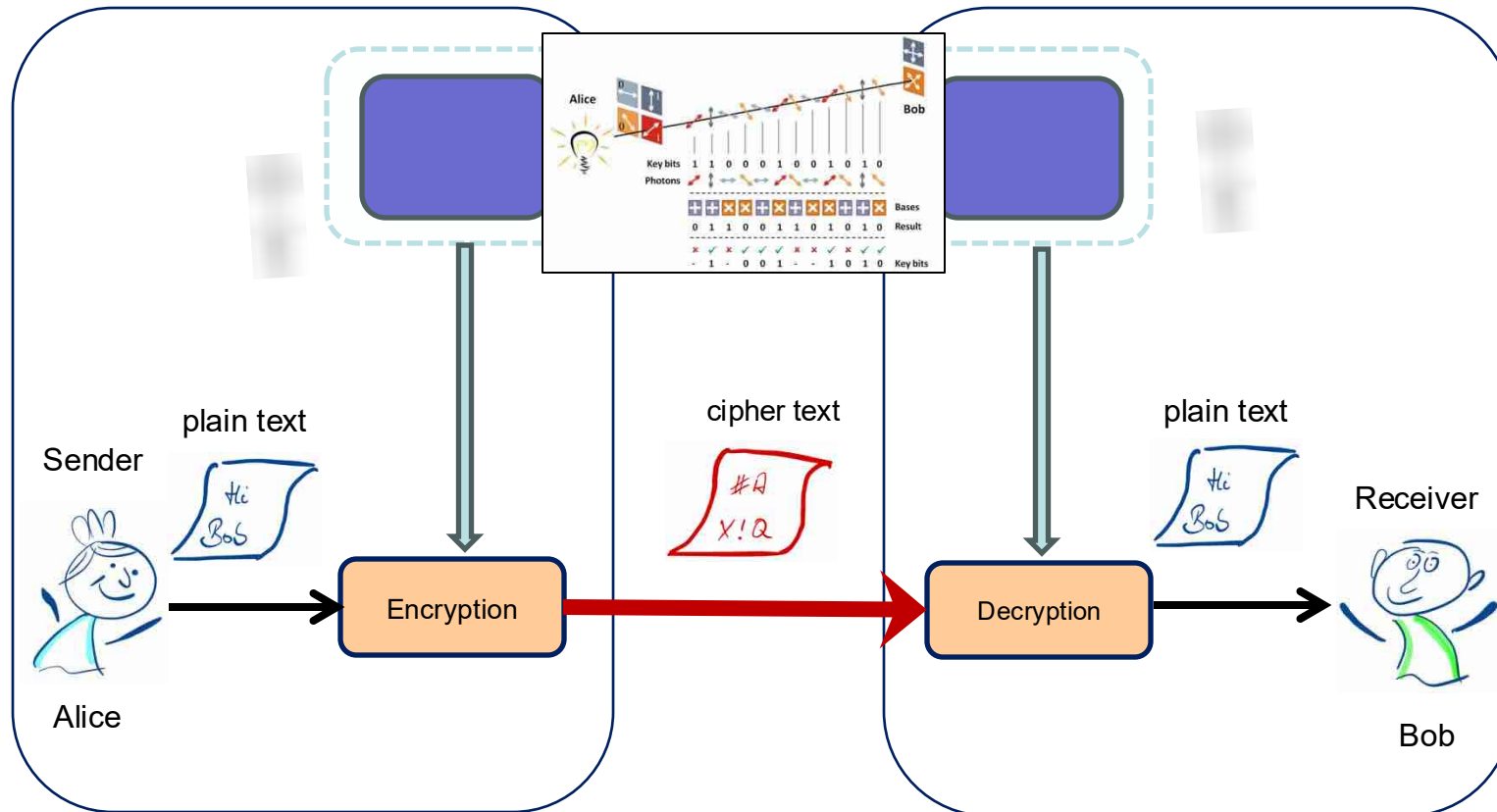


Source: CNSA - Commercial National Security Algorithm Suite

https://media.defense.gov/2022/Sep/07/2003071834/-1/-1/0/CSA_CNSA_2.0_ALGORITHMS_.PDF

NEUE QUANTENWELT

Quantum Key Distribution (QKD)

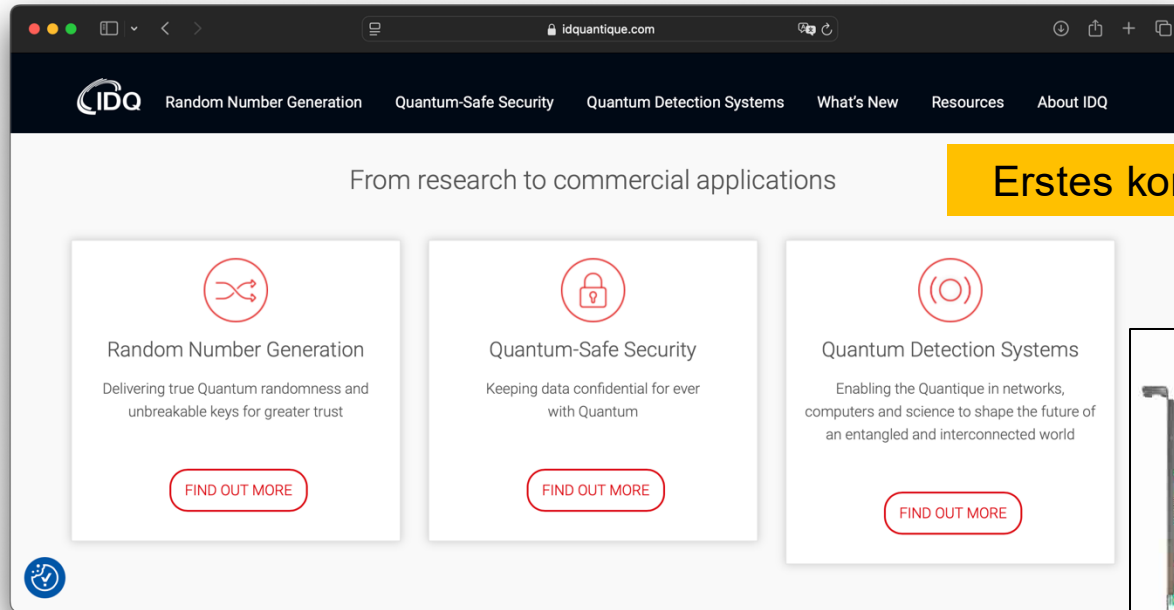


„Klassische“ BB84 basiert auf den folgenden Prinzipien

- No-Cloning-Theorem
- Ununterscheidbarkeit von Quantenzuständen
(Unschärferelation von Heisenberg)


„Modernere Protokolle“ (*Device Independent Quantum Key Distribution*)
nutzen

- Verschränkung
- Bell- bzw. CHSH-Ungleichung
- Teleportation



The screenshot shows the IDQ website homepage. The navigation bar includes: IDQ, Random Number Generation, Quantum-Safe Security, Quantum Detection Systems, What's New, Resources, and About IDQ. The main content area features the headline "From research to commercial applications" and three service cards: "Random Number Generation" (delivering true Quantum randomness and unbreakable keys), "Quantum-Safe Security" (keeping data confidential), and "Quantum Detection Systems" (enabling the Quantique in networks). Each card has a "FIND OUT MORE" button.

Erstes kommerzielles System 2004



Quantis QRNG PCIe

High performance PCIe Quantum Random Number Generator, with embedded NIST compliant post-processing

- ▶ Based on latest Quantis QRNG technology (IDQ20MC1 chip)
- ▶ 40 Mbps and 240 Mbps entropy source
- ▶ Embedded NIST 800-90 A/B/C compliant DRBG
- ▶ Health monitoring and failure detection mechanism at chip level
- ▶ Easy integration with Quantis Software & compatible with other Quantis PCIe and USB



Clavis XG KQD System

Quantum Key Distribution for production environments requiring high key transmission rate or extended range interconnection

- ▶ Long range (up to 150 km)
- ▶ Higher key rate on short distances: typical 14'000 AES-256 Keys per hour @ 24 dB
- ▶ Complex network topologies (ring, hub and spoke, meshed, star)
- ▶ Controlled and monitored centrally
- ▶ Interoperability with major Ethernet and OTN encryptors

Aktueller Stand:

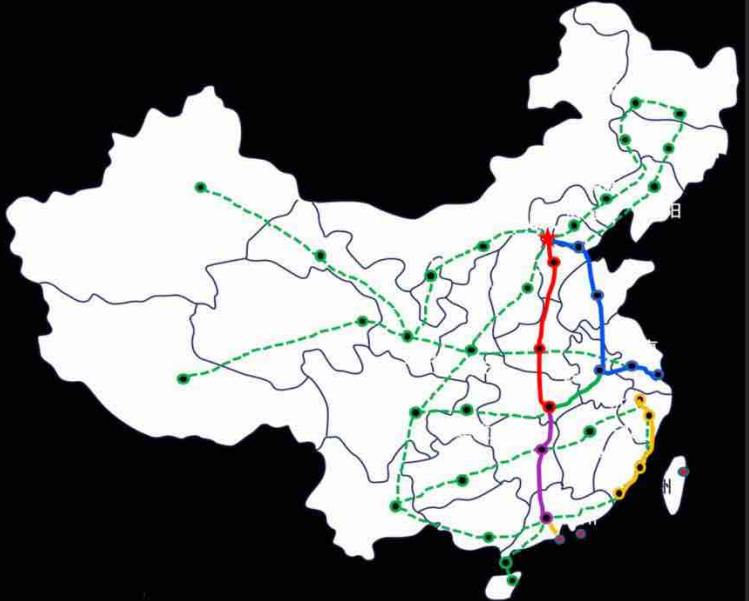
Upgrade the Current Fiber QKD Network in China

More safer, greater distance, and wider coverage

National Quantum Backbone Network

Lines	Longitudinal Backbone			
	Beijing-Shanghai	Beijing-Wuhan	Wuhan-Guangzhou	Extension of Shanghai-Hefei
Length(km)	1979	1700	1400	1400
Phase	Completed	Under Construction	Under Construction	Under Construction

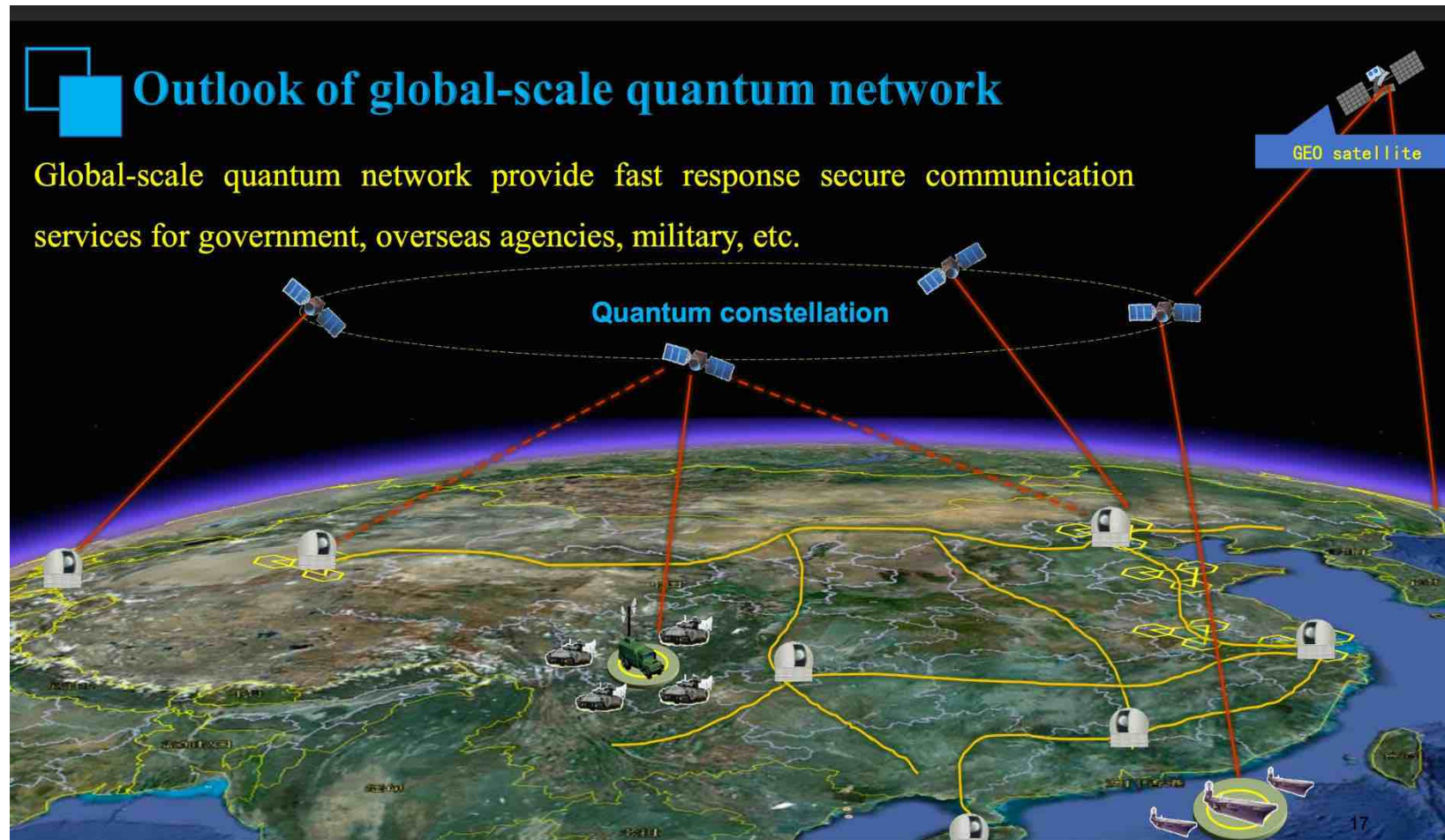
Lines	Transverse Backbone			
	Wuhan-Hefei	Guangdong-Hong Kong-Macao	Shandong	Jianghuai
Length(km)	693	180+180	/	2200
Phase	Completed	Under Construction	Under Review	Proposal



Folie stammt aus:

https://accelconf.web.cern.ch/icalpcs2021/talks/tukl01_talk.pdf

Plan bzw. Vision:



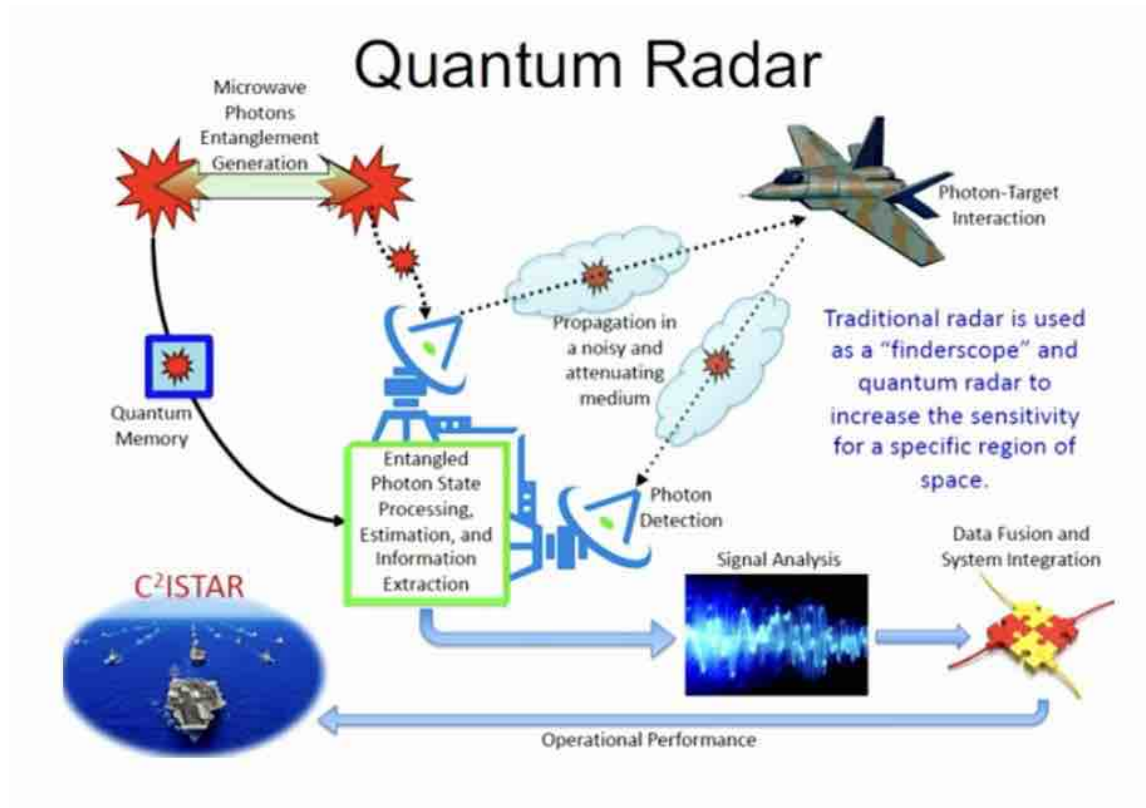


Figure 2.13. In a 2018 address to the National Academies, Dr. Marco Lanzagorta, explained how quantum radar might detect low-observable aircraft. Image courtesy US Naval Research Laboratory.

Quelle: Hoofnagle CJ, Garfinkel SL. *Law and Policy for the Quantum Age*. Cambridge University Press; 2022.

Detecting the Gravitational Quadrupole of a Simple Submarine Model

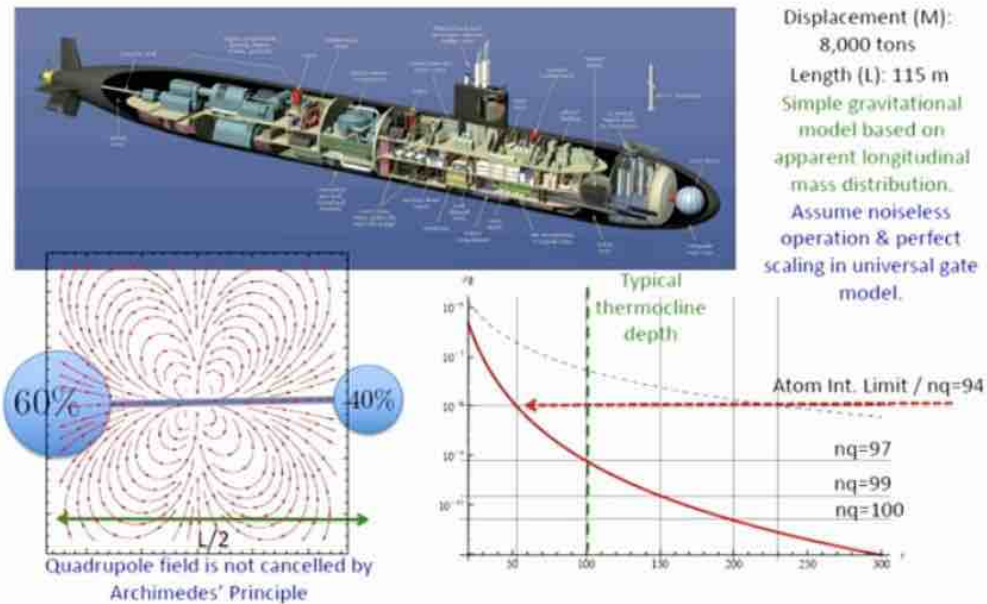


Figure 2.7. In a 2018 address to the National Academies, Dr. Marco Lanzagorta, explained how quantum gravimeters might detect a submarine. Image courtesy US Naval Research Laboratory.

Quelle: Hoofnagle CJ, Garfinkel SL. *Law and Policy for the Quantum Age*. Cambridge University Press; 2022.

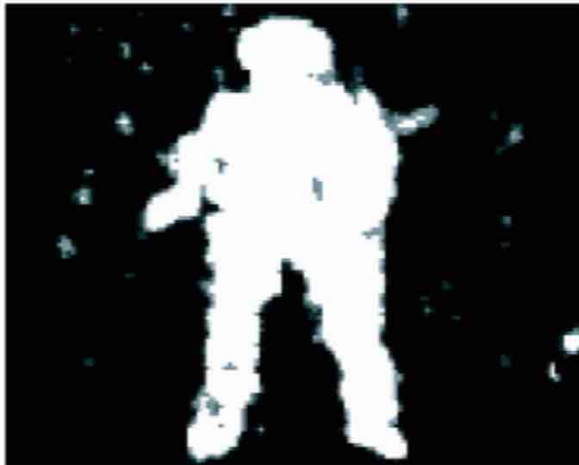


Figure 2.9. Carefully counting scattered and reflected photons from this toy soldier created a “ghost image” of it – an image constructed of an object that was outside the view of a camera. Courtesy of Office of the Secretary of Defense Public Affairs.

Meyers, Ronald E., Keith S. Deacon, and Yanhua Shih (April 2008).

“Ghost-Imaging Experiment by Measuring Reflected Photons.”

Physical Review A 77.4, pp. 041801.

link.aps.org/doi/10.1103

[/PhysRevA.77.041801.](https://link.aps.org/doi/10.1103/PhysRevA.77.041801)

Quelle: Hoofnagle CJ, Garfinkel SL. *Law and Policy for the Quantum Age*. Cambridge University Press; 2022.

Eine Quantenwelt bietet mehr als „nur“ sicheren Schlüsseltausch!

Anwendungen in einer „Quantenwelt“

- Quanten-Computing
- Quanten-Internet
 - Mit globalem sicheren Schlüsselaustausch
- Überprüfbare Löschung von Daten
- Sichere Signaturen/Kopierschutz
 - Privat und öffentlich verifizierbar
- Blind Computing
- Quanten-Grid-Computing
- „Vernetzte“ Quantenmessungen (z. B. Quantenteleskope)
- usw.



Über die **Quantum Computing** Kursreihe

Dieser Kanal bringt Ihnen gebündelt die gewaltigen Potenziale der aufkommenden Schlüsseltechnologie des Quantum Computing näher. Hier können Sie durch kostenlose offene Onlinekurse schnell und einfach einsteigen in die vielversprechende Welt der Quantencomputer und deren Nutzung in der Wirtschaft. Lassen Sie sich durch erfahrene Fachwissenschaftlerinnen und -wissenschaftler einen Wissensvorsprung vermitteln – sowohl zu den theoretischen Grundlagen der Technologie als auch zu prototypischen Anwendungsmöglichkeiten in Unternehmen.

Zusätzlich bietet das **Quantum Computing Forum** allen Interessierten und Teilnehmenden der MOOCs eine Plattform zum gegenseitigen Austausch.

Suchen Sprache **Alle** Niveau **Alle** Thema **Alle**

Kurse im Selbststudium

<p>Quantum Computing for Natural Sciences (with IBM Quantum) SEIT 26. SEP. 2022 IM... # QUANTUM COMPUTING LEISTUNGSNACHWEIS EN</p>	<p>Quantenalgorithmen und -implementierung Teil 2 und 3 SEIT 23. JUL. 2022 IM... # QUANTUM COMPUTING LEISTUNGSNACHWEIS DE</p>	<p>Quantum Optimization (with IBM Quantum) SEIT 6. JUN. 2022 IM... # QUANTUM COMPUTING LEISTUNGSNACHWEIS EN</p>	<p>Quanteninformation und -Kryptographie - Teil 3 SEIT 23. MAI. 2022 IM... # QUANTUM COMPUTING LEISTUNGSNACHWEIS DE</p>
<p>Einführung in das Quantencomputing - Teil 3 SEIT 3. MAI. 2022 IM... # QUANTUM COMPUTING LEISTUNGSNACHWEIS DE</p>	<p>Vom Bit zum Qubit SEIT 7. MÄR. 2022 IM... # QUANTUM COMPUTING LEISTUNGSNACHWEIS DE</p>	<p>Quantum Machine Learning (with IBM Quantum) SEIT 29. JAN. 2022 IM... # QML DATA AND AI, QUANTUM... LEISTUNGSNACHWEIS EN</p>	<p>Quantum Computing Forum SEIT 31. DEZ. 2022 IM... # QUANTUM COMPUTING LEISTUNGSNACHWEIS DE</p>
<p>Quantenalgorithmen und -implementierung Teil 1 SEIT 21. DEZ. 2022 IM... # QUANTUM COMPUTING LEISTUNGSNACHWEIS DE</p>	<p>Quanteninformation und -Kryptographie - Teil 2 SEIT 21. NOV. 2022 IM... # QUANTUM COMPUTING LEISTUNGSNACHWEIS DE</p>	<p>Einführung in das Quantencomputing - Teil 2 SEIT 31. NOV. 2022 IM... # QUANTUM COMPUTING LEISTUNGSNACHWEIS DE</p>	<p>Introduction to Quantum Computing with QISKIT (with IBM Quantum) SEIT 31. NOV. 2022 IM... # QUANTUM COMPUTING LEISTUNGSNACHWEIS EN</p>
<p>Quanteninformation und -kryptographie - Teil 1 SEIT 29. JUN. 2022 IM... # QUANTUM COMPUTING LEISTUNGSNACHWEIS EN</p>	<p>Einführung in das Quantencomputing - Teil 1 SEIT 14. JUN. 2022 IM... # QUANTUM COMPUTING LEISTUNGSNACHWEIS DE</p>		

Grundlagen des Quantencomputing, 1-3

Prof. Dr. Bettina Just, THM



Quantenkryptographie 1-3

Prof. Dr. Jörg Hettel,
HS Kaiserslautern



Quantenalgorithmen und Implementierung 1-3

Prof. Dr. Gerhard Hellstern,
DHBW Stuttgart



Kontakt

Prof. Dr. Jörg Hettel

Hochschule Kaiserslautern
Fachbereich Informatik und Mikrosystemtechnik

joerg.hettel@hs-kl.de
