Post-quantum cryptography

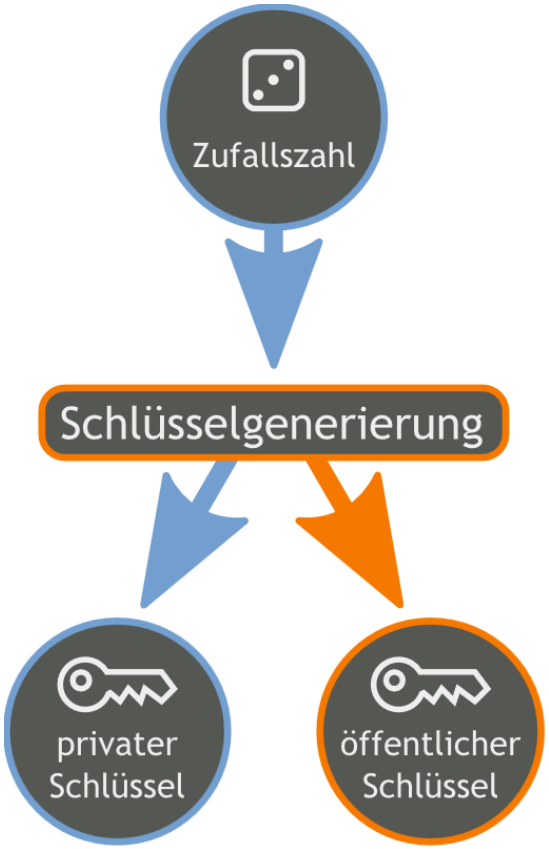
Post-quantum cryptography

1. Komplexität & Quantencomputer
2. Kryptografie in Gittern
3. FHE – Eine Revolution im Datenschutz?

WIESO?

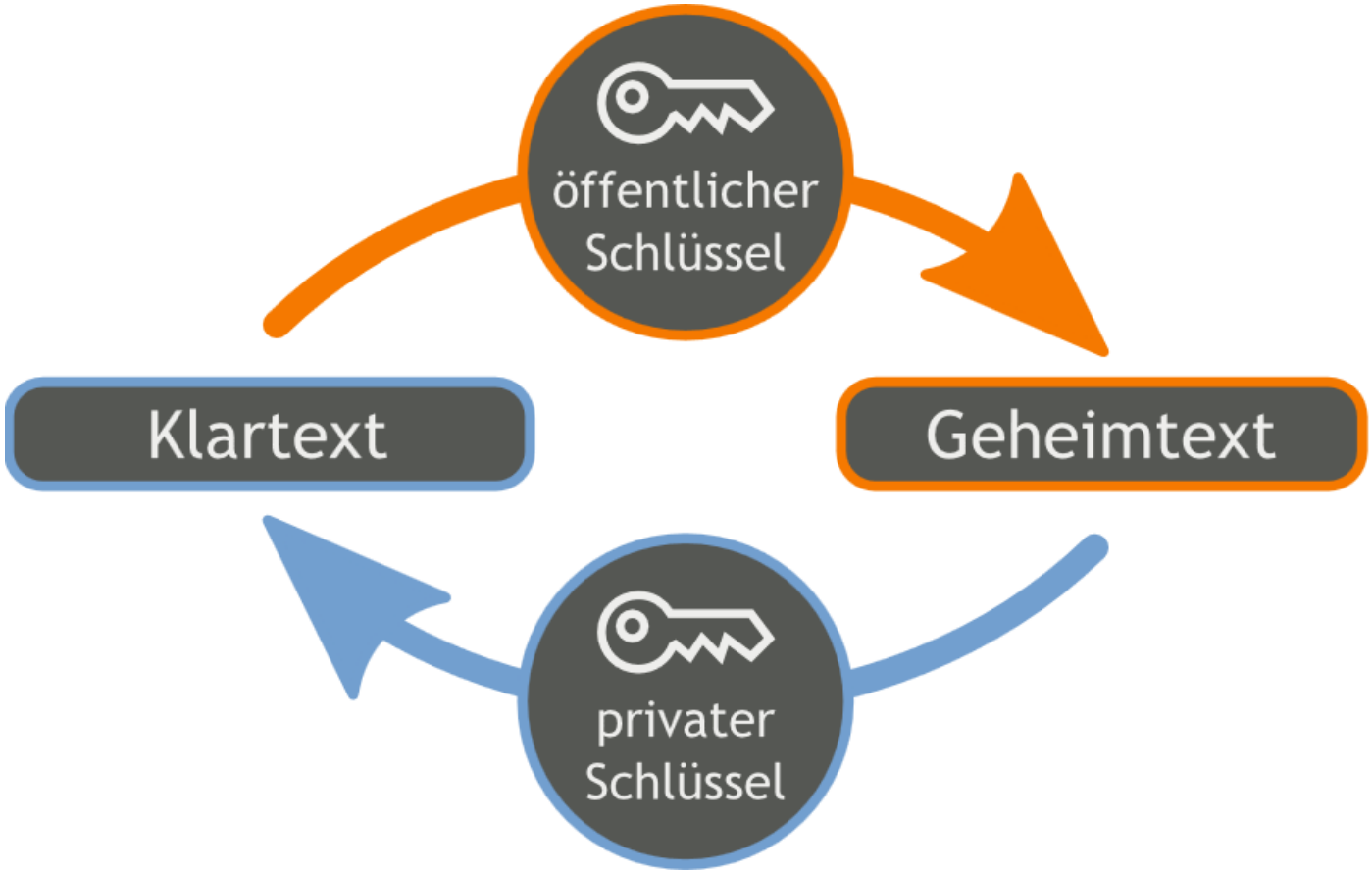
KOMPLEXITÄT

Public-Key-Kryptografie



Grafiken: http://de.wikipedia.org/wiki/Asymmetrisches_Kryptosystem

Public-Key-Kryptografie



Ziel

Schlüsselgenerierung, Ver- und Entschlüsslung sollen leicht durchführbar sein.

Entschlüsslung ohne Kenntnis des privaten Schlüssels soll nicht sinnvoll möglich sein

Effizienz

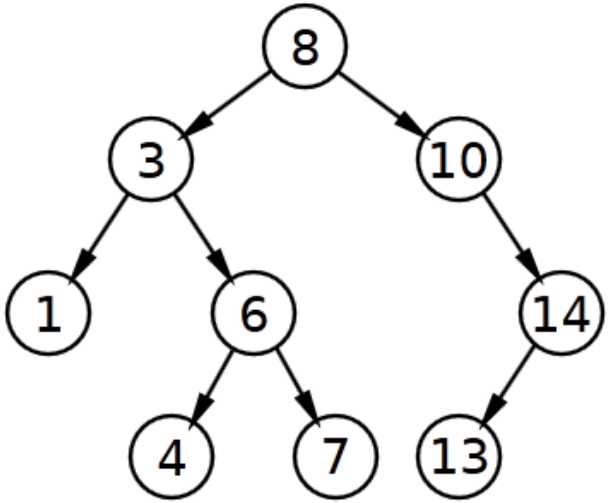
Schlüsselgenerierung, Ver- und Entschlüsslung sollen *effizient* durchführbar sein.

Entschlüsslung ohne Kenntnis des privaten Schlüssels soll *nicht effizient* möglich sein.

Komplexität

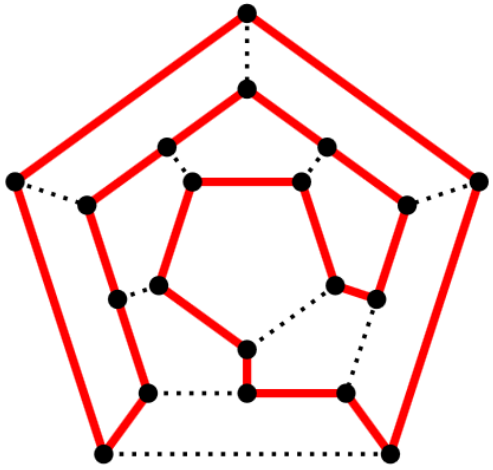
- Wie viele Rechenschritte sind zur Lösung eines Problems nötig?

Binäre Suche



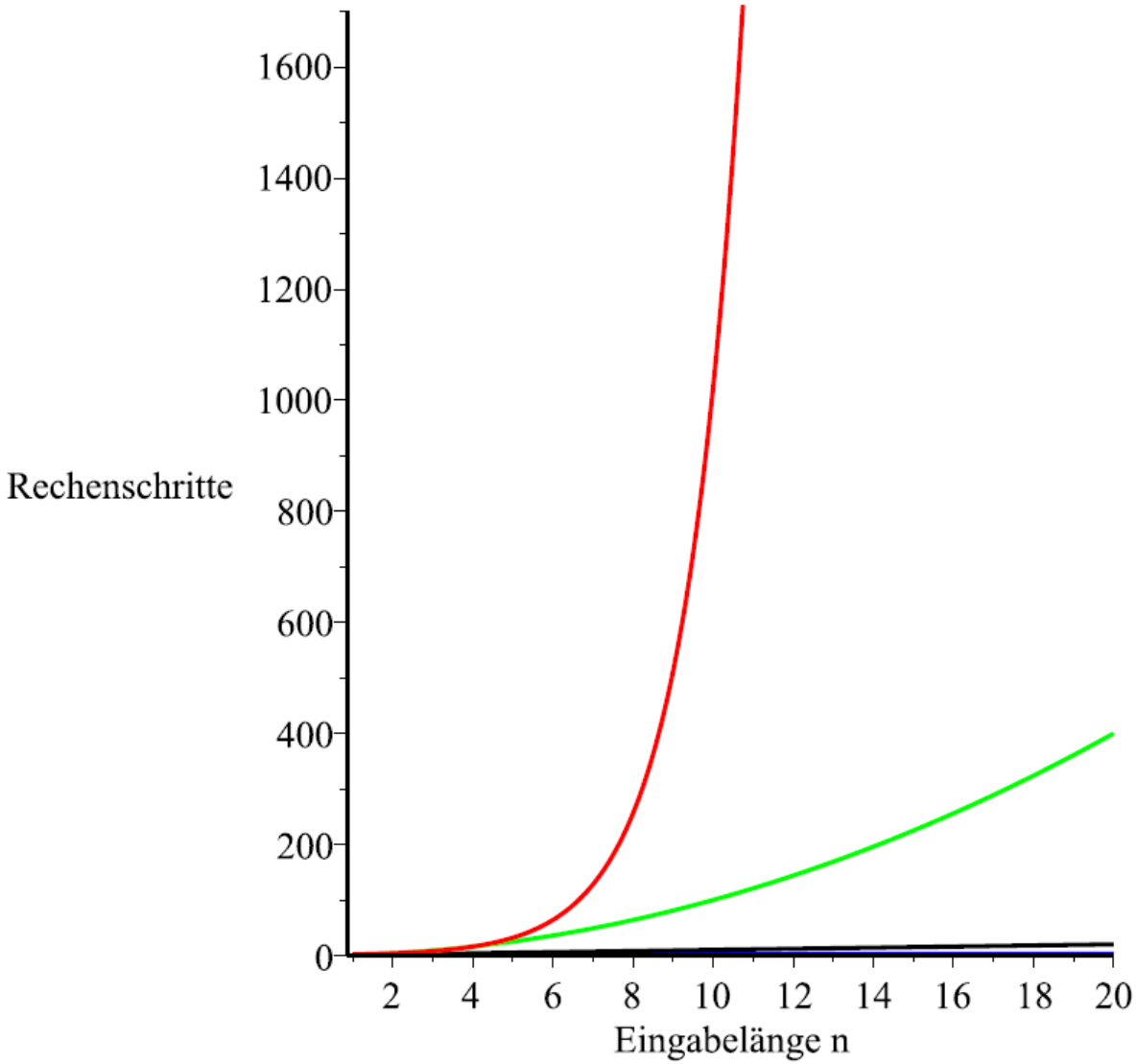
$O(\log n)$

Hamiltonkreise



$O(2^n)$

Grafik: http://en.wikipedia.org/wiki/File:Hamiltonian_path.svg
http://en.wikipedia.org/wiki/Binary_search_tree


Rechenschritte
1600
1400
1200
1000
800
600
400
200
0
2
4
6
8
10
12
14
16
18
20
Eingabelänge n

Komplexe Probleme in der Kryptografie

Faktorisierung von ganzen Zahlen (RSA)

$$N = p \cdot q$$

Finde Faktoren p, q.

Komplexe Probleme in der Kryptografie

Diskrete Logarithmen (ElGamal)

Sei G eine endliche Gruppe,

gegeben $x, y \in G \; mit \; y = x^n$.

Finde n.

Verwendete Gruppen:

- Ganze Zahlen modulo p
- Elliptische Kurven

QUANTENCOMPUTER

Quantencomputer

- Physikalische Repräsentation von Qubits
- Superposition von Zuständen 0 und 1
- Durch Messung kollabieren die Qubits in einen klassischen Zustand

Theoretisch können n Qubits 2^n Zustände gleichzeitig repräsentieren.

- Die Entwicklung von Quanten-Algorithmen ist sehr speziell
- Zahlreiche Anwendungen

Grovers Algorithmus

- Suche in einer unsortierten Liste mit N Einträgen

Klassisch: $O(n)$

Quantenalgorithmus: $O(\sqrt{n})$

- Grovers Algorithmus verdoppelt die Länge aller Passwörter/Schlüssel, die man brechen kann

Abelsches HSP, Shors Algorithmus

- Quanten-Fouriertransformation in endlichen Gruppen

- Shors Algorithmus liefert ausgerechnet eine Lösung in Polynomialzeit für
  - Faktorisierung von ganzen Zahlen
  - Diskrete Logarithmen

Quantencomputer brechen alle populären Public-Key-Verfahren

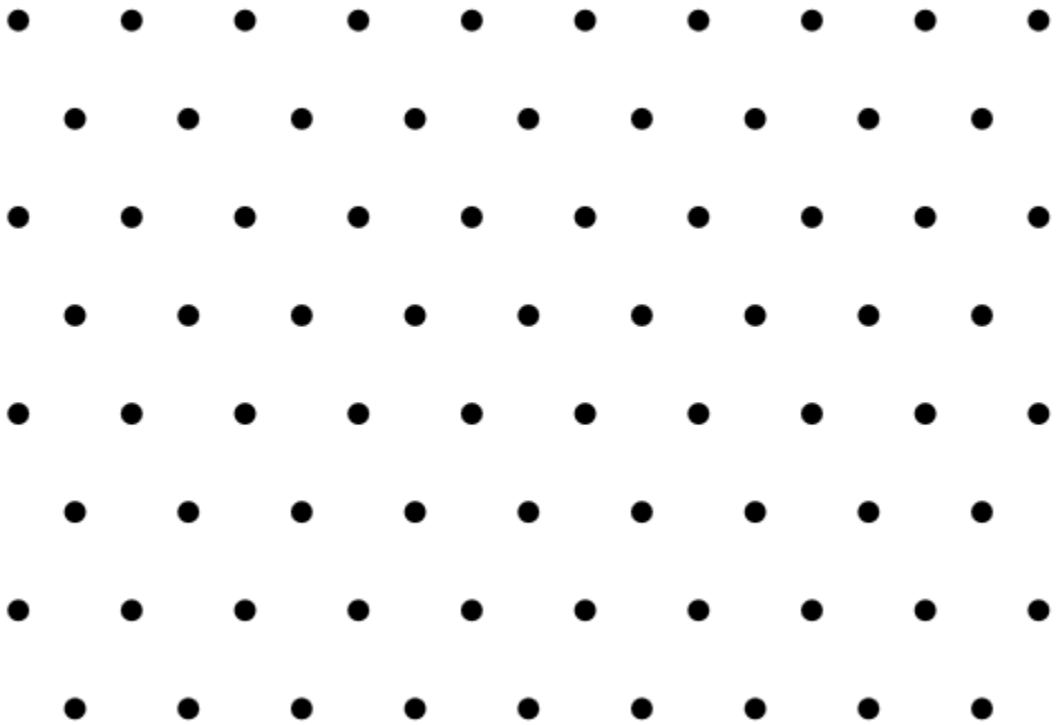
Quantencomputer – Stand der Technik

- Seit 1980 werden Quanten-Algorithmen theoretisch erforscht

- Einzelne Quantenberechnungen wurden erfolgreich durchgeführt
  - 2011: Faktorisierung von 143 mit 4 Qubits
  - Heute: 512 Qubits mit Einschränkungen
- Technische Hürden
- Offiziell: $79.7 Millionen Forschungsetat der NSA

KRYPTOGRAFIE IN GITTERN

Ganzzahlige Gitter



Ganzzahlige Gitter

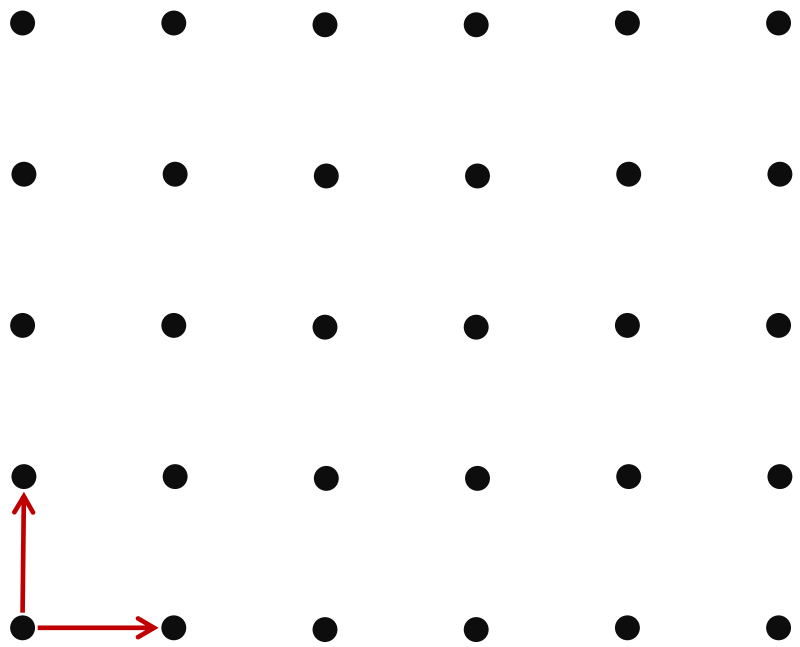
Definition:

Ein ganzzahliges Gitter ist eine Menge von Punkten mit ganzzahligen Koordinaten, die unter Summen und Differenzen abgeschlossen ist.

Gitterbasen

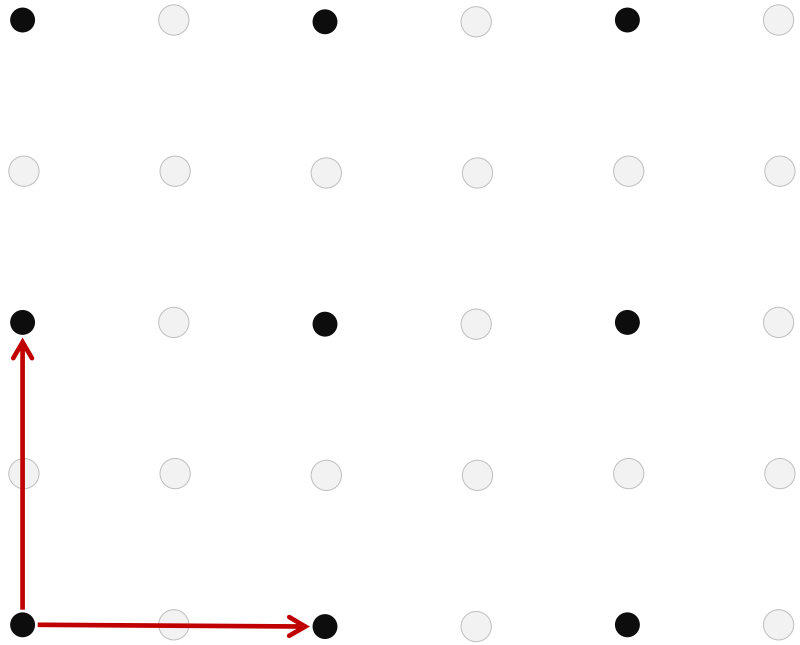
Jedes d-dimensionale Gitter wird durch eine Basis aus d linear unabhängigen Vektoren erzeugt.

Gitterbasen



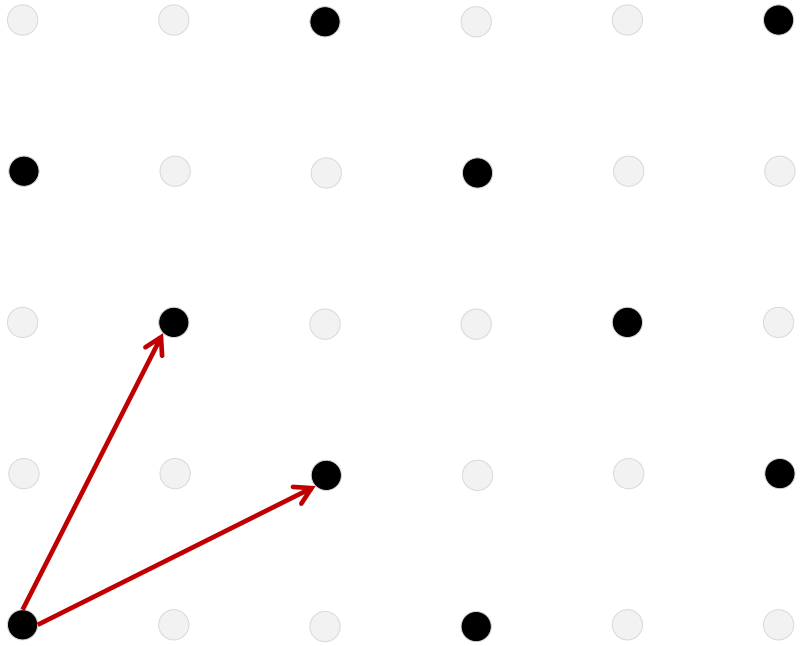
$v_1 = \binom{1}{0}, v_2 = \binom{0}{1}$

Gitterbasen



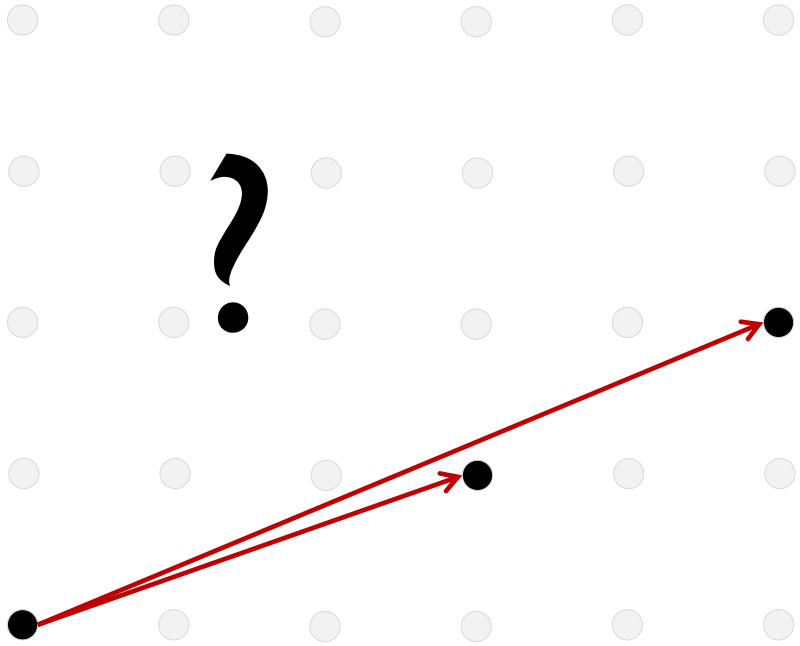
$v_1 = \binom{2}{0}, v_2 = \binom{0}{2}$

Gitterbasen



$v_1 = \binom{2}{1}, v_2 = \binom{1}{2}$

Gitterbasen



$v_1 = \binom{3}{1}, v_2 = \binom{5}{2}$

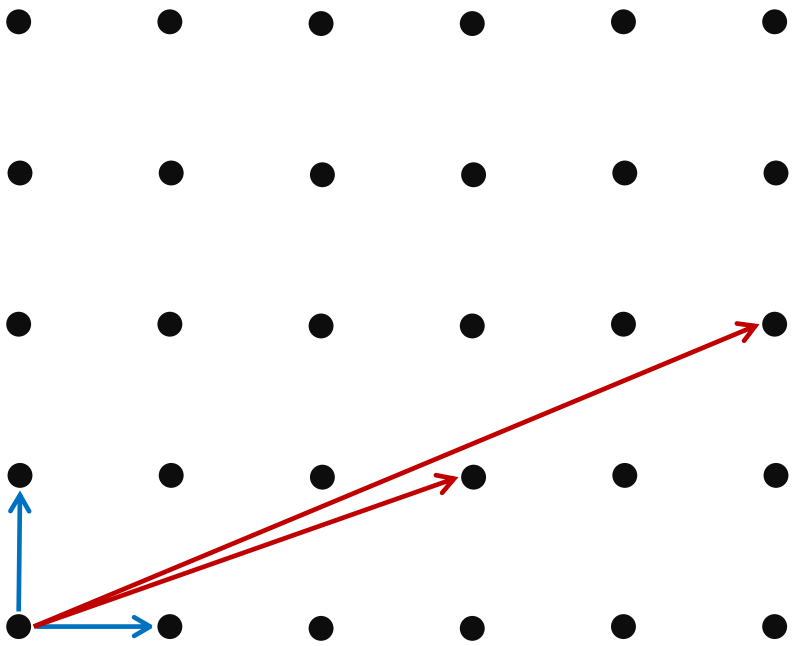
Basiswechsel

$$2 \cdot \begin{pmatrix} 3 \\ 1 \end{pmatrix} - 1 \cdot \begin{pmatrix} 5 \\ 2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$-5 \cdot \begin{pmatrix} 3 \\ 1 \end{pmatrix} + 3 \cdot \begin{pmatrix} 5 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Gitterbasen

Verschiedene Basen desselben Gitters!



Basiswechsel und Matrizen

Invertieren von ganzzahligen Matrizen

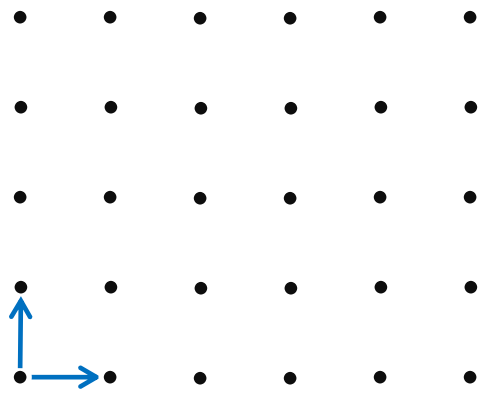
$$\begin{pmatrix} 3 & 5 \\ 1 & 2 \end{pmatrix}^{-1} = \begin{pmatrix} 2 & -5 \\ -1 & 3 \end{pmatrix}$$

Bedingung:

$$\left|\det \begin{pmatrix} 3 & 5 \\ 1 & 2 \end{pmatrix}\right| = 1$$

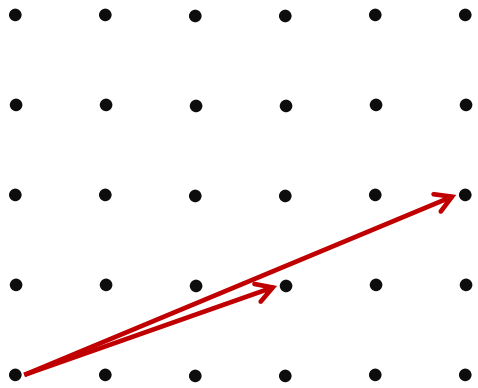
Gute und schlechte Basen

Gute Basis



- Sehr kurze Vektoren
- Nahezu orthogonal
- $H \approx 1$

Schlechte Basis

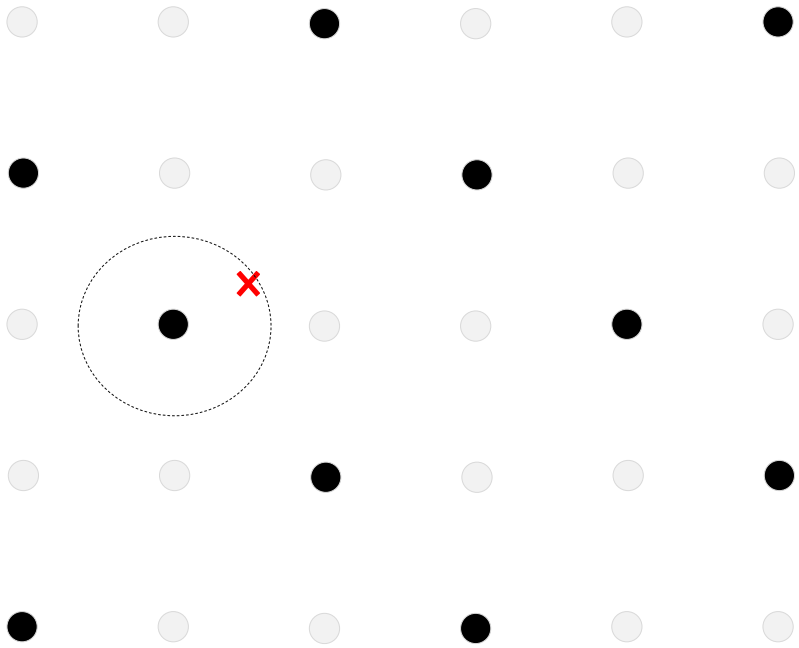


- Unnötig lange Vektoren
- Nahezu parallel
- $H \ll 1$

Hadamard-Quotient H: $1 \geq \sqrt[n]{\frac{\det L}{||v_1|| \cdots ||v_n||}} \geq 0$

Probleme in Gittern

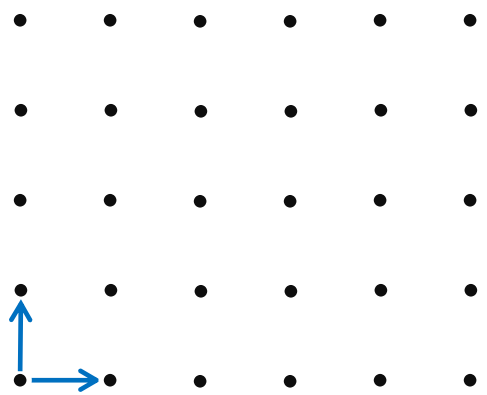
- SVP: Shortest Vector Problem
- CVP: Closest Vector Problem



Probleme in hochdim. Gittern

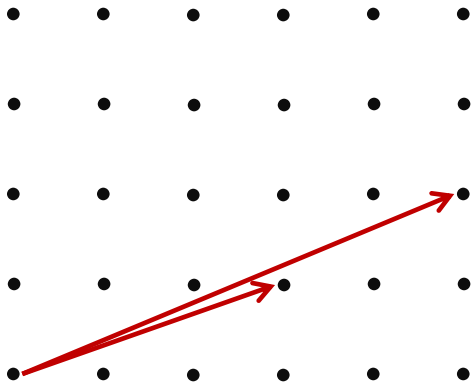
Multiplikation mit zufälligen Matrizen, |det| = 1

Gute Basis



- CVP/SVP sehr einfach (LGS lösen, runden)

Schlechte Basis



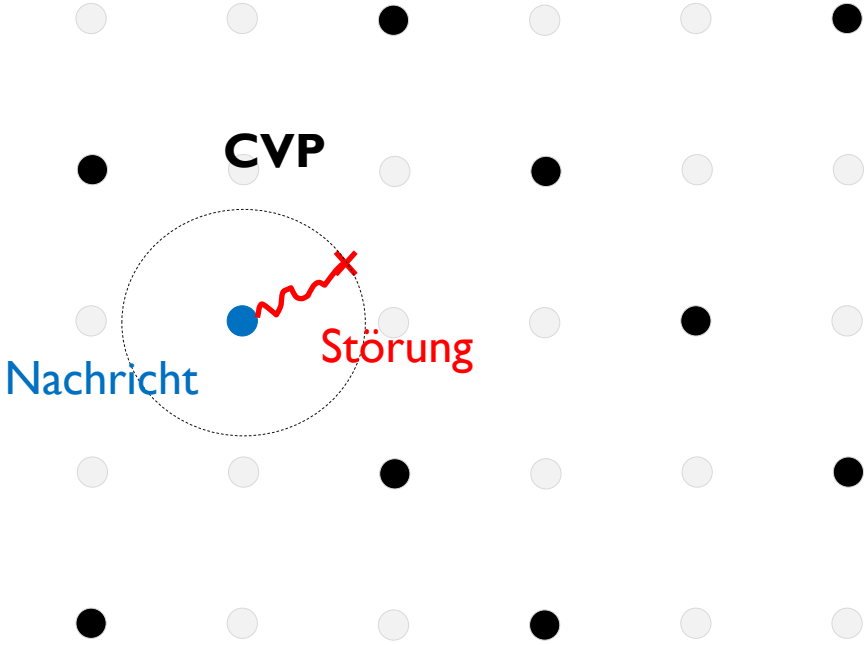
- Kein effizientes Verfahren für CVP/SVP bekannt

„Lattice reduction“, kein effizientes Verfahren bekannt

Public-Key-Kryptografie in Gittern

Öffentlicher Schlüssel	Privater Schlüssel
Schlechte Basis eines hochdimensionalen Gitters	Gute Basis des Gitters

- Nachrichten als Gitterpunkte
- Wer eine gute Basis kennt, kann die Störung eliminieren



Beispiel

```
Lgood := ⟨100, 0, 1; 2, 500, 0; 3, 3, 999⟩^%T;
```

$$\begin{bmatrix} 100 & 2 & 3 \\ 0 & 500 & 3 \\ 1 & 0 & 999 \end{bmatrix}$$

```
U := ⟨45839, 6243, 1384991642; 439, 49823, 999; 1234, 4391, 36158890⟩;
Determinant(U);
```

$$\begin{bmatrix} 45839 & 6243 & 1384991642 \\ 439 & 49823 & 999 \\ 1234 & 4391 & 36158890 \end{bmatrix}$$

$$1$$

```
Lbad := Lgood.U;
```

$$\begin{bmatrix} 4588480 & 737119 & 138607642868 \\ 223202 & 24924673 & 108976170 \\ 1278605 & 4392852 & 37507722752 \end{bmatrix}$$

```
msg := ⟨1, 4, 0⟩;
pt := Lbad.msg;
err := ⟨-18, 20, 15⟩ :
crypt := pt + err :
```

$$\begin{bmatrix} 1 \\ 4 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 7536956 \\ 99921894 \\ 18850013 \end{bmatrix}$$

```
cv := Lgood round~(LinearSolve(Lgood, crypt));
msg2 := LinearSolve(Lbad, cv);
```

$$\begin{bmatrix} 7536956 \\ 99921894 \\ 18850013 \end{bmatrix}$$

$$\begin{bmatrix} 1 \\ 4 \\ 0 \end{bmatrix}$$

Public-Key-Kryptografie in Gittern

Vorteile

- Sehr effizient
- Einfach zu implementieren
- Kein effizienter Quantenalgorithmus bekannt

Nachteile

- Sehr große Schlüssel
- d-dim. Gitter → d x d - Matrix

NTRU

- Zusammenhang zwischen Gittern und Polynomen, z.B.

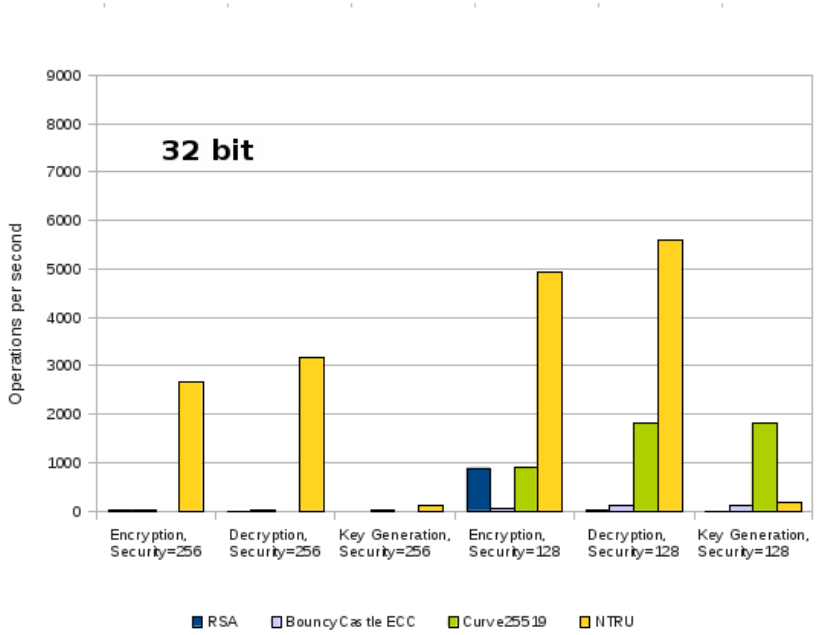
$$\mathbb{Z}^n \cong \mathbb{Z}[x]/\langle f \rangle$$

Praktisches Verfahren:

- Schlüssel sind Polynome
- Verwandt mit Gitterreduktion in N Dimensionen (max. Security N=503)
- O(n log n) Schlüsselgröße

NTRU in der Praxis

- 1996 patentiert von Hoffstein, Pipher, Silverman
- Algorithmen NTRUEncrypt, NTRUSign
- Open-Source-Implementierung verfügbar (http://tbuktu.github.io/ntru/)



Beispiel (Offizielle Demo)

```
private static void encrypt() {
System.out.println("NTRU encryption");

// create an instance of NtruEncrypt with a standard parameter set
NtruEncrypt ntru = new NtruEncrypt(EncryptionParameters.APR2011_439_FAST);

// create an encryption key pair
EncryptionKeyPair kp = ntru.generateKeyPair();

String msg = "The quick brown fox";
System.out.println(" Before encryption: " + msg);

// encrypt the message with the public key created above
byte[] enc = ntru.encrypt(msg.getBytes(), kp.getPublic());

// decrypt the message with the private key created above
byte[] dec = ntru.decrypt(enc, kp);

// print the decrypted message
System.out.println(" After decryption: " + new String(dec));
}
```

FULLY HOMOMORPHIC ENCRYPTION (FHE)

Milliardärsproblem

Wie können zwei Milliardäre feststellen, wer von beiden der reichere ist, ohne ihr Vermögen zu offenbaren?

Fully homomorphic encryption

Beliebige Berechnungen auf verschlüsselten Daten ausführen:

Ein Programm erhält verschlüsselte Eingaben und berechnet die Verschlüssslung der Ausgabe, ohne zu entschlüsseln oder überhaupt den Schlüssel zu kennen.

Revolution im Datenschutz?

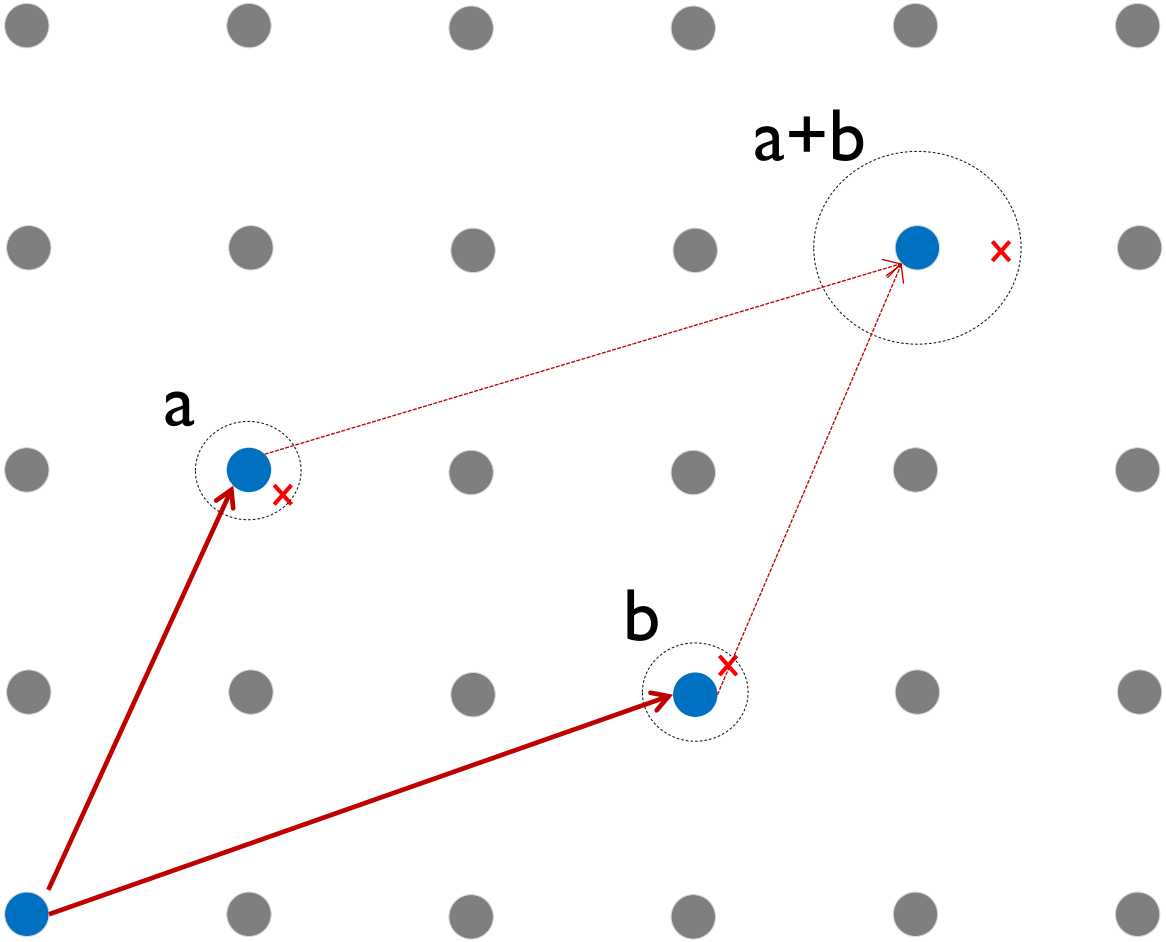
- Auslagerung von Berechnungen
- Cloud-Computing, Mehrparteien-Berechnung
- Private Information Retrieval
- Sichere und geheime Abstimmungen

Implementierung

- 2009 hat Craig Gentry das erste funktionierende FHE-Kryptosystem vorgestellt
- *Noch* unpraktikabel

- Basierend auf Idealgittern (Polynomringe)

Idee



Idee

Gitter-Kryptosysteme erlauben Operationen auf den verschlüsselten Daten, akkumulieren aber die Störungen, bis das CVP nicht mehr die richtige Lösung liefert.

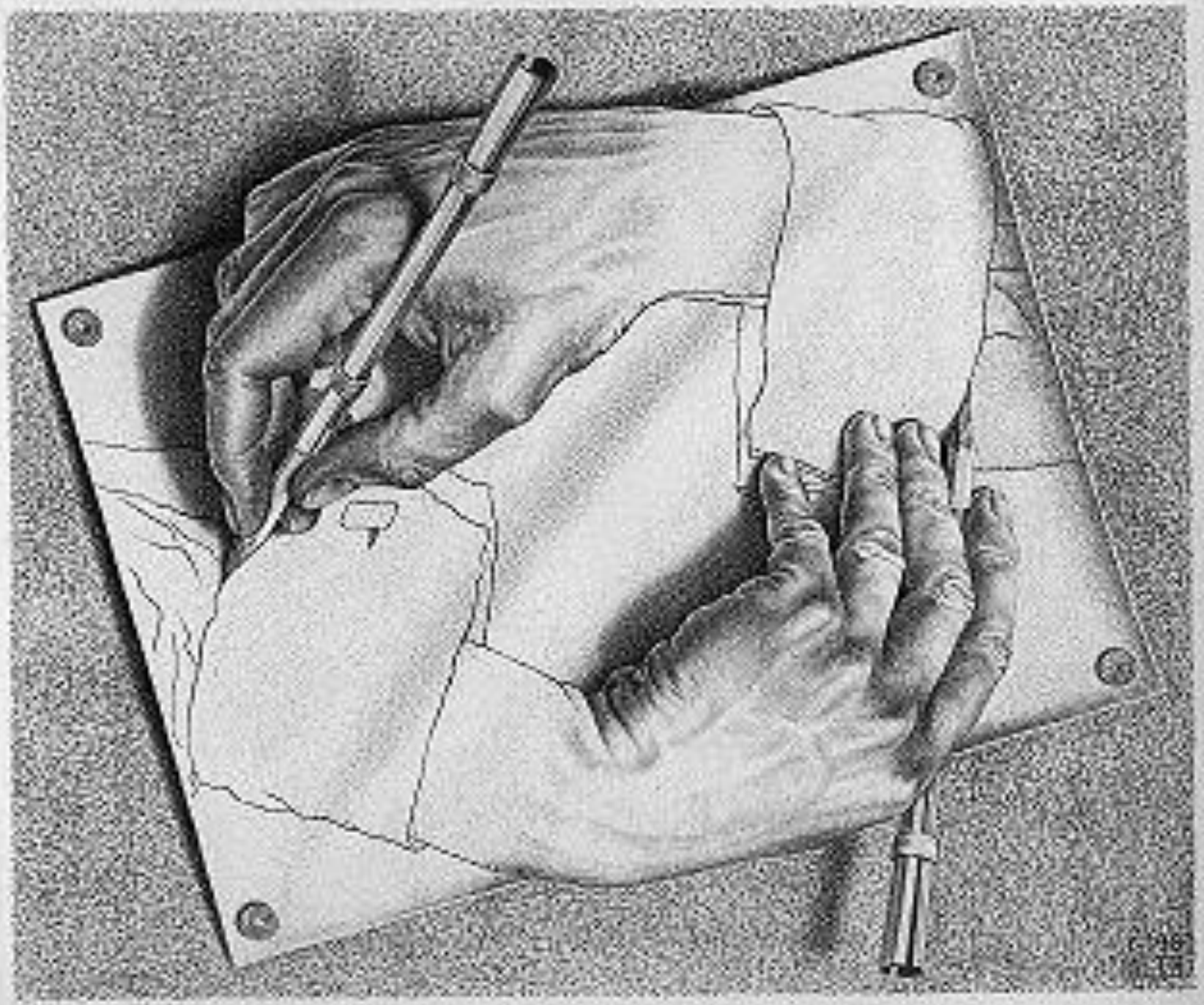
Lösung: Alle N Schritte die Störung reduzieren ➔ Neu verschlüsseln

Wie ohne privaten Schlüssel?

Die Lösung

Entschlüsslungsalgorithmus homomorph auf verschlüsselten Daten ausführen!

Dazu muss die Entschlüsslung in unter N Schritten durchführbar sein. Aber Gitter-Kryptografie ist sehr effizient zu implementieren ☺



Take home message

- Quantencomputer rücken in greifbare Nähe. Sie brechen ausgerechnet alle heute populären Public-Key-Verfahren.
- Kryptografie in Gittern (bspw. NTRU) wird durch kein heute bekanntes Verfahren auf Quantencomputern gebrochen
- FHE wurde erstmals implementiert und verspricht, den Datenschutz zu revolutionieren