Gegenmassnahmen
Was können wir tun?

Martin Kaufmann, Urs Wagner

cnlab Herbsttagung 2023: Bits & Qubits
Gleisarena, Zürich, 6. September 2023

cnlab

Quantencomputer brechen unsere Public-Key-Verfahren

- Gefährdet sind unter anderem:
  - Schlüsselaustausch
  - Public-Key-Verschlüsselung
  - Signaturen
- Was können wir tun?
  - Quanten-kryptografische Verfahren
    (Verfahren basierend auf Quanten-Effekten)
  - Post-Quanten-kryptografische Verfahren
    (Klassische Verfahren basierend auf mathematischen Problemen, welche auch Quantencomputer nicht effizient lösen können)

Wieso sind quanten-kryptografische Verfahren interessant?

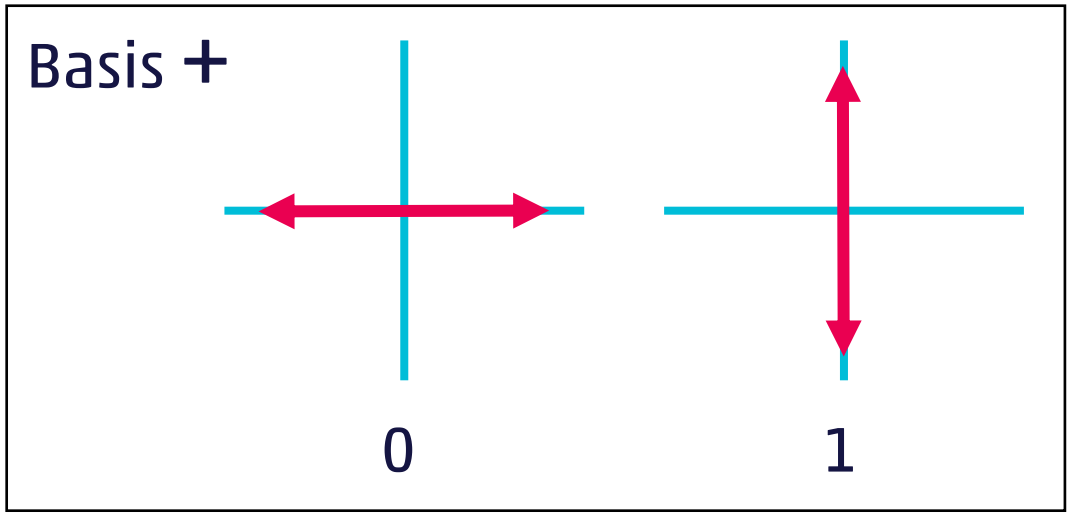
- Die Sicherheit von klassischen kryptografischen Verfahren basiert auf der (mutmasslichen) Schwierigkeit von mathematischen Problemen.
- Die Sicherheit von quanten-kryptografischen Verfahren basiert auf Gesetzen der Quanten-Physik.
- Beispiele
  - Quantum Random Number Generator (QRNG)
  - Quantum Key Distribution (QKD): BB84 Protokoll, E91 Protokoll

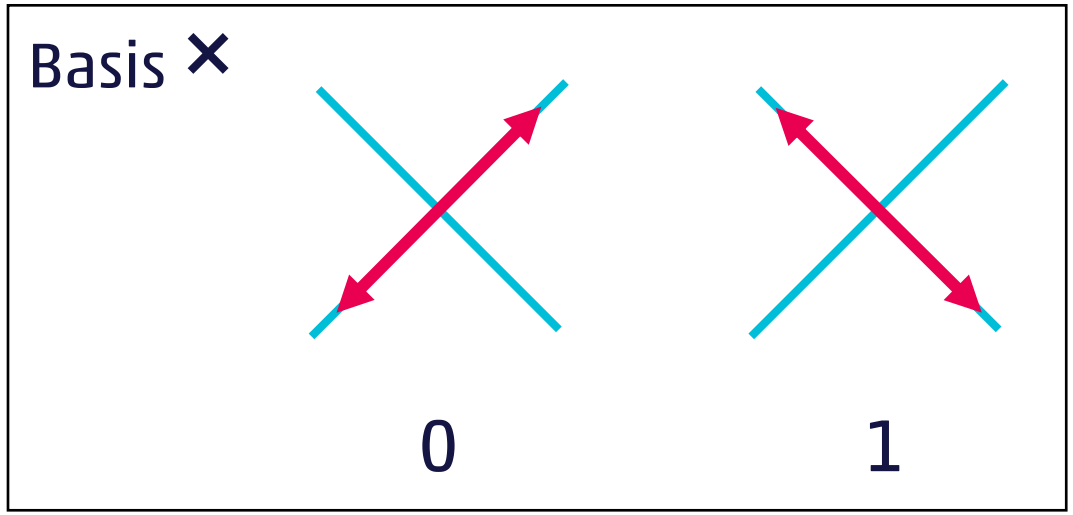
Quantum Key Distribution (QKD): BB84 Protokoll

- Erfinder: Charles H. Bennett und Gilles Brassard
- Jahr: 1984
- Annahme:

- Ziel: Alice und Bob etablieren einen gemeinsamen geheimen Schlüssel (Key).

Quantum Key Distribution (QKD): BB84 Protokoll





- Falls man in der falschen Basis misst, so erhält man einen zufälligen Wert
- Messungen beeinflussen die Polarisierung

Quantum Key Distribution (QKD): BB84 Protokoll

- Alice wählt zufällige Bits
- Alice wählt für jedes Bit zufällig eine Basis
- Alice polarisiert Photonen entsprechend den gewählten Bits und Basen, und sendet sie Bob
- Bob wählt für jedes Photon eine zufällige Basis und misst die Polarisierung. Falls er nicht dieselbe Basis wie Alice wählt, so erhält er bei der Messung einen zufälligen Wert.

Alice

Bits	1	0	0	1	0	1	1
Basis	+	+	×	×	+	×	×

Basis	0	1
+	↔	↕
×	↗	↘

Bob

Basis	+	+	+	×	×	+	×
Messung	1	0	1	1	0	0	1

Quantum Key Distribution (QKD): BB84 Protokoll

- Bob sendet Alice für jedes Bit die verwendete Basis über den authentischen Kommunikationskanal
- Alice weiss nun welche Bits von Bob korrekt dekodiert wurden.

Alice

Bits	1	0	0	1	0	1	1
Basis	+	+	×	×	+	×	×
Bobs Basis	+	+	+	×	×	+	×

korrekt dekodierte Bits

Bob

Basis	+	+	+	×	×	+	×
Messung	1	0	1	1	0	0	1

Quantum Key Distribution (QKD): BB84 Protokoll

- Alice teilt Bob mit, welche Bits er korrekt dekodiert hat.
- Wenn Eve die Bits auf dem Quanten-Kommunikationskanal abfängt und misst, dann führt dies zu Abweichungen in den korrekt dekodierten Bits von Alice und Bob.
- Alice sendet einen Teil der korrekt dekodierten Bits über den authentischen Kommunikationskanal. Bob prüft diese und teilt Alice mit, ob er dieselben Werte erhalten hat.
- Falls die Werte übereinstimmen, verwenden Alice und Bob die übrigen korrekt dekodierten Bits als gemeinsamen Schlüssel.

Alice

Bits	1	0	0	1	0	1	1
Basis	+	+	×	×	+	×	×
Bobs Basis	+	+	+	×	×	+	×

korrekt dekodierte Bits

Bob

Basis	+	+	+	×	×	+	×
Messung	1	0	1	1	0	0	1

Jetzt auf Quantum Key Distribution umsteigen?

Wohl eher (noch) nicht …

- Vorteile
  - Sicherheit basiert auf Gesetzen der Quanten-Physik
- Nachteile
  - (klassischer) authentischer Kommunikationskanal erforderlich
  - zusätzliche spezielle Hardware erforderlich → limitiertes Einsatzgebiet
  - relativ teuer (~100’000 CHF pro QKD-System mit zwei Endpunkten)
  - limitierte Distanz (~100 km, gewisse Systeme bis zu 600 km)
  - fehlende Standardisierung → fehlende Interoperabilität
  - Sicherheit der Systeme nicht so gut erforscht

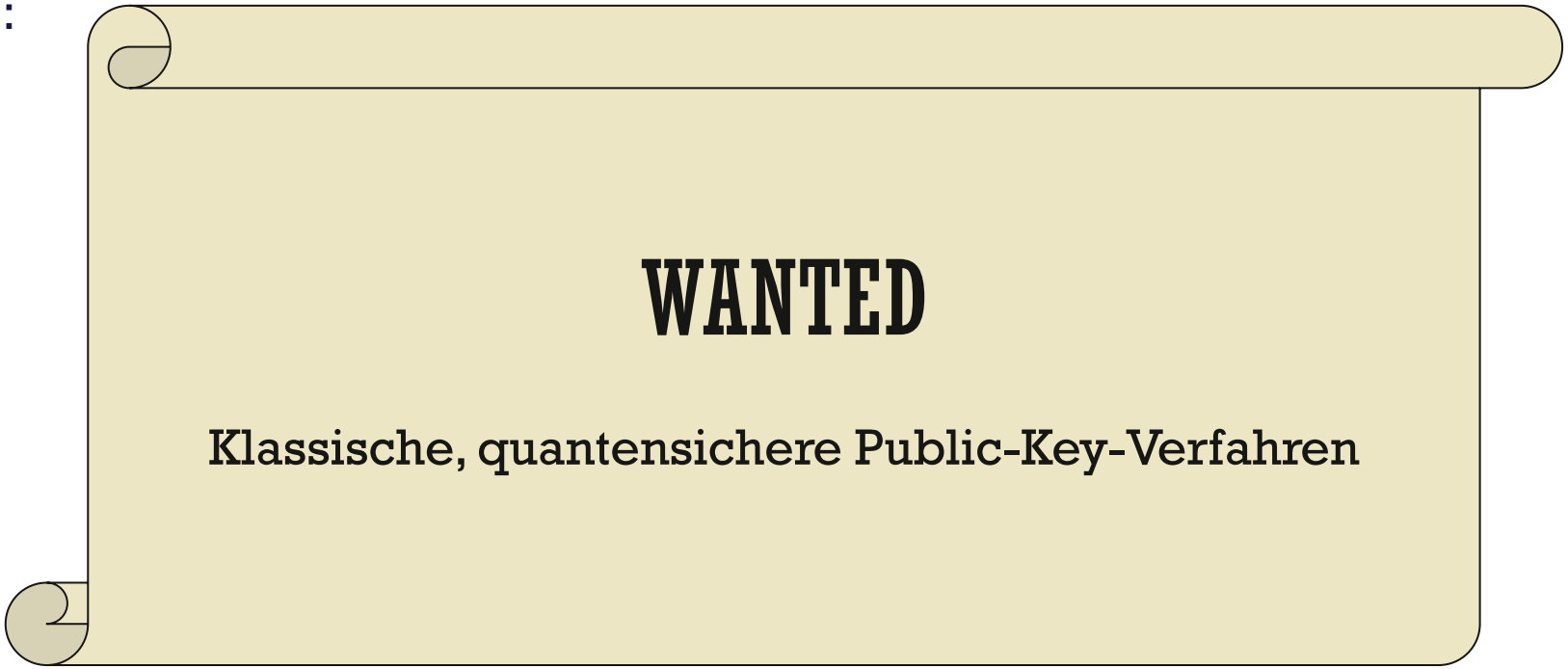
Jetzt auf Quantum Key Distribution umsteigen?

"... NSA views quantum-resistant (or post-quantum) cryptography as a more cost effective and easily maintained solution than quantum key distribution. For all of these reasons, NSA does not support the usage of QKD or QC to protect communications in National Security Systems, and does not anticipate certifying or approving any QKD or QC security products for usage by NSS customers unless these limitations are overcome."

https://www.nsa.gov/Cybersecurity/Quantum-Key-Distribution-QKD-and-Quantum-Cryptography-QC/

Post-Quantum Cryptography (PQC)

- Quantencomputer brechen gängige asymmetrische (Public-Key) Verfahren
- Deshalb:



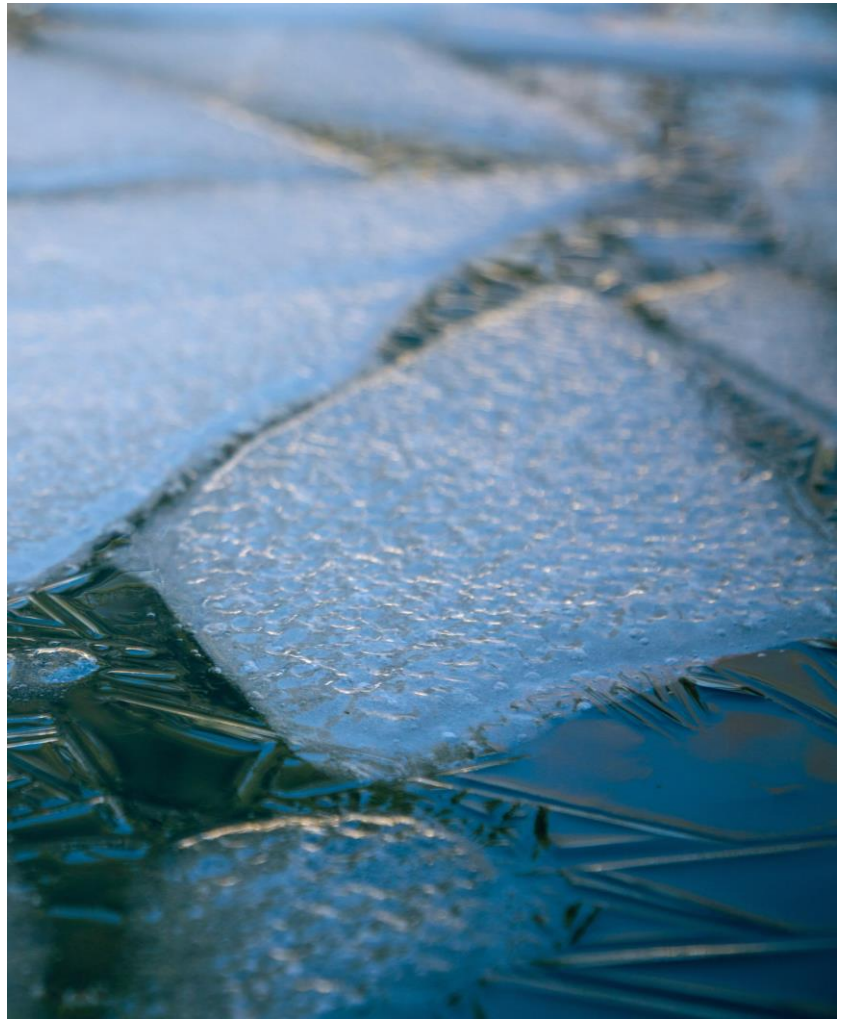
- Note: Quantencomputer brechen gängige symmetrische Verfahren **nicht**

  → Kein Handlungsbedarf (ausser Verdoppelung der Schlüssellängen)

Public-Key-Kryptografie

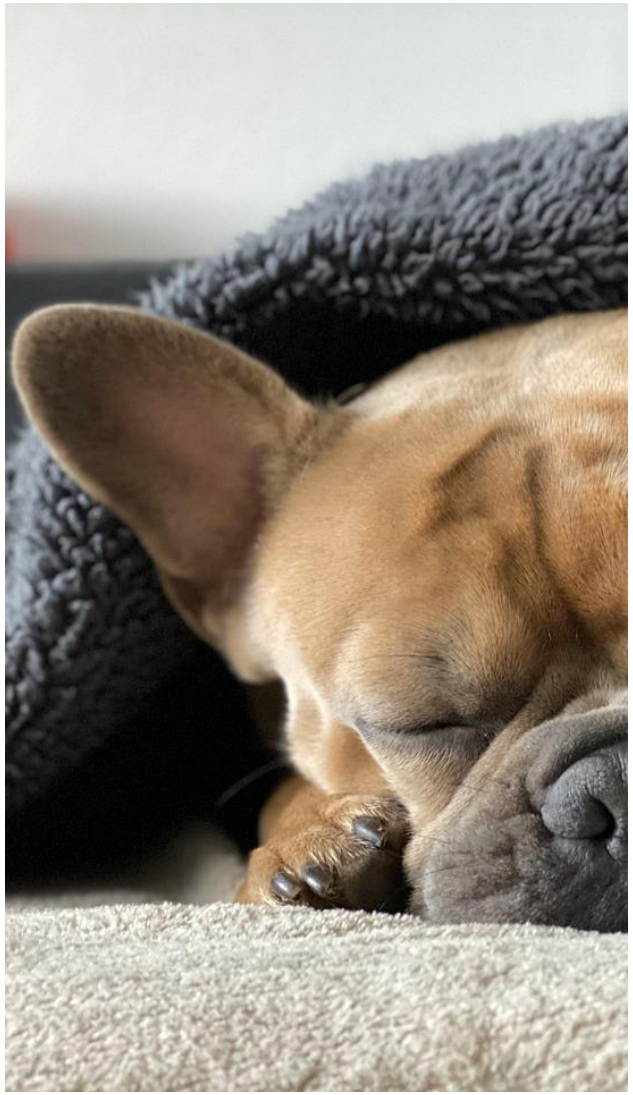
- Für Public-Key-Verfahren brauchen wir eine Funktion
  - einfach zu berechnen («verschlüsseln»)
  - schwierig zu invertieren («dechiffrieren») ausser man kennt die «Trapdoor» (bzw. den privaten Schlüssel)

- Die gängigsten Public-Key-Verfahren basieren auf dem Diskreten-Logarithmus- oder Faktorisierungsproblem:
  - nicht quantensicher
  - nicht beweisbar schwierig (auch klassisch nicht!)



Kryptografen-Traum

- Kryptografen-Traum (schon vor der Gefahr durch Quantencomputer):
  - Ein Krypto-Verfahren, dessen Sicherheit auf einem beweisbar schwierigen Problem basiert
- Es gibt solche Probleme, z.B.:
  - Decoding General Linear Codes
  - Gitter-Probleme (Lattice-problems)
  - Knapsack-Problem
- Auch darauf basierende Verfahren gibt es schon, z.B.:
  - McEliece (1978, unbroken)
  - Merkle-Hellman (1978, broken 1983)
- Glücklicherweise gelten diese auch in der Quantenwelt als sicher



NIST PQC-Standardisierung

WANTED

Klassische, quantensichere

- Signieralgorithmen
- Asymmetrische Chiffrierung / Key Encapsulation

Start 2017 / End (hopefully) 2024

NIST PQC: Stand nach drei Runden (Juli 2022)

Table 4. Algorithms to be Standardized

Public-Key Encryption/KEMs	Digital Signatures
CRYSTALS–KYBER	CRYSTALS–Dilithium
	FALCON
	SPHINCS$^+$

Table 5. Candidates advancing to the Fourth Round

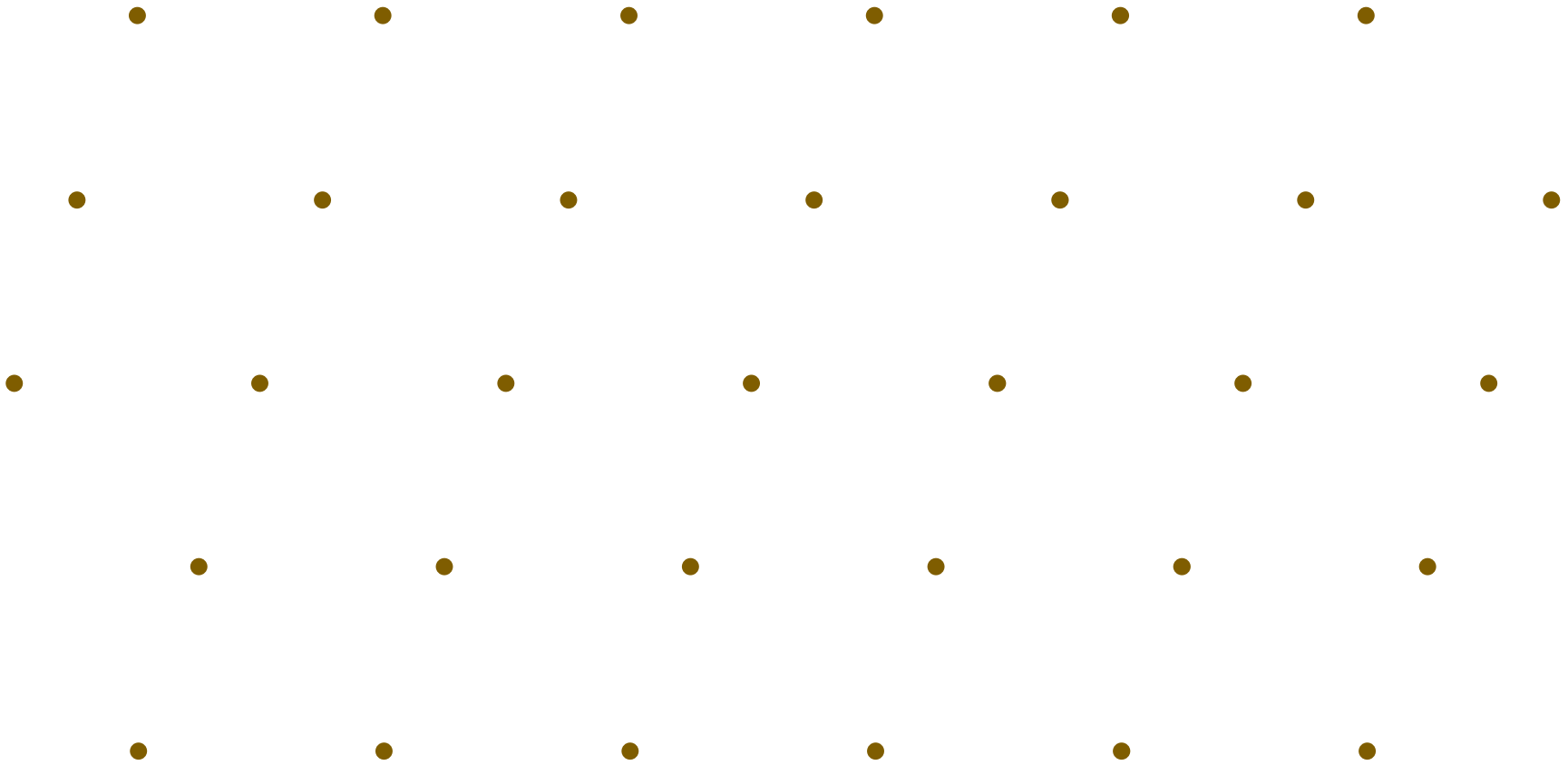
Public-Key Encryption/KEMs	Digital Signatures
BIKE	
Classic McEliece	
HQC	
SIKE	

- Lattice-based
- Hash-based
- Code-based
- Isogeny-based

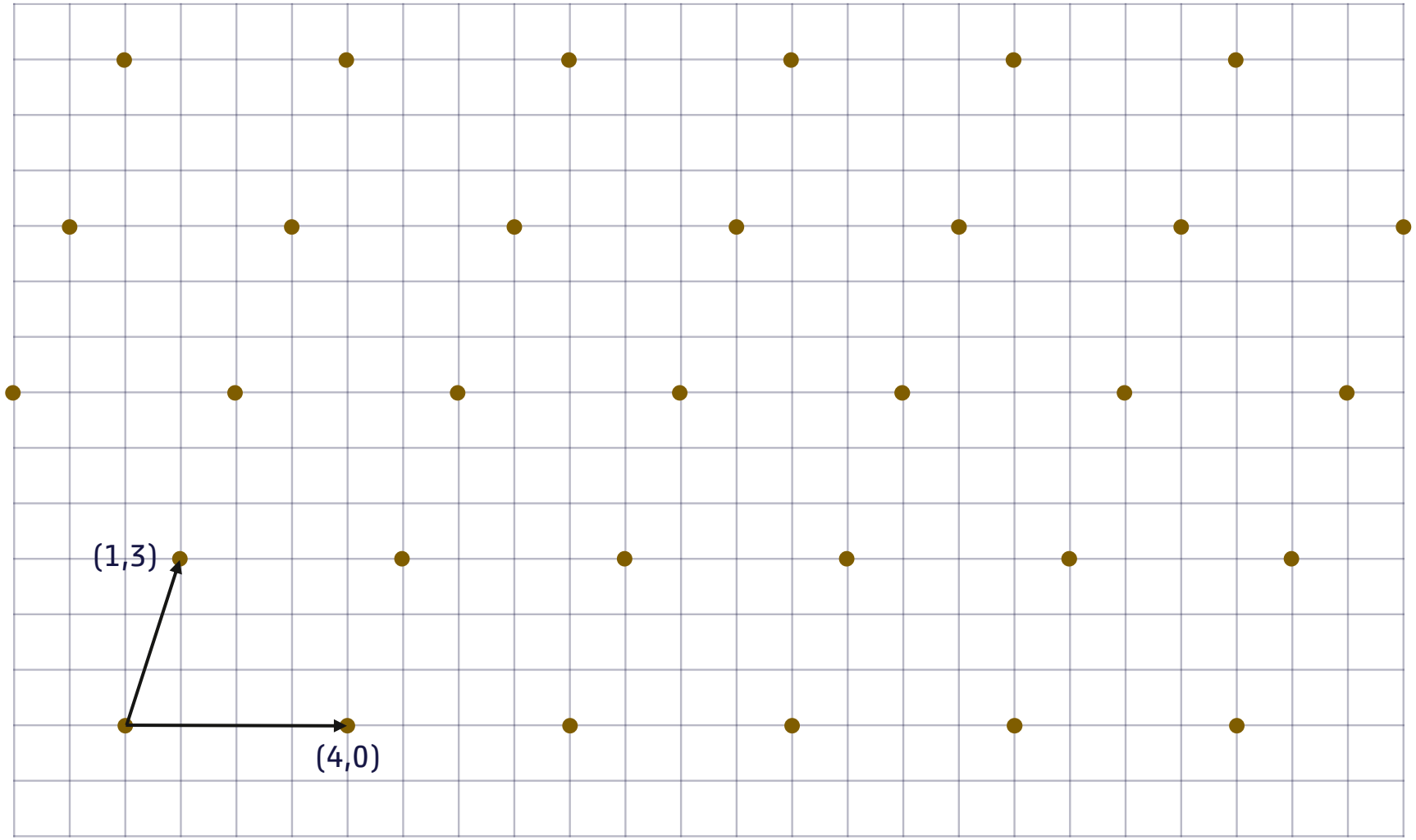
- Algorithmen zur Standardisierung ausgewählt:
  - Request for Comments für FIPS-Drafts (24. August 2023)
- Basieren grösstenteils auf Gitterproblemen: Klumpenrisiko
- SPHINCS+ trotz Nachteilen gegenüber den gitterbasierten Systemen ausgewählt
- call for additional digital signature proposals (closed July 2023) welche in der vierten Runde angeschaut werden
- Alternativen für CRYSTALS-KYBER werden weiter evaluiert

Quelle: Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Prozess

Gitter (Lattice)

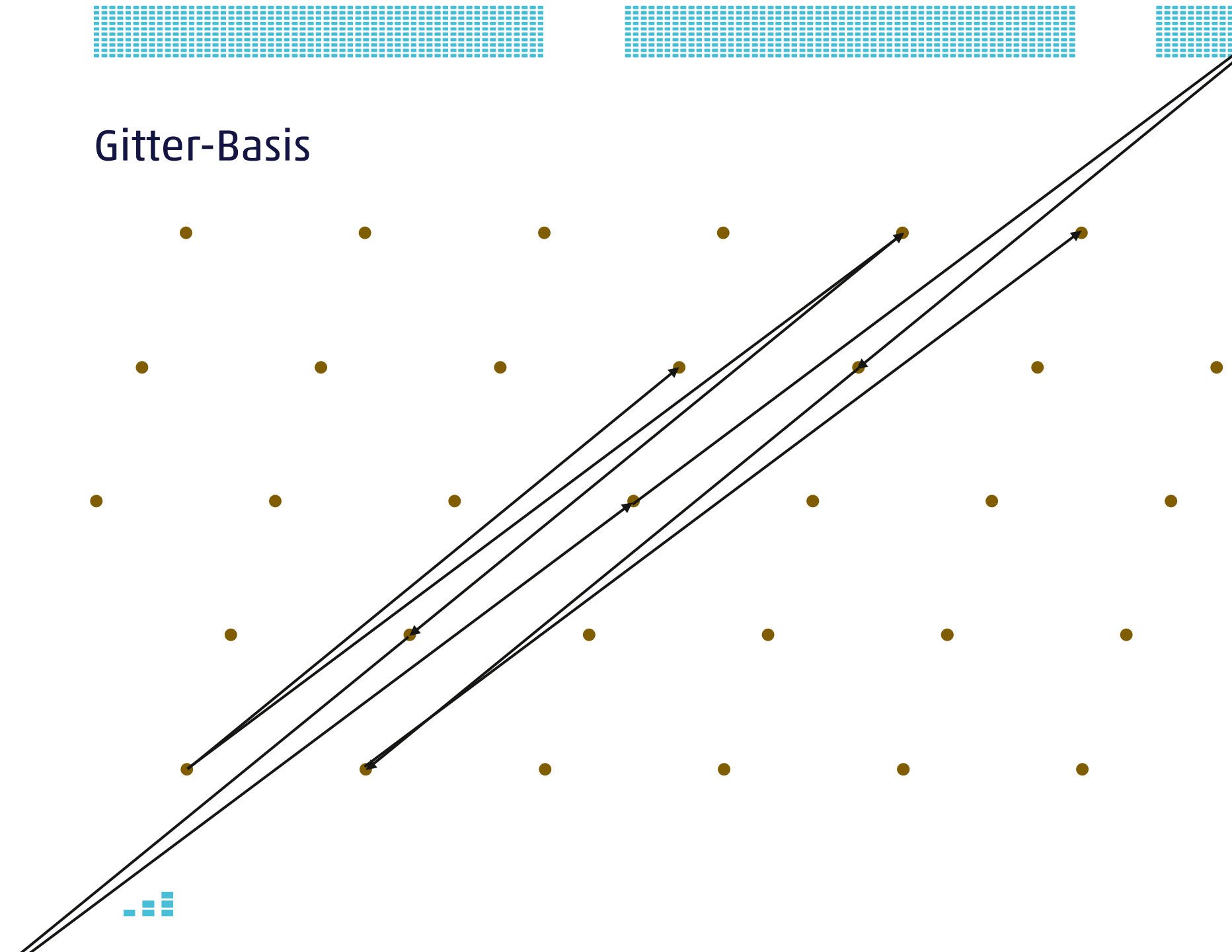


Gitter-Basis



- (1,3), (4,0) bilden eine Basis für das Gitter
- Jeder Punkt kann als Integer-Linearkombination der zwei Basisvektoren dargestellt werden

Gitter-Basis



- (1,3), (4,0) bilden eine Basis für das Gitter
- (11,9), (16,12) bilden eine alternative Basis für dasselbe Gitter

SVP und CVP

Shortest Vector Problem (SVP)

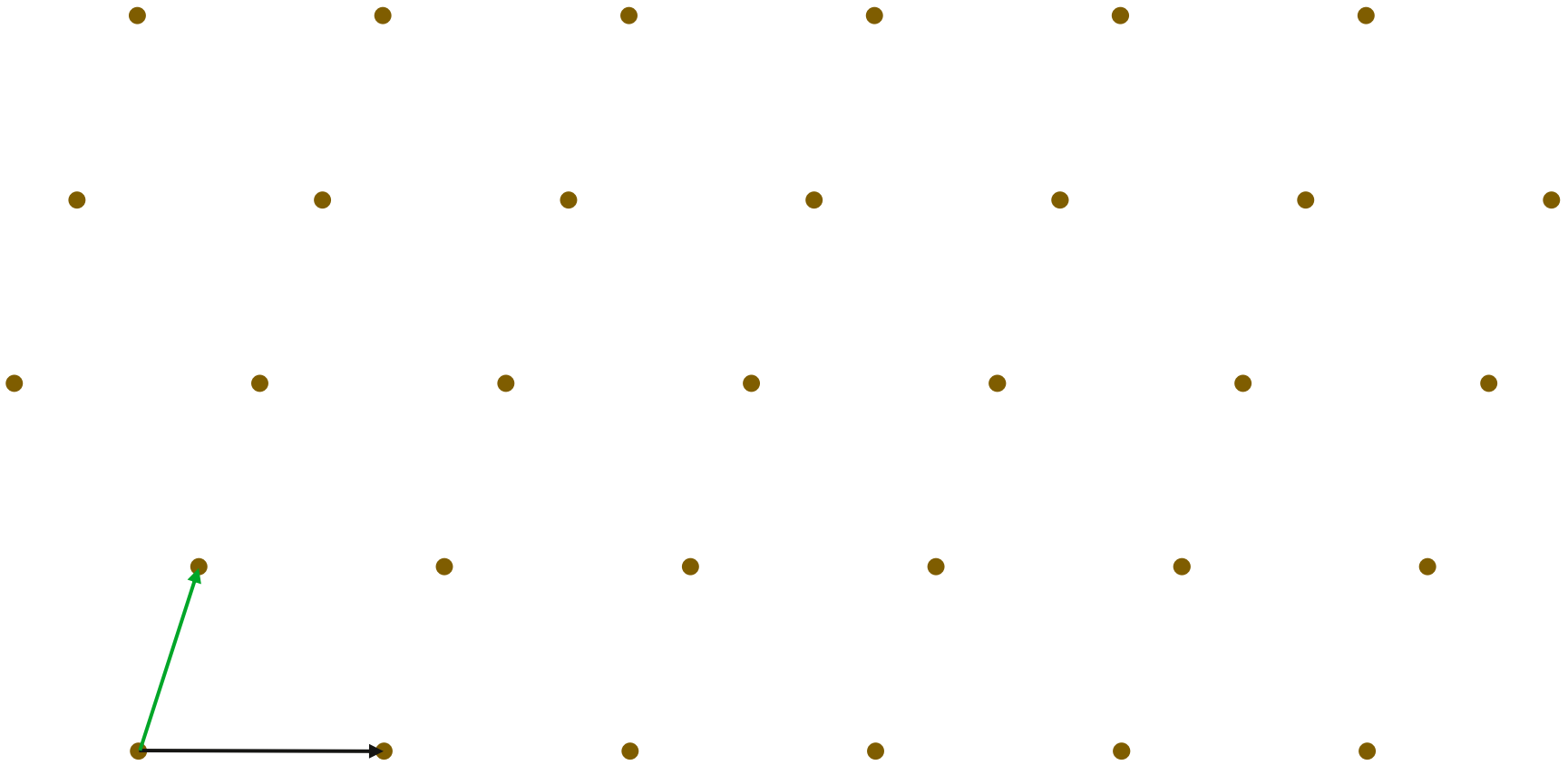
finde den kürzesten Abstand zwischen zwei Gitterpunkten

Closest Vector Problem (CVP)

gegeben einen Punkt im Raum, finde den nähesten Gitterpunkt

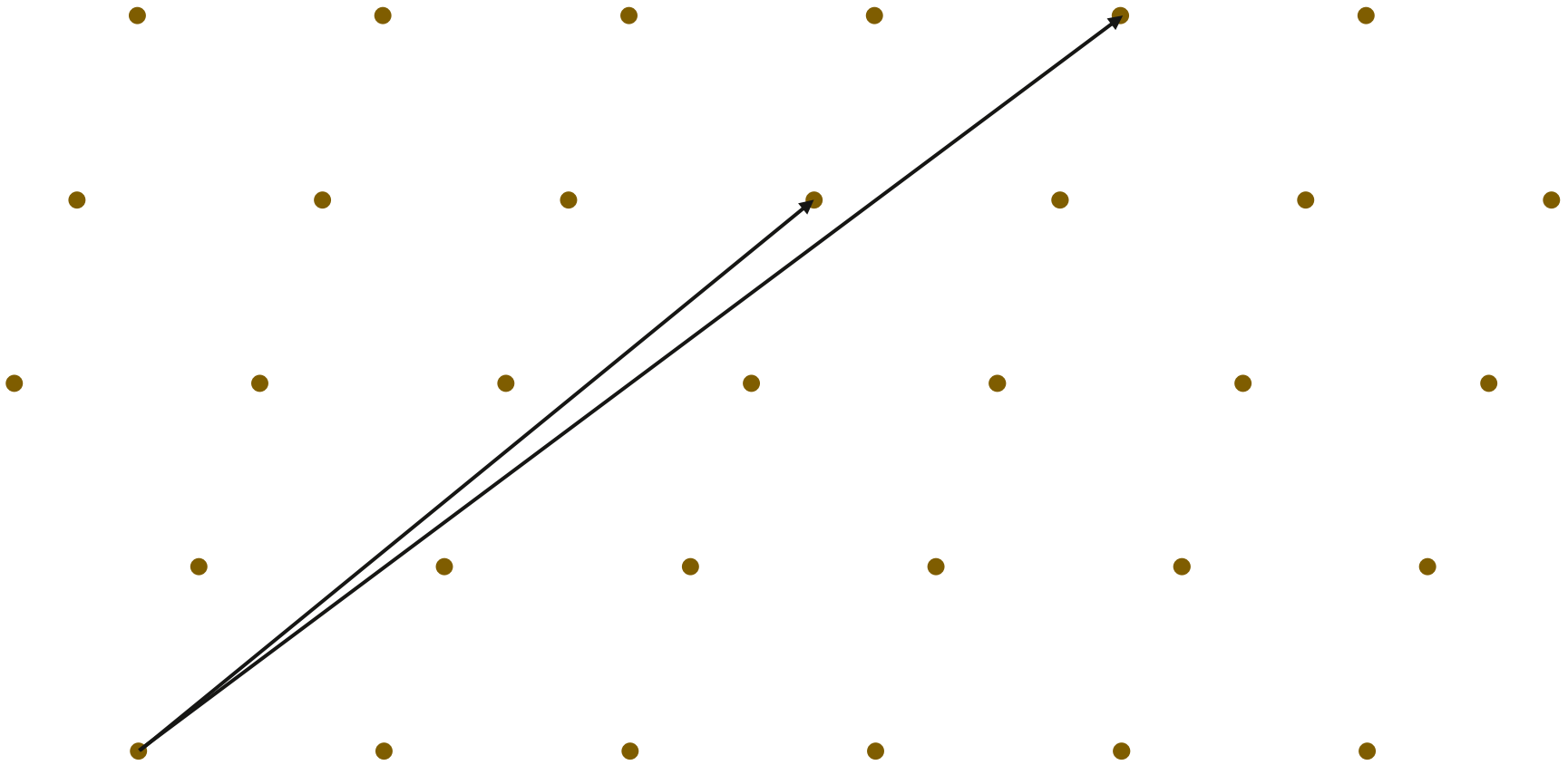
Beweisbar schwierige Probleme!

Shortest Vector – «gute Basis»



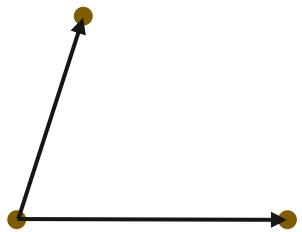
- Mit Basis (1,3), (4,0) ist es einfach, den kürzesten Vektor zu bestimmen
- (1,3), (4,0) ist eine **«gute Basis»**

Shortest Vector – «schlechte Basis»

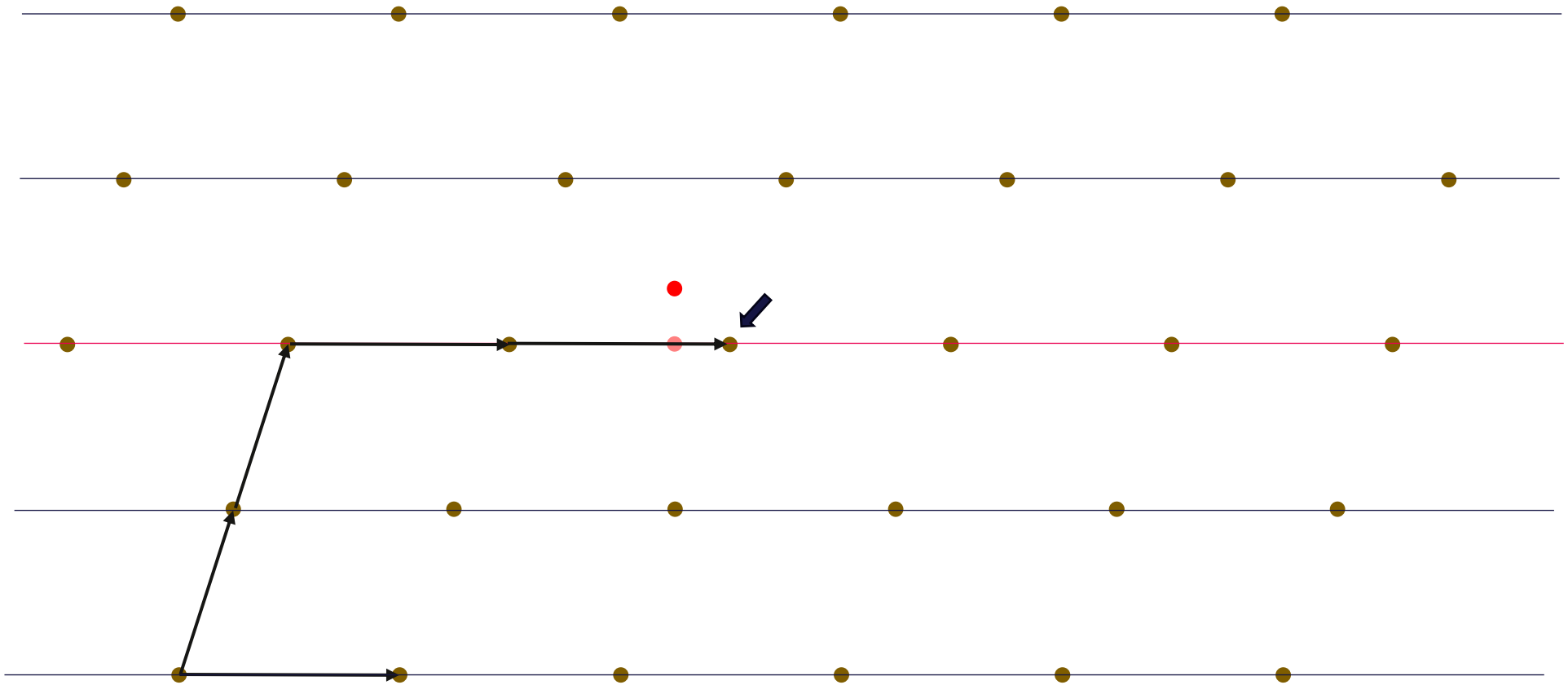


- Mit Basis (11,9), (16,12) ist es **nicht** so einfach, den kürzesten Vektor zu bestimmen
- (11,9), (16,12) ist eine **«schlechte Basis»**
- **Achtung:** Die Schwierigkeit ist in der Dimension des Gitters!

Closest Vector – «gute Basis»

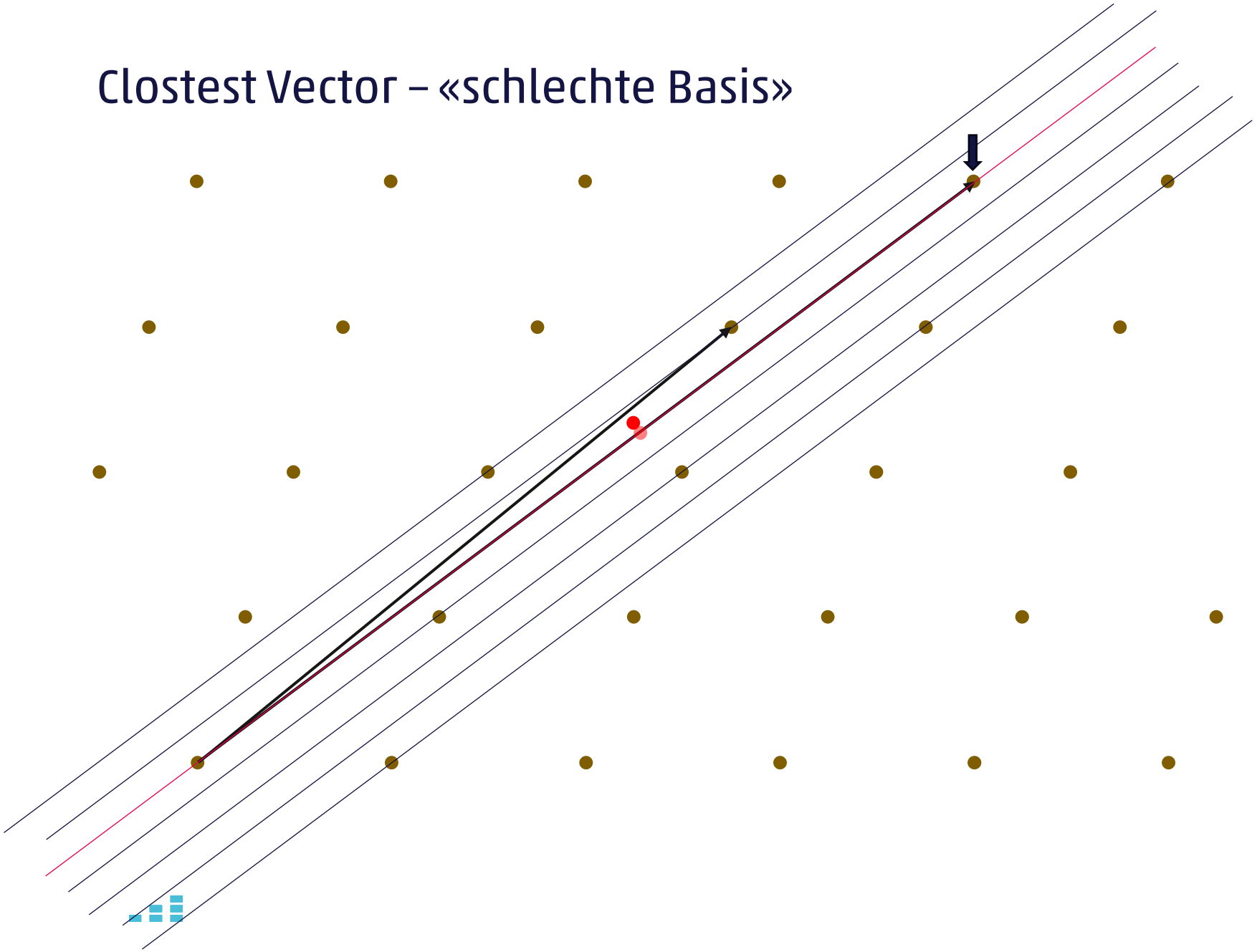


Closest Vector – «gute Basis»



- Mit einer guten Basis findet «Babai's Nearest Plane Algorithm» den nähesten Gitterpunkt

Clostest Vector – «schlechte Basis»



- Mit einer **schlechten Basis** findet «Babai's Nearest Plane Algorithm» den nähesten Gitterpunkt **nicht**

Gitter-basiertes Kryptosystem

- Recall: Schwierigkeit von CVP und SVP abhängig von der «Güte» der Basis
- Idee für ein Gitter-basiertes Kryptosystem (GGH, 1997):
  - **Public Key**: Schlechte Basis für ein Gitter
  - **Secret Key**: Gute Basis für ein Gitter
  - **Verschlüsselung**:
    - Encodiere die Nachricht als Gitterpunkt
    - Füge einen kleinen Fehler hinzu
  - Nur der Besitzer einer guten Basis kann die Nachricht **entschlüsseln** (das CVP lösen)
- Dieses System wurde gebrochen (1999)
- Mittlerweile bessere Konstruktionen. «Brechen» des Systems bedeutet ein CVP zu lösen: «Beweisbar sicher».

Ruhig schlafen?

«Theoretisch gibt es keinen Unterschied zwischen Theorie und Praxis. Praktisch aber schon.»

(Autor uneindeutig)

Vielen Dank für Ihre Aufmerksamkeit_

Martin Kaufmann, Urs Wagner

info@cnlab-security.ch
+41 55 214 33 40

cnlab security AG
Obere Bahnhofstrasse 32b
CH-8640 Rapperswil-Jona
Switzerland

cnlab