"T.I.S.P. Community Meeting 2021"

Berlin, 03.-04.11.2021

Post-quantum Cryptography

Marian Margraf, Freie Universität Berlin

Gefährdung verschlüsselter Kommunikation durch Quantencomputer

Timeline

1994	Peter Shor entdeckt Quanten-Algorithmus, der Faktorisierung und diskreten Logarithmus effizient berechnen kann.
2013	Snowden-Leaks: Die Konstruktion eines kryptographisch nutzbaren Quantencomputer ist Teil eines 80 Mio. Forschungsprogramms der NSA mit dem Namen “*Penetrating Hard Targets”.*
?	NSA nutzt Quantencomputer, um verschlüsselte Nachrichten von politischer, wirtschaftlicher oder gesellschaftlicher Relevanz auszuwerten.

Wann wird es Quantencomputer von hinreichender Größe geben?

- Um RSA zu brechen, benötigt Shors Algorithmus ~2(N+1) Qbits, wobei N der Bitlänge des RSA Moduls entspricht. Um RSA-2048 zu brechen, werden also 4098 logische Qbits benötigt, für RSA-3072 logische 6146 Qbits.
- Da Quantenzustände fehleranfällig sind (Dekohärenz), müssen logische Qbits aus einer Vielzahl von physischen Qbits konstruiert werden.
- Die Konstruktion von 4098 logischen Qbits würde vermutlich Millionen von physischen Qbits benötigen.

Wann wird es Quantencomputer von hinreichender Größe geben?

- *Quantum Supremacy* beschreibt den Zeitpunkt ab dem Quantencomputer Berechnungen durchführen können, die für klassische Computer nicht effizient ausführbar sind.
- Dieser Punkt wurde 2019 erreicht, als Google mit einem 53-Qbit-Computer speziell für diesen Zweck konstruierte Aufgaben gelöst hat, die für einen klassischen Computer nicht effizient berechenbar sind.
- Aufgrund der Struktur der durchgeführten Berechnungen mussten die 53 Qbits nicht fehler-korrigiert werden. Ein Quantencomputer, der 53 *logische* Qbits zur Verfügung stellt, ist also noch weiter entfernt.
- Quantencomputer-Experte **Michele Mosca** hat 2015 folgende Einschätzung gegeben**:** *“At present [...] I estimate a **1/7 chance of breaking RSA-2048 by 2026 and a 1/2 chance by 2031.**”*

Welche Gegenmaßnahmen können zum Schutz der Verschlüsselung ergriffen werden?

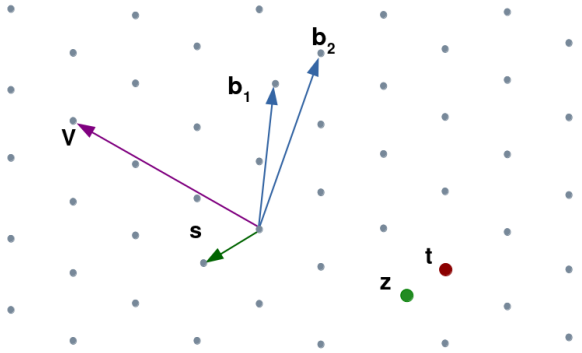
- Lässt sich RSA härten, indem wir die Schlüssellänge erhöhen? Nein. Shor's Algorithmus hat asymptotisch die gleiche Laufzeit-Komplexität wie die Schlüsselgenerierung von RSA.
  - N = Bitlänge des RSA Moduls
  - Laufzeit Shor's Algorithmus: $O(N^2 \log N \log \log N)$
  - Laufzeit Schlüsselgenerierung RSA: $O(N^2)$
- Das Gleiche gilt für Kryptographie die auf dem diskreten Logarithmus basiert. Um weiterhin sicher verschlüsseln zu können, müssen daher Post-Quanten-Kryptographische Protokolle eingesetzt werden, die auf anderen mathematischen Primitiven basieren, z.B. Gitter, Codes oder Isogenien.

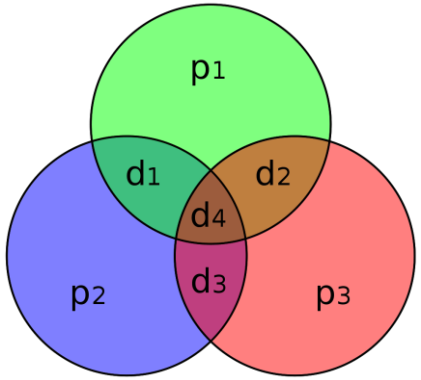
Gegenmaßnahme: Entwicklung quantensicherer Kryptosysteme

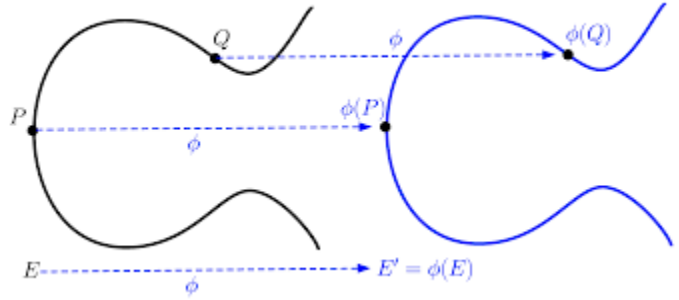
NIST Post-Quantum Cryptography Competition

- In 3 Runden werden innerhalb von ~6-8 Jahren Quantencomputer-resistente Kryptosysteme zur Standardisierung ausgewählt (gestartet 12/2016) .

- Mathematische Primitive:
  a. Gitter-basierte Verfahren (Sicherheit reduzierbar zu SVP)
  b. Code-basierte Verfahren (Sicherheit reduzierbar zu Dekodierungsproblem)
  c. Isogenie-basierte, Hash-basierte und andere







NIST Post-Quantum Cryptography Standardization
Runde 3

Schlüsselaustausch

- Finalisten
  - Classic McEliece (Codes)
  - CRYSTALS-KYBER (Gitter)
  - NTRU (Gitter)
  - SABER (Gitter)
- Alternative Kandidaten
  - BIKE (Codes)
  - FrodoKEM (Gitter)
  - HQC (Codes)
  - NTRU Prime (Gitter)
  - SIKE (Isogenien)

Signaturen

- Finalisten
  - CRYSTALS-DILITHIUM (Gitter)
  - FALCON (Gitter)
  - Rainbow (Multivariate)
- Alternative Kandidaten
  - GeMSS (Multivariate)
  - Picnic (Zero-Knowledge-Proofs)
  - SPHINCS+ (Hash)

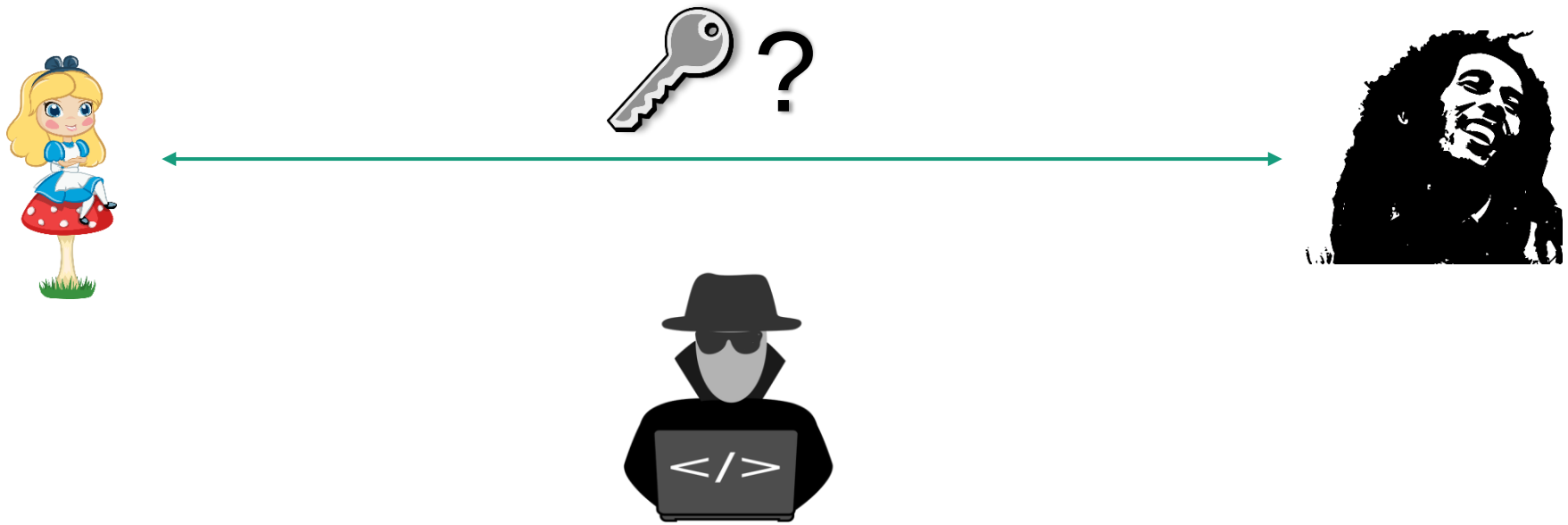
Aktuelle Forschungsansätze @ Fraunhofer AISEC

- PQDB: Vergleich der NIST Kandidaten.
- Quantencomputer-basierte Kryptoanalyse der neuen Primitive.
- Erweiterung der Kryptobibliothek Botan mit quantencomputer-resistenten Kryptoalgorithmen (gefördert durch BMBF).
- Quantencomputer-resistente Kryptoverfahren für hoheitliche Dokumente (gefördert durch BMWi).
- Entwurf einer quantencomputer-resistenten PKI (gefördert durch BMBF).

Supersingular Isogeny Diffie-Hellman (SIDH)

- Einzig bekanntes PQC-Verfahren, das Diffie-Hellman Struktur hat
- Ist die Basis für das Verfahren „SIKE“ in Runde 3 der NIST PQC Standardisierung
- 2011 von De Feo, Jao und Plut veröffentlicht
- Vorteile:
  - Diffie-Hellman Struktur
  - Vergleichsweise kleine Schlüssel
- Nachteile
  - Operationen dauern (derzeit) vergleichsweise lang
  - Verfahren ist noch nicht so alt

Motivation



Auffrischung: Diffie-Hellman Schlüsselaustausch

Öffentliche Parameter:
Primzahl p, Generator g

wählt x zufällig — $A = g^x \bmod p$ →

← $B = g^y \bmod p$ — wählt y zufällig

berechnet $B^x \bmod p$ | berechnet $A^y \bmod p$

$$A^y = (g^x)^y = g^{xy} = \boldsymbol{K} = g^{yx} = (g^y)^x = B^x$$

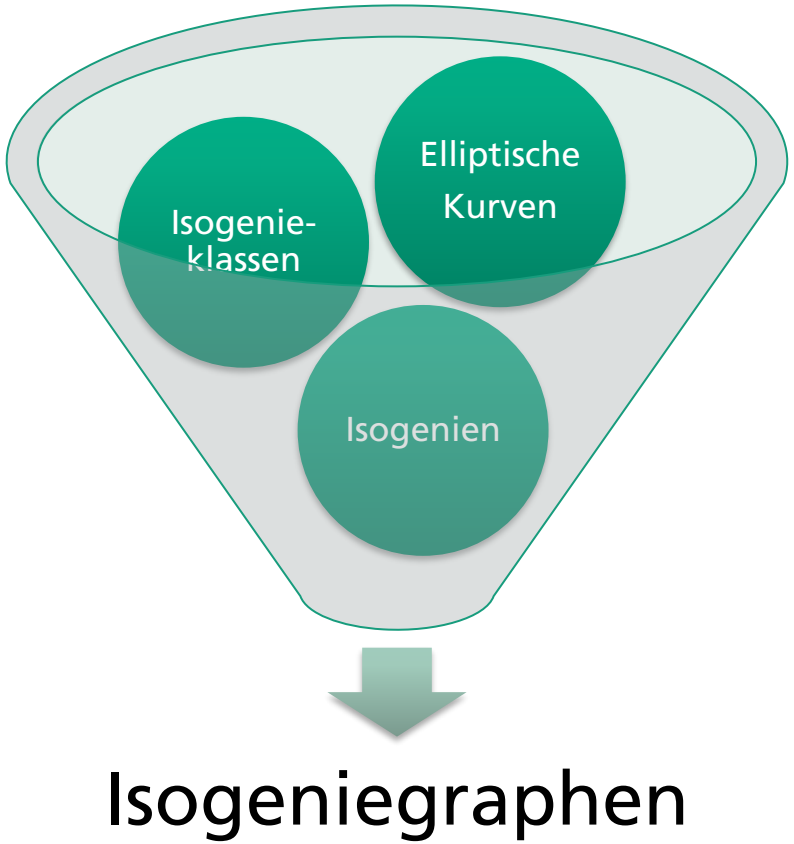
Das Problem

$$A = g^x \bmod p$$

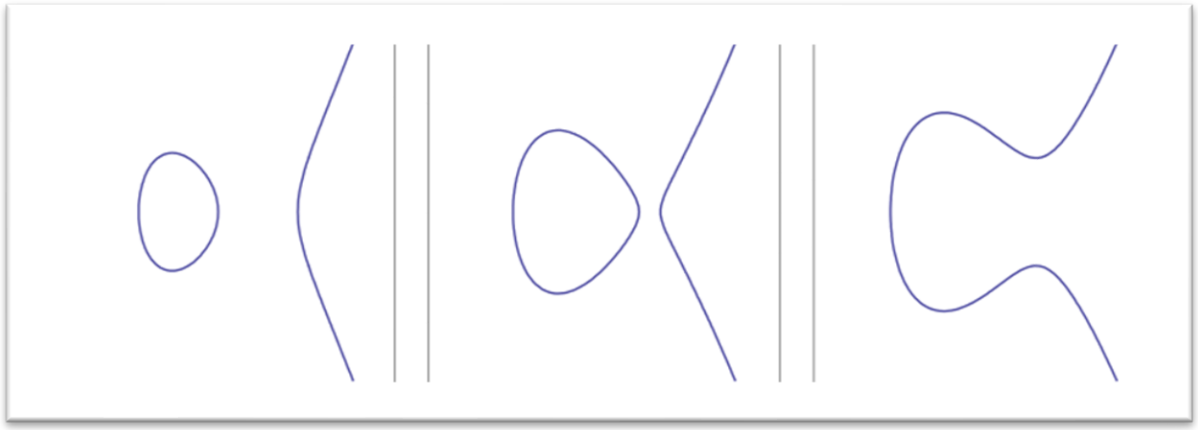
$$\log_g A = x$$

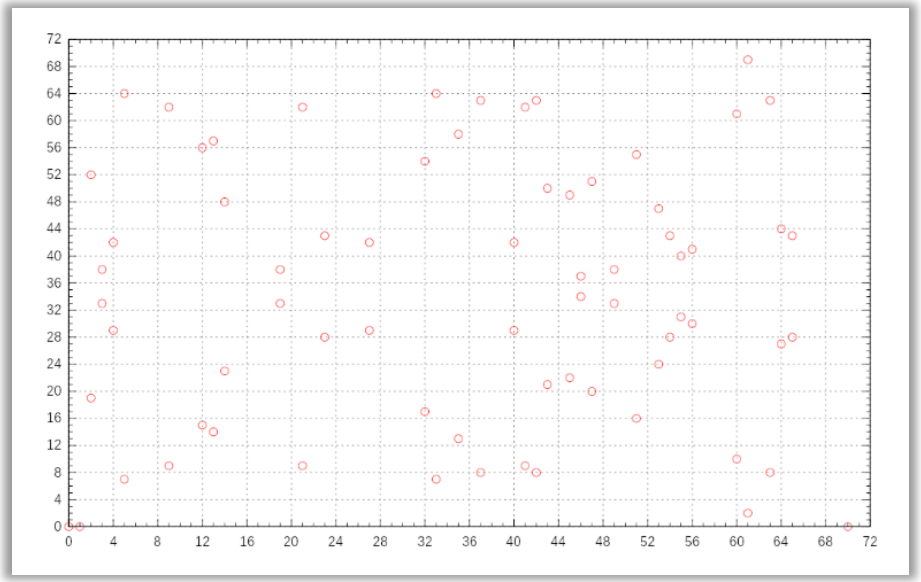
Der diskrete Logarithmus ist auf einem hinreichend großen Quantencomputer effizient berechenbar.

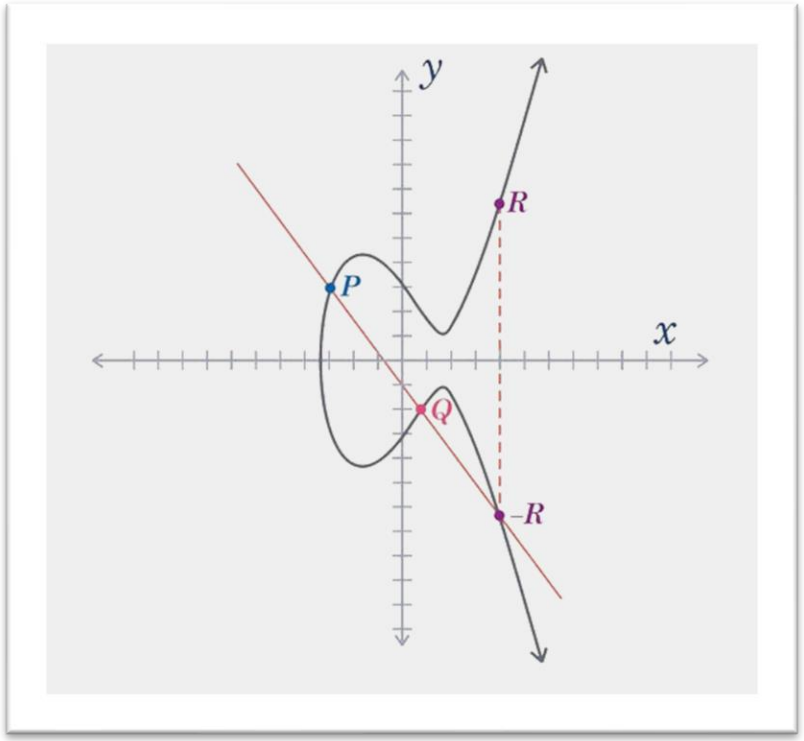
Die Zutaten für SIDH



Elliptische Kurven

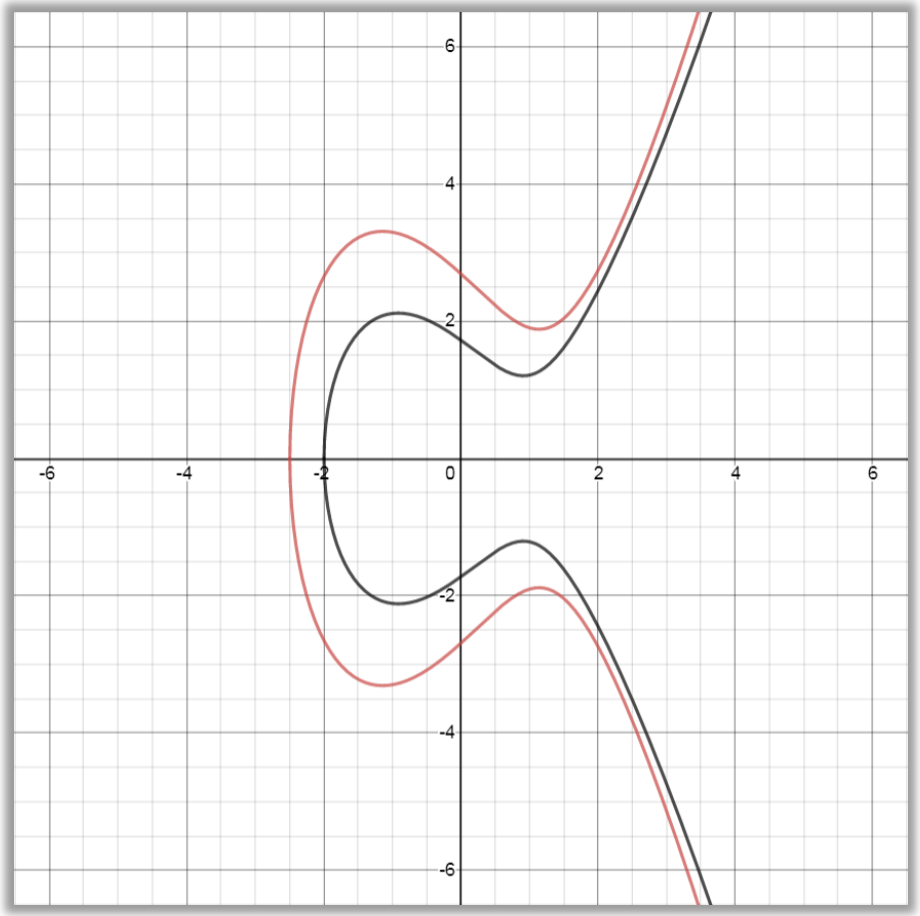




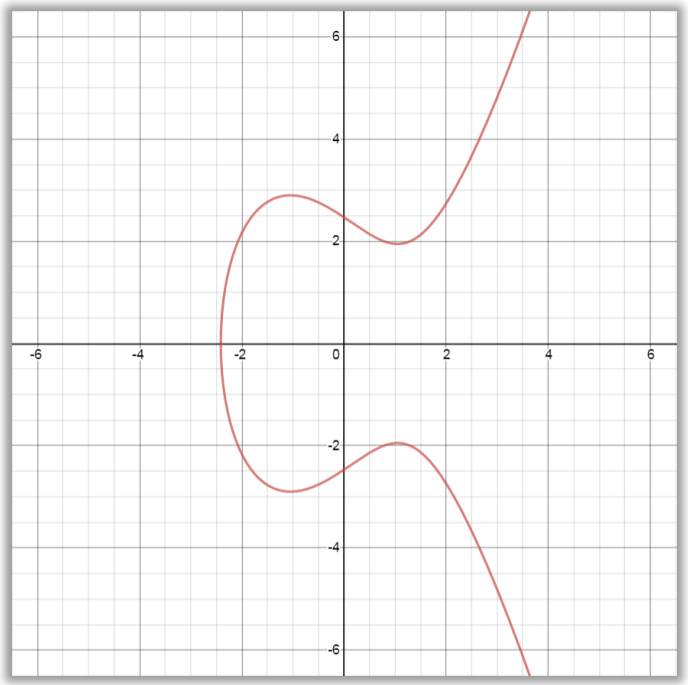


Isogenieklassen

- Isomorphe Kurven mit Kennziffer eindeutig identifizierbar (j-Invariante)
- Eine Isogenieklasse enthält alle Kurven mit derselben Kennziffer j

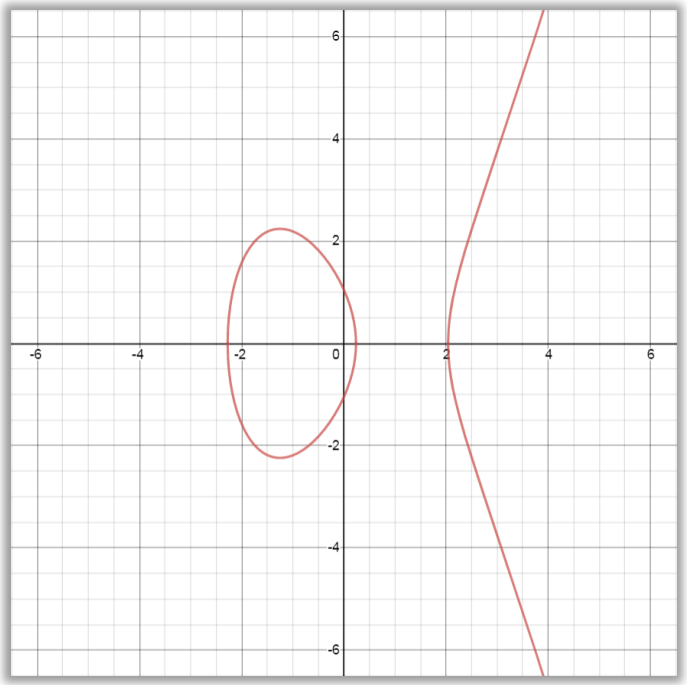


Isogenien auf supersingulären elliptischen Kurven



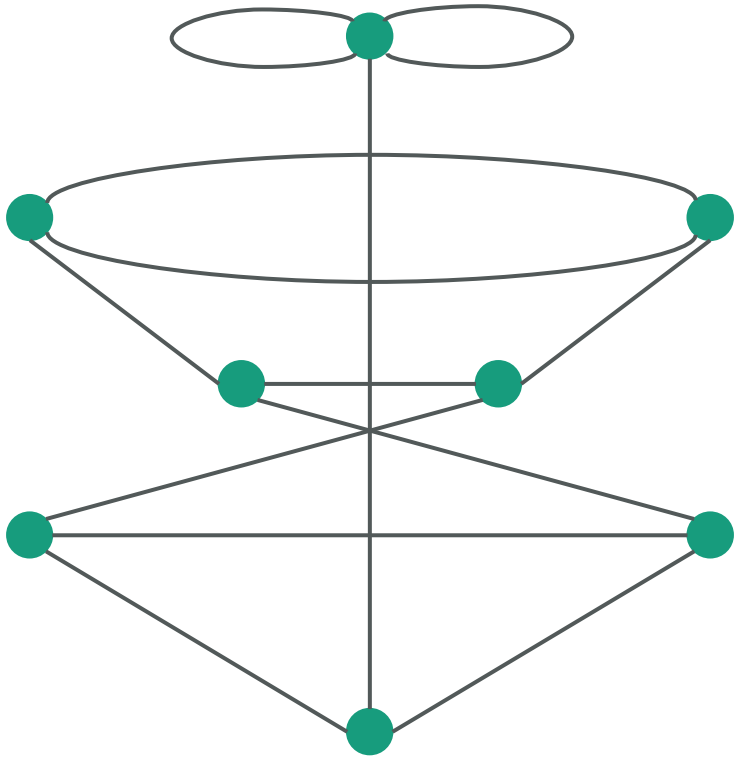
E

ϕ

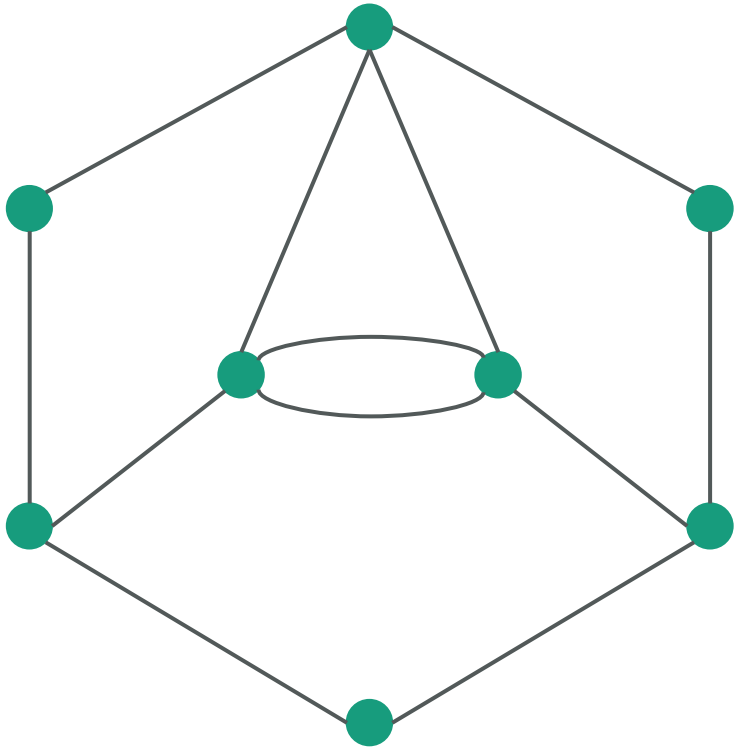


$\phi(E)$

Isogeniegraphen



Grad 2 Isogeniegraph über $\mathbb{F}_{97^2}$



Grad 3 Isogeniegraph über $\mathbb{F}_{97^2}$

Supersingular Isogeny Diffie-Hellman (SIDH)

Öffentliche Parameter:
Basispunkte P_A, P_B
Elliptische Kurve E

wählt ϕ_A zufällig
unter Benutzung von P_A

$E_A = \phi_A(E); \phi_A(P_B)$ →

← $E_B = \phi_B(E); \phi_B(P_A)$

wählt ϕ_B zufällig
unter Benutzung von P_B

berechnet ϕ'_A
unter Benutzung von ϕ_A und $\phi_B(P_A)$

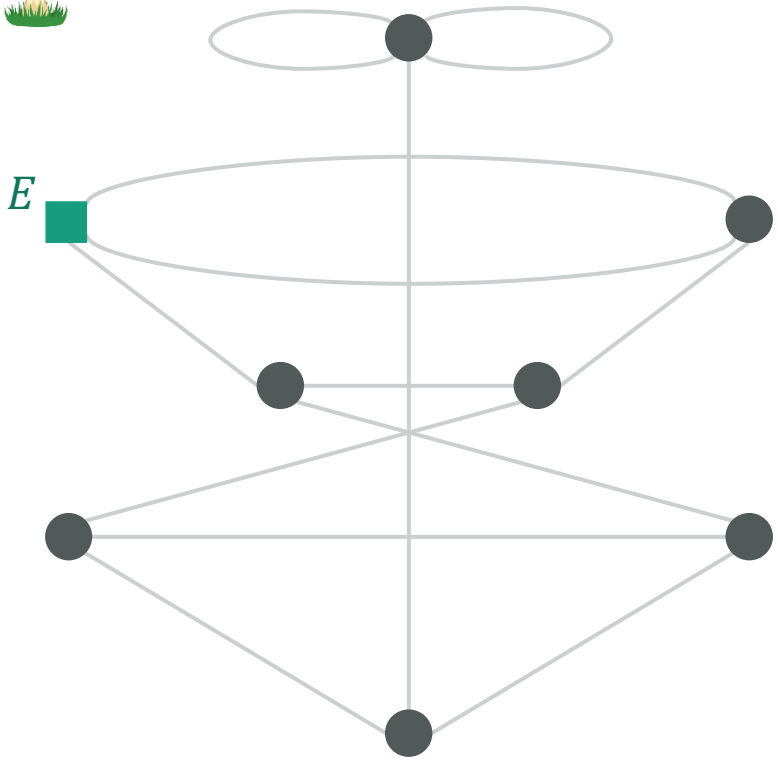
berechnet ϕ'_B
unter Benutzung von ϕ_B und $\phi_A(P_B)$

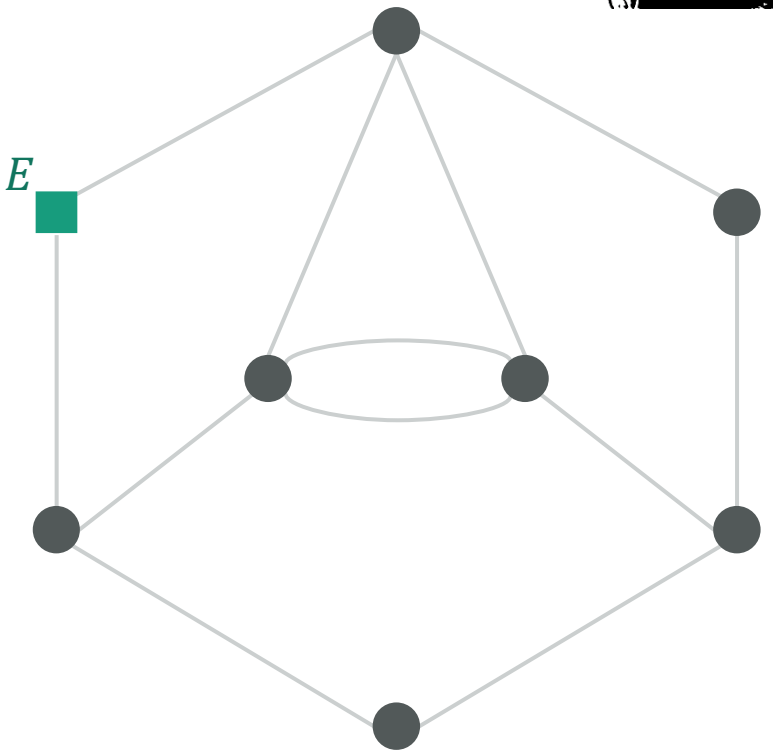
berechnet $\phi'_A(E_B)$

Berechnet $\phi'_B(E_A)$

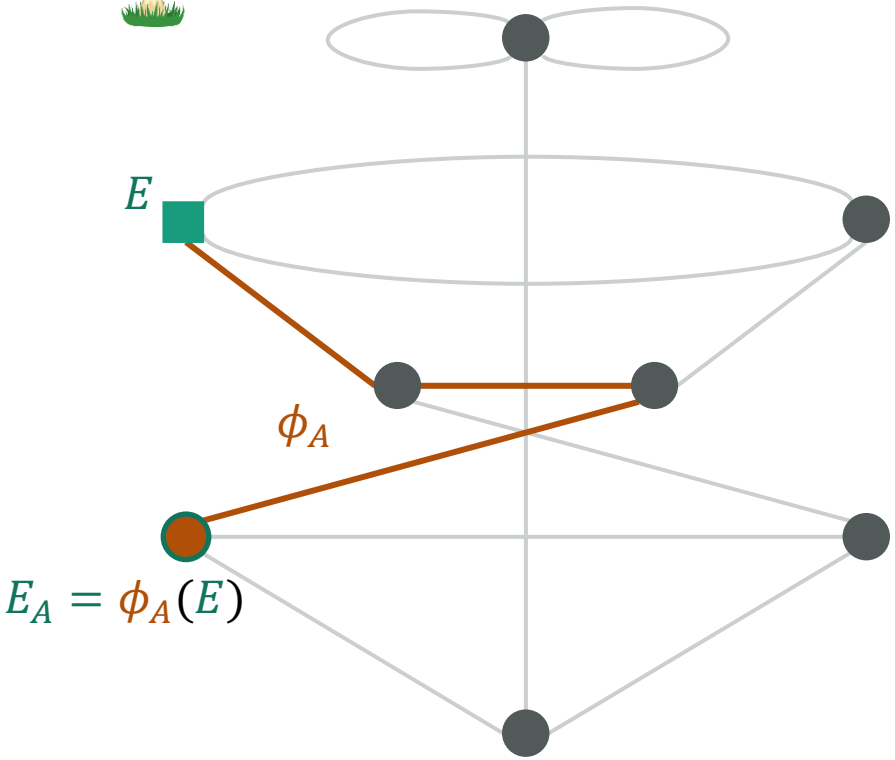
$$\phi'_A(E_B) = \phi'_A(\phi_B(E)) = \boldsymbol{E_{AB}} = \phi'_B(\phi_A(E)) = \phi'_B(E_A)$$

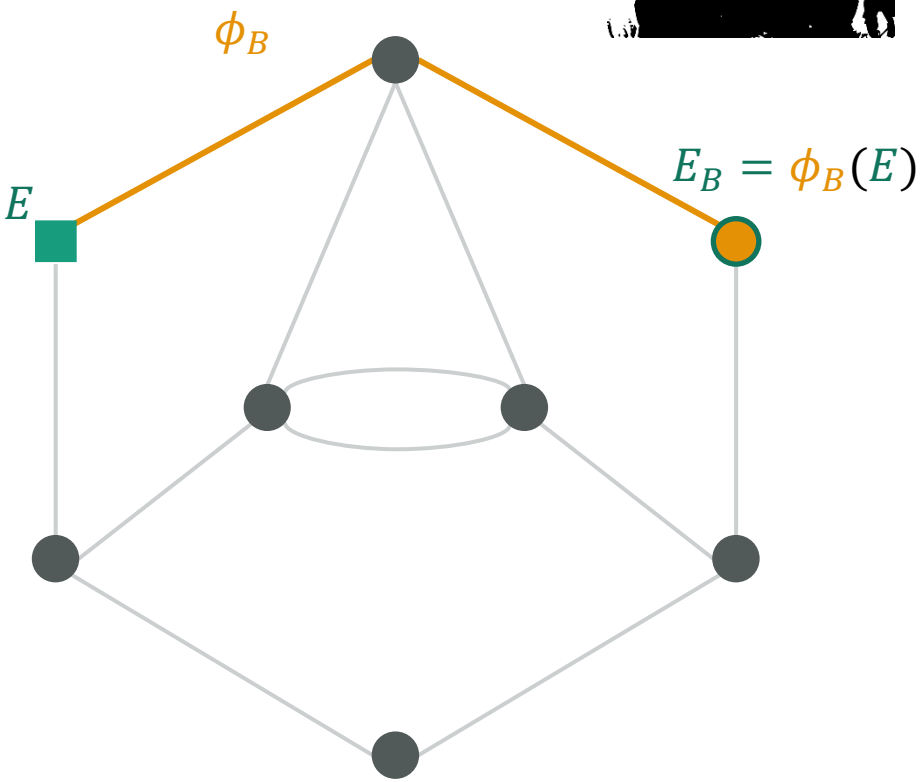
Supersingular Isogeny Diffie-Hellman (SIDH)



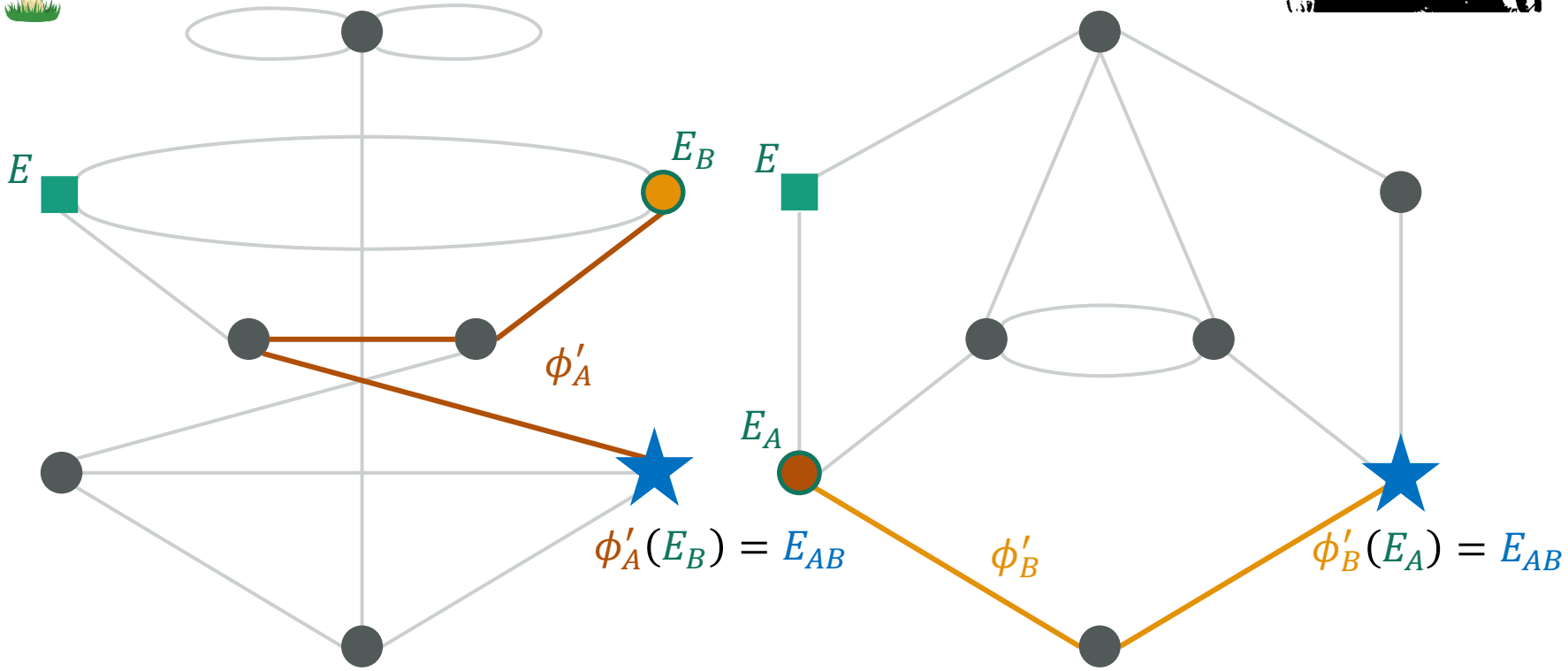


Supersingular Isogeny Diffie-Hellman (SIDH)





Supersingular Isogeny Diffie-Hellman (SIDH)



Diffie-Hellman mit verschiedenen Gruppen

	DH	ECDH	SIDH
Elemente	Ganzzahlen g modulo Primzahl	Punkte P in Kurvengruppe	Kurven E in Isogenieklasse
Geheimnisse	Exponenten x	Skalare k	Isogenien ϕ
Berechnungen	$g, x \mapsto g^x$	$k, P \mapsto [k]P$	$\phi, E \mapsto \phi(E)$
Schwere Probleme	Gegeben g, g^x finde x	Gegeben $P, [k]P$ finde k	Gegeben $E, \phi(E)$ finde ϕ
Post-Quanten sicher	Nein	Nein	Ja