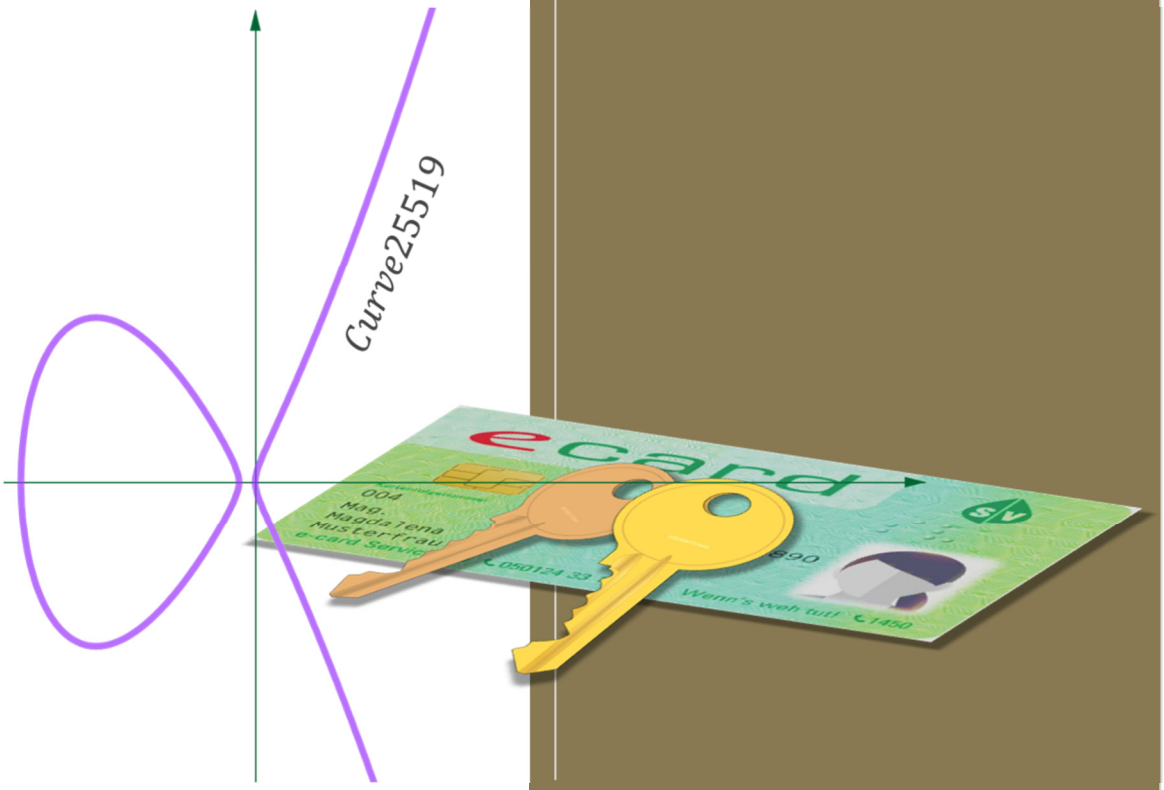
ECC

Elliptic-Curve-Cryptography



UHLMANN

Inhalt

Abkürzungsverzeichnis

A

AES *Advanced Encryption Standard*

C

CA *Certification Authority*

D

DH *Diffie-Hellman-Schlüsseltausch*
DLP *Discrete Logarithm Problem*
DSA *Digital Signature Algorithm*

E

EC *Elliptic Curve*
ECC *Elliptic-Curve-Cryptography*
ECDH *Elliptic Cryptography Diffie-Hellman*
ECDLP *Elliptic Curve Discrete Logarithm Problem*
ECDSA *Elliptic Curve Digital Signature Algorithm*

R

RSA *Rivest–Shamir–Adleman*

Einleitung

Elliptic-Curve-Cryptography (ECC) ist eine Methode asymmetrischer Verschlüsselung, mit der sich kryptographische Verfahren wie der Deffie-Hellman-Schlüsseltausch (DH) effizient realisieren lassen. Wie der Name schon sagt, beruht der Algorithmus auf sogenannten elliptischen Kurven. Die Bezeichnung ist etwas irreführend, es handelt sich nämlich nicht um Ellipsen, sondern um algebraische Kurven, die im Zuge der mathematischen Behandlung elliptischer Bogenlängen aufgetaucht sind.

Das Knacken einer ECC-Verschlüsselung ist deutlich schwieriger als das sonst übliche Faktorisieren großer Zahlen, auf die ansonsten viele Verschlüsselungsalgorithmen beruhen. Kryptosysteme auf Grundlage von elliptischen Kurven kommen daher bei vergleichbarer Sicherheit mit wesentlich kürzeren Schlüsseln aus.

Die Verwendung elliptischer Kurven für die Kryptografie wurde 1985 unabhängig voneinander von Neal Koblitz und Victor S. Miller vorgeschlagen. Heute sind sie in vielen Anwendungen verwirklicht. Einige Beispiele (Wikipedia):

- Windows-Betriebssysteme
- Firefox Browser
- e-card
- Bankomatkarte
- Reisepässe und Personalausweise zur Sicherung vor Zugriff auf den Chip
- Messenger Dienste wie WhatsApp
- Digitale Unterschrift in der Bitcoin Blockchain
- etc.

Elliptische Kurven

Eine sehr gebräuchliche elliptische Kurve trägt die Bezeichnung **Curve25519** und ist durch die Gleichung

$$E_{Curve25519}: \ y^2 = x^3 + 486662x^2 + x$$

gegeben. Sie ist im Messenger Dienst WhatsApp implementiert und schematisch in Abbildung 1 dargestellt. Im Reellen besteht die Kurve aus zwei getrennten Teilen, im Komplexen sind sie aber verbunden. Abbildung 1 zeigt auch schematisch **secp256k1**, die in der Public-Key-Kryptographie von Bitcoin verwendete elliptische Kurve

$$E_{secp256k1}: \ y^2 = x^3 + 7$$

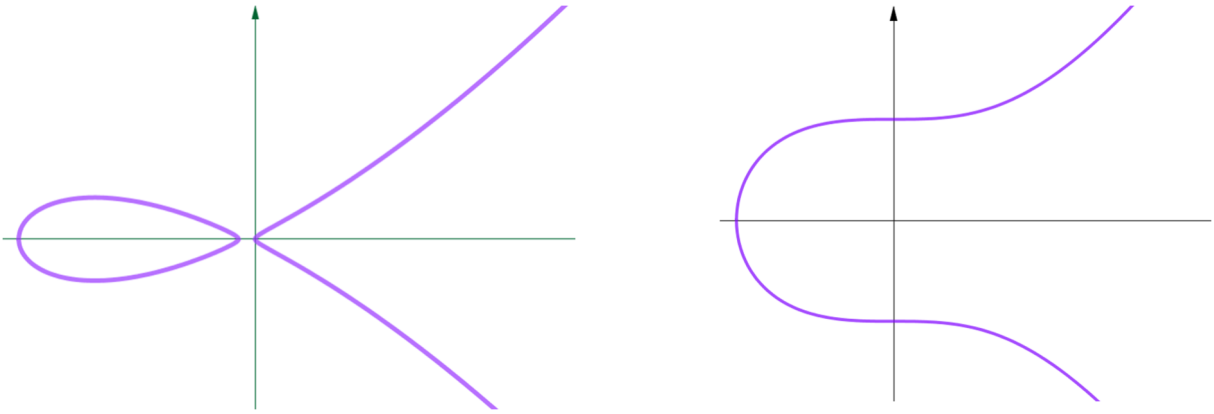


Abbildung 1: Curve25519 (links) und secp256k1 (rechts)

Curve25519 gehört zur Gruppe der sogenannten Montgomery-Kurven

$$By^2 = x^3 + Ax^2 + x \qquad (B \neq 0\,, A \neq \pm 2)\,,$$

während **secp256k1** eine sogenannte Weierstraß-Kurve

$$y^2 = x^3 + ax + b \qquad (4a^3 + 27b^2 \neq 0)$$

ist. Vergleicht man jedoch zum Beispiel die Montgomery-Kurve $y^2 = x^3 + 5x^2 + x$ mit der Weierstraß-Kurve $y^2 = x^3 - \frac{22}{3}x + \frac{205}{27}$, so erkennt man lediglich eine Parallelverschiebung (Abbildung 2).

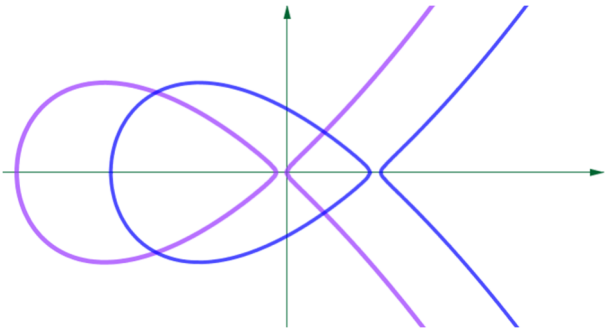


Abbildung 2

Es kann gezeigt werden, dass jede Montgomery-Kurve in eine bezüglich der Kryptographie gleichwertige Weierstraß-Form transformiert werden kann (siehe Anhang).

Dies gilt auch für noch andere Formen elliptischer Kurven, weshalb ***die Weierstraß-Form die allgemeinste Form elliptischer Kurven darstellt***.

Eine elliptische Kurve (EC) in Weierstraß-Form muss die Bedingung $4a^3 + 27b^2 \neq 0$ erfüllen. Ansonsten besitzt sie nämlich einen Knoten, was sie für die Anwendung in einem Kryptosystem ausschließt.

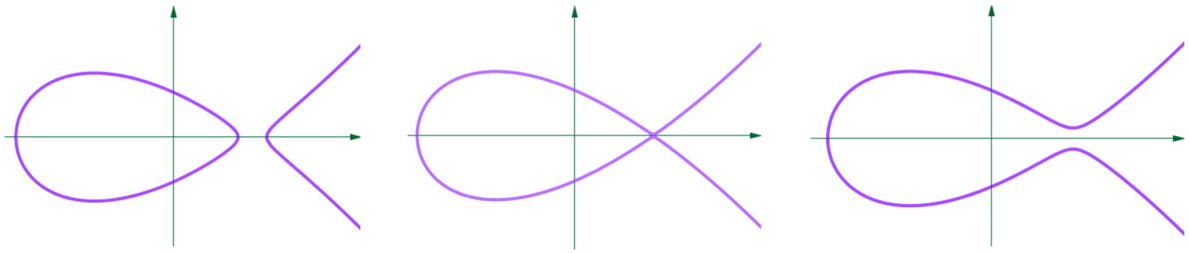


Abbildung 3: $y^2 = x^3 - 3x + 1,9$; $y^2 = x^3 - 3x + 2$!!! ; $y^2 = x^3 - 3x + 2,1$

Elliptischen Kurven über einem Körper

Betrachten wir elliptische Kurven der allgemeinen Weierstraß-Form

$$y^2 = x^3 + a \cdot x + b.$$

Dabei muss noch geklärt werden, wofür die Variablen x, y, a, b und das Additions- bzw. Multiplikationszeichen stehen. Selbstredend haben wir angenommen, dass der Körper der reellen Zahlen, also $(\mathbb{R}, +, \cdot)$, die Grundlage aller Berechnungen ist. Ansonsten hätten wir die Graphen nicht so einfach darstellen können. Der vollständige „Titel" lautet also:

Elliptische Kurven über dem Körper der reellen Zahlen $(\mathbb{R}, +, \cdot)$.

Bezüglich $(\mathbb{R}, +, \cdot)$ gibt es unendlich viele Lösungen (x, y). Grafisch dargestellt liegen sie dicht aneinander und ergeben somit die Linien. Zum Beispiel

$$\underbrace{y^2}_{LS(linke\ Seite)} = \underbrace{x^3 + 2x + 2}_{RS(rechte\ Seite)}$$

Im Gegensatz dazu:

Elliptische Kurven über dem Restklassenkörper $(\mathbb{Z}_p, +, \cdot)$

Für p wird – wie üblich in der Kryptographie – gerne eine Primzahl verwendet.
Über dem Körper $(\mathbb{Z}_{17}, +, \cdot)$ gibt es nur 18 Lösungen (x, y). Grafisch dargestellt ergeben sie eine Punktwolke. Die strichlierte Verbindungslinie in Abbildung 4 ist willkürlich und gehört nicht zur Lösung.

$$\underbrace{y^2}_{LS(linke\ Seite)} \equiv_{\mathrm{mod}\,17} \underbrace{x^3 + 2x + 2}_{RS(rechte\ Seite)}$$

$LS_{17} = RS_{17}$	1	1	1	1	1	1	2	2	2	2	2	2	9	9	15	15	16	16
x	3	3	5	5	9	9	0	0	7	7	10	10	6	6	13	13	16	16
y	1	16	1	16	1	16	6	11	6	11	6	11	3	14	7	10	4	13
Nur 18 Pkte	(3;1)	(3;16)	(5;1)	(5;16)	(9;1)	(9;16)	(0;6)	(0;11)	(7;6)	(7;11)	(10;6)	(10;11)	(6;3)	(6;14)	(13;7)	(13;10)	(16;4)	(16;13)

Tabelle 1: In $\mathbb{Z}_{17}$ gibt es nur endlich viele Lösungen.

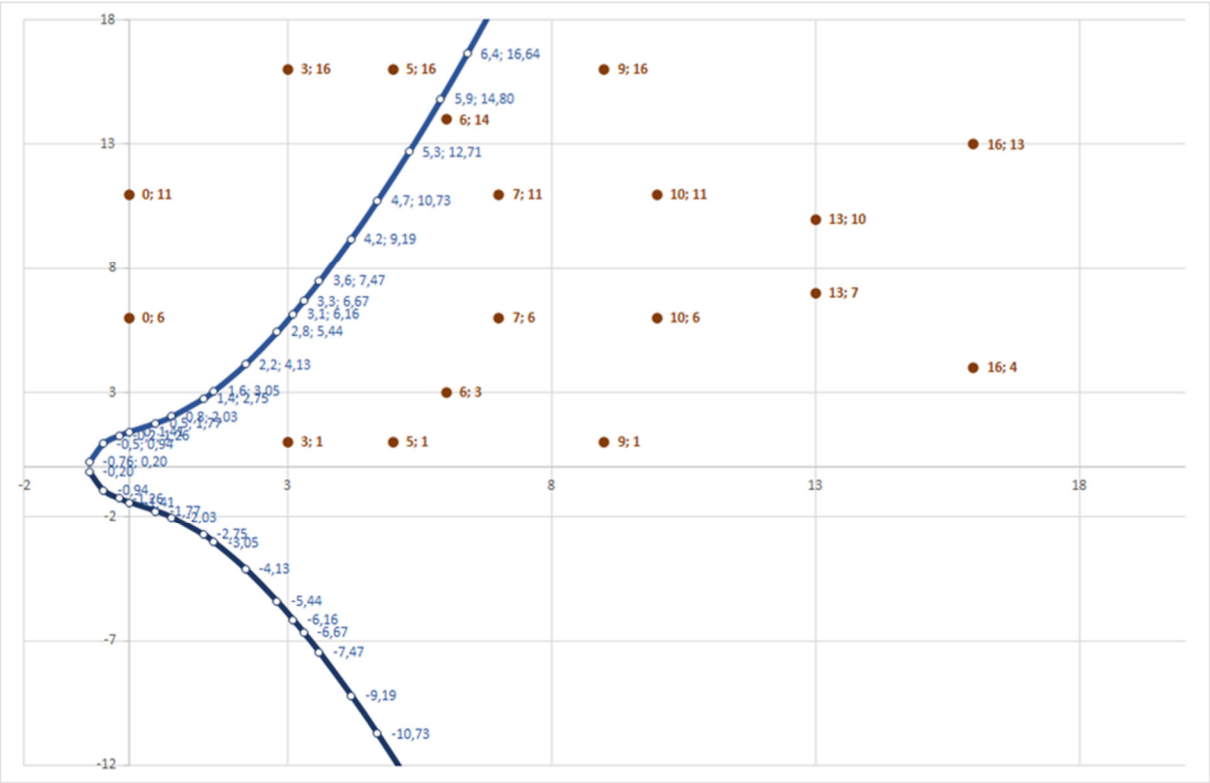


Abbildung 4: In $\mathbb{R}$ (blaue Kurve) gibt es unendlich viele Lösungen, einige davon sind markiert. In $\mathbb{Z}_{17}$ sind es genau 18.

Rechnen auf elliptischen Kurven

Elliptische Kurven haben die Eigenschaft, dass eine Sekante durch zwei Punkte P und Q der Kurve, diese noch ein drittes Mal schneidet (vorausgesetzt $4a^3 + 27b^2 \neq 0$). Man muss dabei nur berücksichtigen, dass die Schnittpunkte von Geraden und Kurve mit der richtigen Vielfachheit gezählt werden (Tangente in einem Punkt ergibt Vielfachheit 2, eine Wendetangente sogar 3).

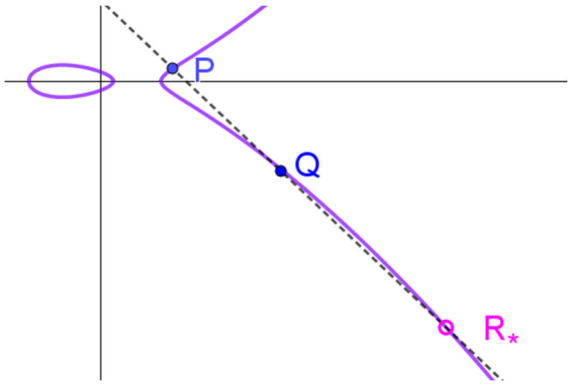


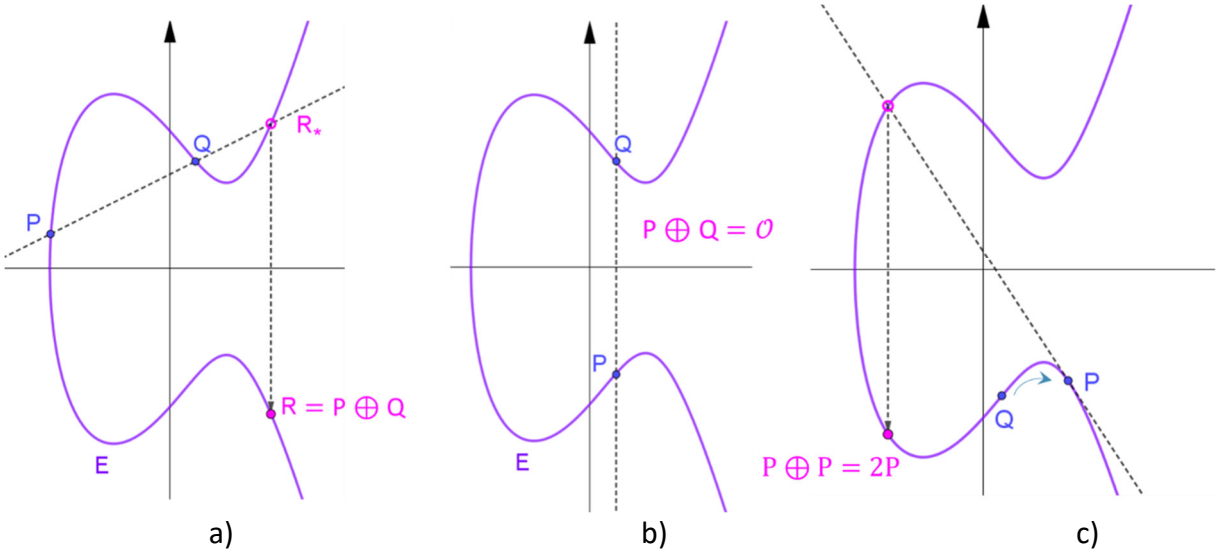
Abbildung 5

Deshalb lässt sich mit den Punkten einer elliptischen Kurve eine Rechenvorschrift mit Gruppeneigenschaften definieren, die von den Eigenschaften her einer Addition ähnlich ist.

Beschränken wir uns vorerst auf elliptische Kurven der Form $y^2 = x^3 + ax + b$ über $\mathbb{R}$.

Die Punktaddition auf elliptische Kurven über $\mathbb{R}$

Abbildung 6 veranschaulicht die sogenannte Punktaddition $\oplus$. Ausgehend von zwei Punkten P und Q der elliptischen Kurve E und deren ***Sekante***, ergibt die ***Spiegelung des dritten Schnittpunktes*** an der x-Achse letztendlich als Ergebnis den Punkt $R = P \oplus Q$.



a) b) c)

Abbildung 6

Es gibt aber ein Problem. Haben die beiden Punkte P und Q die gleiche x-Koordinate (Abbildung 6b), dann liegt der dritte Schnittpunkt und damit auch sein Spiegelpunkt quasi im Unendlichen (∞). Um die ***Abgeschlossenheit*** der Punktaddition zu gewährleisten, wird in diesem Fall der Addition ein koordinatenloser Punkt $\mathcal{O}$ im „Unendlichen“ zugeordnet.

Von dieser Punktaddition kann gezeigt werden, dass sie sowohl ***kommutativ*** (trivialerweise) als auch ***assoziativ*** ist.

Punktverdopplung: Ist $\boldsymbol{Q = P}$, so wird die *Sekante zur Tangente* (Abbildung 6c) und $P \oplus Q = P \oplus P = 2P$.

Das ***neutrale Element*** ist $\mathcal{O}$.

$$P \oplus \mathcal{O} = \mathcal{O} \oplus P = P \quad \text{(Abbildung 7 links)}$$

Das ***inverse Element*** wird wie bei Addition üblich mit $-P$ bezeichnet und ist der an der x-Achse gespiegelte Punkt.

$$-P \oplus P = P \oplus -P = \mathcal{O} \quad \text{(Abbildung 7 rechts)}$$

$$P = (x_P\,, y_P) \Longrightarrow -P = (x_P\,, -y_P)$$

Die ***Subtraktion*** entspricht der Addition mit der Inversen: $P \oplus -Q$

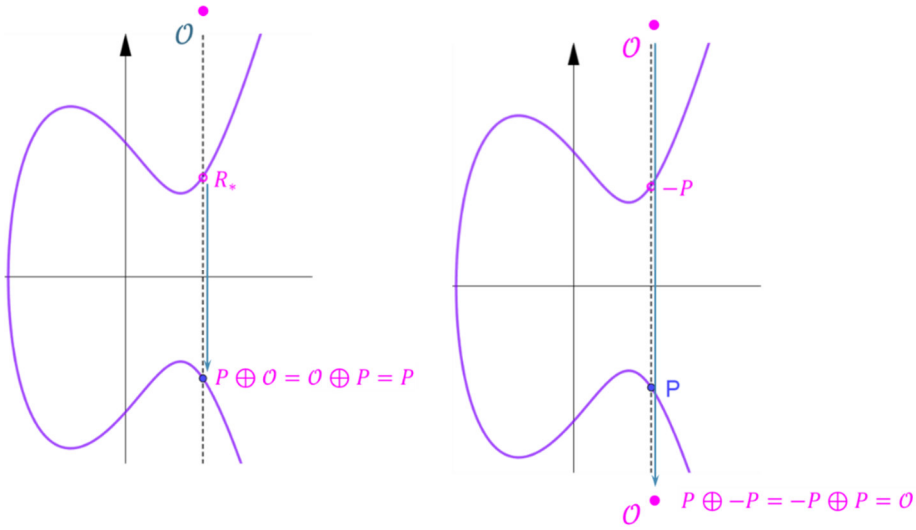


Abbildung 7

ADDITIONSFORMEL

Für die Steigung $\boldsymbol{s}$ der Sekante ($\boldsymbol{P \neq Q}$) bzw. der Tangente ($\boldsymbol{P = Q}$) gilt:

$$s := \begin{cases} (y_Q - y_P)(x_Q - x_P)^{-1} & P \neq Q \ (Sekante) \\ (3x_P^2 + a)(2y_P)^{-1} & P \oplus P = 2P \ (Tangente) \end{cases}$$

$$P \oplus Q = R = (x_R, y_R) \equiv \left(s^2 - x_P - x_Q \ , \ s \cdot (x_P - x_R) - y_P\right)$$

$$P \oplus Q = R = (x_R, y_R) \equiv \left(s^2 - x_P - x_Q \ , \ s \cdot (x_Q - x_R) - y_Q\right)$$

$$P \oplus P = 2P = (x_R, y_R) \equiv \left(s^2 - 2x_P \ , \ s \cdot (x_P - x_R) - y_P\right)$$

(Begründung siehe Anhang)

Das Besondere an elliptischen Kurven ist, dass ihre (rationalen) Punkte in natürlicher Weise eine abelsche Gruppe bilden, insofern man im Falle einer senkrechten Geraden $\mathcal{O}$ als dritten Schnittpunkt interpretiert.

Satz

Sei $E(K) = \{(x, y) \in K^2 : y^2 = x^3 + ax + b \ , (a, b \in K \wedge 4a^3 + 27b^2 \neq 0)\} \cup \{\mathcal{O}\}$.

Die algebraische Struktur $(E(K) \ , \ \oplus)$ bildet eine abelsche Gruppe.

Der Nachweis, vor allem des Assoziativgesetzes, ist nicht trivial und sehr umfangreich, da man es vielfach mit dem Lösen von Gleichungen dritten Grades zu tun hat. Insofern auch die Einschränkung $4a^3 + 27b^2 \neq 0$ bei kubischen Gleichungen.

Punktaddition auf elliptische Kurven über endliche Körper ($\mathbb{Z}_p$)

Elliptische Kurven über endlichen Körpern sind geometrisch keine „Kurven“ mehr, vielmehr diskrete Punkte. Wir beschränken uns dabei auf den Körper $(\mathbb{Z}_p, +, \cdot)$ mit Modulus p prim.

Mit Addition und Multiplikation sind Restklassen-Addition bzw. Restklassen-Multiplikation gemeint.

Gemäß Tabelle 1 gibt es für $\boldsymbol{E(\mathbb{Z}_{17})}$ nur 19 Lösungen, die $\mathcal{O}$ mit eingerechnet.

$(\mathcal{O})$	(3 , 1)	(3 , 16)	(5 , 1)	(5 , 16)	(9 , 1)	(9 , 16)	(0 , 6)	(0 , 11)	(7 , 6)
	(7 , 11)	(10 , 6)	(10 , 11)	(6 , 3)	(6 , 14)	(13 , 7)	(13 , 10)	(16 , 4)	(16 , 13)

Für die Punkt-Addition gelten die gleichen Additionsformeln, allerdings $\bmod p$ *gerechnet.*

Insbesondere ist in $(y_Q - y_P)(x_Q - x_P)^{-1}$ und $(3x_P^2 + a)(2y_P)^{-1}$ über die jeweiligen Inversen zu rechnen, da ja so die Division definiert ist.

•	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2	0	2	4	6	8	10	12	14	16	1	3	5	7	9	11	13	15
3	0	3	6	9	12	15	1	4	7	10	13	16	2	5	8	11	14
4	0	4	8	12	16	3	7	11	15	2	6	10	14	1	5	9	13
5	0	5	10	15	3	8	13	1	6	11	16	4	9	14	2	7	12
6	0	6	12	1	7	13	2	8	14	3	9	15	4	10	16	5	11
7	0	7	14	4	11	1	8	15	5	12	2	9	16	6	13	3	10
8	0	8	16	7	15	6	14	5	13	4	12	3	11	2	10	1	9
9	0	9	1	10	2	11	3	12	4	13	5	14	6	15	7	16	8
10	0	10	3	13	6	16	9	2	12	5	15	8	1	11	4	14	7
11	0	11	5	16	10	4	15	9	3	14	8	2	13	7	1	12	6
12	0	12	7	2	14	9	4	16	11	6	1	13	8	3	15	10	5
13	0	13	9	5	1	14	10	6	2	15	11	7	3	16	12	8	4
14	0	14	11	8	5	2	16	13	10	7	4	1	15	12	9	6	3
15	0	15	13	11	9	7	5	3	1	16	14	12	10	8	6	4	2
16	0	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

Tabelle 2: Multiplikationstabelle für $\mathbb{Z}_{17}$

Beispiel $(\mathbb{Z}_{17})$ ***:***

$$R = P \oplus Q = (10\,,6) \oplus (5\,,1)$$

$$\begin{aligned} s &\equiv (y_Q - y_P)(x_Q - x_P)^{-1} \\ &\equiv -5 \cdot (-5)^{-1} \equiv 12 \cdot 12^{-1} \\ &\equiv 12 \cdot 10 \equiv 1 \end{aligned}$$

$$x_R \equiv s^2 - x_P - x_Q \equiv -14 \equiv 3$$

$$y_R \equiv s \cdot (x_P - x_{\boldsymbol{R}}) - y_P \equiv 1$$

$$R = P \oplus Q = (3\,,1)$$

$$(10\,,6) \oplus (5\,,1) = (3\,,1)$$

Man erkennt in der Multiplikationstabelle (Tabelle 2) die zu einander inversen Elemente, da sie das Einselement ergeben. In Restklassenkörpern modulo einer Primzahl hat jedes Element sein Inverses, ausgenommen wie immer das neutrale Element der Addition, die Null.

In Abbildung 8 sind alle 18 Elemente $(\neq \mathcal{O})$ der EC $E(\mathbb{Z}_{17})$ blau markiert. Bei der Punktaddition schneiden die Punkte P, Q, R_* die elliptische Kurve entlang einer Geraden. Der Spiegelpunkt von R_* ist dann das Ergebnis $R = P \oplus Q =^{z.B.} (3\,,1)$.

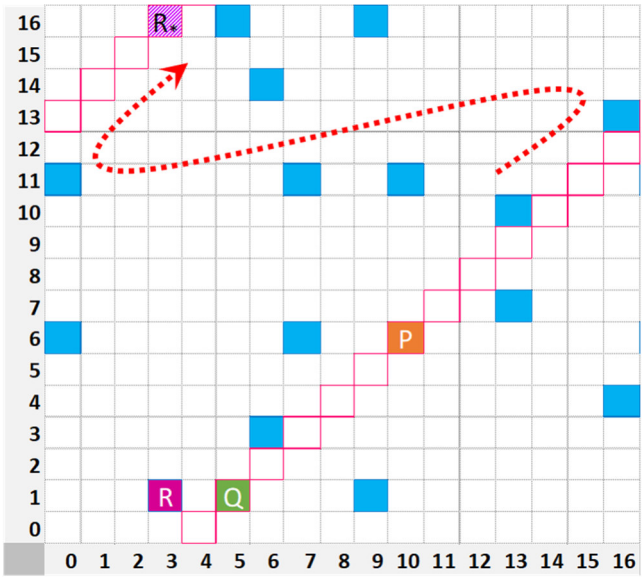


Abbildung 8

Die Gerade, das sind die roten Kästchen in Abbildung 8. In $\mathbb{Z}_p$ ist sie natürlich $\bmod p$ gebrochen. Dass P, Q und R_* auf einer „Geraden“ liegen sieht man, wenn man weitere Repräsentanten der Punkte mit höheren Koordinaten betrachtet (Abbildung 9).

Beispiel: $P = (\mathbf{10}, \mathbf{6}) \quad Q = (\mathbf{5}, \mathbf{1}) \rightarrow \; R = (3, 1) \rightarrow R_* = (3, -1) \equiv (\mathbf{20}, \mathbf{16})$
Die Punkte $(10, 6)$, $(5, 1)$ und $(20, 16)$ liegen direkt auf einer Geraden.

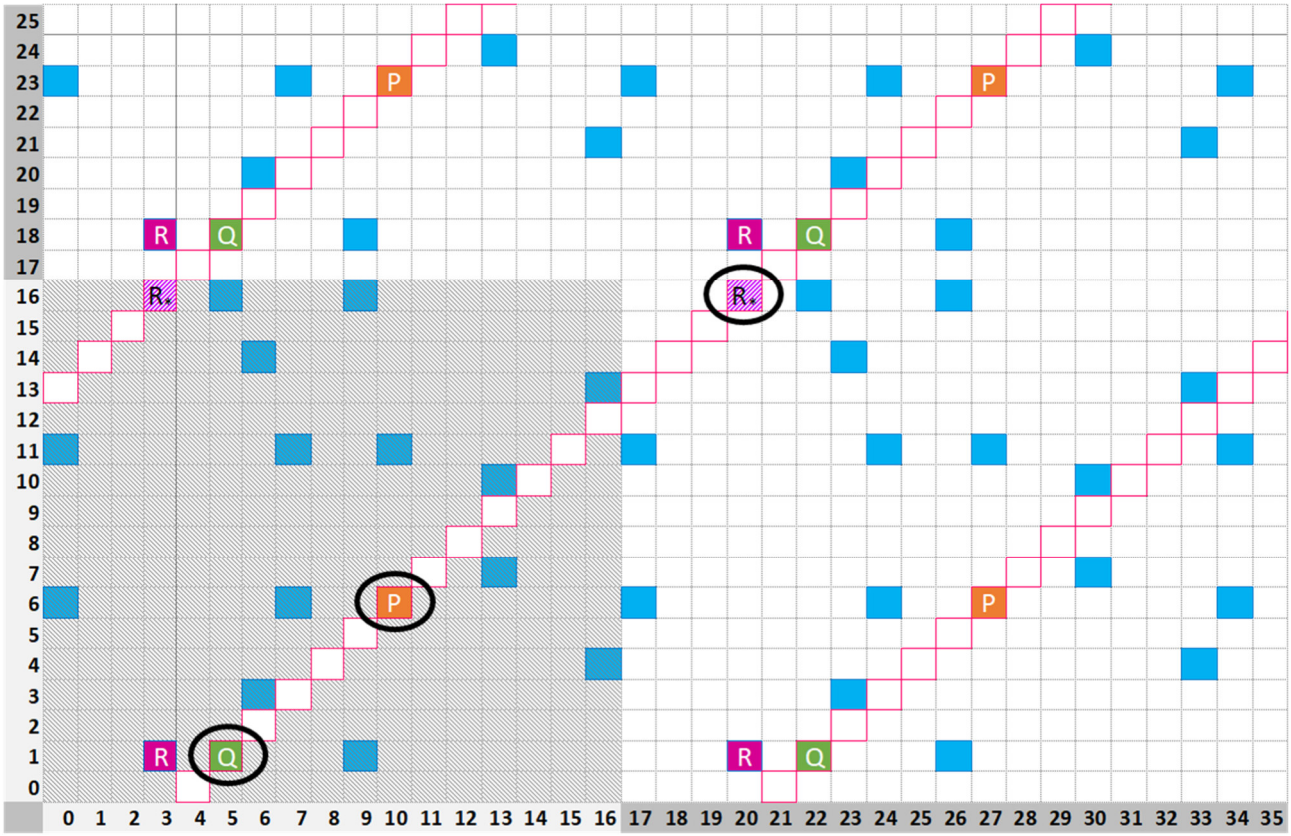


Abbildung 9: P, Q und $-R = R_*$ liegen auf einer Geraden.

Zusammenfassend hat die Punktaddition auf einer elliptischen Kurve E über dem Körper K folgende Eigenschaften:

$P, Q \in E(K)\,; \quad x_P \neq x_Q$

$$P \oplus Q = Q \oplus P$$

$$(P \oplus Q) \oplus S = P \oplus (Q \oplus S)$$

$$P \oplus \mathcal{O} = \mathcal{O} \oplus P = P$$

$$P \oplus -P = \mathcal{O} \text{ mit } P = (x_p\,, y_p) \text{ und } -P = (x_p\,, -y_p)$$

$P, Q \in E(K)\,; \quad x_P = x_Q$

$$P \oplus Q = P \oplus -P = \mathcal{O}$$

Skalare Multiplikation in $E(\mathbb{Z}p)$

Für die Kryptographie von besonderer Bedeutung ist die skalare Multiplikation nP.

Definition (Skalarmultiplikation)
Sei P ein Punkt einer elliptischen Kurve über einem Primkörper (z.B. $\mathbb{Z}_p$). Die ***skalare Multiplikation*** von P mit $n \in \mathbb{N}$ ist definiert als die n-fache Addition des Punktes P.

$$nP = \underbrace{P \oplus P \oplus \ldots \oplus P}_{n-mal} \quad , \qquad 0P \coloneqq \mathcal{O}\ (\text{inf})$$

Beispiel

$E\colon\ y^2 \equiv x^3 - 6x + 2 \equiv x^3 + 7x + 2 \mod 13\,, \quad \boldsymbol{P = (7\,, 11)}$

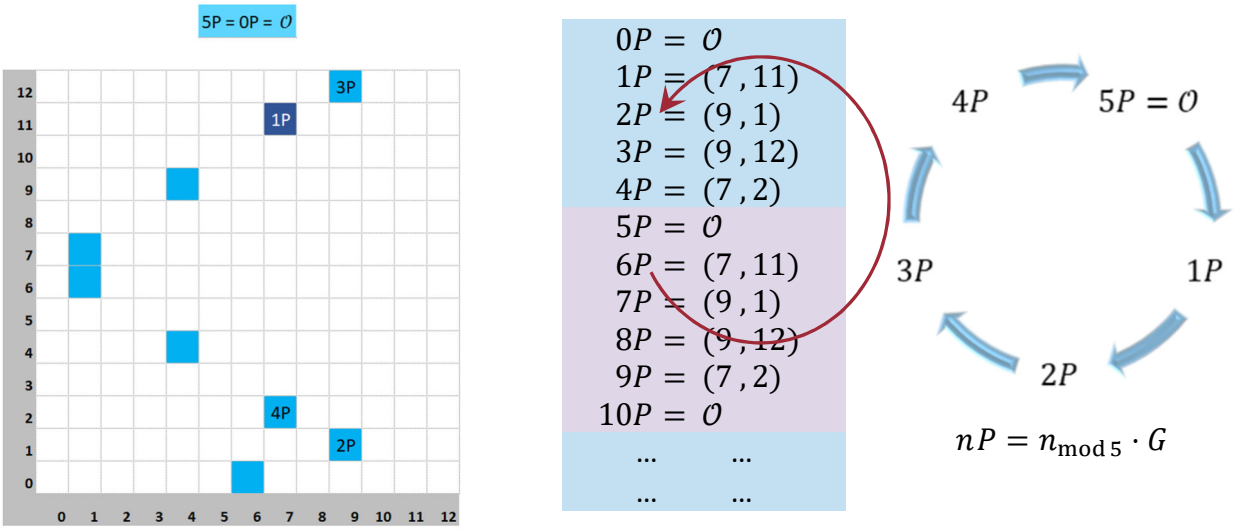


Abbildung 10

Die Elemente von $E(\mathbb{Z}_{13})$ sind in Abbildung 10 links blau eingefärbt dargestellt. Der Punkt P und alle seine Vielfachen ergeben zyklisch immer dieselben Elemente. Sie bilden sozusagen eine

zyklische Untergruppe $(U,\oplus)$ innerhalb der Gruppe $(E(\mathbb{Z}_{13}),\oplus)$. Gruppe ist hier mathematisch gemeint. Alle Gruppeneigenschaften einer abelschen Gruppe sind in dieser Untergruppe erfüllt. Assoziativität und Kommutativität sind von der „Großgruppe“ vererbt, das neutrale Element ist mit $0P$ mit an Bord und jedes Element hat sein inverses, da man ja 'im Kreis' rechnet und damit immer auf $\mathcal{O}$ kommen kann.

Ausblick

Mit zyklischen Gruppen lässt sich ein dem diskreten Logarithmus-Problem (DLP) adäquates entwickeln und damit Kryptographie betreiben!!!

In unserem Beispiel erzeugt der Punkt $P = (7\,,11)$ eine zyklische Untergruppe mit 5 Elementen (Kardinalität). Der Erzeugerpunkt wird ***Generator*** genannt und im Folgenden mit ***G*** bezeichnet. Die ***Kardinalität*** $\#\mathrm{U}$ (= Anzahl der Elemente) seiner erzeugten Untergruppe ist die ***Ordnung von*** G.

Der Generator $G = (7\,,11)$ besitzt somit die Ordnung 5. Ideal wäre ein Generatorpunkt, der gleich die ganze Gruppe $E(\mathbb{Z}_p)$ erzeugt. Die Vielfachen dieses Punktes würden alle Punkte der elliptischen Kurve durchlaufen.

Beispiel

$E\colon\ y^2 \equiv x^3 - 6x + 2 \equiv x^3 + 7x + 2 \mod 13\,,\ \ \boldsymbol{G = (1\,,7)}$

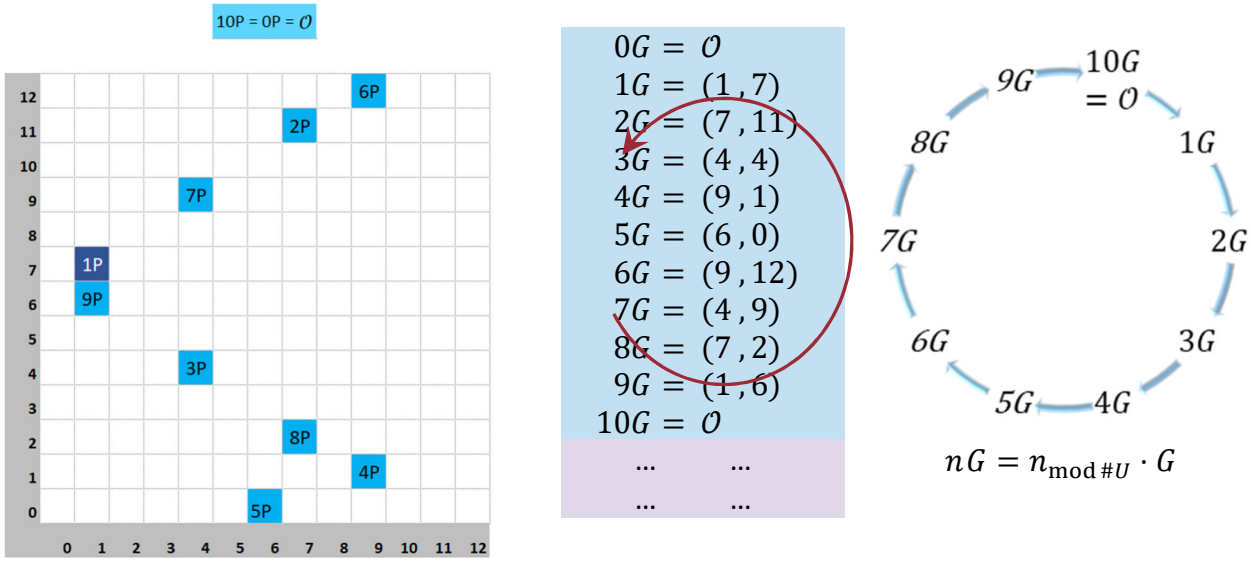


Abbildung 11

Der Punkt $G = (1\,,7)$ generiert die ganze Gruppe der elliptischen Kurve. Er ist ein Generator von $E(\mathbb{Z}_{13})$. Und es gibt deren mehrere: $(1\,,7)\ \ (4\,,9)\ \ (4\,,4)\ \ (1\,,6)$

Es gibt elliptische Kurven, deren Punkte alle (ausgenommen $\mathcal{O}$) Generatoren sind.

Satz

Ist die Kardinalität $\#E$ der elliptischen Kurve $E(\mathbb{Z}_p)$ selbst auch eine Primzahl, dann sind alle ihre Elemente $P \in E(\mathbb{Z}_p)$ (ausgenommen $\mathcal{O}$) Generatoren.

Die Kardinalität, also die Anzahl der Punkte einer elliptischen Kurve, zu bestimmen, kann eine sehr umständliche und zeitaufwändige Prozedur sein. 1936 zeigte Helmut Hasse die Abschätzung

$$\#E(\mathbb{Z}_p) = (p+1) \pm 2\sqrt{p} \quad (\textbf{\textit{Satz von Hasse}})$$

René Schoof entdeckte 1986 den nach ihm benannte Algorithmus, der $\#E$ auch bei großen Zahlen in annehmbarer Zeit berechnet.

Für die Anwendung in der Kryptographie will man zyklische Untergruppen U mit großer Kardinalität, wenn möglich $E(\mathbb{Z}_p)$ selber. $\#U$ ist ein Teiler von $\#E$. Im Allgemeinen wird also eine elliptische Kurve gewählt, ihre Ordnung $\#E$ bestimmt, ein hoher Teiler $\#U$ gewählt und in der gefundenen Untergruppe ein geeigneten Basispunkt P ermittelt.

Beispiel (Curve-ID: brainpoolP256r1)

$E(\mathbb{Z}_p)$: $y^2 = x^3 + ax + b$

Hexadezimal

$p = A9FB57DBA1EEA9BC3E660A909D838D726E3BF623D52620282013481D1F6E5377$

$a = 7D5A0975FC2C3057EEF67530417AFFE7FB8055C126DC5C6CE94A4B44F330B5D9$

$b = 26DC5C6CE94A4B44F330B5D9BBD77CBF958416295CF7E1CE6BCCDC18FF8C07B6$

$G(x, y)$:
$x = 8BD2AEB9CB7E57CB2C4B482FFC81B7AFB9DE27E1E3BD23C23A4453BD9ACE3262$
$y = 547EF835C3DAC4FD97F8461A14611DC9C27745132DED8E545C1D54C72F046997$

$\#U_G = \#E = A9FB57DBA1EEA9BC3E660A909D838D718C397AA3B561A6F7901E0E82974856A7$

Dezimal

$p = 76884956397045344220809746629001649093037950200943055203735601445031516197751$

$a = 56698187605326110043627228396178346077120614539475214109386828188763884139993$

$b = 17577232497321838841075697789794520262950426058923084567046852300633325438902$

$G(x, y)$:
$x = 63243729749562333355292243550312970334778175571054726587095381623627144114786$
$y = 38218615093753523893122277964030810387585405539772602581557831887485717997975$

$\#U_G = \#E = 76884956397045344220809746629001649092737531784414529538755519063063536359079$

Tabelle 3

DLP und ECDLP

Das diskrete Logarithmusproblem über elliptische Kurven (ECDLP) ist das Analogon zum echten diskreten Logarithmusproblem (DLP) in endlichen Körpern wie $\mathbb{Z}_p$.
Es handelt sich dabei um eine sogenannte Trapdoor-Funktion, die leicht berechenbar, aber schwer umzukehren ist. Kryptosysteme verwenden dazu gerne modulares Potenzieren.

$$a = b^k \bmod p \rightarrow a$$

Die Umkehrung erfordert modulares Logarithmieren,

$$b^k = a \mod p \rightarrow k = ?$$

was mit größeren Zahlen zunehmend nur mit Brute Force möglich ist.
(Beispiel: $47^k = 347 \bmod 607 \rightarrow k = ?$)

In $(E(K), \oplus)$ haben wir es mit einer additiven Gruppe zu tun, somit ist die Schreibweise eine andere.

Definition (DLP über EC: ECDLP)

Gegeben sei die abelsche Gruppe $(E(K), \oplus)$ mit der elliptischen Kurve E über dem endlichen Körper K (z.B. $\mathbb{Z}_p$).
$G \in E(K)$ sei Generator einer zyklischen Untergruppe U und $T \in U(K), T \neq G$ ein weiterer Punkt darin. Somit gibt es ein $n \in \mathbb{N}$, sodass

$$nG = T.$$

Das DLP in elliptischen Kurven (ECDLP) ist die Bestimmung dieser natürlichen Zahl n.

Anwendungen

Elliptische Kurven finden in der Kryptographie vor allem dort Anwendung, wo der Rechenaufwand sehr groß ist. ECC ist nämlich erheblich schneller als RSA, da die benötigten Schlüssellängen bei gleichem Sicherheitsniveau für ECC deutlich kürzer sind als bei klassischen asymmetrischen Verfahren. Das liegt daran, dass das DLP-Problem in der Theorie der elliptischen Kurven deutlich schwieriger ist als bei vergleichbaren anderen Kryptosystemen.

Ein bekannter Algorithmus ist der Diffie-Hellman- Schlüsselaustausch. Symmetrische Verschlüsselungsverfahren sind einfacher und schneller als asymmetrische, weswegen sie bevorzugt verwendet werden. Dabei müssen Sender und Empfänger über denselben Schlüssel verfügen. Um einen gemeinsamen Schlüssel auch über unsichere Kanäle vereinbaren zu können bedient man sich eines sicheren Schlüsselaustauschverfahrens. Whitfield Diffie und Martin Hellman haben im Jahr 1976 dafür einen Algorithmus entwickelt.

Abbildung 12: Whitfield Diffie und Martin Hellman

Diffie-Hellman-Schlüsselaustausch mit EC (ECDH)

Wie in der Kryptographie üblich, werden Sender und Empfänger gerne „Alice“ und „Bob“ genannt. Wollen Alice und Bob über ein symmetrisches Verschlüsselungsverfahren (z.B. AES) kommunizieren, so müssen sie bzw. ihre Endgeräte zuvor einen Austausch des dafür benötigten Schlüssels vornehmen. Beim sogenannten Diffie-Hellman-Verfahren (DH) wird genau genommen der Schlüssel nicht ausgetauscht, sondern es werden gleichzeitig auf Sender- und Empfängerseite identische Schlüssel errechnet.

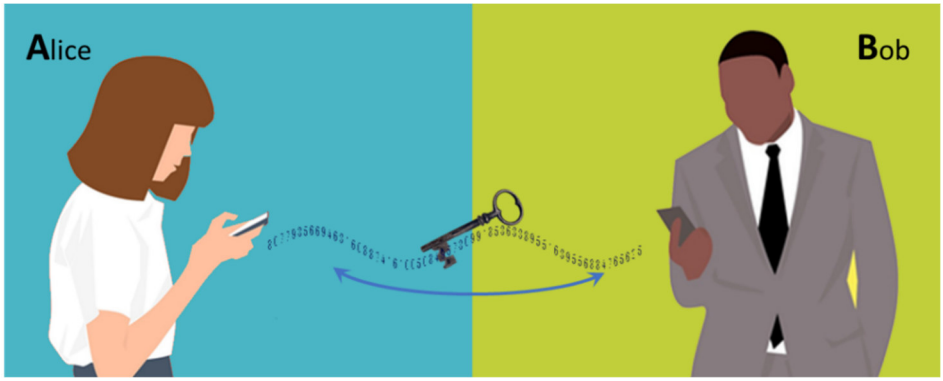


Abbildung 13

Der gewünschte gemeinsame Schlüssel wird hier mit Hilfe ECDLP über die Formel

$$nG = T$$

errechnet. Dazu muss eine elliptische Kurve $E(\mathbb{Z}_p)$ inklusive großem Primzahl-Modulus bestimmt werden, sowie ein passender Generator G.
E, p und G werden *nicht geheim* gehalten, sie gehören zum sogenannten „öffentliche Schlüssel“ ***(Public-Key)***.

Andererseits wählt die Software bei Alice und Bob jeweils intern einen *geheimen* ***Multiplikator d (Private-Key)***. Den bei Alice nennen wir a, den bei Bob b.

Alice:	Bob:
$\alpha \in \{2, 3, \dots, \#U - 1\}$	$\beta \in \{2, 3, \dots, \#U - 1\}$

Beide Endgeräte berechnen ihren Teil der Multiplikation und erhalten die Punkte A bzw. B.

$$\alpha G = \underbrace{G \oplus \dots G}_{\alpha - mal} = A = (x_A, y_A) \qquad \beta G = \underbrace{G \oplus \dots \dots G}_{\beta - mal} = B = (x_B, y_B)$$

Diese Berechnungen werden innerhalb der Punktmenge der EC modulo p durchgeführt. Die Punktkoordinaten werden danach untereinander ausgetauscht.

$$A = \alpha G \rightarrow$$

$$\leftarrow B = \beta G$$

Dabei kann es passieren, dass diese Übertragungen abgehört werden. Der gemeinsame Schlüssel entsteht aber erst danach bei Alice und Bob. Die Software multiplizieren jeweils den erhaltenen Teil

des Schlüssels wieder mit dem eigenen geheimen Multiplikator und erhalten beide so dasselbe Ergebnis S.

$$S = \alpha B = \alpha(\beta G) = (\alpha\beta)G = (x_S\,, y_S) \qquad S = \beta A = \beta(\alpha G) = (\alpha\beta)G = (x_S\,, y_S)$$

Beide Koordinaten von S sind in der Regel große Binärzahlen. Man könnte zum Beispiel für die weitere Verwendung in symmetrischen Verfahren die ersten 128 Bit-Stellen der Koordinate x_S verwenden.

Beispiel ECDH

In einer Messenger-Software implementierte EC: $a = 2 \quad b = 5 \quad p = 47 \quad G = (33\,, 37)$
$\#E = 42 \quad \#U_G = 21$

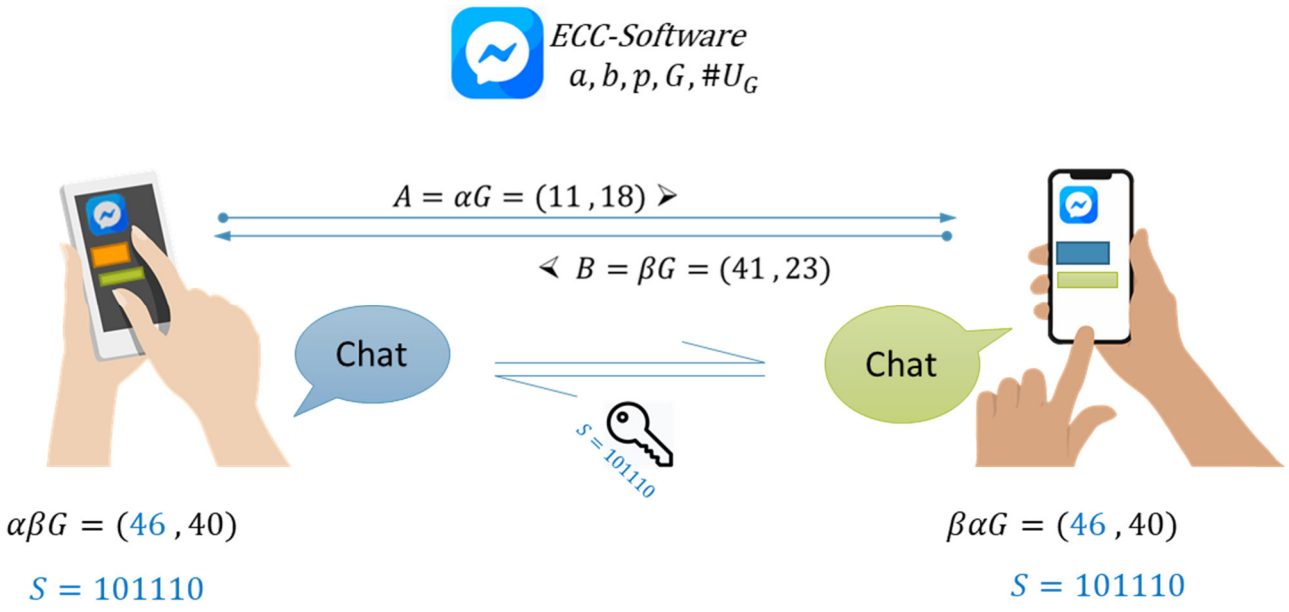


Abbildung 14

Für den Schlüsselaustausch wählt die Software als privaten Schlüssel (private key) einen zufälligen Wert $\alpha \in \{2, \dots, 20\}$ und erzeugt damit den öffentlichen Schlüssel (public key) $\alpha G =_{z.B.} 15G = (11\,, 18)$. Analog erzeugt die Software beim Gesprächspartner den privaten Schlüssel z.B. $\beta = 11$ und damit dessen öffentlichen Schlüssel $\beta G =_{z.B.} 11G = (41\,, 23)$.

Beide Geräte multiplizieren den erhaltenen Code erneut mit dem eigenen und geheimen Multiplikator: $\beta(\alpha G) = 11(11\,, 18) = (46\,, 40)$ bzw. $\alpha(\beta G) = 15(41\,, 23) = (46\,, 40)$

Beide erhalten den gleichen Schlüssel und können damit Nachrichten untereinander verschlüsselt austauschen.

Beispiel WhatsApp

curve25519: $y^2 = x^3 + 486662x^2 + x$ (Montgomery)

$p = 57896044618658097711785492504343953926634992332820282019728792003956564819949$

$G = (9\,, 14781619447589544791020593568409986887264606134616475288964881837755586237401)$

$\#U_G = 72370055773322622139731865630429942408571163593799076060019509382854542509 89$

$\#E = 8 \cdot \#U$

Digitale Signatur

(Digital Signature Algorithm DSA)

Die digitale Signatur gewährleistet die Authentizität von Daten, dass sie also jederzeit ihrem Ursprung bzw. Verfasser zugeordnet werden können. Das können eine persönliche E-Mail, ein behördliches digitales Dokument oder aber auch die Echtheit einer Webseite („https“) sein. Die digitale Unterschrift funktioniert ähnlich wie früher das Siegel auf einem Dokument, was die Echtheit gewährleisten sollte.

Abbildung 15: Einst und heute

Beim Signieren geht es nicht darum, den Inhalt der Message zu verschlüsseln. Nur die Unterschrift wird verschlüsselt und beim Empfänger wieder entschlüsselt.

Die Signatur hat die Bedeutung eines Personalausweises. Das heißt, die Echtheit wird von Amts wegen bestätigt. Ich muss mir meine Signatur von einer sogenannten Zertifizierungsstelle (Certification Authority, kurz: CA) ausstellen lassen. In Österreich ist das die A-Trust, in Deutschland z.B. die D-Trust (Bundesdruckerei).

Sind meine Personalien überprüft, so werden 2 Schlüssel bezüglich eines asymmetrischen Verschlüsselungsverfahrens generiert. Der eine dient zum Verschlüsseln und bleibt geheim auf meinem Rechner. Das ist der private Schlüssel (Private Key).
Der zweite Schlüssel dient zum Entschlüsseln und ist öffentlich (Public Key). Er ist in der Zertifizierungsstelle hinterlegt und jederzeit abrufbar.

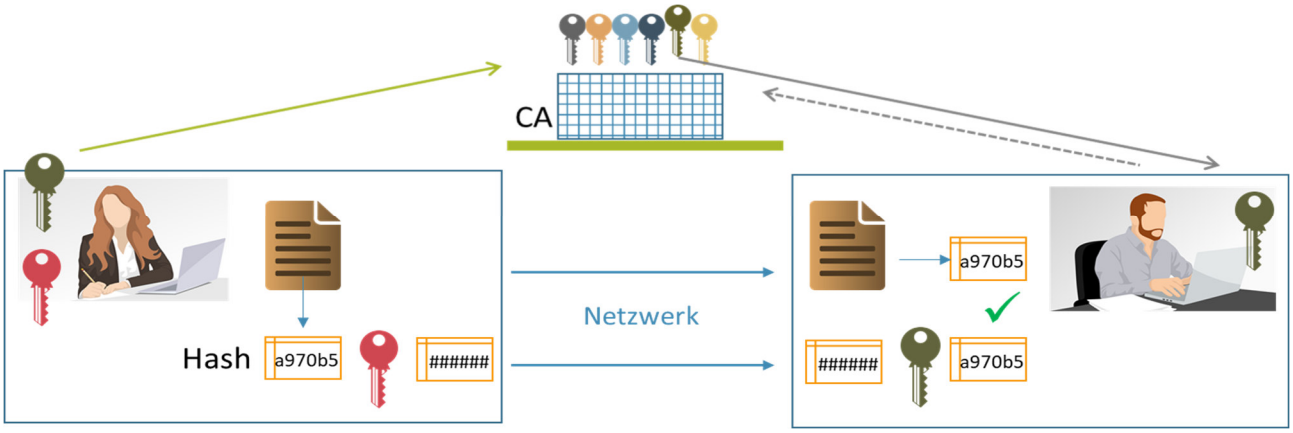


Abbildung 16

Sende ich ein signiertes Dokument per E-Mail an einen Empfänger, so erzeugt der Algorithmus zuerst den Hashwert des Dokuments. Der Hashwert ist eine für dieses Dokument charakteristisch eindeutige Zahl (z.B. hex 4ce311519756c6dd9435fefcd61c19a84e7a8c92). Auf die Funktionsweise wird hier nicht näher eingegangen. Nur soviel: eine einzige Buchstabenänderung im Dokument erzeugt einen völlig anderen Hashwert und aus dem Hashwert kann nicht auf das zugehörige Dokument rückgeschlossen werden.

Dieser Hashwert wird mit meinem privaten Schlüssel verschlüsselt und entspricht so meiner Unterschrift. Das E-Mail-Programm sendet nun mein Dokument (unverschlüsselt) inklusive des verschlüsselten Hashwerts und diverser Daten (CA, Hashwert-Algorithmus etc.) an den Empfänger. Das E-Mail-Programm des Empfängers erkennt die Signatur und überprüft automatisch ihre Echtheit.
Dazu lädt es sich von der Zertifizierungsstelle (CA) meinen öffentlichen (Ent-) Schlüssel herunter und entschlüsselt den verschlüsselten Hashwert des Dokuments. Gleichzeitig wird auch vom E-Mail-Programm des Empfängers der Hashwert des zugesandten Dokuments neu berechnet.

Ist das Dokument tatsächlich von mir, dann müssen jetzt beide Hashwerte identisch sein. Dabei garantiert die Zertifizierungsstelle die Zugehörigkeit des öffentlichen Schlüssels zu meiner Person.

Die digitale Signatur auf elliptischen Kurven (ECDSA)

(Elliptic Curve Digital Signature Algorithm ECDSA)

Die digitale Signatur auf elliptischen Kurven über endlichen Körpern (z.B. $E(\mathbb{Z}_p)$) ist noch ein bisschen trickreicher. Wir werden den Algorithmus hier mit sehr kleinen Zahlen demonstrieren. Die auftretenden Berechnungen können online unter https://andrea.corbellini.name/ecc/interactive/modk-mul.html (bzw. offline siehe Dateien) nachgerechnet werden. In der Realität werden die Zahlen sehr groß gewählt um die gewünschte Sicherheit zu gewährleisten.

SIGNATURERSTELLUNG

Wir benötigen:

1. Eine EC:$E(\mathbb{Z}_p)$ inklusive zyklischen Untergruppe U_G mit Primzahl-Kardinalität.
2. Vier Zahlen: m (der Hash des Dokuments), k_{prv} (der private Schlüssel), sowie zwei Zufallszahlen k und ρ.
3. Daraus berechnet sich eine fünfte Zahl σ.
4. Die beiden Zahlen $(\rho\,,\sigma)$ bilden die Signatur.
5. $k_{prv}G$ bildet den öffentlichen Schlüssel K_{pub}.

Die Voraussetzungen an die EC:

Die elliptische Kurve $\boldsymbol{E(\mathbb{Z}_p)}, \boldsymbol{a}, \boldsymbol{b}$ muss einen Generatorpunkt $\boldsymbol{G}$ mit zugehöriger zyklischer Untergruppe $\boldsymbol{U_G}$ besitzen. Dabei ist wichtig, dass die Kardinalität der Untergruppe ebenfalls eine Primzahl ist ($\#\boldsymbol{U_G} =: \boldsymbol{q} \in \mathbb{P}$).

p und q müssen also beide Primzahlen sein, da man überall dividieren können muss!

Bevorzugt sind elliptische Kurven, die selbst eine zyklische Gruppe bilden! ($\#E = q = p$)

Beispiel

$E\colon\ y^2 = x^3 + 3x + 2$ besitzt für $p = 47$ eine von $G = (2\,,4)$ erzeugte zyklische Untergruppe mit $q = 23$ Punkten. Somit sind p und q beide Primzahlen.

Man muss sich bewusst sein, dass sowohl p als auch q jeweils einen eigenen modularen Kreis definieren. Für die *Berechnung von Punkten* P bzw. ihren Koordinaten auf der elliptischen Kurve wird $\mathrm{mod}\ p$ gerechnet, für die *Multiplikatoren* λ gilt der Zyklus der Untergruppe $\mathrm{mod}\ q$.
(Siehe Abbildung 10 und Abbildung 11)

Beispiel bezüglich der oben gewählten Parameter:

$$\lambda P =^{zB} 31\,(56\,,104) \equiv 8_{\mathrm{mod}\,23}\,(9\,,10)_{\mathrm{mod}\,47} \equiv (7\,,15)_{\mathrm{mod}\,47}$$

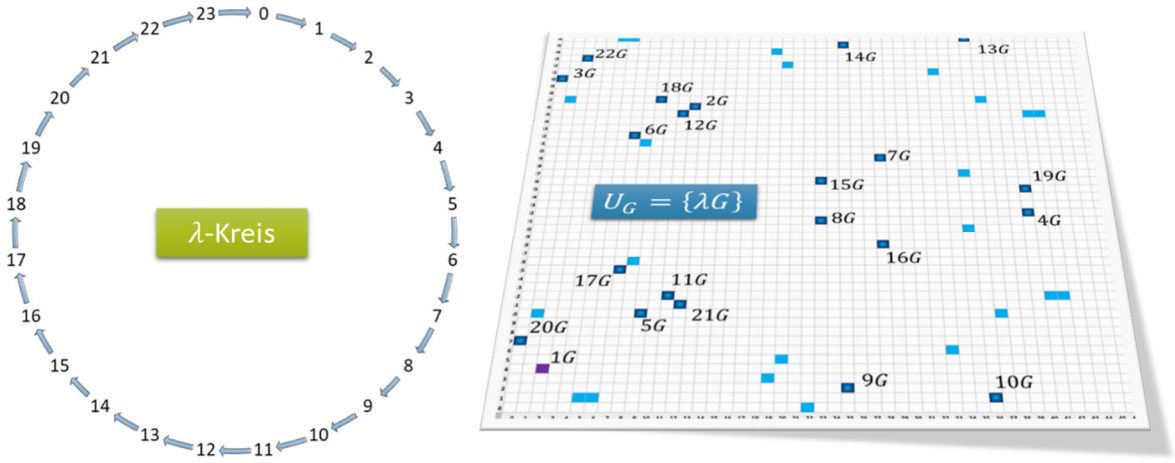


Abbildung 17: Die Multiplikatoren λ der Untergruppe $U_{G=(2\,,4)}$ (links) und die $U_{G=(2\,,4)}$ in der $EC\colon E(\mathbb{Z}_{47}), a = 3, b = 2$ (rechts)

Der Algorithmus:

In den Algorithmus eingegeben werden müssen:

- Der ***Hashwert*** der Nachricht

$$m = 4$$

- Der, aus dem λ-Kreis der Multiplikatoren der Untergruppe $\{2, \dots, q-1\}$ gewählte, eigene ***private Schlüssel***

$$k_{prv} = 2$$

Das Signaturprogramm gibt zwei Zahlen vor, die aus dem Multiplikator-λ-Kreis $\bmod q$ der Untergruppe $\{2, \dots, q-1\}$ stammen:

- Eine geheime(!) ***Zufallszahl*** k

$$k =_{zB} 18$$

Sie wird als *gewünschtes Endergebnis* interner Berechnungen vorgegeben und *muss für jeden Signaturprozess geändert werden*.

k wird zu $P_k = kG = 18 \cdot (2\,,4) = (\mathbf{9}\,, 37)$ verschlüsselt! $\boldsymbol{P_k}$ ***dient später der Verifikation!****Fehler! Verweisquelle konnte nicht gefunden werden.*

- Eine ***Koordinate*** ρ des Punktes P_k, standardmäßig ist es die x-Koordinate

$$\rho = 9$$

(ρ dient als weiterer Multiplikator in der folgenden Rechnung)

Für die Berechnung der Signatur (Abbildung 18)

- wird zum Hash $m = 4$ der private Schlüssel $k_{prv} = 2$ insgesamt $\rho = 9$mal addiert!

$$k^* = m + \rho k_{prv} \equiv 22 \mod q_{=23}$$

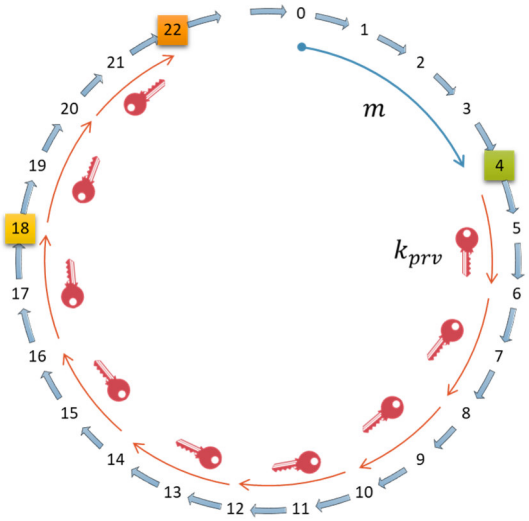
k^* ist ein Zwischenergebnis. *Hier soll das Endergebnis aber $k = 18$ lauten*.

- Berechnung des ***Korrektur-Quotienten*** σ, sodass

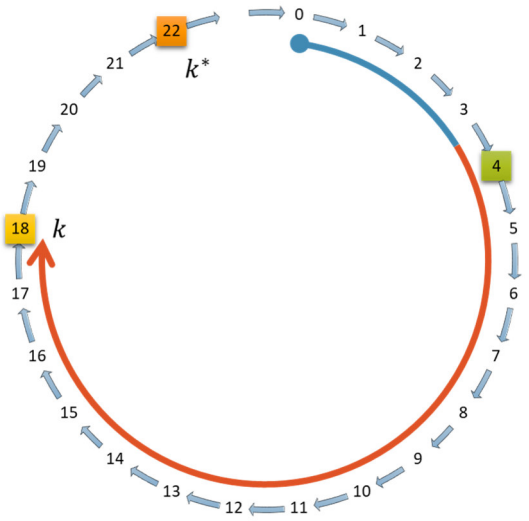
$$\frac{\boldsymbol{m + \rho k_{prv}}}{\boldsymbol{\sigma}} =: \mathbf{18} = \boldsymbol{k}$$

Daraus folgt für σ: $\sigma = \underbrace{(m + \rho k_{prv})}_{=22} \cdot k^{-1} \equiv 22 \cdot 9 \equiv 14 \bmod q$

[Probe: $(\boldsymbol{m + \rho k_{prv}})\boldsymbol{\sigma^{-1}} = (4 + 9 \cdot 2) \cdot 14^{-1} = 22 \cdot 14^{-1} \equiv 22 \cdot 5 = 110 \equiv \mathbf{18}$]



$k^* = m + \rho \cdot k_{prv} = 4 + 9 \cdot 2 = 22$



$k = (m + \rho \cdot k_{prv})\, \sigma^{-1} = (4 + 9 \cdot 2) \cdot 5 \equiv 18$

Abbildung 18: Multiplikator-Kreis mod q=23

Berechnung des öffentlichen Schlüssels

- Der ***öffentliche Schlüssel*** ist der EC-Punkt des privaten Schlüssels in $E(\mathbb{Z}_p)$.

$$K_{pub} = k_{prv} G$$
$$K_{pub} = (12\,, 36)$$

Trotz Kenntnis von K_{pub} und G ist die Lösung nach k_{prv} ein DLP und bietet deshalb *bei genügend großen Zahlen ein hohes Maß an Sicherheit*.

Ausgabe der Signatur

- EC mit a, b, p, q, G
- Die Hashfunktion (SHA-1, SHA-256, MD5, etc.)
- Die Multiplikatoren ρ und σ
- ev. der öffentliche Schlüssel K_{pub}, falls noch nicht veröffentlicht.

VERIFIKATION

Während die Erstellung der Signatur hauptsächlich in der Menge der Multiplikatoren der Untergruppe abläuft, findet die Verifikation im Raum der Punktmenge der elliptischen Kurve statt.

Die Software auf Seiten des Empfängers überprüft die Signatur folgendermaßen:

1. Berechnung des Hashwertes $\hat{m}$ der empfangenen Nachricht. Wurde die Nachricht unverfälscht übertragen, dann ist $m = \hat{m}$. Das ist aber im Moment noch nicht klar.

2. Berechnung von $\hat{m}/\sigma = \hat{m}\sigma^{-1}$ und $\rho/\sigma = \rho\sigma^{-1}$

   Bei Voraussetzung, dass $\hat{m} = m$ ist

$$\hat{m}\sigma^{-1} = 4 \cdot 14^{-1} = 4 \cdot 5 \equiv_{23} 20$$
$$\rho\sigma^{-1} = 9 \cdot 14^{-1} = 9 \cdot 5 \equiv_{23} 22$$

3. Zur Verifikation wird der Punkt $P_k = kG$ ohne Kenntnis von k und k_{prv} berechnet.

   Der private Schlüssel ist zwar unbekannt, seine Verschlüsselung durch $k_{prv}G = K_{pub}$ und der Generatorpunkt G jedoch schon.

$$P_k = kG = \left((\hat{m} + \rho \cdot k_{prv}) \cdot \sigma^{-1}\right) G = \left(\underbrace{\hat{m}\sigma^{-1}}_{20} + \underbrace{\rho\sigma^{-1}}_{22} k_{prv}\right) G = \hat{m}\sigma^{-1} G \oplus \rho\sigma^{-1} \underbrace{k_{prv} G}_{K_{pub}}$$

$$P_k =^{?} 20G \oplus 22K_{pub} = 20(2\,, 4) \oplus 22(12\,, 36) = (0\,, 7) \oplus (12\,, 11) = (9\,, 37) \quad ✓✓✓$$

Die Signatur ist echt in dem Sinn, dass das Dokument unverfälscht angekommen ist und dass der verwendete öffentliche Schlüssel zum unbekannten privaten Schlüssel passt.

Die Sicherheit des ECDLP

Die Sicherheit beruht auf der Schwierigkeit, aus G und nG den Multiplikator n zu berechnen.

Bei symmetrischer Verschlüsselung ist die Angriffskomplexität gleich $2^{\text{Schlüssellänge}}$. Bei einem Schlüssel $\text{key} = 101101$ gibt es für Brute Force an jeder Stelle 2 Möglichkeiten (0 oder 1), also insgesamt $2^6 = 64$ mögliche Schlüssel. *AES* (Advanced Encryption Standard) ist das *symmetrische* Standardverfahren für verschlüsselten Nachrichtenaustausch. Es hat aufgrund der darin verwendeten Schlüssellänge eine ***Angriffskomplexität*** von 2^{128}. Es gibt also 2^{128} Schlüssel, die mit der Brute-Force-Methode auszuprobieren sind.

In *asymmetrischen* Verfahren werden die Schlüssel berechnet, das schließt manche Möglichkeiten aus, die Angriffskomplexität des Verfahrens verringert sich für Brute-Force-Algorithmen.

Um ein ECDH zu knacken müssen Hacker aus den abgefangenen Punkten A und B die darin enthaltenen geheimen Multiplikatoren α und β ermitteln. Mit Brute-Force müssen dazu alle möglichen Werte für α und β der verwendeten zyklische Untergruppe durchprobiert werden. Somit ist die sogenannte ***Angriffskomplexität*** von der Größe $\#\mathrm{U}$, also gleich der Anzahl der darin enthaltenen Punkte.

Ist die Untergruppe sehr groß oder wünschenswerterweise gleich $E(\mathbb{Z}_p)$, dann ist die Angriffskomplexität in der Größenordnung von $\#\mathrm{U} \approx \#E$.

Die besten Angriffe auf ein *asymmetrisches* ECDLP-Verfahren sind die sogenannten ***Quadratwurzel-Attacken***. Sie reduzieren die Angriffskomplexität auf $\sqrt{\#\mathrm{U}} \approx \sqrt{\#\mathrm{E}}$.

Um z.B. den *asymmetrischen* DH-Schlüsselaustausch mit derselben Sicherheit durchführen zu können wie das symmetrische Verfahren *AES* danach, muss die Gruppen-Kardinalität

$$\#\mathrm{U} \approx \#E \approx (2^{128})^2 = 2^{256}$$

betragen. Das ist heute (2022) die Standard-Bitlänge für asymmetrische Kryptoverfahren wie den ECDH-Schlüsselaustausch.

$\#E$ wird im Wesentlichen von der Größe der Primzahle p bestimmt (Satz von Hasse). Also muss die Primzahl von $\mathbb{Z}_p$ eine Länge in der Größenordnung von 256 Bit haben.

Der Double-and-Add-Algorithmus

Die privaten Schlüssel a und b sind möglicherweise auch 2^{256}-Bit-Zahlen. Dadurch wird die skalare Punktmultiplikation

$$nG = \underbrace{G \oplus \ldots \oplus G \oplus \ldots \oplus G}_{n \approx 2^{xxx}\ mal}$$

für αP und βP sehr aufwändig. Auch wenn moderne Computer problemlos eine Million solcher Operationen pro Sekunde ausführen können, würde dies ungefähr 10^{64} Jahre dauern.

Der Double-and-Add-Algorithmus verkürzt das vor allem im Binärsystem des Computers wesentlich.

Beispiel

Gegeben G Gesucht $53G$

Das Beispiel hat jetzt nicht die Komplexität, von der die Rede war, es soll damit auch nur der Algorithmus erklärt werden.

Ziel ist, den Multiplikator vollständig in $+1$ und $\cdot 2$-Operationen zu zerlegen. Von ungeraden Anteilen wird die Zahl 1 abgespalten, gerade Anteile werden halbiert:

$$
\begin{aligned}
&53 = \\
&52 + 1 = \\
&26 \cdot 2 + 1 = \\
&13 \cdot 2 \cdot 2 + 1 = \\
&12 + 1) \cdot 2 \cdot 2 + 1 = \\
&6 \cdot 2 + 1) \cdot 2 \cdot 2 + 1 = \\
&3 \cdot 2 \cdot 2 + 1) \cdot 2 \cdot 2 + 1 = \\
&2 + 1) \cdot 2 \cdot 2 + 1) \cdot 2 \cdot 2 + 1 = \\
&\mathbf{+1} \cdot \mathbf{2} + \mathbf{1}) \cdot \mathbf{2} \cdot \mathbf{2} + \mathbf{1}) \cdot \mathbf{2} \cdot \mathbf{2} + \mathbf{1}
\end{aligned}
$$

A ***D*** **A** ***D*** ***D*** **A** ***D*** ***D*** **A**

$$
\begin{aligned}
&53 = \\
&52 + 1 = \\
&26 \cdot 2 + 1 = \\
&13 \cdot 2^2 + 1 = \\
&12 + 1) \cdot 2^2 + 1 = \\
&6 \cdot 2 + 1) \cdot 2^2 + 1 = \\
&3 \cdot 2^2 + 1) \cdot 2^2 + 1 = \\
&2 + 1) \cdot 2^2 + 1) \cdot 2^2 + 1 = \\
&+1 \cdot 2 + 1) \cdot 2^2 + 1) \cdot 2^2 + 1
\end{aligned}
$$

A … Add 1 ***D***…Double

(Die linken Klammern wurden der Übersicht wegen weggelassen.)

Damit erhält man $53G$ in nur 9 Berechnungsschritten:

$$53G = G \cdot 53 = \Big(\big((G \cdot 1) \cdot 2 + 1\big) \cdot 2^2 + 1\Big) \cdot 2^2 + 1$$

A D A DD A DD A

$$\underset{A}{\rightarrow} G \underset{D}{\rightarrow} 2G \underset{A}{\rightarrow} 3G \underset{D}{\rightarrow} 6G \underset{D}{\rightarrow} 12G \underset{A}{\rightarrow} 13G \underset{D}{\rightarrow} 26G \underset{D}{\rightarrow} 52G \underset{A}{\rightarrow} 53G$$

Double-and-Add-Algorithmus und Seitenkanalangriffe

Abseits mathematischer Versuche, eine EC-Verschlüsselung zu knacken, bergen auch sogenannte Seitenkanalangriffe Gefahrenpotenzial. So können Beobachtungen von Schwankungen der Berechnungszeiten, des Stromverbrauchs oder der elektromagnetischen Abstrahlung.

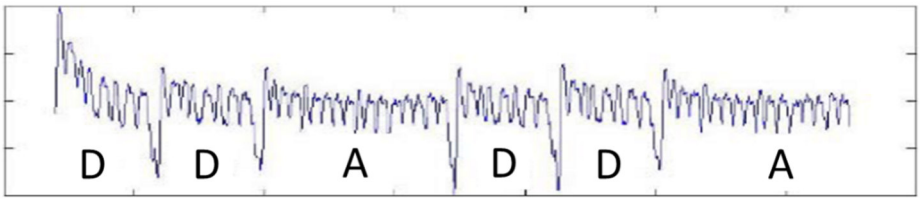
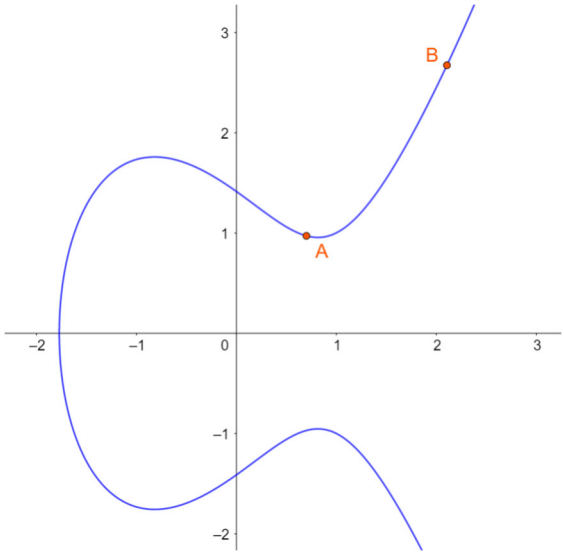


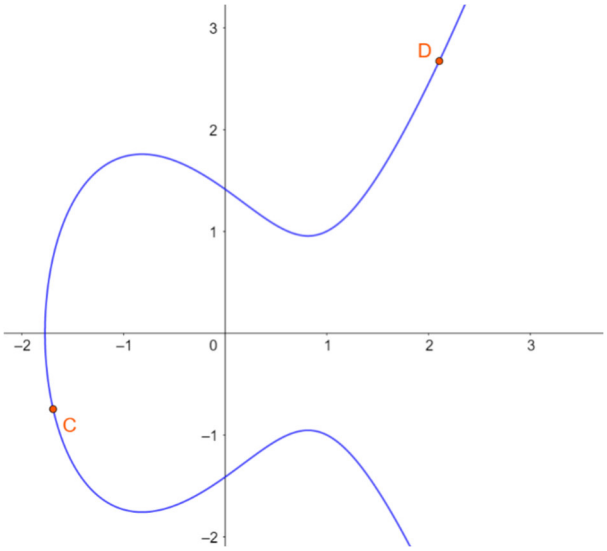
Abbildung 19: Stromverbrauchsprofil einer Sequenz von Punktadditionen (A) und -verdopplungen (D) auf einer Weierstraß-Kurve

Rückschlüsse auf einen privaten Schlüssel zulassen. Am Stromverbbrauch kann ein Schadprogramm Addition und Verdopplung erkennen (siehe Abbildung 19). Um den privaten Schlüssel geheim zu halten, muss der Algorithmus so programmiert werden, dass die Rechenzeit vom Rechenprozess unabhängig ist.

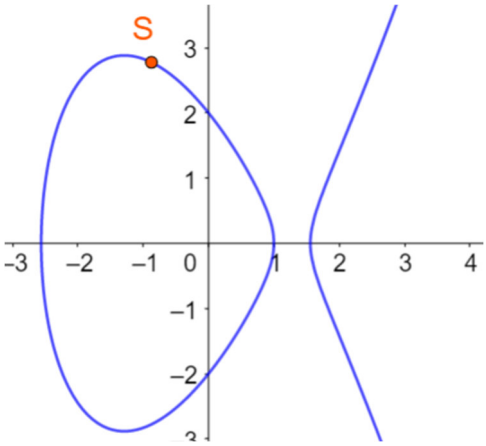
Aufgaben

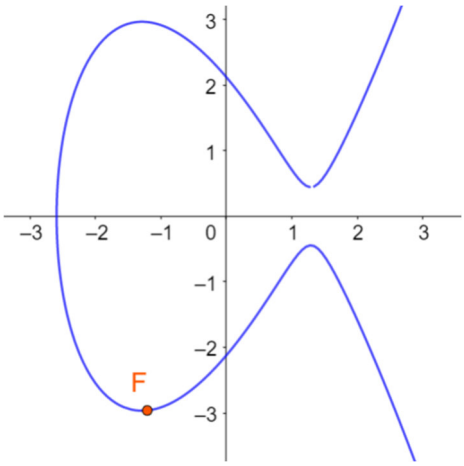
A1 Addiere graphisch $A \oplus B$ und $C \oplus D$!





A2 Verdopple S graphisch und konstruiere $3F$!

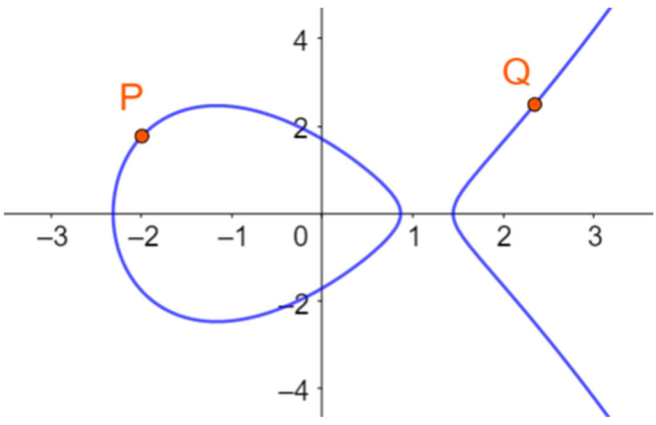




A3 Löse graphisch die Gleichung

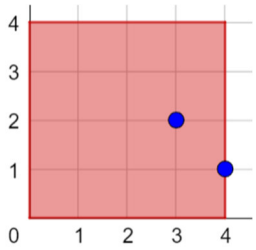
a) $X \oplus P = Q$

b) $X \oplus Q = P$



A4 Gegeben ist $E(\mathbb{Z}_5)$ mit der Bitcoin-Kurve $E_{secp256k1}: \; y^2 = x^3 + 7$ und zwei darin enthaltenen Punkte $P = (4\,,1)$ und $Q = (3\,,2)$.

a) Zeichne in nebenstehende Grafik alle Elemente von $E(\mathbb{Z}_5)$ außer $\mathcal{O}$ ein!
b) Berechne $P \oplus Q$!
c) Berechne $2P$ und $3Q$!
d) Berechne $-P$!
e) Löse $X \oplus Q = P$ nach X!



Lösungen
a) (vergrößern)
b) $(4\,,4)$
c) $(3\,,3)$ bzw. $(2\,,0)$
d) $(4\,,-1) \equiv (4\,,4)$
e) $X = P \oplus -Q = (2\,,0)$

A5 Berechne bezüglich der Angaben von Aufgabe A4 die Untergruppe von Q!

$0Q = \mathcal{O}$ $\quad 1Q = (3\,,2)$ $\quad 2Q = (??,??)$ $\quad$ Lsg: $\mathcal{O}\,,(3\,,2),(3\,,3)$
$3Q = (??,??)$ $\quad 4Q = (??,??)$ $\quad 5Q = (??,??)$

A6 Aus wie vielen Punkten besteht die elliptische Kurve $E(\mathbb{Z}_{211})$: $y^2 = x^3 + 671x + 120$?
Kreuze die richtig mögliche Variante an!

☐ 125 ☐ 225 ☐ 325 ☐ 425 ☐ 525 ☐ 625

Lösung: $212 \pm 2\sqrt{211}$

A7 EC Parameter (Weierstraß)

$a = 3$, $b = 22$, $p = 23$, Generator $G = (6\,,16)$

Kreuze an, welche Punkte der Untergruppe U_G angehören!
Verwende dazu z.B. https://andrea.corbellini.name/ecc/interactive/modk-mul.html

☐ (11 , 12) ☐ (5 , 22) ☐ (20 , 3) ☐ (16 , 16) ☐ (4 , 12) ☐ (8 , 11) ☐ (22 , 15) ☐ (3 , 9)
☐ (21 , 13) ☐ (17 , 15) ☐ (1 , 16) ☐ (13 , 21) ☐ (2 , 17) ☐ (14 , 5) ☐ (6 , 7) ☐ (7 , 15)
☐ (7 , 8) ☐ (6 , 16) ☐ (14 , 18) ☐ (2 , 6) ☐ (13 , 2) ☐ (1 , 7) ☐ (17 , 8) ☐ (21 , 10)
☐ (3 , 14) ☐ (22 , 8) ☐ (8 , 12) ☐ (4 , 11) ☐ (16 , 7) ☐ (20 , 20) ☐ (5 , 1) ☐ (11 , 11)
☐ $(\mathcal{O})$

A8 Konstruiere unter Voraussetzung der in A7 gegebenen Parameter private Schlüssel sowohl für Alice als auch für Bob und demonstriere damit einen DH-Schlüsseltausch.

Hilfsmittel: https://andrea.corbellini.name/ecc/interactive/modk-mul.html auch offline in den Dateien bzw. Excel-Datei auch ebendort.

A9 Montgomery-Kurven sind elliptische Kurven der Form

$$By^2 = x^3 + Ax^2 + x \;, \qquad B(A^2 - 4) \neq 0$$

Leite die Formel der Punktaddition für Montgomery-Kurven her. Gehe dabei analog der Herleitung für Weierstraß-Kurven vor (siehe Anhang).

Lösung: $s_{P \neq Q} = (y_Q - y_P)(x_Q - y_Q)^{-1}$; $s_{2P} = (3x_P^2 + 2Ax_P + 1)(sBy_P)^{-1}$
$P \oplus Q = R$; $x_R = Bs^2 + A - x_P - x_Q \quad y_R = s(x_P - x_R) - y_P$

A10 Erstellung einer Signatur

Die zu signierende Nachricht: Der eigene Name z.B. Karin Musterfrau

EC domain:

Parameters	
p	hex 000000000000040F
a	hex 0000000000000000
b	hex 0000000000000007
G	hex(00000000000002ED, 0000000000000173)
q	hex 000000000000015D

Hilfsmittel H1: https://andrea.corbellini.name/ecc/interactive/modk-mul.html bzw. in den Dateien.
Hilfsmittel H2: Wolfram Alpha
Hilfsmittel H3: Online-Converter Hex – Dec
Hilfsmittel H4: Hashfunktion MD5 (Onlinetool)

a) Umwandlung der angegebenen Hexadezimalzahlen in Dezimalzahlen und Vergleich mit den unten abgegebenen Werten. (H3)

$p = 1039$ $a = 0$ $b = 7$ $G = (749\,,\,371)$ $q = 349$

b) Überprüfe, ob p und q Primzahlen sind! (H2: „is ##### prime?“)

c) Bestimme den Hashwert deines Namens mit Hilfe der Hashfunktion MD5 (H4). Verwende für die weiteren Berechnungen die ersten 3 signifikanten Stellen.
z.B.: m = c23072682d1cae66bd35a14384e888b6 = **258**~~12178187986856779530710653785397880 6~~

d) Bestimme eine Zufallszahl für deinen privaten Schlüssel $k_{prv} < q$!
z.B. $k_{prv} = 299$. Denk dir aber eine eigene aus!

e) Berechne den zugehörigen öffentlichen Schlüssel! (H1) $\rightarrow^{zB} K_{pub} = (804\,,74)$

f) Bestimme die ephemere Zufallszahl $k < q$!
z.B. $k = 131$. Denk dir aber eine eigene aus!

g) Bestimme R und damit ρ! (H1) $\rightarrow^{zB} R = (496\,,650)$

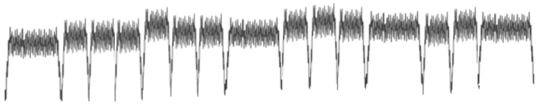
h) Berechne $\sigma = (m + \rho k_{prv})/k$ (H2: „$\underbrace{\#\#\#}_{Zahl}$ ^(−1) mod $\underbrace{\#\#\#}_{Modulus}$ “
$\rightarrow^{zB} \sigma = (258 + \rho \cdot 299) \cdot 8 \equiv 151$

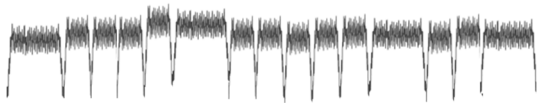
i) Woraus besteht jetzt die Signatur? Was muss zusätzlich zur Signatur noch öffentlich sein?

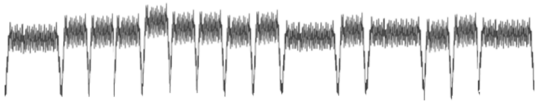
j) Verifiziere die Signatur!
$\rightarrow^{zB}$ $m = 258$; $m/\sigma = 41$; $\rho/\sigma = 68$; $\Longrightarrow (207\,,106) \oplus (629\,,548) = (496\,,650)$ ■

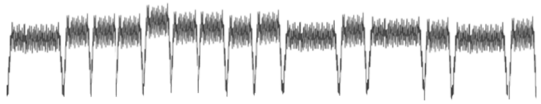
A11 Double and Add

Welches der folgenden Stromprofile passt zum privaten Schlüssel $k_{prv} = 2061$?

☐ 

☐ 

☐ 

☐ 

A12 Schlüssellänge und Sicherheit

BrainpoolP192r1: (Die Präfix „0x“ steht für „hexadezimal“)

p 0x c302f41d932a36cda7a3463093d18db78fce476de1a86297
a 0x 6a91174076b1e0e19c39c031fe8685c1cae040e5c69a28ef
b 0x 469a28ef7c28cca3dc721d044f4496bcca7ef4146fbf25c9
G (0x c0a0647eaab6a48753b033c56cb0f0900a2f5c4853375fd6,
0x 14b690866abd5bb88b5f4828c1490002e6773fa2fa299b8f)
q 0x c302f41d932a36cda7a3462f9e9e916b5be8f1029ac4acc1

a) Wie groß ist die Kardinalität $\#E$ von BrainpoolP192r1?
b) Wie groß ist die Kardinalität $\#U$ der verwendeten Untergruppe von BrainpoolP192r1?
c) Bestimme die Schlüssellänge!
d) Bestimme die Sicherheit bzw. Angriffskomplexität von BrainpoolP192r1!
e) Mit welcher Sicherheit symmetrischer Schlüssel ist das vergleichbar?

Lsgn:
a)b) $\#U = q$ per Definition. Aus $q = p$ folgt hier auch $\#U = \#E$.
$\#U = \#E \approx$
0b 11000011 00000010 11110100 00011101 10010011 00101010 00110110 11001101 10100111 10100011 01000110 00110000 10010011 11010001 10001101 10110111 10001111 11001110 01000111 01101101 11100001 10101000 01100010 10010111 =
0d 4 781 668 983 906 166 242 955 001 894 344 923 773 259 119 655 253 013 193 367
c) 192-Bit
d) 2^{192}
e) 2^{384} (384-Bit)

Anhang

Punktadditionsformeln (Herleitung)

Gegeben sind die elliptische Kurve $E: y^2 = x^3 + ax + b$ sowie die Punkte $P = (x_P\,, y_P)$ und $Q = (x_Q\,, y_Q)$. In Abbildung 20 ist die Punktaddition grafisch dargestellt.

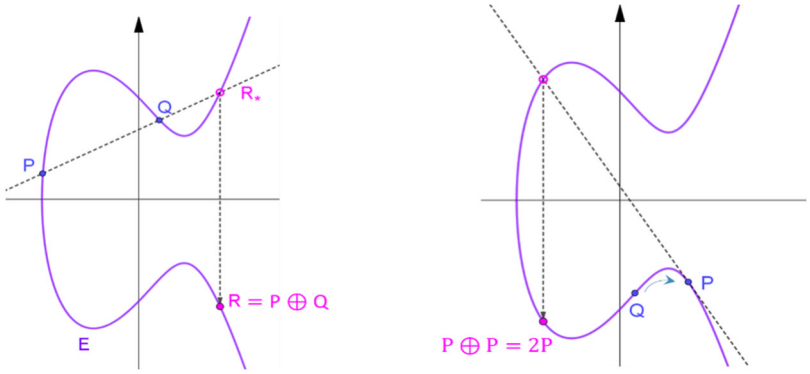


Abbildung 20: Links $(P \neq Q)$ $P \oplus Q$, rechts $2P = P \oplus P$

Fall $P \neq Q$:

$$g(P,Q): y = s \cdot x + d \text{ mit } s = \frac{y_Q - y_P}{x_Q - x_P} \cdot x + d.$$

$$P \in g(P,Q) \Longrightarrow y_P = s \cdot x_P + d \Longrightarrow \quad *)\ d = y_P - s \cdot x_P$$

$$g \cap E: (s \cdot x + d)^2 = x^3 + a \cdot x + b$$

$$s^2x^2 + 2sdx + d^2 = x^3 + a \cdot x + b$$

$$**)\quad x^3 - s^2x^2 + (a - 2sd)x + (b - d^2) = 0$$

Die Lösungen sind die bekannten x_P und x_Q und das noch unbekannte x_{R_*}. Somit ließe sich die Gleichung auch mit den Linearfaktoren anschreiben.

$$(x - x_P)(x - x_Q)(x - x_{R_*}) = 0$$

Ausmultipliziert ergibt das

$$***)\quad x^3 - (x_P + x_Q + x_{R_*}) \cdot x^2 + (x_Px_Q + x_Px_{R_*} + x_Qx_{R_*}) \cdot x - x_Px_Qx_{R_*} = 0.$$

Die Gleichungen $**)$ und $***)$ sind ja äquivalent und beide normiert (der Leitkoeffizient ist 1). Die Koeffizienten müssen einander also entsprechen. Vergleicht man den Koeffizienten von x^2, so ergibt sich

$$s^2 = x_P + x_Q + x_{R_*} \Longrightarrow x_{R_*} = s^2 - x_P - x_Q$$

Die y-Koordinate entnehmen wir der Geradengleichung unter Berücksichtigung von $*)$

$$y_{R_*} = s \cdot x_{R_*} + y_P - s \cdot x_P = y_P - s \cdot (x_P - x_{R_*})$$

$$\Longrightarrow P \oplus Q = R = (x_{R_*}\,, -y_{R_*}) = (s^2 - x_P - x_Q\ ,\ s \cdot (x_P - x_R) - y_P) \quad \blacksquare$$

Fall $2P = P \oplus P$ (Punkt-Verdopplung)

Der Fall ist identisch dem vorherigen, mit einer Ausnahme. Der Anstieg s muss anders berechnet werden, da die Gerade g jetzt eine Tangente von E darstellt.

$$E: y^2 = x^3 + ax + b$$

Die Gleichung implizit differenziert ergibt

$$2yy' = 3x^2 + a$$

und damit speziell für den Punkt P

$$2y_P y'_P = 3x_P^2 + a \Longrightarrow y'_P = s = (3x_P^2 + a)(2y_P)^{-1} \quad \blacksquare$$

Das Ergebnis ist bewusst nicht als Division angeschrieben, da diese eigentlich, und insbesondere in $\mathbb{Z}_p$, über das inverse Element definiert ist.

Transformation Montgomery → Weierstraß

Die in Messenger-Diensten wie WhatsApp, Signal etc. verwendete elliptische Kurve *Curve25519* hat die Gleichung

$$E:\ y^2 \equiv x^3 + 486662x^2 + x \ \text{ über } \mathbb{Z}_p$$

und gilt in dieser Form als besonders schnell. Die Primzahl ist in diesem Fall $p = 2^{255} - 19$, was der Kurve den Namen gegeben hat.

Gemäß ihrer Gleichung ist *Curve25519* als Montgomery-Kurve $By^2 \equiv x^3 + Ax^2 + x$ klassifiziert.

Satz
Jede Montgomery-Kurve E kann in eine bezüglich der Kryptographie gleichwertige Weierstraß-Form $\tilde{E}$ transformiert werden kann.

Bemerkung: Die Umkehrung ist nur unter gewissen Bedingungen möglich.

Beweis
Der Einfachheit halber zeigen und veranschaulichen wir die Transformation für E über $\mathbb{R}$. Für den endlichen Körper $\mathbb{Z}_p$ müssen lediglich die Bruchstrich-Divisionen durch multiplikativ Inverse ersetzt werden.

$$E:\ By^2 = x^3 + Ax^2 + x \ \mid : B^3$$

$$\left(\frac{y}{B}\right)^2 = \left(\frac{x}{B}\right)^3 + \frac{A}{B}\left(\frac{x}{B}\right)^2 + \frac{1}{B^2}\left(\frac{x}{B}\right)$$

Neuskalierung in x- und y-Richtung: $\frac{y}{B} \to y \quad \frac{x}{B} \to x$

$$y^2 = x^3 + \frac{A}{B}x^2 + \frac{1}{B^2}x$$

Translation um $A/_{3B}$ in x-Richtung: $\quad y \to y \quad x \to \left(x - \frac{A}{3B}\right)$

$$y^2 = \left(x - \frac{A}{3B}\right)^3 + \frac{A}{B}\left(x - \frac{A}{3B}\right)^2 + \frac{1}{B^2}\left(x - \frac{A}{3B}\right)$$

ergibt ausmultipliziert eine äquivalente Form $\tilde{E}$ von E:

$$y^2 = x^3 + \frac{3 - A^2}{3\,B^2}x + \frac{A(2A^2 - 9)}{27B^3}$$

$$y^2 = x^3 + ax + b$$

Montgomery $E: By^2 = x^3 + Ax^2 + x \;\rightarrow$ Weierstraß $\tilde{E}: y^2 = x^3 + ax + b$

mit $a = (3 - A^2)(3B^2)^{-1}$ *und* $b = A(2A^2 - 9)(27B^3)^{-1}$

Beispiel

Montgomery E: $5y^2 = x^3 + 7x^2 + x$

Weierstraß $\tilde{E}$: $y^2 = x^3 - \frac{46}{75}x + \frac{623}{3375}$

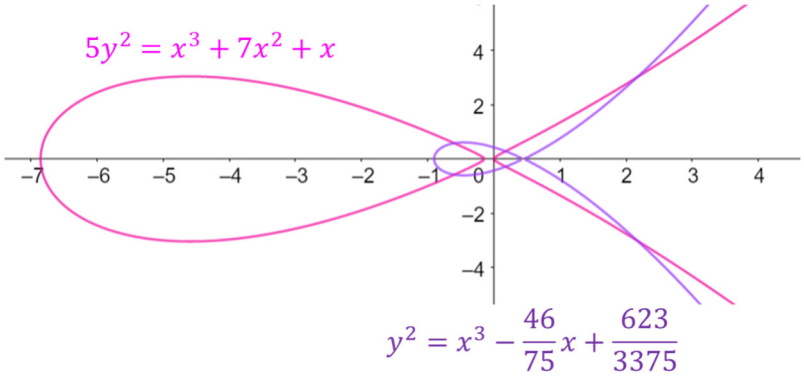


Abbildung 21