Humboldt-Universität zu Berlin
Mathematisch-Naturwissenschaftliche Fakultät
Institut für Informatik

Kryptographie mit elliptischen Kurven

Seminararbeit

eingereicht von: Hannes Benne
geboren am: 21.06.1989
geboren in: Magdeburg

Gutachter/innen: Dr. Wolf Müller

Diese Arbeit führt in die Kryptographie mit elliptischen Kurven ein. Zunächst werden die mathematischen Grundlagen, sprich elliptische Kurven über den reellen Zahlen und über endlichen Körpern eingeführt. Davon ausgehend werden Rechenoperationen auf elliptischen Kurven, insbesondere die Punktaddition und die Berechnung des diskreten Logarithmus untersucht. Die Darstellung der mathematischen Grundlagen orientiert sich an [Wer02].

Anschließend wird gezeigt, wie sich elliptische Kurven für die Kryptographie nutzen lassen. Dafür werden die Verfahren Elliptic-Curve Diffie–Hellman (ECDH) und Elliptic Curve Digital Signature Algorithm (ECDSA) vorgestellt. Die Darstellung dieser kryptographischen Verfahren orientiert sich an [Mir03] und [Fed].

Inhaltsverzeichnis

1 Elliptische Kurven

1.1 Elliptische Kurven auf $\mathbb{R}$

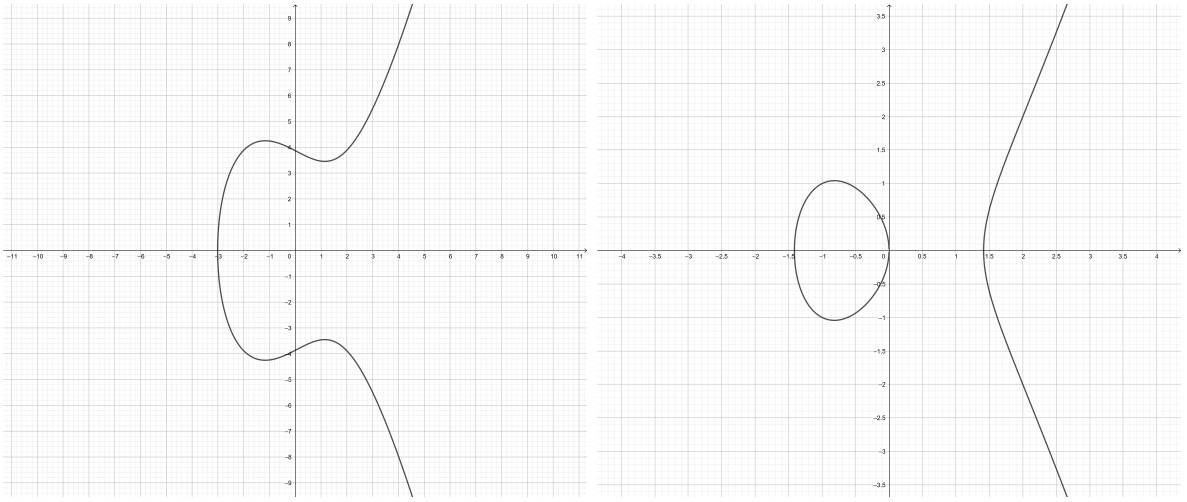
Als Motivation und für die geometrischen Anschauung von Arithmetik auf elliptischen Kurven führen wir zunächst elliptische Kurven über den reellen Zahlen ein.

Definition 1 *Die Nullstellenmenge des Weierstraßpolynoms*

$$f(x, y) = y^2 - x^3 - ax - b$$

wird als Elliptische Kurve $E(\mathbb{R})$ über $\mathbb{R}$ bezeichnet.

Es handelt sich bei elliptischen Kurven also um eine Menge von Punkten, welche die Gleichung $y^2 = x^3 + ax + b$ erfüllen. Diese können wir in der Ebene zeichnen.



(a) $y^2 = x^3 - 4x + 15$ (b) $y^2 = x^3 - 2x$

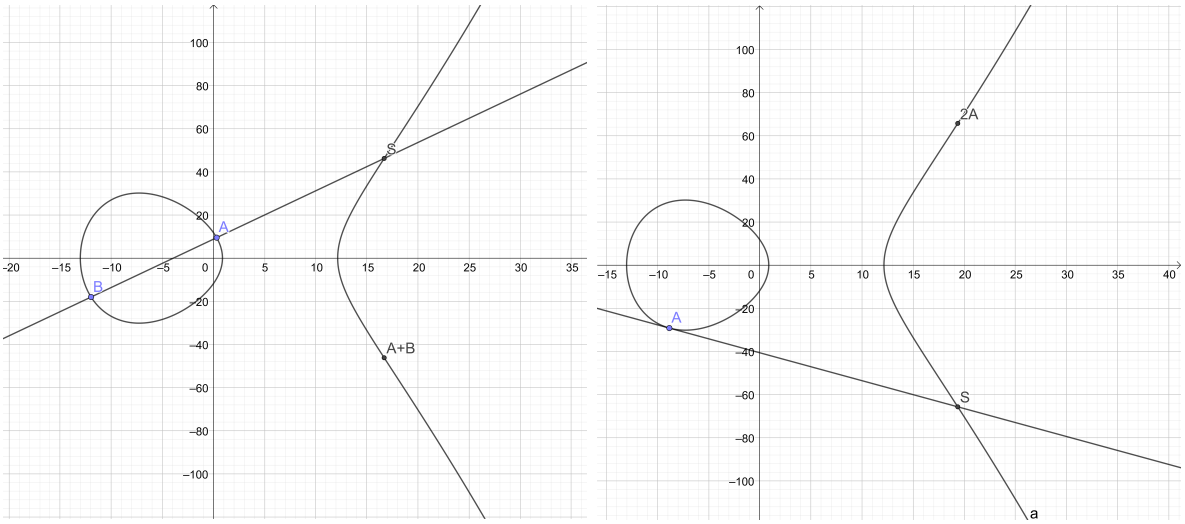
Abbildung 1: Beispiele für elliptische Kurven über $\mathbb{R}$

Für zwei Punkte $A = (x_A, y_A), B = (x_B, y_B)$ auf $E(\mathbb{R})$ definieren wir eine Addition $+ : E(\mathbb{R}) \times E(\mathbb{R}) \to E(\mathbb{R})$ auf folgende Weise:

- **Fall 1:** Sei $A \neq B$ und $y_A \neq -y_B$. Wir legen eine Sekante durch die Punkte A und B. Diese hat einen dritten Schnittpunkt S mit der Kurve $y^2 = x^3 + ax + b$. Diesen Schnittpunkt S spiegeln wir an der x - Achse und definieren den daraus resultierenden Punkt als $A + B$.

- **Fall 2:** Sei $A = B$ und $x_A \neq 0$. Wir legen eine Tangente durch A. Diese hat einen weiteren Schnittpunkt S mit $y^2 = x^3 + ax + b$. Diesen Schnittpunkt S spiegeln wir an der x - Achse und definieren den daraus resultierenden Punkt als $A + A = 2A$.

– **Fall 3:** Sei $y_A = -y_B$. In diesem Fall hat Sekante durch A und B keine weiteren Schnittpunkt mit $y^2 = x^3 + ax + b$ und wir führen einen projektiven Punkt $\mathcal{O}$ ein und setzen $A + B = \mathcal{O}$.



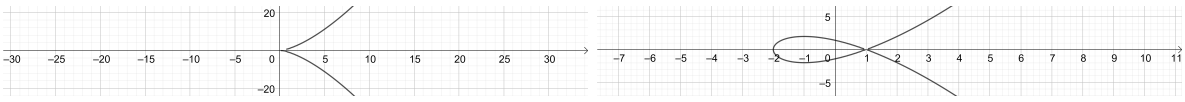
(a) Addition von A und B (b) Punktverdopplung von A

Abbildung 2: Arithmetik auf elliptischen Kurven

Im Fall der Punktverdopplung stellt sich die Frage, ob es in überhaupt möglich ist, in jedem Punkt der Kurve eine Tangente anzulegen. In Punkten in denen die Ableitungen des Weierstraßpolynoms verschwinden, lässt sich keine Tangente anlegen.

Definition 2 *Eine elliptische Kurve heißt singulär im Punkt $A = (x, y)$, falls gilt:*

$$f(x, y) = 0, \quad \frac{\partial f(x, y)}{\partial x} = 0, \quad \frac{\partial f(x, y)}{\partial y} = 0.$$



(a) $y^2 = x^3$ ist singulär in $(0, 0)$ (b) $y^2 = x^3 - 3x + 2$ ist singulär in $(1, 0)$

Abbildung 3: Beispiele für singuläre Punkte

Damit wir auf einer Kurve rechnen können, soll diese keine singulären Punkte besitzen. Eine derartige Kurve wird *nicht singuläre* elliptische Kurve genannt. Ein hinreichendes Kriterium für die Nichtsingularität ist, dass für die Koeffizienten des Weierstraßpolynoms gilt $4a^2 + 27b^2 \neq 0$.

In der folgenden Sektion wird näher darauf eingegangen, dass elliptische Kurven mit der Punktaddition eine Gruppenstruktur tragen. Einige Eigenschaften abelscher Gruppen werden wir uns jedoch schon anhand der geometrischen Darstellung der Punktaddition veranschaulichen.

- **Kommutativgesetz:** Wenn wir für zwei Punkte $A, B \in E(\mathbb{R})$ eine Sekante durch A und B legen, ist diese offenbar gleich der Sekante durch B und A. Das bedeutet wir erhalten in beiden Fällen den gleichen dritten Schnittpunkt auf der Kurve und es gilt $A + B = B + A$.
- **Inverses Element:** Wir können einen Punkt invertieren, indem wir seine y-Komponente invertieren. Das heißt der inverse Punkt zu $A = (x, y)$ ist $-A = (x, -y)$. Wenn wir durch A und $-A$ eine Sekante legen, erhalten wir nach Definition als dritten Schnittpunkt den projektiven Punkt $\mathcal{O}$. Dieser wird das neutrale Element der Gruppe sein. Also gilt $A + (-A) = \mathcal{O}$.
- **Inverses Element:** Für eine Gerade durch A und den projektiven Punkt $\mathcal{O}$ erhalten wir $-A$ als dritten Schnittpunkt. Spiegeln wir diesen an der x-Achse, erhalten wird wieder A. Es gilt also $A + \mathcal{O} = A$.
- **Assoziativgesetz:** Die Assoziativität ist nicht direkt aus der geometrischen Anschauung ersichtlich. Wir werden aber in der folgenden Sektion Formeln für die Punktaddition angeben mit denen sich nachrechnen lässt, dass für Punkte $A, B, C \in E(\mathbb{R})$ gilt: $(A + B) + C = A + (B + C)$.

1.2 Endliche Körper

Elliptische Kurven über den reellen Zahlen liefern eine geometrische Anschauung für Punktarithmetik auf Kurven, sind aber für die Kryptografie ungeeignet. Denn reelle lassen Zahlen sich im Computer nicht exakt darstellen und sind bei der Arithmetik rundungsfehlerbehaftet. Besser geeignet sind Galoiskörper (auch endliche Körper genannt). Unter diesen sind zwei Klassen für die Kryptografie von Interesse:

- $GF(2^n)$ mit $n \in \mathbb{N}$. Körper mit 2^n Elementen, sind für die Hardwareimplementierung von kryptografischen Verfahren von Bedeutung.
- $GF(p)$ mit $p \in \mathbb{P}$. Primkörper sind für die Softwareimplementierung von Kryptografischen Verfahren von Bedeutung.

Wir werden uns im Folgenden auf die Körper $GF(p)$ beschränken. Da es – bis auf Isomorphie – genau einen Körper mit p Elementen gibt, gilt $GF(p) \cong \mathbb{Z}_p = \{0, \dots p-1\}$. Für diesen Restklassenkörper lassen sich Rechenoperationen einfach implementieren.

Von der National Security Agency (NSA) und dem National Institute of Standards and Technology (NIST) werden für Elliptische Kurven Kryptographie Körper mit folgenden Bitlängen für p empfohlen: 192, 224, 256, 384, und 521 bit.

1.3 Elliptische Kurven auf $GF(p)$

Definition 3 *Sei $GF(p)$ ein Galoiskörper mit Primzahlcharakteristik. Für $a, b \in GF(p)$ mit $4a^2 + 27b^2 \not\equiv 0$ mod p bezeichnen wir die Menge*

$$E(GF(p)) = \{(x, y) \in GF(p) \times GF(p) \mid y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\}$$

als elliptische Kurve über $GF(p)$.

Zeichnerisch lassen sich elliptische Kurven über endlichen Körpern als Punktwolken in der ebene Darstellen. Ein Beispiel dafür wird im Abschnitt über das diskrete Logarithmusproblem gegeben.

1.4 Gruppenstruktur auf E(GF(p))

Wie auf $E(\mathbb{R})$ lässt sich auch auf $E(GF(p))$ eine Addition einführen. Das geometrische Verfahren, wie für Punkte auf $E(\mathbb{R})$ vorgestellt, hat den Nachteil, dass es sich schlecht für die Implementierung von Algorithmen eignet. Wir geben daher Formeln für die Addition von Punkten auf elliptischen Kurven über $GF(p)$.

Für $A, B \in E(GF(p))$ mit $A = (x_1, y_1), B = (x_2, y_2)$ definiere wir:

1. $-A = (x_1, -y_1)$,
2. $A + B = \mathcal{O}$, falls $A = -B$,
3. $A + B = (x_3, y_3), fallsA \neq B$

Wobei die Koordinaten (x_3, y_3) wie folgt berechnet werden:

$$x_3 = \lambda^2 - x_1 - x_2$$
$$y_3 = \lambda(x_1 - x_3) - y_1$$
$$\lambda = \begin{cases} \dfrac{y_1 - y_2}{x_1 - x_2} & \text{falls } P \neq Q \\ \dfrac{3x_1^2 + a}{2y_1} & \text{falls } P = Q \end{cases}$$

Bei der Addition und Multiplikation handelt es sich um Arithmetik im Körper $(GF(p), +, \cdot)$, das bedeutet es wird modulo p gerechnet.

Wie im reellen Fall, so tragen auch elliptische Kurven über endlichen Körpern mit der oben definierten Addition eine Gruppenstruktur. Der Beweis dafür, dass die Addition auf $E(GF(p))$ eine abgeschlossene Verknüpfung ist, wird in Kapitel 2 von [Wer02] mit Hilfe der projektiven Geometrie geführt. Auch der Nachweis von Inversen und des neutralen Elementes erfordern theoretische Grundlagen, welche in dieser Arbeit nicht vorgestellt werden. Lediglich das Kommutativgesetz und das Assoziativgesetz lassen sich mit den vorgestellten Formeln nachrechnen.

1.5 Endlichkeit von E(GF(p))

Anders als im reellen Fall, hat die Gruppe $E(GF(p))$ endliche Ordnung. Das heißt, nur endlich viele Punkte erfüllen die Gleichung $y^2 = x^3 + ax + b$. Eine allgemeine Formel für die exakte Anzahl der Elemente in $E(GF(p))$ ist nicht bekannt, allerdings liefert der Satz von Hasse eine Abschätzung:

$$-2\sqrt{p} + p + 1 \leq |E(GF(p))| \leq 2\sqrt{p} + p + 1$$

Die Ordnung von $E(GF(p))$ ist also ungefähr gleich der Anzahl der Elemente in $GF(p)$.

Die Arbeit [Mir03] widmet sich hauptsächlich der Bestimmung der Punktzahl auf elliptischen Kurven. Dort werden mit dem Schoof–Algorithmus und dem SEA–Algorithmus zwei Verfahren vorgestellt, mit denen sich dir Ordnung von $E(GF(p))$ in Polynomialzeit bestimmen lässt.
Ob ein kryptografisches Verfahren mit elliptischen Verfahren sicher ist, hängt unter anderem von der Gruppenordnung der verwendeten Kurve ab, wie wir später sehen werden.

1.6 Skalarmultiplikation auf E(GF(p))

Auf folgende Weise können wir Punkte auf elliptischen Kurven mit einer natürlichen Zahl multiplizieren:

$$n \cdot P = \underbrace{P + P + \ldots + P}_{n \text{ mal}}$$

Wenn n eine s–stellige Binärzahl ist, hat die Berechnung von $n \cdot P$ eine Zeitkomplexität von $\mathcal{O}(2^s)$. Die Skalarmultiplikation auf diese Weise zu implementieren ist für die Praxis daher ungeeignet. Allerdings existieren schnellere Verfahren, wie der Double–and–add–Algorithmus. Die Idee zur schnellen Skalarmultiplikation liefert die little-endian Binärdarstellung des Skalars, hier exemplarisch für $n = 101$.

$$101_{10} = (1010011)_2 = 1 \cdot 2^0 + 1 \cdot 2^2 + 1 \cdot 2^5 + 1 \cdot 2^6.$$

Sei nun b_k das k-te Bit in der Binärdarstellung einer s-Bit Zahl n, dann berechnet der Double–and–add–Algorithmus $n \cdot P$ wie folgt:

Input : P, n
Output : $Result = n \cdot P$
$Result \leftarrow 0$;
for $k = 1 \ldots s$ **do**
  if $b_k == 1$ **then**
    $Result \leftarrow Result + P$;
  end
  $P \leftarrow 2 \cdot P$
end

Für unser Beispiel erhalten wir mit dem Algorithmus:

- $k = 1$**:**

$$\begin{aligned} b_k &= 1 \\ Result &\leftarrow Result + P \quad (= 2^0 \cdot P) \\ G &\leftarrow 2 \cdot P \end{aligned}$$

- $k = 2$**:**

$$\begin{aligned} b_k &= 0 \\ P &\leftarrow 2 \cdot P \quad (= 4 \cdot P) \end{aligned}$$

- $k = 3$**:**

$$\begin{aligned} b_k &= 1 \\ Result &\leftarrow Result + P \quad (= 2^0 \cdot P + 2^2 \cdot P) \\ P &\leftarrow 2 \cdot P \quad (= 8 \cdot P) \end{aligned}$$

- $k = 4$**:**

$$\begin{aligned} b_k &= 0 \\ P &\leftarrow 2 \cdot P \quad (= 16 \cdot P) \end{aligned}$$

- $k = 5$**:**

$$\begin{aligned} b_k &= 1 \\ P &\leftarrow 2 \cdot P \quad (= 32 \cdot P) \end{aligned}$$

- $k = 6$**:**

$$\begin{aligned} b_k &= 1 \\ Result &\leftarrow Result + P \quad (= 2^0 \cdot P + 2^2 \cdot P + 2^5 \ cdotP) \\ P &\leftarrow 2P \quad (= 64 \cdot P) \end{aligned}$$

- $k = 7$**:**

$$\begin{aligned} b_k &= 1 \\ Result &\leftarrow Result + P \quad (= 2^0 \cdot P + 2^2 \cdot P + 2^5 \cdot P + 2^6 \cdot P = 101 \cdot P) \end{aligned}$$

Statt 100 Additionen haben wir nur sechs Punktverdopplungen und vier Additionen benötigt. Wenn Addition und Punktverdopplung eine Zeitkomplexität von $\mathcal{O}(1)$ haben, so liegt der Double–and-add-Algorithmus in $\mathcal{O}(s)$.

2 Das Diskrete Logarithmusproblem

In diesem Abschnitt werden wir das diskrete Logarithmusproblem auf Elliptischen Kurven (ECDLP) betrachten. Die Sicherheit der elliptischen Kurven Kryptografie beruht darauf, dass dieses Problem schwer zu lösen ist.

Als Vorbereitung erinnern wir uns an die Definition von durch P erzeugten Untergruppen. Die vom P erzeugte Untergruppe enthält gerade die Potenzen von P. Weil $E(GF(p))$ endlich ist, ist $\langle P \rangle$ nach dem Satz von Lagrange ebenfalls endlich und wir können schreiben:

$$\langle P \rangle = \{k \cdot P \mid k \in \mathbb{Z}\} = \{k \cdot P \mid 0 \leq k \leq ord(P)\}.$$

Definition 4 (ECDLP) *Gegeben sei eine Elliptische Kurve $E(GF(p))$ und ein Punkt P der Ordnung n auf E. Weiter sei Q ein Punkt aus der von P erzeugten Untergruppe $\langle P \rangle$ gegeben, für den gilt: $Q = k \cdot P$, $0 \leq k \leq n-1$. Zu bestimmen ist dann die Zahl k.*

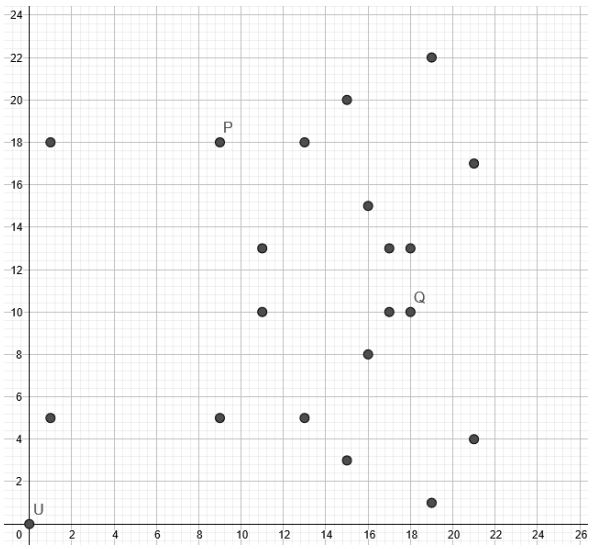


Abbildung 4: Die Kurve $y^3 = x^3 + x$ über $GF(23)$

Die Multiplikation eines Punktes P mit $k \in \mathbb{N}$ entspricht graphisch k chaotisch wirkenden Sprüngen von Punkt zu Punkt in der Punktwolke. Das umgekehrte Problem – die Bestimmung des diskreten Logarithmus – kann wie folgt veranschaulicht werden: Wir haben zwei Punkte P und Q in der Punktwolke gegeben. Gefragt ist nun die Anzahl k der Sprünge, die wir benötigen um von Punkt P nach Punkt Q zu gelangen.

Die schnellsten derzeit bekannten Algorithmen zur Lösung des ECDLP haben exponentielle Laufzeit. Exemplarisch sei hier der Pohlig-Hellman-Algorithmus zu nennen. Dieser

nutzt Faktorisierungen der Gruppenordnung n von $\langle P \rangle$. Anstatt den Logarithmus k direkt zu berechnen, bestimmt der Algorithmus k modulo p_i, wobei p_i die Primfaktoren der Gruppenordnung n sind. Den Logarithmus selbst berechnet man dann mit Hilfe des Chinesischen Restsatzes.
Dieses Vorgehen ist effizient, wenn die Gruppenordnung $|E(GF(p))|$ relative kleine Primfaktoren hat. Es muss $E(GF(p))$ also so gewählt werden, dass die Primfaktoren von $|E(GF(p))|$ sehr groß sind. Empfohlen werden Primfaktoren, die mindestens 160 Bit lang sind. Im Idealfall ist $|E(GF(p))|$ selbst bereits prim.

3 Anwendungen

3.1 Elliptic Curve Diffie–Hellman

Der Diffie-Hellmann-Schlüsselaustausch ist ein Verfahren, mit dem zwei Kommunikationspartner (Alice und Bob) über öffentliche Kanäle einen gemeinsamen Schlüssel vereinbaren können. Dieser kann dann für symmetrische Verschlüsselungsverfahren eingesetzt werden. Für den Schlüsselaustausch gehen Alice und Bob wie folgt for:

- Zunächst einigen sich Alice und Bob auf die Parameter a, b, p, P. Dabei definieren a, b, p eine elliptische Kurve und P ist ein Punkt auf dieser Kurve.
- Alice wählt eine zufällige Natürliche Zahl x mit $1 \leq x \leq ord(P)$ als privaten Schlüssel. Dann sendet sie den Punkt $x \cdot P$ an Bob.
- Bob wählt eine zufällige Natürliche Zahl y mit $1 \leq y \leq ord(P)$ als privaten Schlüssel. Dann sendet sie den Punkt $y \cdot P$ an Alice.
- Alice berechnet mit ihrem privaten Schlüssel den Punkt $x \cdot (y \cdot P) = x \cdot y \cdot P$.
  Bob berechnet mit seinem privaten Schlüssel den Punkt $y \cdot (x \cdot P) = x \cdot y \cdot P$

Bob und Alice besitzen nun den Punkt $x \cdot y \cdot P$ als gemeinsamen Geheimnis. In der Praxis wird die x-Koordinate oder ein Hashwert des Punktes als Schlüssel verwendet.

Ein Angreifer der den Schlüsseltausch mithört, kann $x \cdot P$ und $y \cdot P$ in Erfahrung bringen. Aber ohne das ECDLP lösen zu können, kann er daraus weder x noch y bestimmen und deswegen den Punkt $x \cdot y \cdot P$ nicht berechnen.

Während der Elliptic Curve Diffie–Hellman mit der Gruppe $E(GF(p))$ arbeitet, wird für den klassischen Diffie–Hellman-Algorithmus die Gruppe $GF(p)$ verwendet. Zunächst scheint die aufwendigere Berechnung von Potenzen in $E(GF(p))$ – um $n \cdot P$ zu berechnen werden mehrere Rechenoperationen in $GF(p)$ benötigt – von Nachteil zu sein. Allerdings ist das Diskrete Logarithmusproblem (DLP) in $E(GF(p))$ noch schwieriger zu lösen als in $GF(p)$. Die Zeitkomplexität des ECDLP nimmt linear mit p zu, die Zeitkomplexität des DLP nur logarithmisch mit p. Das führt dazu, dass wir bei gleichbleibendem Sicherheitsniveau mit wesentlich kürzeren Schlüssellängen auskommen, wenn wir den Elliptic

Curve Diffie–Hellman-Algorithmus verwenden. Die daraus resultierende Einsparung an Rechenzeit wird meist um den Faktor 10 angegeben.

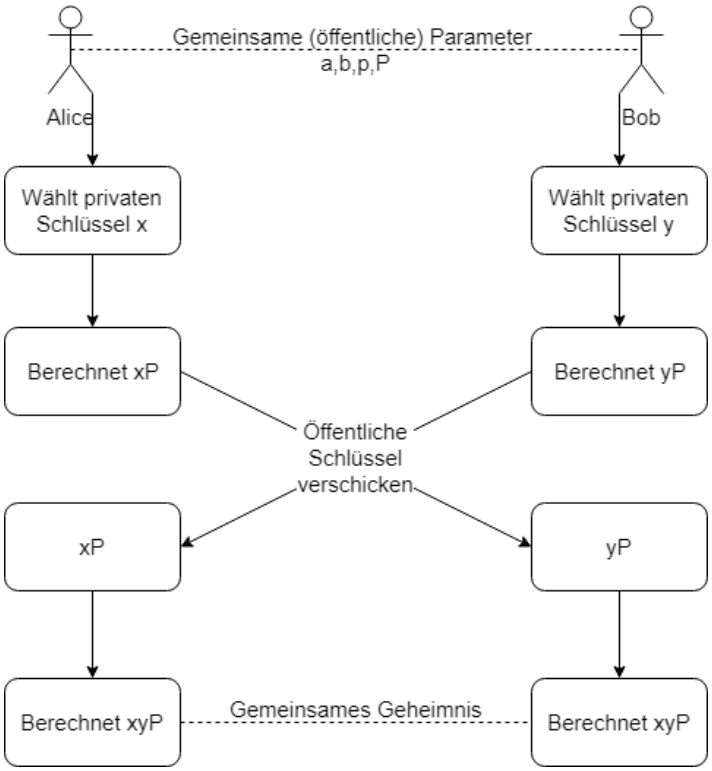


Abbildung 5: Elliptic Curve Diffie–Hellman

3.2 Elliptic Curve Digital Signature Algorithm

Digitale Signaturen sind ein asymetrisches Kryptosystem. Der Absender einer Nachricht berechnet mit Hilfe seines privaten Schlüssels einen Wert, welcher als Signatur bezeichnet wird. Diesen Wert versendet er gemeinsam mit der Nachricht. Jeder der den öffentlichen Schlüssel des Absenders besitzt, kann mit der Signatur und öffentlichem Schlüssel die Urheberschaft der Nachricht verifizieren.

Die bekanntesten Signaturverfahren sind das RSA-Kryptosystem, welches auf dem Faktorisierungsproblem für große Primzahlen basiert und der Digital Signature Algorithm (DSA), welcher auf dem diskreten Logarithmusproblem in $GF(p)$ basiert. Wir werden eine Version des DSA betrachten, welche auf elliptischen Kurven basiert.

Alice möchte Bob eine signierte Nachricht schicken. Dazu werden zunächst die öffentlichen Parameter (a, b, p, P, n, h) festgelegt. Dabei definieren die Parameter a, b, p eine elliptische Kurve. h ist eine Hashfunktion. Weiter ist P ein Punkt auf dieser Kurve und $n = ord(P)$. Dabei soll n eine Primzahl sein und den Abschätzungen $n > 2^{160}$ und $n > 4\sqrt{p}$ genügen.

Zudem sollte n für $k = 1, \ldots, 20$ kein Teiler von $q^k - 1$ sein. Diese Forderungen an n stellen sicher, dass der diskrete Logarithmus auf $E(GF(p))$ schwer zu bestimmen ist.

Alice möchte eine Nachricht m signieren. Zunächst benötigt sie ein öffentlich–privates Schlüsselpaar. Als privaten Schlüssel wählt sie eine Zahl $d \in \{1, \ldots, n-1\}$. Den öffentlichen Schlüssel berechnet sie mit dem privaten Schlüssel und dem öffentlichem Punkt P als $d \cdot P$. Um aus dem öffentlichen Schlüssel den privaten Schlüssel zurück zu rechnen, müsste das ECDLP gelöst werden. Für die Berechnung der Signatur geht sie wie folgt vor:

1. Sie wählt eine zufällige Zahl k mit $1 \leq k \leq n-1$ als privaten Schlüssel.
2. Aus der Zufallszahl k und dem Punkt P berechnet sie $k \cdot P$.
3. Aus der x–Koordinate P_x von P berechnet Alice den Wert $r \equiv P_x \bmod n$ in $\mathbb{Z}_n$. Falls $r \equiv 0$ ist, so beginnt sie wieder in Schritt 1.
4. Sie berechnet k^{-1} in $\mathbb{Z}_n$. Das ist immer möglich, da n eine Primzahl und $\mathbb{Z}_n$ ein Körper ist.
5. Alice berechnet den Hashwert $h(m)$ ihrer Nachricht.
6. Mit Hilfe des privaten Schlüssels berechnet sie $s \equiv k^{-1}(h(m) + rd) \bmod n$. Falls $s \equiv 0$, so beginn Alice wieder in Schritt 1.

Die Signatur ist das Tupel (r, s).

Hat Bob die Nachricht m mit Signatur (r, s) von Alice erhalten, so geht er – nachdem er die öffentlichen Parameter (a, b, p, P, n, h) und den öffentlichen Schlüssel $k \cdot P$ erhalten hat – wie folgt vor, um die Signatur zu überprüfen:

1. Bob prüft, ob gilt $1 \leq r, s \leq n-1$. Wenn das nicht der Fall ist, ist die Signatur ungültig.
2. Aus der Nachricht m berechnet er den Hashwert $h(m)$.
3. In $\mathbb{Z}_n$ berechnet Bob den Wert $w = s^{-1}$. In $E(GF(p))$ berechnet er den Punkt $R = w(h(m)P + rd \cdot P)$.
4. Bob prüft, ob $R = \mathcal{O}$. Wenn das der Fall ist, ist die Signatur ungültig.
5. Bob prüft, ob für die x-Koordinate R_x von R gilt $r \equiv R_x \bmod n$. Wenn das der Fall ist, ist die Signatur gültig. Andernfalls ist die Signatur ungültig.

Die Korrektheit des Algorithmus wird in [Wer02] beweisen.
Neben der Wahl einer geeigneten elliptischen Kurve muss bei dem Signaturverfahren sichergestellt werden, dass ein geeigneter Zufallszahlengenerator verwendet wird. Weiterhin ist es für die Sicherheit des Verfahren essentiell, dass Alice nie für zwei unterschiedliche

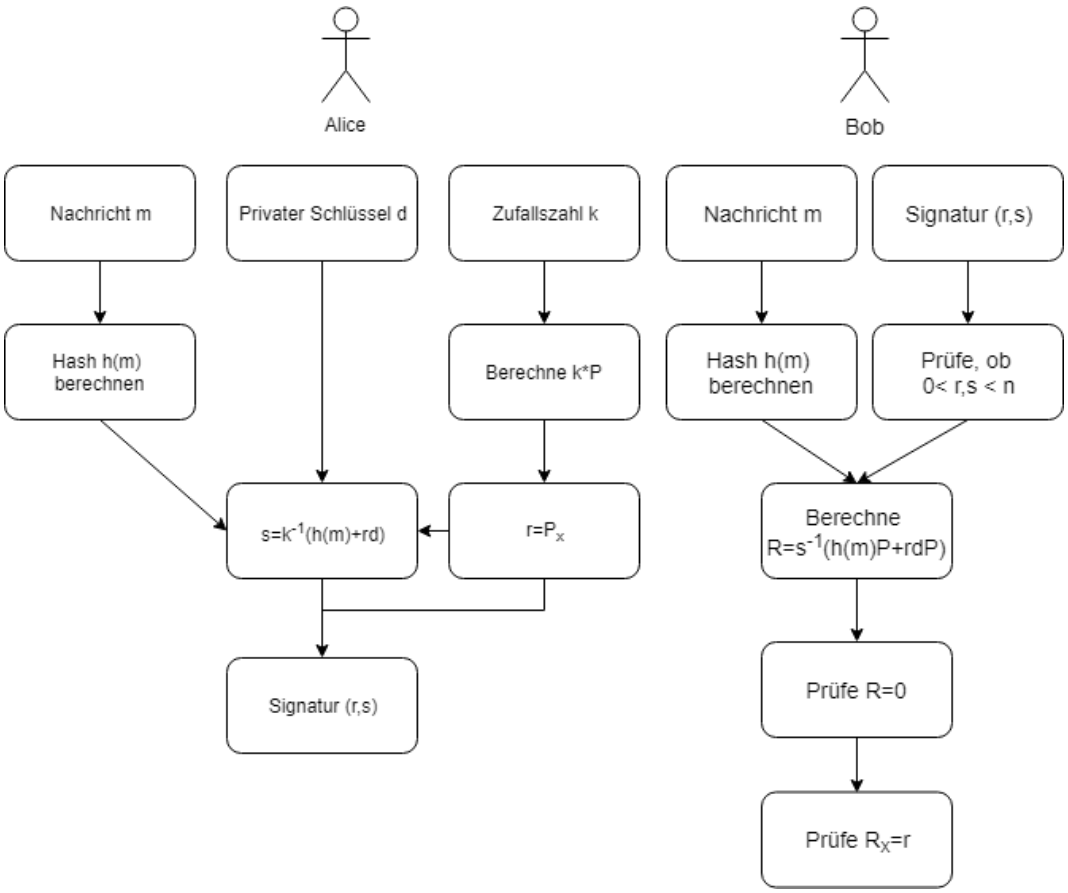


Abbildung 6: Erstellung und Verifikation von Signaturen

Nachrichten den gleichen Wert k verwendet, um die Signatur zu berechnen. Denn wenn sie für Nachrichten m_1, m_2 zweimal den gleichen Wert k wählt, gilt:

$$ks_1 \equiv h(m_1) + rd \bmod n$$
$$ks_2 \equiv h(m_2) + rd \bmod n.$$

Daraus lässt sich mit $k(s_1 - s_2) \equiv h(m_1) - h(m_2) \bmod n$ der Wert k berechnen. Ist k bekannt, kann ein Angreifer gefälschte Signaturen erstellen.

3.3 Vor- und Nachteile

Vorteile von elliptischer Kurven Kryptographie:

- Die Berechnung von privat–öffentlichen Schlüsselpaaren ist sehr einfach. Zu einem privaten Schlüssel d erhält man für einen Punkt P auf einer elliptischen Kurve den öffentlichen Schlüssel als $d \cdot P$.
- Elliptische Kurven Verfahren benötigen vergleichsweise kurze Rechenzeiten. So ist

bspw. der Elliptic Curve Diffie-Hellman Schlüsseltausch etwa um den Faktor 10 schneller als der Diffie-Hellman Schlüsseltausch.

– Bei gleichbleibendem Sicherheitsniveau kommt die elliptische Kurven Kryptographie im Vergleich zu RSA, DH und DSA mit der sehr kurzen Schlüssellängen aus.

Zeit den Schlüssel zu knacken in MIPS-Jahren	RSA-Schlüssellänge (in Bits)	ECC-Schlüssellänge (in Bits)	Verhältnis
10^4	512	102	4,8:1
10^8	768	132	5,8:1
10^{12}	1024	160	6,4:1
10^{20}	2048	210	9,8:1

Tabelle 1: Vergleich von Schlüssellängen

Nachteile von elliptischer Kurven Kryptographie:

– Die Berechnung von geeigneten elliptischen Kurven ist aufwändig.

– Algorithmen für Arithmetik auf vom NIST herausgegebenen Kurven sind patentbehaftet.

– Verfahren und Kurven die von amerikanischen Behörden – wie NSA und NIST – veröffentlich werden, wird von einigen Personen wenig Vertrauen entgegen gebracht.

Literatur

[Fed] Federal Office for Information Security. Bsi tr-03111 elliptic curve cryptography, version 2.10.

[Mir03] Andreas Mirbach. *Elliptische Kurven. Die Bestimmung ihrer Punktzahl und Anwendungen in der Kryptographie.* Verlagshaus Monsenstein und Vannerdat, 2003.

[Wer02] Anette Werner. *Elliptische Kurven in der Kryptographie.* Springer-Verlag, Berlin Heidelberg New York, 2002.