

Grundlagen der Elliptic Curve Cryptography

Mathias Schmalisch · Dirk Timmermann

Universität Rostock



Übersicht

- Motivation
- Elliptische Kurven
- Punktoperationen
- Körper
- Hardwareimplementierung
- Zusammenfassung

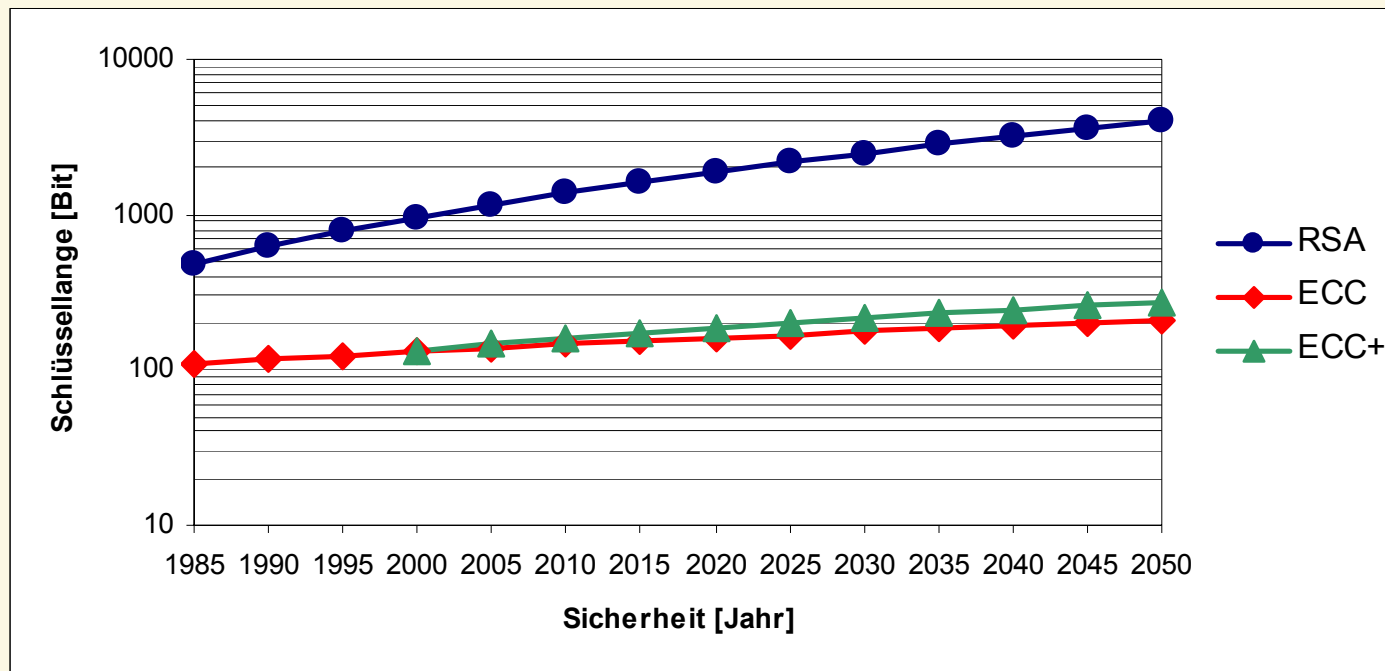


Motivation

- für asymmetrische Verfahren werden lange Schlüssel benötigt
- lange Schlüssel benötigen viel Rechenzeit
- meist nur für Schlüsselaustausch und/oder Signatur benötigt, daher viel Overhead
- wenn der Schlüssel einmal geknackt wurde, können alle damit verschlüsselten Nachrichten entschlüsselt werden

Vergleich mit RSA Verfahren

- Schlüssellänge wesentlich kürzer als bei RSA Verfahren
- RSA Schlüssellänge steigt schneller an
- ECC Berechnungen sind allerdings aufwendiger

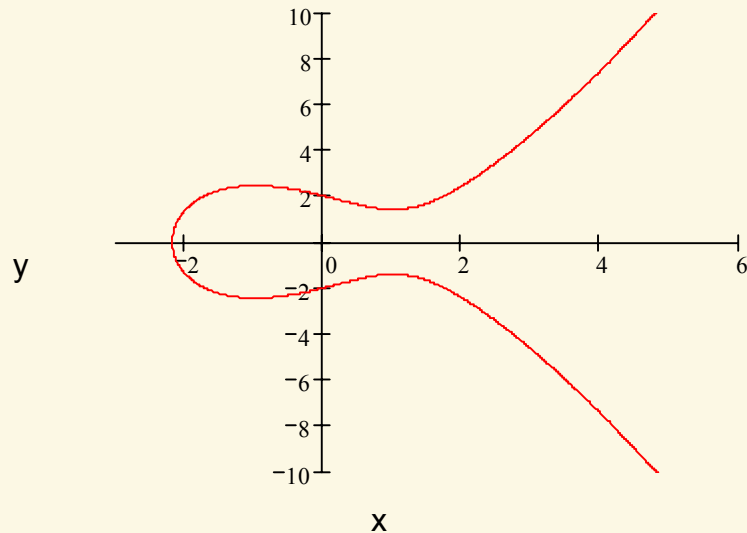


Beispiel

- Diffie-Hellmann Schlüsselaustausch
- Alice wählt geheime Zahl a und Bob die Zahl b
- Alice und Bob einigen sich auf eine ell. Kurve E und Punkt F
- Alice berechnet $PA = a * F$
- Bob berechnet $PB = b * F$
- PB und PA werden ausgetauscht
- Alice berechnet Schlüssel x mit $x = a * PB$
- Bob berechnet Schlüssel x mit $x = b * PA$

$$b * PA = b * (a * F) = b * (F * a) = (b * F) * a = a * PB$$

Elliptische Kurven



- Weierstrass Normalform:

$$E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

$$x, y, a_i \in K$$

- Charakteristik von $K \neq 2, 3$

$$E: y^2 = x^3 + a_4x + a_6$$

- Charakteristik von $K = 2$

$$E_1: y^2 + xy = x^3 + a_2x^2 + a_6$$

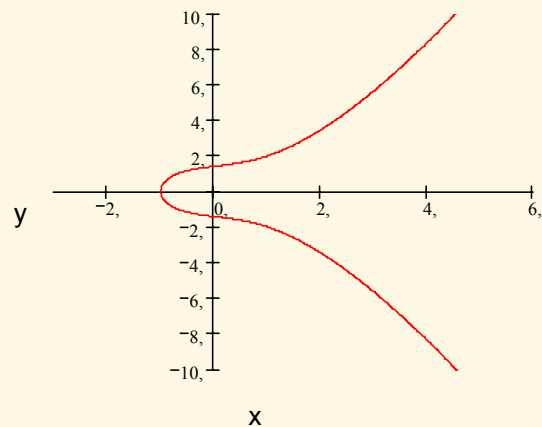
$$E_2: y^2 + a_3y = x^3 + a_4x + a_6$$

Elliptische Kurven

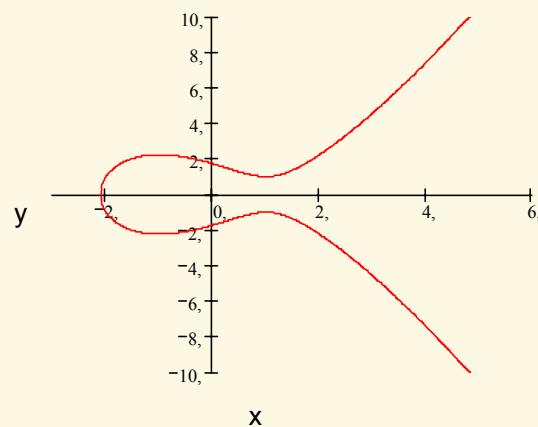
- Kurve E/K , mit $K = \mathbf{R}$ (reelle Zahlen)

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

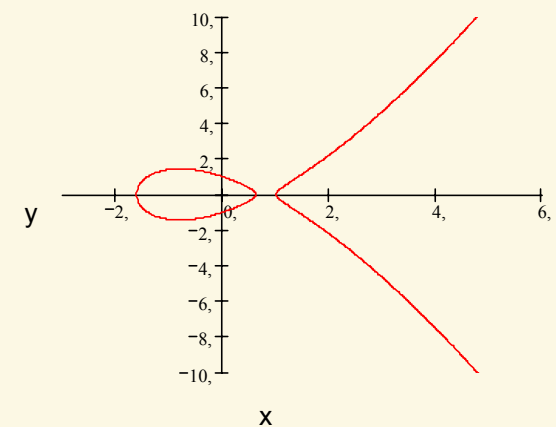
- Koeffizienten a, b müssen ganze Zahlen sein
- Kurven mit einer oder drei Nullstellen



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



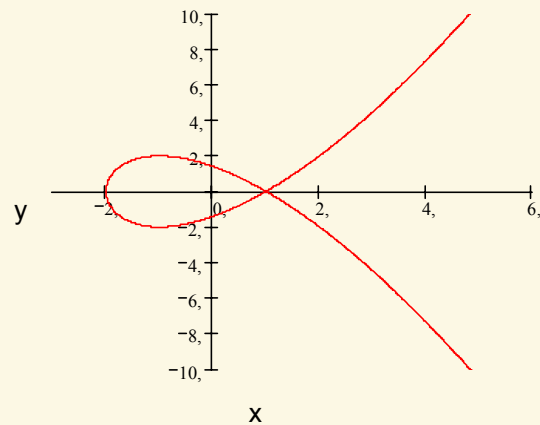
$$y^2 = x^3 - 3x + 3$$



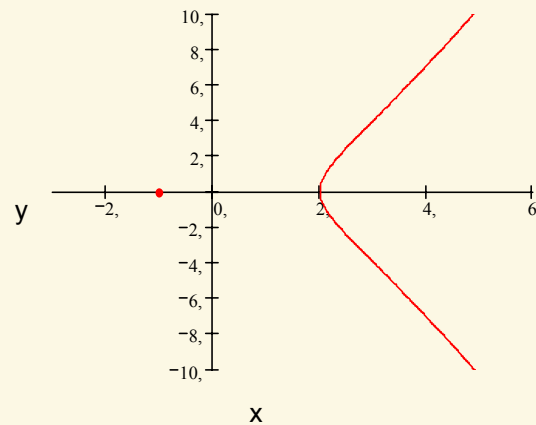
$$y^2 = x^3 - 2x + 1$$

Elliptische Kurven

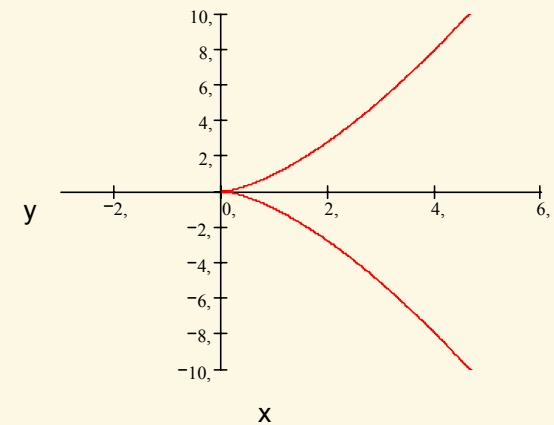
- Kurven mit doppelten Nullstellen
- werden auch singuläre/degenerierte Kurven genannt
- Punktoperationen sind auf diesen Kurven nicht definiert
- für Kryptographie ungeeignet



$$y^2 = x^3 - 3x + 2$$



$$y^2 = x^3 - 3x - 2$$



$$y^2 = x^3$$

Diskriminante Δ

- mit der Diskriminante Δ kann eine singuläre Kurve ermittelt werden
- bei singulären Kurven ist $\Delta = 0$
- Charakteristik von $K \neq 2,3$

$$E: y^2 = x^3 + a_4x + a_6$$

$$\Delta = -16 (4a_4^3 + 27a_6^2)$$

- Charakteristik von $K = 2$

$$E_1: y^2 + xy = x^3 + a_2x^2 + a_6$$

$$\Delta = a_6$$

$$E_2: y^2 + a_3y = x^3 + a_4x + a_6$$

$$\Delta = a_3^4$$

Punktoperationen

Punkte auf einer Kurve E bilden eine abelsche Gruppe
für alle Punkte $P, Q \in E$ gilt:

1. $0 + P = P$ und $P + 0 = P$ (0 ist das neutrale Element)
2. $-0 = 0$
3. wenn $P = (x_1, y_1) \neq 0$, dann $-P = (x_1, -y_1 - a_1x_1 - a_3)$
4. wenn $Q = -P$, dann $P + Q = 0$
5. Punktaddition: $P + Q = R, R \in E$

Punktaddition

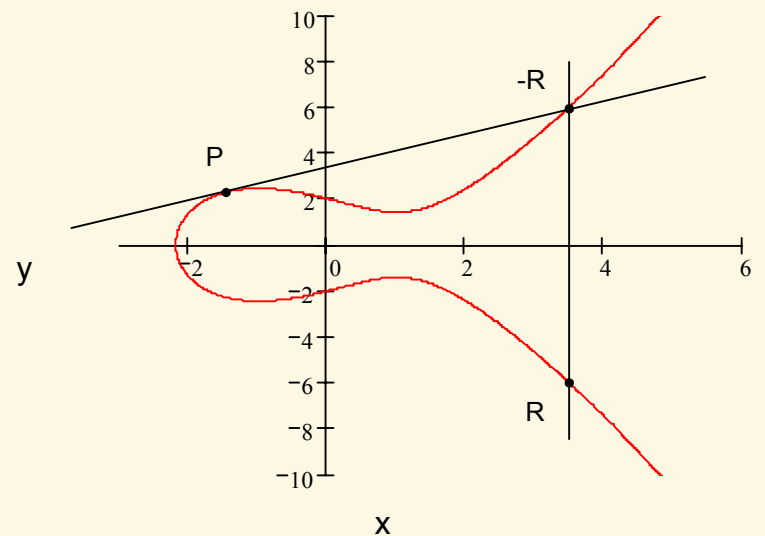
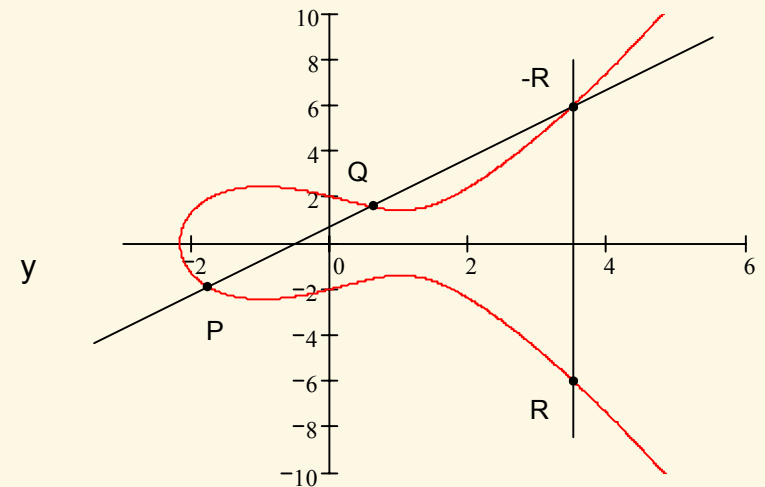
- $P \neq Q$
- Linie durch P und Q

- $P = Q$
- Tangente an P

- Berechnung von R

$$x_3 = \lambda^2 + a_1\lambda - a_2 - x_1 - x_2$$

$$y_3 = -(\lambda + a_1)x_3 - \beta - a_3$$



Skalarmultiplikation

- Multiplizieren mit einem Skalar

$$k * P = P + P + P + \dots + P \quad (k-1 \text{ Punktadditionen})$$

- Vereinfachung mit Punktverdopplung (z.B. $k = 9 = [1001]_2$)

$$2P = P + P \text{ (Punktverdopplung)}$$

$$4P = 2P + 2P$$

$$8P = 4P + 4P$$

$$9P = P + 8P \text{ (Punktaddition)}$$

- für $k = 2^{150}$, im Durchschnitt etwa 225 Punktoperationen

Körper

- K ist eine nichtleere Menge
- auf K sind zwei Operanden definiert $(+,*)$ für die gilt:

1. Assoziativgesetz

$$a + (b + c) = (a + b) + c$$

$$a * (b * c) = (a * b) * c$$

2. Kommutativgesetz

$$a + b = b + a$$

$$a * b = b * a$$

3. Distributivgesetz

$$a * (b + c) = a * b + a * c$$

4. neutrales Element

$$0, \text{ mit } a + 0 = a$$

$$1, \text{ mit } a * 1 = a$$

5. inverses Element

$$-a, \text{ mit } a + (-a) = 0$$

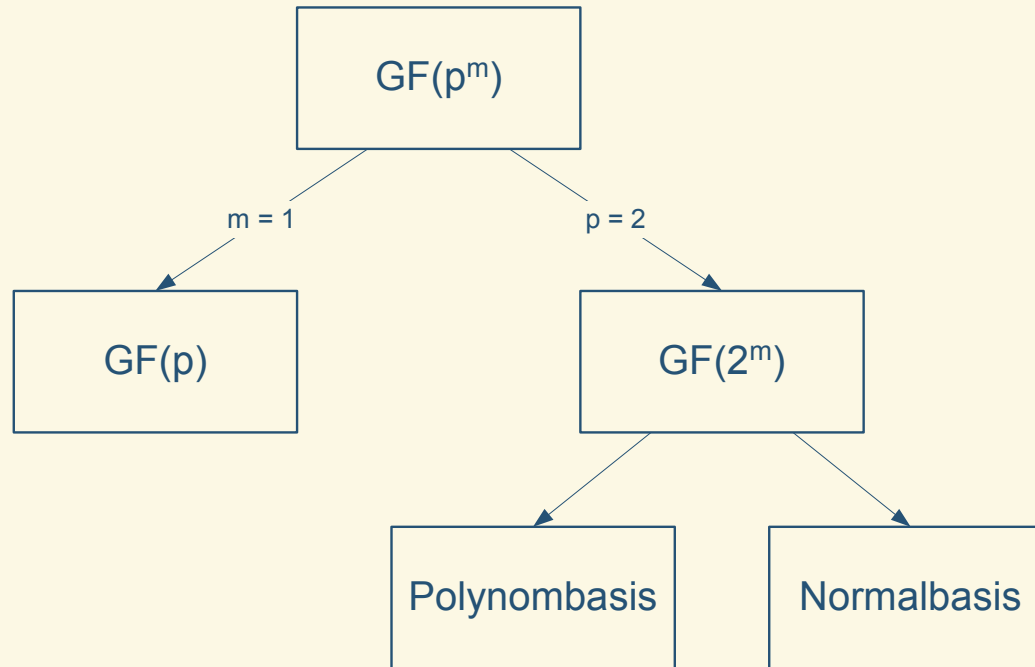
$$a^{-1}, \text{ mit } a * a^{-1} = 1$$

Körper

- K kann kein Körper sein über:
 - ▶ \mathbf{Z} – ganze Zahlen (kein inverses Element für $*$)
 - ▶ \mathbf{N} – natürliche Zahlen (kein inverses Element für $*$)
- K kann ein Körper sein über:
 - ▶ \mathbf{R} – reelle Zahlen
 - ▶ \mathbf{C} – komplexe Zahlen
 - ▶ \mathbf{Q} – irrationale Zahlen
 - ▶ \mathbf{F}_q – endlicher Körper ($\mathbf{GF}(q)$ – Galois Feld) mit q Elementen

endlicher Körper $GF(q)$

- $q = p^m$ und p muß eine Primzahl sein
- Charakteristik von $K = p$
- für Kryptographie von Interesse sind die Spezialfälle
 $m = 1$ und $p = 2$



endlicher Körper GF(p)

- Modulare Mathematik
- Beispiel: $y^2 = x^3 + 3x$ über GF(5)

$$y^2 = x^3 + 3x \pmod{5}$$

- Addition

$$3 + 3 \pmod{5} = 1$$

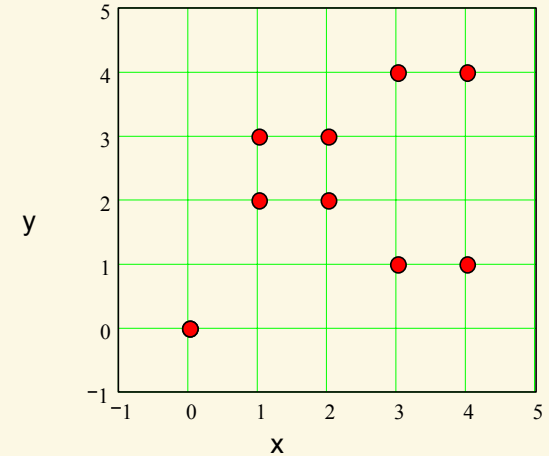
- Multiplikation

$$3 * 3 \pmod{5} = 4$$

- Modulo Inverses

$$x^{-1} \pmod{p} = 1$$

$$\text{z.B. } 3^{-1} \pmod{5} = 2, 3^{-1} = 2$$



endlicher Körper $GF(2^m)$

- Addition im $GF(2)$
 - ▶ $a + a = 2a = 0$
- Addition im $GF(2^m)$
 - ▶ Addition der einzelnen Bitstellen mod 2 ($1 + 1 = 0$)
 - ▶ $(a_0, \dots, a_{m-1}) + (b_0, \dots, b_{m-1}) = (a_0 + b_0, \dots, a_{m-1} + b_{m-1})$
- es gilt im $GF(2^m)$:
 - ▶ $a = -a$
 - ▶ $(a + b)^2 = a^2 + b^2$

Vergleich Polynom- und Normalbasis

- Basis $B = (\beta^0, \beta^1, \beta^2, \dots, \beta^{m-1})$

- irreduzibles Polynom $m(x)$

 - ▶ z.B. für $m = 8$

 - ▶ $m(x) = x^8 + x^4 + x^3 + x + 1$

- Multiplikation

 - ▶ schieben und addieren

 - ▶ Reduktion mit irr. Polynom $m(x)$

- Modulo Inverses

 - ▶ Euklidischer Algorithmus

- Basis $B = (\beta, \beta^2, \beta^{2^2}, \dots, \beta^{2^{m-1}})$

 - ▶ Nullelement $(0, \dots, 0)$

 - ▶ Einselement $(1, \dots, 1)$

- Multiplikation

 - $(a_0, \dots, a_{m-1}) * (b_0, \dots, b_{m-1}) =$

 - (c_0, \dots, c_{m-1}) , mit

$$c_k = \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} a_i + k^{b_j} + k^{l_{ij}^{(0)}}$$

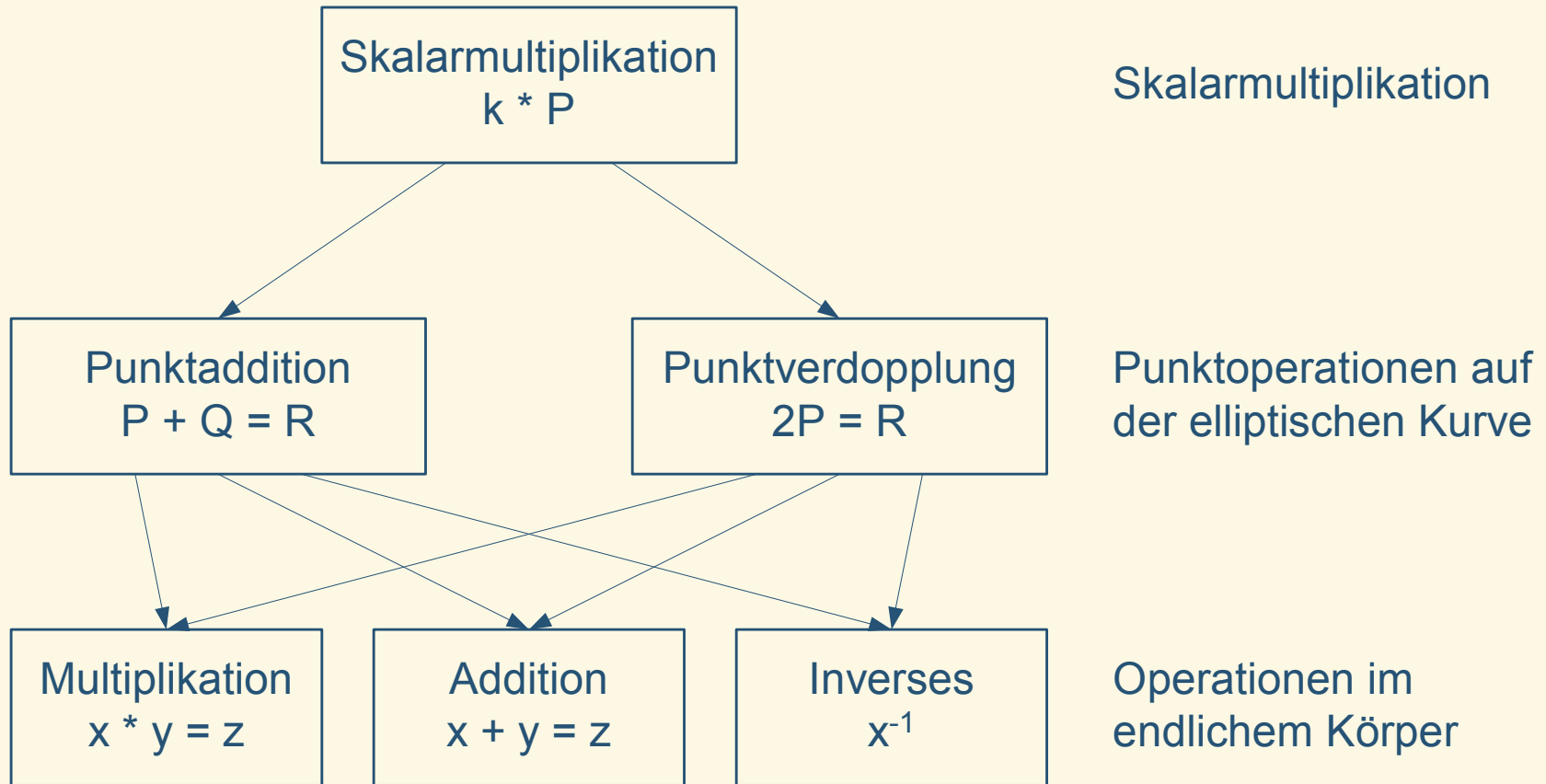
- Modulo Inverses

 - ▶ Fermatsches Theorem

- Quadrierung x^2

 - $(a_0, \dots, a_{m-1})^2 = (a_{m-1}, a_0, \dots, a_{m-2})$

Hardwareimplementierung



Hardwareimplementierung

- Elliptische Kurven über endliche Körper $GF(2^m)$ lassen sich besonders gut in Hardware implementieren
- Beispiel Polynombasis:
- Addition
 - ▶ einfachen XOR-Verknüpfung (schnell da kein Überlauf)
- Multiplikation
 - ▶ schieben und XOR-Verknüpfung
- Modulo Inverses
 - ▶ euklidischer Algorithmus, nach $2m$ Takten fertig

Zusammenfassung

- Elliptische Kurven über $GF(2^m)$ sind für Hardwareimplementierung besonders gut geeignet
- kürzere Schlüssellänge als bei anderen Verfahren (z.B. RSA)
- keine komplizierte Berechnung der Schlüssel notwendig
- für jede Nachricht kann ein neuer Schlüssel gewählt werden
- Mathematik wesentlich aufwendiger als bei anderen Algorithmen