Seminar zur Kryptologie*

Thema: Elliptische Kurven

Ralf Zimmermann

22. Januar 2007

Zusammenfassung

Diese Seminararbeit gibt eine Übersicht über den Umgang mit Elliptischen Kurven in der heutigen Kryptographie. Es wird dabei auf die mathematischen Grundlagen eingegangen, auf die Frage, wie Elliptische Kurven erzeugt und Punkte der Kurve gefunden werden, wie man mit diesen Punkten rechnen und dies in kryptographischen Verfahren (Schlüsselerzeugung, Signaturverfahren, Verschlüsselungsverfahren) einsetzen kann.

*Technische Universität Braunschweig, Wintersemster 2006/2007

Inhaltsverzeichnis

Abbildung 1: Vergleich benötigter Bitlänge für gleiche Seicherheit

RSA modulus n	1024	2048	3072	8192	15360
EC parameter n	160	224	256	384	512

1 Einleitung

Dieses Seminar basiert auf dem Kapitel 4 „Cryptographic Protocols“ des Buchs [3]. Es wurde auf Grund der Komplexität des Themas und den Voraussetzungen an die anderen Seminarteilnehmer nicht das gesamte Kapitel behandelt sondern versucht, eine Einführung in die Kryptographie mit Elliptischen Kurven zu geben.

1.1 Warum Elliptische Kurven

Elliptische Kurven wurden bereits seit Mitte des 19. Jahrhunderts von Mathematikern untersucht, jedoch lange Zeit nicht mit Kryptographie in Verbindung gebracht. Sie spielten u.a. eine entscheidende Rolle beim Beweis der Farmat'schen Vermutung und sind ansonsten in der Funktionentheorie, der algebraischen Geometrie sowie der Zahlentheorie wichtig. Als 1976 das Prinzip der Public-Key Kryptographie durch Whitfield Diffie und Martin Hellman entdeckt wurde, basierte der erste Public-Key Algorithmus 1977 - der RSA Algorithmus - auf der Schwierigkeit der Primfaktorisierung großer Zahlen.

Im Jahr 1984 wurde von Hendrik Lenstra der subexponentielle Algorithmus ECM[1] entwickelt, der Elliptische Kurven zur Faktorisierung benutzt. Daraufhin wurde der Nutzen von Elliptischen Kurven für die Kryptographie weiter untersucht. Neal Koblitz und Victor Miller entdeckten 1985 das Elliptische-Kurven-Kryptosystem[2] (ECC). ECC ist ein vollständiges Public-Key Kryptosystem, dessen Sicherheit jedoch auf dem Diskreten Logarithmusproblem der Elliptischen Kurve[3] (ECDLP, siehe Kap. 5) basiert.

Der dadurch entstehende Vorteil des Elliptische-Kurven-Kryptosystems ist die wesentlich kürzere Schlüsselgröße zum Erreichen äquivalenter Sicherheit zu anderen Systemen basierend auf dem Diskreten Logarithmus Problem (DLP) oder dem Faktorisierungsproblem. Das Faktorisierungsproblem lässt sich subexponentiell mit der *Number Field Sieve* (NFS) in $L_n[\frac{1}{3}, 1.923]$[4], das DLP entweder mit NFS in $L_p[\frac{1}{3}, 1.923]$ oder mit dem Pollard's rho Algorithmus in $O(\sqrt{\frac{\pi q}{2}})$ lösen, je nachdem welcher Aufwand höher ist. Für das ECDLP sind keine subexponentiellen Algorithmen bekannt, der bestmögliche Algorithmus, der Pollard's rho Algorithmus, löst das Problem in $O(\sqrt{\frac{\pi n}{2}})$.

Die Tabelle in Abb. 1 soll einen Vergleich der benötigten Bitlänge zwischen dem Modulus n des RSA Verfahrens und der Ordnung n einer Elliptischen Kurve bei gleicher Sicherheit des Systems bieten.

2 Mathematische Grundlagen

Um Elliptische Kurven zu betrachten, ist es wichtig, einige mathematische Grundlagen zu wiederholen. Diese sollten sowohl aus den Vorlesungen *(Lineare) Algebra* und *Kryptographie I* bekannt sein und werden dementsprechend kurz gehalten.

2.1 Gruppe und Körper

Für die Betrachtung von Elliptischen Kurven spielen Gruppen eine große Rolle. Für die Verfahren muss eine neue Addition definiert werden, die auf Punkten der Kurve arbeitet. Daher ist es wichtig, sich zu erinnern, was die Gruppenaxiome sind:

[1] elliptic curve factorization method, siehe [4]
[2] elliptic curve cryptography
[3] elliptic curve discrete logarithm problem
[4] für Eingabelänge $\log n$ gilt $L_n[\alpha, c] = O(e^{(c+o(1))(\log n)^{\alpha}(\log\log n)^{1-\alpha}}$

Definition 2.1 *Eine Gruppe $(G, *)$ besteht aus einer Menge G und einer binären Operation $* : G \times G \to G$ mit folgenden Eigenschaften:*

1. *(Assoziativität)* $a * (b * c) = (a * b) * c \quad \forall\, a, b, c \in G$
2. *(Existenz des Neutralen Elements)* $\exists e \in G$: $a * e = e * a = a \quad \forall\, a \in G$
3. *(Existenz des Inversen Elements)* $\quad \forall\, a \in G\ \exists b \in G$: $a * b = b * a = e$

Definition 2.2 *Eine Gruppe $(G, *)$ heißt abelsch, genau dann wenn $(G, *)$ die Gruppenaxiome erfüllt und $*$ ist zusätzlich kommutativ ist:*

$$a * b = b * a \quad \forall\, a, b \in G$$

Nach der Definition der Gruppe wird nun der Körperbegriff betrachtet. Ein Körper ist eine Abstraktion einer bekannten Grundmenge $\mathbb{F}$, z.B. $\mathbb{F} = \mathbb{R}$ und dessen Verbindung mit zwei verschiedenen Operationen.

Definition 2.3 *Sei $\mathbb{F}$ eine Grundmenge und $+$ eine additive, $\cdot$ eine multiplikative Operation. Dann heißt $\mathbb{F}$ Körper, wenn gilt:*

1. $(\mathbb{F}, +)$ *abelsche Gruppe, neutrales Element* 0
2. $(\mathbb{F} \setminus \{0\}, \cdot)$ *abelsche Gruppe, neutrales Element* 1
3. *Distributivgesetz:* $(a + b) \cdot c = a \cdot c + b \cdot c \quad \forall\, a, b, c \in \mathbb{F}$

Definition 2.4 *Ein Körper $\mathbb{F}$ heißt genau dann endlicher Körper, wenn gilt:* $|\mathbb{F}| < \infty$

Im folgenden werden nur noch endliche Körper betrachtet. Diese für die reine Mathematik weniger interessante Klasse von Körpern ist für die Kryptographie sehr wichtig. Für den Umgang mit diesen sind einige Begriffe notwendig, die nun definiert werden.

Definition 2.5 *Sei $\mathbb{F}$ ein Körper mit $q = |\mathbb{F}|$ Elementen. Dann gilt:*

1. *q heißt Ordnung von $\mathbb{F}$*
2. *$\mathbb{F}$ ist Körper der Ordnung $q :\Leftrightarrow q = p^m$ mit p prim, $m \geq 1$*
3. *p heißt die Charakteristik des Körpers*
4. *$\mathbb{F}$ heißt Primkörper für $m = 1$*
5. *$\mathbb{F}$ heißt Erweiterugskörper für $m \geq 2$*
6. *$\mathbb{F}$ heißt Binärkörper für $q = 2^m$*

Bemerkung:

- Alle Körper $\mathbb{F}$ der gleichen Ordnung q sind isomorph, man schreibt daher $\mathbb{F}_q$.
- Jeder endliche Körper besitzt mindestens ein erzeugendes Element $g \in \mathbb{F}_q$, welches die Untergruppe $\mathbb{F}_q^* = \{g^i \mid 0 \leq i \leq q - 2\}$ erzeugen.

2.2 Legendre und Jacobi-Symbol

Für die Bestimmung von Punkten der Kurve sind das Legendre bzw. das Jacobi-Symbol wichtig. Dafür wird zuerst der Begriff des quadratischen Rest modulo p definiert:

Definition 2.6 (Quadratischer Rest) *Sei $a \in \mathbb{N}$ mit ggT(a, n) = 1. a heißt quadratischer Rest modulo n: $\Leftrightarrow \exists x \in \mathbb{N} : x^2$ mod $n = a$. Andernfalls heißt a quadratischer Nichtrest modulo n.*

Zur Berechnung der quadratischen Wurzel modulo p gibt es verschiedene Algorithmen[5], die den Rahmen diese Arbeit sprengen würden. Wichtig ist jedoch, effizient bestimmen zu können, ob eine Zahl $m \in \mathbb{Z}$ modulo p ein quadratischer Rest oder Nichtrest ist.

[5] siehe [5] S. 96

Definition 2.7 (Legendre-Symbol) *Sei p prim und $m \in \mathbb{Z}$. Dann heißt*

$$\left(\frac{m}{p}\right) = \begin{cases} 0 & \text{falls } m \equiv 0 \text{ mod } p \\ 1 & \text{falls } m \text{ quadratischer Rest modulo } p \\ -1 & \text{falls } m \text{ quadratischer Nichtrest modulo } p \end{cases}$$

Legendre-Symbol von m und p.

Das Jacobi-Symbol ist eine Verallgemeinerung des Legendre-Symbols und beschränkt sich nicht auf eine Primzahl p:

Definition 2.8 (Jacobi-Symbol) *Sei $m, n \in \mathbb{Z}$, $n \geq 1$, $n \equiv 1$ mod 2 und Primzerlegung $n = \prod_i p_i^{e_i}$. Es gilt:*

$$\left(\frac{m}{n}\right) = \prod_i \left(\frac{m}{p_i}\right)^{e_i}$$

Dabei ist $\frac{m}{p_i}$ das Legendre-Symbol. Es gilt weiterhin

$$\left(\frac{m}{n}\right) = \left(\frac{m \bmod n}{n}\right)$$

sowie die Multiplikativität

$$\left(\frac{m_1 m_2}{n}\right) = \left(\frac{m_1}{n}\right)\left(\frac{m_2}{n}\right).$$

Mit weiteren Formeln für die Berechnung von $\left(\frac{-1}{n}\right)$, $\left(\frac{2}{n}\right)$ sowie dem quadratischen Reziprozitätsgesetz lässt sich das Legendre-Symbol schnell berechnen.

3 Elliptische Kurven

In diesem Kapitel wird zunächst die allgemeine Form der Elliptische Kurve betrachtet. Anschließend wird für die weitere Anwendung der zu Grunde liegende Körper auf Primkörper beschränkt und über den Punkten dieser Kurven eine abelsche Gruppe gebildet. Diese ist wichtige Grundlage für die kryptographischen Protokolle, die in Kapitel 6 vorgestellt werden.

3.1 Allgemeine Elliptische Kurven

Definition 3.1 *Eine Elliptische Kurve E über einem endlichen Körper K ist definiert durch die Weierstrass-Gleichung*

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{3.1}$$

mit $a_1, a_2, a_3, a_4, a_6 \in K$ und der Diskriminante $\Delta \neq 0$.

Wie aus der Algebra bekannt, lassen sich durch Betrachtung der Diskriminante einer algebraischen Gleichung Aussagen über die Zahl und Art der Lösungen treffen. Für Elliptische Kurven ist die Bedingung $\Delta \neq 0$ eine wichtige Voraussetzung. Das bedeutet, dass die Kurve nicht singulär ist, d.h. die partiellen Ableitungen des von x und y abhängigen Polynoms sind nicht Null. Das hat zur Folge, dass keine Knoten, Spitzen oder abgetrennte Punkte existieren können.

Definition 3.2 *Die Diskriminante Δ einer Elliptischen Kurve E ist definiert durch*

$$\begin{aligned} \Delta &= -d_2^2 d_8 - 8d_4^3 - 27d_6^2 + 9d_2d_4d_6 \\ d_2 &= a_1^2 + 4a_2 \\ d_4 &= 2a_4 + a_1a_3 \\ d_6 &= a_3^2 + 4a_6 \\ d_8 &= a_1^2 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2 \end{aligned} \tag{3.2}$$

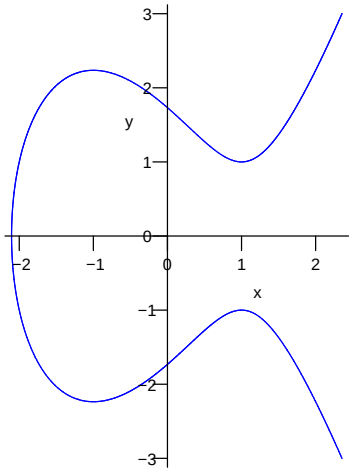

Abbildung 2: Graph der Kurve $y^2 = x^3 - 3x + 3$ in $\mathbb{R}$

Beispiel: Die Gleichung $y^2 = x^3 - 3x + 3$ erfüllt $\Delta \neq 0$. Abb. 2 auf S. 6 zeigt einen Ausschnitt der Kurve.

Um eine Elliptische Kurve nicht nur als Konstrukt zu betrachten sondern damit arbeiten zu können, muss man zunächst die Menge der Punkte definieren, die zu der Kurve gehören:

Definition 3.3 *Ist L eine Körpererweiterung zu K, dann ist die Menge der L-rationalen Punkte von E:*

$$E(L) = \{(x, y) \in L \times L \mid x, y \text{ erfüllen } (3.1)\} \cup \{ \mathcal{O} \} \tag{3.3}$$

mit $\mathcal{O}$als Punkt im Unendlichen.

Definition 3.4 *$|E(K)|$ heißt Ordnung einer Elliptischen Kurve über K.*

Diese Definition ist bis auf den Punkt im Unendlichen nicht weiter überraschend. Die Bedeutung des Punktes erschließt sich erst bei der Konstruktion der Additions-Operation.

Zuvor wird jedoch betrachtet, wieviele Elemente diese Kurve enthält. Der naive Ansatz ist, für alle Elemente $x \in K$ zu überprüfen, ob ein oder mehrere $y \in K$ exisiteren, die die Gleichung (3.1) lösen. Es ist zwar möglich, mit weitaus weniger Aufwand festzustellen, ob zu einem gegebenen Element $x \in K$ genau 0, 1 oder 2 Lösungen der Kurvengleichung existieren (siehe Kapitel 4), dies ist für große Körper trotzdem nicht effizient. Eine Abschätzung der ungefähren Anzahl ergibt sich durch das Theorem von Hasse:

Theorem 3.5 (Hasse) *Sei E eine Elliptische Kurve über $\mathbb{F}_q$. Dann gilt*

$$q + 1 - 2\sqrt{q} \leq |E(\mathbb{F}_q)| \leq q + 1 + 2\sqrt{q} \tag{3.4}$$

Demnach ist die Ordnung $|E(\mathbb{F}_q)| \approx q$. Die exakte Bestimmung ist jedoch für die kryptographischen Verfahren notwendig. Dazu hat Schoof 1985 einen polynomialen Algorithmus[6] zur exakten Berechnung angegeben. Es wird zuerst $|E(K)| \bmod l$ für viele kleine Primzahlen l berechnet und anschließend die exakte Anzahl mit dem chinesischen Restsatz bestimmt. Eine Weiterentwicklung ist der SEA (Schoof-Elkies-Atkin) Algorithmus, der derzeit beste Algorithmus zur Berechnung der Ordnung von Elliptische Kurven über Primkörpern. Für Elliptische Kurven über Binärkörpern wurde 1999 von Satoh eine neue Methode eingeführt, die in ihren Varianten SST und AGM sehr effizient ist.

[6] siehe [6] S. 105f

3.2 Elliptische Kurven über Primkörpern

Im folgenden werden nur noch Elliptische Kurven über Primkörpern, also $K = \mathbb{F}_p$ mit p prim betrachtet. Dadurch lassen sich die Gleichungen (3.1) und (3.2) transformieren. Man erhält dadurch isomorphe Kurven nach folgender Definition:

Definition 3.6 *Eine Elliptische Kurve E über einem Primkörper $\mathbb{F}_p$ ist definiert durch die Weierstrass-Gleichung*

$$E : y^2 = x^3 + ax^2 + b \tag{3.5}$$

mit $a, b \in \mathbb{F}_p$ und der Diskriminante $\Delta := -16(4a^3 + 27b^2) \neq 0$.

Betrachtet man die Diskriminante lediglich zum Ausschluss mehrfacher Nullstellen, genügt also die Sicherstellung von $4a^3 + 27b^2 \neq 0$. Wie bereits erwähnt ist das Ziel die Konstruktion einer Operation, so dass mit Punkten der Elliptischen Kurve gerechnet werden kann. Im folgenden wird eine Addition $+$ in $E(F_p)$ zweier Punkte P, Q eingeführt. Insbesondere heißt die Summe $P + Q$ Verdoppelung von P, falls $P = Q$ ist.

Bemerkung: $\forall s \in \mathbb{Z} : sP = P + P + \cdots + P$ (s mal).

Die Geometrische Konstruktion dieser Addition erfolgt in drei Schritten. Dazu betrachtet man beide Fälle, $P \neq Q$ (Abb. 3) und $P = Q$ (Abb. 4 auf S. 8) getrennt.

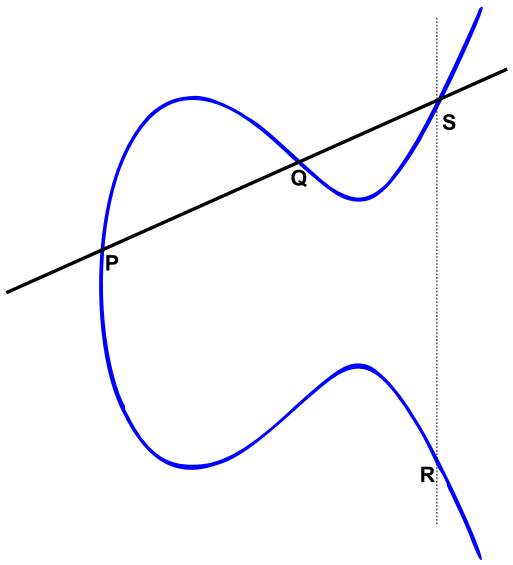

Abbildung 3: Konstruktion von $R = P + Q$ für $P \neq Q$

1. Definiere Gerade: G $= \begin{cases} \text{Gerade durch } P \text{ und } Q & \text{falls } P \neq Q \\ \text{Tangente durch } P & \text{falls } P = Q \end{cases}$
2. weiterer Schnittpunkt $S \in G \cap E(\mathbb{F}_p), S \notin \{P, Q\}$
3. R ist Spiegelung von S an der x-Achse

Wie man leicht sieht gibt es Fälle, in denen es keinen neuen Schrittpunkt der Geraden mit der Kurve gibt. In diesem Fall wird $S = R = \mathcal{O}$ gesetzt.

Nach der geometrischen Konstruktion muss nun die Addition algebraisch definiert werden. Dafür wird zuerst $R = P + Q$ für den Fall $P = (x_1, y_1)$, $P \neq Q = (x_2, y_2)$, $P, Q \in E(\mathbb{F}_p)$ betrachtet.

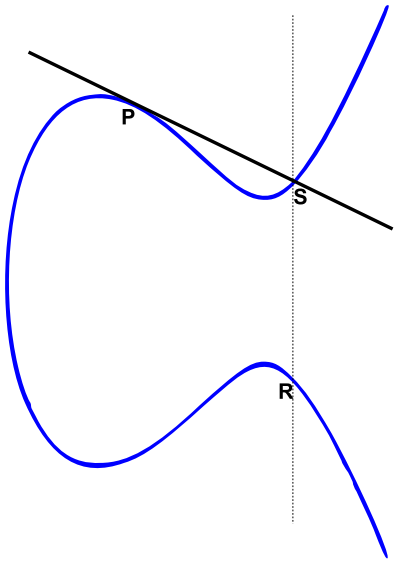

Abbildung 4: Konstruktion von $R = 2P$

Desweiteren folgt aus $x_1 = x_2 \Rightarrow y_2 = y_1$ sofort, dass $R = \mathcal{O}$ gilt. Für $x_1 \neq x_2$ folgt die Geradengleichung sowie die Steigung der Geraden:

$$\begin{aligned} m &= \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x1} && (3.6) \\ y &= m(x - x_1) + y_1 \\ &= mx + n \text{ mit } (n := y_1 - mx_1) && (3.7) \end{aligned}$$

Formt man die Gleichung (3.5) der Elliptischen Kurve um, erhält man direkt

$$0 = x^3 + ax + b - y^2 \tag{3.8}$$

Einsetzen der Geradengleichung (3.7) führt zu

$$\begin{aligned} 0 &= x^3 + ax + b - (mx + n)^2 \\ &= x^3 - m^2x^2 + (a - 2mn)x + (b - n^2) && (3.9) \end{aligned}$$

Die Gleichung (3.9) lässt sich faktorisieren, wobei die zwei der drei Nullstellen bereits bekannt sind ($x_1 und x_2$:

$$0 = (x - x_1)(x - x_2)(x - x_3) \tag{3.10}$$

Durch Koeffizientenvergleich erhält man für die Summe $R = P + Q$ schließlich

$$\begin{aligned} x_3 &= m^2 - x_1 - x_2 && (3.11) \\ y_3 &= m(x_3 - x_1) + y_1. && (3.12) \end{aligned}$$

Nun fehlt der zweite Fall, in dem $P = Q$ ist. Es wird eine Tangente durch P konstruiert (Abb. 4). Es gilt: $y_1 = 0 \Rightarrow G \cap E(\mathbb{F}_p) = \emptyset \Rightarrow R = \mathcal{O}$. Betrachte daher $y_1 \neq 0$. Die Tangentensteigung ist offensichtlich

$$m = \frac{3x_1^2 + a}{2y_1} \tag{3.13}$$

Setzt man diese Steigung in (3.9) ein und betrachtet (3.10), jedoch mit x_1 als doppelter Nullstelle. Durch Koeffizientenvergleich erhält man ebenfalls (3.11) und (3.12). Daher lässt sich nun die Additionsoperation wie folgt definieren:

Definition 3.7 (Addition von Punkten) *Sei $P = (x_1, y_1) \neq Q = (x_2, y_2)$, $P, Q \in E(\mathbb{F}_p)$. Dann ist die Operation $+$ eindeutig festgelegt durch*

- $P + \mathcal{O} = \mathcal{O} + P = P \quad \forall\, P \in E(\mathbb{F}_p)$
- $(-P) := (x_1, -y_1) \Rightarrow P + (-P) = \mathcal{O}$
- *Für $Q \neq -P$ existiert $R \in E(F_p)$, $R = (x_3, y_3) = P + Q$ mit:*

$$m = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & x_1 \neq x_2 \\ \frac{3x_1^2 + a}{2y_1} & x_1 = x_2 \end{cases}$$

$$x_3 = m^2 - x_1 - x_2 \quad , \quad y_3 = -m(x_3 - x_1) - y_1$$

Bemerkung: Es ist sofort ersichtlich, dass $(E(\mathbb{F}_p), +)$ eine Gruppe bildet. Durch Nachrechnen lässt sich zeigen, dass es eine abelsche Gruppe ist.

Definition 3.8 *Sei $P \in E(\mathbb{F}_p)$. Dann $\exists n \in \mathbb{Z} : nP = \mathcal{O}$. n heißt die Ordnung des Punkts P.*

Bemerkung: Der Punkt P erzeugt die Untergruppe $\langle P \rangle$ der Ordnung n.

4 Finden von Punkten der Elliptischen Kurve

Bislang wurde sich noch nicht damit beschäftigt, wie man einen Punkt der Kurve erhält. Für die Konstruktion einer Kurve zur Verwendung als kryptographisches System benötigt man jedoch einen Basispunkt. Daher werden in diesem kurzen Kapitel Überlegungen zur Bestimmung von Punkten der Kurve gemacht.

4.1 Berechnung eines Punkts

Ein Punkt der Elliptischen Kurve muss die Gleichung (3.5) erfüllen. Das ist gleichbedeutend damit, dass $x^3 + ax + b$ für ein festes $x = x_0 \in \mathbb{F}_p$ ein quadratischer Rest in $\mathbb{F}_p$ ist. Wenn das gegeben ist, lassen sich mit Algorithmen[7] die Quadratwurzeln y_i effizient bestimmen und man erhält so die Lösungen (x_0, y_i). Um zu überprüfen, ob es Lösungen gibt und wie viele, betrachtet man das Legendre-Symbol $L_{x_0} := \left(\frac{x_0^3 + ax_0 + b}{p}\right)$. Man sieht sofort aus Def. 2.7, dass die Anzahl der Lösungen $L_{x_0} + 1$ entspricht. Genauer gilt: ist $L_{x_0} = 0$ ist der Punkt $(x_0, 0) \in E(\mathbb{F}_p)$, für $L_{x_0} = 1$ lassen sich die Punkte $(x_0, y), (x_0, -y) \in E(\mathbb{F}_p)$ bestimmen.

4.2 Wahrscheinlichkeitsabschätzung

Auf den ersten Blick erscheint es unwahrscheinlich, dass man bei p Elementen im Körper durch zufälliges Raten ohne Kenntnis weiterer Kurveneigenschaften effizient einzelne Punkte bestimmen kann. Daher wird im Folgenden die Wahrscheinlichkeit des Misserfolgs, sprich dass man keinen Punkt der Kurve erzeugen kann, nach oben hin abgeschätzt. Da nur der Fall $K = \mathbb{F}_p$ betrachtet wird, gilt für das Theorem 3.5 $q = p$.

Der endliche Körper $\mathbb{F}_p$ besitzt genau p Elemente. Die Anzahl der Punkte der Elliptischen Kurve $E(\mathbb{F}_p)$ sei k. Es gibt maximal drei Elemente $x_i \in \mathbb{F}_p$ gibt mit Legendre-Symbol $L_{x_i} = 0$ gibt. Demnach hat man noch höchstens weitere $p - 3$ Elemente im Körper, die $k - 3$ Punkte der Kurve durch die x-Koordinate festlegen.

Es gilt: $(x, y) \in E(\mathbb{F}_p) \Rightarrow (x, -y) \in E(\mathbb{F}_p)$. Daraus folgt, dass die Anzahl der Elemente $x \in \mathbb{F}_p$ mit der Eigenschaft $LS(x) = 1$ genau $\frac{k-3}{2}$ ist. Also gilt für die Wahrscheinlichkeit eines Misserfolgs:

[7] z.B. [6] Algorithmus 9.1, S. 160

$$
\begin{aligned}
\frac{p-3-\frac{k-3}{2}}{p} &\leq \frac{p-3-\frac{(p+1-2\sqrt{p})-3}{2}}{p} && (4.1)\\
&= \frac{2p-6-p-1+2\sqrt{p}+3}{2p}\\
&= \frac{p+2\sqrt{p}-4}{2p}\\
&= \frac{1}{2}+\frac{1}{\sqrt{p}}-\frac{2}{p}\\
&\leq \frac{1}{2}+\frac{1}{\sqrt{p}} && (4.2)
\end{aligned}
$$

Die erste Abschätzung (4.1) nach oben beruht auf dem Theorem von Hasse (3.5) für die untere Schranke. Dadurch wird $-\frac{k-3}{2}$ minimiert, so dass die Wahrscheinlichkeit, einen Misserfolg zu erhalten, maximiert wird. Für $p \to \infty$ nähert sich die obere Schranke eines Misserfolgs bei zufälliger Wahl eines Elements $x \in \mathbb{F}_p$ also von oben gegen $\frac{1}{2}$ an.

5 Erzeugung von Elliptischen Kurven

In diesem Kapitel geht es um die Erzeugung von Elliptischen Kurven unter bestimmten Sicherheitskriterien zur Verwendung als Kryptographisches System. Es wird kurz auf das ECDLP eingegangen, auf dem die Sicherheit der Verfahren basiert sowie Angriffe gegenüber diesem. Anschließend wird definiert, welche Parameter zur Umgebung für das spätere Kryptosystem zählen und wie sie erzeugt werden.

5.1 ECDLP

Die Verfahren der Elliptische-Kurven-Kryptographie basieren auf dem Problem, dass der diskrete Logarithmus in der additiven Gruppe $(E(\mathbb{F}_p), +)$ berechenbar praktisch unmöglich ist.

Definition 5.1 *Sei $P \in E(\mathbb{F}_p)$ ein Punkt der Ordnung n und $Q \in \langle P \rangle$. Die Zahl $s \in \mathbb{Z}_n : Q = sP$ heißt diskreter Logarithmus von Q zur Basis P.*

Bemerkung: Man schreibt $s = \log_P Q$.

Definition 5.2 *Das Diskrete Logarithmus Problem über Elliptischen Kurven (ECDLP) ist die Bestimmung des Diskreten Logarithmus s bei gegebenen Punkten P, Q nach Voraussetzung von Def. 5.1.*

Es wurden verschiedene Angriffe auf das ECDLP entwickelt, aus denen man für die Sicherheit wichtige Informationen über die Kurvenparameter gewonnen hat. Diese Angriffe sind nicht auf alle Kurven anwendbar sondern beziehen sich in der Regel auf kleine Gruppen, die daher bei der Erzeugung ausgeschlossen werden können, ohne eine große Einschränkung darzustellen.

Der bekannte *Pohlig-Hellman Angriff* auf das Diskrete Logarithmus Problem[8] berechnet den Diskreten Logarithmus über ein Kongruenzsystem und betrachtet dazu das Problem in den Untergruppen über den Primfaktoren von $\phi(p)$. Äquivalent dazu reduziert der Pohlig-Hellman Angriff das ECDLP auf Untergruppen mit Primordnung, betrachtet also die Primfaktoren von n und berechnet über ein Kongruenzsystem mit dem Chinesischen Restsatz den Diskreten Logarithmus. Beim DLP ist die effiziente Gegenmaßnahme die Verwendung einer sicheren Primzahl, damit $\phi(p)$ einen großen Primfaktor hat. Diese Maßnahme nutzt man auch für das ECDLP: die Ordnung n von P muss einen großen Primfaktor besitzen, damit die Untergruppen-Bildung keine Reduktion erbringt.

[8] siehe [6] S. 110f

Ein anderer Ansatz - der sogenannte *Pollard's rho Angriff* - ist, unterschiedliche Paare $(c, d), (\tilde{c}, \tilde{d})$ mit $c, \tilde{c}, d, \tilde{d} \in \mathbb{Z}_n$ zu finden, mit

$$cP + dQ = \tilde{c}P + \tilde{d}Q. \tag{5.1}$$

Hat man erfolgreich ein solches Paar gefunden, lässt sich der Diskrete Logarithmus durch

$$\begin{aligned} (c - \tilde{c})P &= (\tilde{d} - d)Q & (5.2)\\ &= (\tilde{d} - d)sP & (5.3)\\ (c - \tilde{c}) &\equiv (\tilde{d} - d)s \bmod n & (5.4) \end{aligned}$$

nach $s = \log_P Q$ auflösen. Eine naive Methode zum Finden solcher Paare ist, zufällig die Zahlen c und d zu wählen und das Triple $(c, d, cP + dQ)$ zu speichern. Sortiert man nach der dritten Komponente, wird nach dem Geburtstagsparadoxon eine Kollision nach ungefähr $I := \sqrt{\frac{\pi n}{2}} \approx 1.2533\sqrt{n}$ Iterationen ergeben. Der Nachteil dieser Methode ist der Bedarf an Speicherplatz für I Triple.

Der Pollard's rho Algorithmus findet zwei Paare in ungefähr der gleichen erwarteten Laufzeit, jedoch mit vernachlässigbarem Speicheraufwand. Die Idee ist, eine Iterationsfunktion[9] $F : \langle P \rangle \to \langle P \rangle$ zu definieren, die effizient bei gegebenem $X \in \langle P \rangle, c, d \in [0, n-1]$ und $X = cP + dQ$ einen neuen Punkt $\tilde{X} = f(X)$ und $\tilde{c}, \tilde{d} \in [0, n-1]$ mit $\tilde{X} = \tilde{c}P + \tilde{d}Q$ berechnet. Da $\langle P \rangle$ endlich ist, wird eine Folge $\{X_i\}_{i \geq 0}$ von Punkten $X_i = f(X_{i-1}), i \geq 1$ an einer Stelle s kollidieren. D.h. $\exists t \in \mathbb{N} : X_s = X_t$ und es entsteht ein Zyklus: $\forall j \in \mathbb{N} : X_{s+j} = X_{t+j}$.

Nachdem ein solcher Zyklus erzeugt wurde, lässt sich eine Kollision von Punkten X_i, X_j mit $X_i = X_j$ und $i \neq j$ mit dem Kreisfindungs-Algorithmus von Floyd finden.

Weiterführendes zu dem Algorithmus, der Parallelisierung und Angriffen auf Kurven anderer Gruppen sowie der Isomorphismus, Index-Kalkulus und der Weil und Tate Paarung Angriff sind in [3] S. 158ff genauer beschrieben und werden an dieser Stelle nicht weiter ausgeführt.

5.2 Sicherheitskriterien

Wichtig ist, aus den Angriffen Kriterien für die Sicherheit abzuleiten. Auf Grund der Algorithmen von Pohlig-Hellman Pollard ist es notwendig, dass die Ordnung n einen großen Primfaktor hat. Außerdem sollte als untere Grenze $n \geq 2^{160}$ gelten. Für eine maximale Sicherheit gegen die beiden Algorithmen sollte man die Kurve E so wählen, dass $|E(\mathbb{F}_p)|$ prim oder bis auf einen sehr kleinen Faktor prim ist: $|E(\mathbb{F}_p)| = hn$ mit n prim und in der Regel $h = 1, 2, 3$, oder 4. Um den weiteren Angriffe entgegenzuwirken, ist es wichtig, dass $|E(F_p)| \neq p$ ist und n kein Teiler von $(p^k - 1) \quad \forall 1 \leq k \leq C$ ist mit C groß genug damit das DLP in $\mathbb{F}^*_{p^C}$ berechnungsmäßig praktisch unmöglich ist. Wählt man $n > 2^{160}$ reicht $C = 20$.

5.3 Domain Parameter

Damit alle Teilnehmer mit exakt der selben Elliptischen Kurve arbeiten, ist es wichtig zu wissen, wie die Kurve eindeutig festgelegt ist. Dies wird durch die sogenannten Domain Parameter erreicht. Sie werden zuerst allgemein für einen Körper $\mathbb{F}_q$ definiert:

Definition 5.3 *Domain Parameter D = (q, FR, S, a, b, P, n, h) sind festgelegt durch:*

1. *Körperordnung q*
2. *Körperrepräsentation FR für Elemente aus* $\mathbb{F}_q$
3. *Seed S einer zufällig generierten Kurve*
4. *Koeffizienten* $a, b \in \mathbb{F}_q$
5. $x_P, y_P \in \mathbb{F}_q$ *mit* $\mathcal{O} \neq P = (x_P, y_P) \in E(\mathbb{F}_q)$. *P ist der Basispunkt und hat Primordnung.*
6. *Ordnung* n *von* P.
7. *Kofaktor* $h = \frac{|E(\mathbb{F}_q)|}{n}$

[9] ein theoretisches Beispiel ist in [3] S. 157-158

Bemerkung: Für Primkörper ist die Körperrepräsentation FR nicht relevant und es gilt $q = p$. FR wäre z.B. für die Repräsentation der Elemente eines Binärkörpers wichtig.

Folgender Algorithmus generiert die Domain-Parameter und damit eine Elliptische Kurve zur kryptographischen Nutzung unter Berücksichtigung der Sicherheitskriterien. Man beachte, dass da weiterhin Primkörper betrachtet werden bereits q durch p ersetzt wurde.

Algorithmus 5.1 (Erzeugung der Domainparameter)
***INPUT**: p, FR, Sicherheitsparameter* $160 \leq L \leq \lfloor \log_2 q \rfloor$ *und* $2^L \geq 4\sqrt{p}$
***OUTPUT**: D = (p, FR, S, a, b, P, n, h).*

1. *Berechne* $a, b \in \mathbb{F}_p$ *überprüfbar zufällig*[10]*. Der Algorithmus liefert zusätzlich* S *zurück.*
2. *Berechne* $N = |E(\mathbb{F}_p)|$
3. *Prüfe* $n|N$ *mit* $n > 2^L$ *prim. (sonst* $\rightarrow$ *1)*
4. *Prüfe* $n \not| (p^k - 1)$ *für* $1 \leq k \leq 20$*. (sonst* $\rightarrow$ *1)*
5. *Prüfe* $n \neq p$*. (sonst* $\rightarrow$ *1)*
6. *Setze* $h = \frac{N}{n}$
7. *Wähle beliebigen Punkt* $\tilde{P} \in E(\mathbb{F}_p)$*. Setze* $P = h\tilde{P}$*. Wiederhole bis* $P \neq \mathcal{O}$
8. $D = (p, FR, S, a, b, P, n, h)$

Bevor mit einer solchen Kurve gearbeitet wird, muss überprüft werden, ob die Parameter wirklich durch diesen Algorithmus erzeugt wurden. Daher gibt es folgenden

Algorithmus 5.2 (Verifikation der Domainparameter)
***INPUT**: Domain Parameter D = (p, FR, S, a, b, P, n, h)*
***OUTPUT**: Akzeptanz oder Ablehnung von D*

1. *Prüfe* p *prim*
2. *Prüfe* $a, b, x_P, y_P \in \mathbb{F}_p$
3. *Prüfe* $\Delta \neq 0$ *für* a, b
4. *Prüfe* $\log_2 S \geq l$ *mit* l *Bitlänge der benutzten Hashfunktion* H
5. *Prüfe* (a, b, S) *auf überprüfbare zufällige Generierung*[11]
6. *Prüfe* $P \in E(\mathbb{F}_p) \setminus \{\mathcal{O}\}$
7. *Prüfe* $n > 2^{160}$ *prim,* $n > 4\sqrt{p}$
8. *Prüfe* $nP = \mathcal{O}$
9. *Berechne* $\tilde{h} = \lfloor (\sqrt{p} + 1)^2 / n \rfloor$*. Prüfe* $\tilde{h} = h$
10. *Prüfe* $n \not| p^k - 1$ *für* $1 \leq k \leq 20$
11. *Prüfe* $n \neq p$

Bei dieser Art der Erzeugung von Domain Parametern mit dem Algorithmus zur Erzeugung überprüfbar zufälliger Kurven wird versucht, Schutz vor noch nicht entdeckten Angriffen zu bieten. Es wird davon ausgegangen, dass diese sich nicht auf allgemeine Elliptische Kurven anwenden lassen sondern sich auf kleine Klassen von Kurven beschränken. Durch Verwendung einer One-Way Hashfunktion (z.B. SHA-1) werden die Koeffizienten so erzeugt, dass möglichst keine mathematisch ausnutzbaren Abhängigkeiten entstehen. Durch den zurückgegebenen Seed S lässt sich dabei dennoch überprüfen, ob die Parameter mit Hilfe des Algorithmus erzeugt wurden.

[10] [3], Algorithmus 4.18, S.176
[11] [3], Algorithmus 4.18, S.176

6 Kryptographische Verfahren

In diesem Kapitel werden Schlüsselpaar-Erzeugung, sowie ein Signatur- und Verschlüsslungsverfahren vorgestellt. Diese geben eine Übersicht über die Funktionalität des Kryptosystems und bieten eine Grundlage für viele weitere Algorithmen.

Grundsätzlich wird in diesem Kapitel davon ausgegangen, dass das ECDLP in der benutzten Untergruppe $\langle P \rangle$ nicht effizient lösbar ist und dass ein Algorithmus sofort abbricht, wenn eine Überprüfung misslingt. Weiterhin bezeichnet H eine kryptographische Hashfunktion mit maximal n bit. Da alle Algorithmen eine bereits erzeugte Elliptische Kurve voraussetzen, haben sie grundsätzlich Zugriff auf die Domain Parameter D.

6.1 Schlüsselpaar-Erzeugung

Wie in jedem Public-Key Kryptosystem muss jeder Teilnehmer über ein gültiges Schlüsselpaar verfügen. Die Erzeugung eines neuen Schlüsselpaares erfolgt durch den

Algorithmus 6.1 (Schlüsselpaar-Erzeugung)
***INPUT**: Domain Parameter D*
***OUTPUT**: Schlüsselpaar (Q, d)*

1. *Wähle zufälliges* $d \in \mathbb{Z}_n^*$
2. *Berechne* $Q = dP$
3. *öffentlicher Schlüssel* Q *der Gruppe* $\langle P \rangle$, *privater Schlüssel* $d = \log_P Q$

Es ist wichtig, überprüfen zu können, ob ein öffentlicher Schlüssel über gewisse arithmetische Eigenschaften verfügt. Dadurch wird sichergestellt, dass ein zugehöriger privater Schlüssel in der Theorie existiert, ohne ihn berechnen zu können. Diese Überprüfung ist essenziell bei Schlüsselaustausch-Protokollen, die auf Diffie-Hellman basieren, bei der ansonsten Informationen über den privaten Schlüssel des Gegenübers erhalten werden können.

Algorithmus 6.2 (Überprüfung des öffentlichen Schlüssels)
***INPUT**: öffentlicher Schlüssel Q*
***OUTPUT**: Akzeptanz oder Ablehnung der Gültigkeit von Q*

1. *Prüfe* $Q \neq \mathcal{O}$
2. *Prüfe* $x_Q, y_Q \in \mathbb{F}_p$
3. *Prüfe* Q *erfüllt Gleichung* $y^2 = x^3 + ax + b$ *mit* a, b *aus D*
4. *Prüfe* $nQ = \mathcal{O}$

Bemerkung: Wenn $h = 1$ ist, folgt aus Schritt 1-3 bereits $nQ = \mathcal{O}$.

6.2 ECDSA Signaturverfahren

Zum Signieren von Nachrichten M gibt es mehrere Algorithmen. In [3] wurden ECDSA (Elliptic Curve Digital Signature Algorithm) und EC-KCDSA (Elliptic Curve Korean Certificate-based Digital Signature Algorithm) erklärt. Beides sind analog zu dem DSA bzw. KCDSA Algorithmus. Im Folgenden wird der ECDSA Algorithmus vorgestellt.

Algorithmus 6.3 (ECDSA Signaturerzeugung)
***INPUT**: privater Schlüssel d, Nachricht M*
***OUTPUT**: Signatur* (r, s) *über* $E(\mathbb{F}_p)$ *für* M

1. *Wähle zufälliges* $k \in \mathbb{Z}_n^*$
2. *Berechne* $A = kP = (x_A, y_A)$
3. *Berechne* $r = x_A \mod n$. *Falls* $r = 0 \rightarrow$ *1*
4. *Berechne* $e = H(M)$

5. *Berechne* $s = k^{-1}(e + dr)$ *mod* n. *Falls* $s = 0 \rightarrow$ *1*

Algorithmus 6.4 (ECDSA Signaturverifikation)
INPUT*: öffentlicher Schlüssel Q, Nachricht M, Signatur (r, s)*
OUTPUT*: Akzeptanz oder Ablehnung der Signatur*

1. *Prüfe* $r, s \in \mathbb{Z}_n^*$
2. *Berechne* $e = H(M)$
3. *Berechne* $w = s^{-1}$ *mod* n
4. *Berechne* $u_1 = ew$ *mod* n, $u_2 = rw$ *mod* n
5. *Berechne* $B = u_1P + u_2Q$, *prüfe* $B \neq \mathcal{O}$
6. *Berechne* $v = x_B$ *mod* n
7. *Prüfe* $v = r$

Sei (r, s) eine gültige Signatur nach Algorithmus 6.3, dann gilt nach Schritt 5:

$$s \equiv k^{-1}(e + dr) \pmod{n}.$$

Daraus folgt

$$k \equiv s^{-1}(e + dr) \equiv s^{-1}e + s^{-1}rd \equiv we + wrd \equiv u_1 + u_2d \pmod{n}. \tag{6.1}$$

Der Verifikation nach Algorithmus 6.4 berechnet in Schritt 5:

$$B = u_1P + u_2Q = (u_1 + u_2d)P = kP = A$$

Daraus folgt, dass $r = x_A = x_B = v$ gilt und die Signatur als gültig akzeptiert wird.

6.3 ECIES Verschlüsselungsverfahren

Überträgt man die bekannten Verschlüsselungsverfahren, z.B. RSA oder ElGamal auf Elliptische Kurven, erwartet man zuerst eine Abbildung der Nachricht M als Punkt der Kurve. Anschließend würden Gruppenoperationen eingesetzt um einen Punkt C, den Ciphertext, zu erhalten. Diese Abbildung ist jedoch nicht trivial, insbesondere, da durch den zu Grunde liegenden Körper $\mathbb{F}_p$ nur diskrete Punkte benutzt werden, nicht alle Punkte einer Kurve wie sie ein Graph über $\mathbb{R}$ oder $\mathbb{C}$ zeigt. Daher wird die Kurve in der Regel nur zur Maskierung verwendet, im Fall der hier vorgestellten Verfahren berechnet man Punkte und nutzt die Koordinaten für weitere Berechnungen außerhalb der Kurve.

Als Vertreter der Verschlüsselungsverfahren wird das Elliptic Curve Integrated Encryption Scheme (ECIES) vorgestellt. Ein weiteres Verfahren ist das Provably Secure Encryption Curve Scheme (PSEC), in Varianten als Kombination von PSEC-KEM[12] und DEM1[13] möglich.

Für diese Verfahren werden weitere Funktionen benutzt: ENC/DEC beschreibt einen schnellen, symmetrischen Ver-/Entschlüsslungsalgorithmus, z.B. AES. Weiterhin beschreibt MAC einen Message Authentication Code Algorithmus, z.B. HMAC. Die interessante Funktion ist die Key Derivation Function KDF. Sie wird über einer Hash-Funktion H definiert und ist die Konkatenation von $H(S, i)$. i ist ein Zähler, so dass so lange neue Hashs erzeugt werden, bis die Konkatenation eine gegebene Bitlänge r erzeugt hat.

Algorithmus 6.5 (ECIES Verschlüsselung)
INPUT*: öffentlicher Schlüssel Q, Nachricht M*
OUTPUT*: Ciphertext (R, C, t) einer Nachricht M*

1. *Wähle zufälliges* $k \in \mathbb{Z}_n^*$
2. *Berechne* $R = kP = (x_R, y_R)$

[12] Key Encapsulation Mechanism
[13] Data Encapsulation Mechanism

3. *Berechne* $Z = hkQ = (x_Z, y_Z)$. *Falls* $Z = \mathcal{O} \rightarrow 1$
4. $(k_1, k_2) \leftarrow KDF(x_z, R)$
5. *Berechne* $C = ENC_{k_1}(M)$ *und* $t = MAC_{k_2}(C)$.

Algorithmus 6.6 (ECIES Entschlüsselung)
***INPUT**: privater Schlüssel d, Ciphertext (R, C, t)*
***OUTPUT**: Nachtricht M oder Ablehnung des Ciphertext*

1. *Prüfe arithmetische Eigenschaften von R als Public Key*
2. *Berechne* $Z = hdR$
3. *Prüfe* $Z \neq \mathcal{O}$
4. $(k_1, k_2) \leftarrow KDF(x_z, R)$
5. *Berechne* $\tilde{t} = MAC_{k_2}(C)$
6. *Prüfe* $\tilde{t} = t$
7. *Berechne* $M = DEC_{k_1}(C)$

Wenn (R, C, t) mit Algorithmus 6.5 erzeugt wurde um M zu verschlüsseln, dann gilt:

$$hdR = hd(kP) = hk(dP) = hkQ \tag{6.2}$$

und es wird in beiden Algorithmen der selbe sogenannte Secret Sharing Point Z bestimmt, also auch das Schlüsselpaar (k_1, k_2) für das symmetrische Verfahren und den MAC. Ungewöhnlich sind die drei Überprüfungen vor der eigentlichen Entschlüsselung. Da R die Form kP hat, muss ein gültiger Punkt die arithmetischen Eigenschaften eines öffentlichen Schlüssels besitzen. In Schritt 3 wird überprüft, dass Z wirklich der Ordnung n ist. Diese beiden Überprüfungen wirken Untergruppen-Angriffen und Angriffen mit ungültigen Kurven entgegen.

7 Schlusswort

Auf Grund der Komplexität des Themas mussten in dieser Arbeit einige interessante Aspekte vernachlässigt werden. Wichtig sind z.B. Algorithmen, um die Berechnung von $k \cdot P$ für $k \in K, P \in E(K)$ effizient durchzuführen (also nicht mit naiver k-facher Addition). Diese Algorithmen können in [3] Kapitel 3.3 nachgelesen werden.

Die Standards for Efficient Cryptography Group (SECG)[14] hat für die Verwendung von Elliptischen Kurven zwei Publikationen (siehe [1] und [2]) herausgebracht. Diese beschreiben sowohl die geforderten Eigenschaften an das Kryptosystem als auch verschiedene Domain Parameter für die Erzeugung von Elliptischen Kurven mit einer Übersicht über die erreichte Sicherheit.

Diese vorgeschlagenen Parameter werden in vielen Bibliotheken verwendet und bieten eine Alternative zu einer vollständigen Generation neuer Kurven. Da das Elliptische-Kurven-Kryptosystem heutzutage in vielen Bereichen neben Embedded Systems eingesetzt wird, gibt es inzwischen vielen Schnittstellen und Bibliotheken wie OpenSSL, MIRACLE, TinyECC und libecc. Dadurch ist die Verwendung in eigenen Programmen ebenso einfach wie die Nutzung von ElGamal oder RSA.

Insgesamt hoffe ich, mit dieser Arbeit eine gute Übersicht über die Möglichkeiten der Kryptographie mit Elliptischen Kurven sowie der dahinterstehenden Mathematik geschaffen und das Interesse für die Verfahren geweckt zu haben.

[14] siehe `http://www.secg.org`

Literatur

[1] Certicom Research, *Standards for Efficient Cryptography (SEC) 1: Elliptic Curve Cryptography*, Version 1.0, 2000, `http://www.secg.org/download/aid-385/sec1_final.pdf`

[2] Certicom Research, *Standards for Efficient Cryptography (SEC) 2: Recommended Elliptic Curve Domain Parameters*, Version 1.0, 2000, `http://www.secg.org/download/aid-385/sec1_final.pdf`

[3] Darrel Hankerson, Alfred Menezes, Scott Vanstone, *Guide to Elliptic Curve Cryptography*, Springer-Verlag New York, Inc, 2004

[4] Hendrik W. Lenstra, Jr., "Factoring integers with elliptic curves", *Annals of Mathematics*, 126:649–673, 1987

[5] Wolfgang M. Ruppert, *Elliptische Kurven und Kryptographie, Sommersemester 1998*, Mathematisches Institut der Universität Bayreuth, `http://gd.tuwien.ac.at/books/skripten/collection-rbenedik/Algebra/Elliptische_Kurven_und_Kryptographie_001_g_123p.ps.gz`

[6] Dietmar Wätjen, *Kryptographie: Grundlagen, Algorithmen, Protokolle*, 1. Auflage, Heidelberg: Spektrum Akad. Verl., 2003