Universität Paderborn
Fakultät für Elektrotechnik, Informatik und Mathematik

Proseminar Public-Key Kryptographie

# Kryptographie auf elliptischen Kurven am Beispiel von ElGamal und Menezes-Vanstone

Frank Nillies
frank@upb.de

Betreuer:
Volker Krummel

Paderborn, Februar 2006

# Inhaltsverzeichnis

**1 Einleitung** **2**
1.1 Historische Einordnung . . . . . 2

**2 ElGamal Kryptosystem** **3**
2.1 ElGamal Verschlüsselungsschema . . . . . 3
2.2 Laufzeit von ElGamal . . . . . 4
2.3 Semantische Sicherheit von ElGamal . . . . . 4
2.4 Angriffe über den diskreten Logarithmus . . . . . 6

**3 ElGamal auf elliptischen Kurven** **7**
3.1 ElGamal Verschlüsselungsschema für elliptische Kurven . . . . 7
3.2 Ein einfaches Beispiel . . . . . 8
3.3 Laufzeit von ElGamal bei elementaren Algorithmen . . . . . 10
3.4 Idee der Codierungsfunktion . . . . . 11

**4 Menezes-Vanstone Kryptosystem** **12**
4.1 Menezes-Vanstone Verschlüsselungsschema . . . . . 12
4.2 Sicherheit von Menezes-Vanstone . . . . . 13

**5 Fazit** **15**

**6 Anhang** **16**
6.1 Geeignete elliptische Kurven . . . . . 16
6.2 Punktkompression . . . . . 17
6.3 Übersicht . . . . . 18

**Literatur** **20**

# 1. Einleitung

Diese Ausarbeit gibt einen Einblick in die *Kryptographie auf ellitpischen Kurven*. Dabei liegt der Schwerpunkt auf ElGamal basierten Verfahren. Die Ausarbeit zum Vortrag *Einführung in elliptische Kurven* von Björn Mühlenfeld wird hierbei als Grundlage für arithmetische Operation auf elliptischen Kurven genommen. Für eine umfangreichere Erläuterung der Mathematik auf elliptischen Kurven sei auf [Wer02] verwiesen.

**1.1. Historische Einordnung.** Die elliptischen Kurven, oder vielmehr die Gleichung der Kurven, entstand aus der Idee, eine Formel zur Berechnung des Umfanges elliptischer Kurven anzugeben.
Elliptische Kurven werden in der Regel durch eine Gleichung gegeben, die man die *Weierstraßsche Normalform* nennt, und die über $\mathbb{Q}$ folgendermaßen aussieht: $y^2 = x^3 + ax + b$ mit rationalen Koeffizienten $a$ und $b$, wobei man noch voraussetzt, dass das Polynom $x^3 + ax + b$ keine mehrfache Nullstelle besitzt. Mit dieser Form der elliptischen Kurven hat sich bereits *Fermat*, später dann *Euler* und *Gauß* beschäftigt. Sie haben unter anderem bewiesen, das die Gleichung $x^3 + y^3 = z^3$ keine nichttriviale Lösung in $\mathbb{Z}$ besitzt. Eine detaillierte historische Betrachtung liefert Franz Lemmermeyer [Lem03] in den beiden Einführungskapiteln seiner alten Homepage.
Die Verwendung elliptischer Kurven in der Kryptografie wurde erstmalig im Jahre 1985 unabhängig von *Neil Koblitz* [Kob84] und *Victor Miller* [Mil85] vorgeschlagen.
Im Laufe der inzwischen vergangenen 20 Jahre wurden, aufbauend auf den Ideen von Koblitz und Miller, viele kryptographische Verfahren auf der Basis elliptischer Kurven zur Erzeugung digitaler Signaturen, zum sicheren Schlüsselaustausch und zur Verschlüsselung entwickelt. Die Verschlüsselungssysteme basierend auf elliptischen Kurven werden oft als *ECCs, Elliptic Curve Cryptosystems*, bezeichnet. Einige Ausgewählte dieser vorgeschlagenen kryptographischen Verfahren haben Eingang in die Standardisierung gefunden [P1303]. Dabei konnten sich insbesondere der *ECDSA* als Signaturverfahren, *ECIES* als Verschlüsselungs- und *EC-DH* als Schlüsselaustauschverfahren durchsetzen. Der *ECIES*, definiert in der [P1303], basiert auf den beiden Verfahren von *ElGamal* und *Menezes-Vanstone* [MV93]. Diese beiden Verfahren werden im Folgenden näher betrachtet.

# 2. ElGamal Kryptosystem

Dieses Kapitel erläutert das *Public Key Verschlüsselungsschema* nach ElGamal. Als Grundlage dient eine beliebige *endliche, zyklische Gruppe* $G$, mit der Operation $\bullet$.

## 2.1. ElGamal Verschlüsselungsschema.

Sei $p$ prim, $(G, \bullet)$ endliche, zyklische Gruppe $G$ mit Ordnung $p$.
$\alpha \in G$ erzeugenes Element, $\beta \in G$

Klartextraum: $\mathcal{P} = G$
Chiffretextraum: $\mathcal{C} = G \times G$
Schlüsselraum: $\alpha, \beta$ öffentlich, $m$ geheim, $0 < m < p - 1$

$$\mathcal{K} = (p, G, m, \alpha, \beta) : \beta = \alpha^m$$

Verschlüsselung:

sei $k$ zufällig gleichverteilt, $0 < k < p - 1$

$$\begin{aligned} \mathcal{C} &= enc(x, k) = (y_1, y_2) \\ y_1 &= \alpha^k \\ y_2 &= x\beta^k \end{aligned}$$

Entschlüsselung:

$$\begin{aligned} \mathcal{P} &= dec(y, m) \\ &= (y_1^m)^{-1} y_2 \\ &= (\alpha^{km})^{-1} x \alpha^{km} \\ &= x \end{aligned}$$

Damit das Verschlüsselungssystem effizient und sicher arbeitet, muss gelten:

- $\forall \beta \; \exists! \; m, 0 < m < p - 1 : \alpha^m = \beta$
- Die Berechnung von $\alpha^m$ ist effizient.
- Die Berechnung von $\log_\alpha \beta$ ist schwer.

**2.2. Laufzeit von ElGamal.** Im Folgenden betrachten wir den Aufwand, sprich die Laufzeit, und die Effizienz des ElGamal Systems. Die Laufzeitanalyse beschränkt sich hierbei auf die arithmetischen Operationen.

Für die Verschlüsselung werden eine Multiplikation und zwei Potenzierungen benötigt. Die Anwendung der Potenzfunktion lässt sich jeweils mit maximal $log(p)$ vielen Multiplikationen abschätzen. So ergibt sich insgesamt ein Aufwand von $\mathcal{O}(log(p)$ Multiplikationen für die Verschlüsselung. Betrachtet man die Länge des Klartextes und die Länge des Chiffretextes, so lässt sich in Etwa eine Verdoppelung der Bitlänge erkennen. Dies liegt daran, dass der Klartext nur aus einem Element der Gruppe $G$ besteht, während der Chiffretext aus einem Element aus $G \times G$ besteht.
Bei der Berechnung der Entschlüsselung kommt zusätzlich zur Potenzierung und Multiplikation noch eine Invertierung. Die Invertierung lässt sich mit Hilfe des *erweiterten euklidischen Algorithmus*, s. [CLRS01], bei einer polynomiellen Laufzeit in $log(p)$ lösen. Somit ergibt sich für die Verschlüsselung eine gesamte Laufzeit von $\mathcal{O}(log^2(p))$.

Die Sicherheit des ElGamal Systems beruht auf dem Lösen des diskreten Logarithmus. Solange der diskrete Logarithmus nicht effizient berechnet werden kann, gilt ElGamal als sicher, genauer gesagt als nicht effizient lösbar. Da sicher immer eine Frage der Definition ist, werden folgend einige Angriffe auf das ElGamal System bei unterschiedlicher Sicherheitsdefinition betrachtet.

**2.3. Semantische Sicherheit von ElGamal.** Hier möchte ich zeigen, dass der allgemeine ElGamal semantisch sicher ist, bzw. welche Voraussetzung hierfür gelten muss.

**Definition Semantische Sicherheit:**
Ein *Orakel* liefere bei Eingabe zweier Nachrichten $x_1$ und $x_2$ die korrekte Verschlüsselung $\mathbb{C} = (y_1, y_2)$ genau einer der beiden Nachrichten, ohne anzugeben, für welche der beiden Nachricht diese Verschlüsselung gilt. Als Orakel sei hier eine Person oder Blackbox gemeint, die in der Lage ist, korrekt zu verschlüsseln. Allerdings wird dem Benutzer, in diesem Fall Eve, nur eine von zwei eingegebenen Nachrichten verschlüsselt zurückgegeben. Die Auswahl der Nachricht erfolgt zufällig gleichverteilt.
Ein System gilt nun als semantisch sicher, falls es für Eve keine Möglichkeit gibt, mit einer Wahrscheinlichkeit $\frac{1}{2} + \epsilon$, mit $0 < \epsilon < \frac{1}{2}$, den entsprechenden Klartext zu erraten, bzw. zu bestimmen.

Man betrachte folgende Beobachtungen:

$$1.)\ \left(\frac{\alpha}{p}\right) = -1\ ,\ \textit{(Legendre Symbol)}$$

2.)$a$ mod 2, $k$ mod 2 können effizient berechnet werden.

$$\left(\frac{\beta}{p}\right) = \left(\frac{\alpha^m}{p}\right) = \left(\frac{\alpha}{p}\right)^m = (-1)^m, \text{ d.h.}$$

$$m = 0 \bmod 2 \Leftrightarrow \left(\frac{\beta}{p}\right) = 1$$

3.) $a \cdot k$ mod 2 ist effizient berechenbar.

Eve wählt nun zwei Nachrichten $x_1$ und $x_2$ so, dass gilt:

$$\left(\frac{x_1}{p}\right) = 1\ ,\ \left(\frac{x_2}{p}\right) = -1$$

Sei $\mathcal{C} = (y_1, y_2)$ die Verschlüsselung der Nachricht $x_c$, $c \in \{1, 2\}$ die das Orakel auf Eves Anfrage zurückliefert. Dann ist:

$$\left(\frac{y_2}{p}\right) = \left(\frac{\beta^k \cdot x_c}{p}\right) = \left(\frac{\beta^k}{p}\right) \cdot \left(\frac{x_c}{p}\right) = \left(\frac{\alpha^{mk}}{p}\right) \cdot \left(\frac{x_c}{p}\right) = (-1)^{mk \bmod 2} \cdot \left(\frac{x_c}{p}\right)$$

Somit gilt:

$$\left(\frac{y_2}{p}\right) = (-1)^{mk \bmod 2} \cdot (1) \Rightarrow c = 1$$

$$\left(\frac{y_2}{p}\right) = (-1)^{mk \bmod 2} \cdot (-1) \Rightarrow c = 2$$

Somit kann Eve genau bestimmen, welche der beiden Nachrichten das Orakel verschlüsselt und zurückgeliefert hat. Die Voraussetzung hierfür ist jedoch, das der *DDH, Decisional Diffie-Hellman* [Bon98] effizient lösbar ist.

**2.4. Angriffe über den diskreten Logarithmus.** Die betrachtete semantische Sicherheit gilt für alle Gruppen als erfüllt, auf denen der DDH nicht effizient lösbar ist. Eine andere Möglichkeit ElGamal zu brechen, ist die Berechnung des *diskreten Logarithmus*. Im Gegensatz zu der semantischen Sicherheit, in der Schwächen des Systems ausgenutzt werden, sucht man hier nach effizienteren Algorithmen, so dass derzeit schwer lösbare Probleme vereinfacht werden. Derzeit geht man davon aus, das sich der diskrete Logarithmus nicht effizient berechnen lässt. Es existieren so genannte generische Algorithmen, die den diskreten Logarithmus in maximal exponentieller Laufzeit in Abhängigkeit der Bitlänge $\log_2(n)$ der Eingabe berechnen können. Diese Algorithmen funktionieren auf jeder beliebigen Gruppe. Hierfür seien *Pollard rho*, *Shanks Baby Step - Giant Step* und *Pohlig-Hellmann* mit einer abgeschätzten Laufzeit von jeweils etwa $O(\sqrt{p})$ genannt.

Für einige Gruppen sind darüber hinaus spezielle *subexponentielle* Algorithmen bekannt, die sich die Struktur der Gruppe zu Nutze machen, und so kürzere Laufzeiten erreichen. Ein Beispiel hierfür ist der *IndexCalculus*. IndexCalculus Algorithmen basieren auf der Tatsache, dass gewisse Gruppen Faktorgruppen von Ringen (oder auch Gruppen) mit Primfaktorisierung und endlich vielen Primelementen beschränkter Größe sind. Oder anders ausgedrückt; der IndexCalculus funktioniert auf allen Gruppen, die sich in faktorisierbare Untergruppen aufteilen lassen.

Für die Kryptography auf ellitpischen Kurven sind keine Algorithmen bekannt, die das *ECDLP*, das Problem des diskreten Logarithmus für elliptische Kurven, durch Ausnutzung der Gruppenstruktur vereinfachen können. Somit können nach derzeitigem Kenntnissstand nur die generischen Algortihmen benutzt werden, was die Wahl der Schlüssellänge bei gleichbleibender Sicherheit erheblich verkürzt. Die Tabelle im Anhang verdeutlicht den Unterschied bei gleichem Sicherheitsaspekt der Schlüssellängen verschiedener Algorithmen. Im folgenden Kapitel möchte nun ich zeigen, wie man das System von ElGamal auf elliptischen Kurven anwenden kann.

# 3. ElGamal auf elliptischen Kurven

In diesem Kaptiel wird das ElGamal Verschlüsselungsschema für elliptische Kurven erläutert. Zum besseren Verständnis folgt ein kleines Beispiel. Abschließend wird die Laufzeit des Algorithmus betrachtet und eine Idee für die *Kodierungsfunktion* gegeben. Die Kodierungsfunktion beschreibt, auf welche Art der Klartext in einen Punkt der elliptischen Kurve codiert wird.

## 3.1. ElGamal Verschlüsselungsschema für elliptische Kurven.

Sei $p, q$ prim, $G = (E(\mathbb{F}_{p^n}), \oplus)$ endliche, zyklische Gruppe mit Ordnung $q$.

Klartextraum: $\mathcal{P} = G$
Chiffretextraum: $\mathcal{C} = G \times G$
Schlüsselraum: $\alpha, \beta \in G$ öffentlich, $m$ geheim, $0 < m < q - 1$

$$\mathcal{K} = (p, E, m, \alpha, \beta) : \beta = m\alpha \mod p$$

Verschlüsselung:

sei $k$ zufällig gleichverteilt, $0 < k < p - 1$
$x$ sei Punkt der Kurve mit $x = cf(N)$ *(Codierungsfunktion)*

$$\begin{aligned} enc(x, k) &= (y_1, y_2) \\ y_1 &= k\alpha \\ y_2 &= x + k\beta = x + km\alpha \end{aligned}$$

Entschlüsselung:

$$\begin{aligned} dec(y_1, y_2, m) &= y_2 - my_1 = y_2 + (-1)(my_1) \\ &= x + k\beta - mk\alpha \\ &= x + mk\alpha - mk\alpha \\ &= x \end{aligned}$$

Invertierung der Codierung $N = cf^{-1}(x)$ liefert den entsprechenden Klartext.

**3.2. Ein einfaches Beispiel.** Das folgende Beispiel soll verdeutlichen, auf welche Art und Weise die Punkte einer elliptischen Kurve bestimmt werden, und wie man mit den gefundenen Punkten Nachrichten verschlüsseln kann. Dabei wird auf eine ausführliche Beschreibung sowohl der skalaren Multiplikation eines Punktes, als auch der Addition zweier Punkte verzichtet. Auf eine detaillierte Beschreibung dieser Operationen wurde bereits im vorangegangenen Vortrag eingegangen.

**Berechnung aller Punkte einer elliptischen Kurve:**
Gegeben sei die elliptische Kurve $E(\mathbb{Z}_{11}) : y^2 = x^3 + 3x + 9$.
Zunächst wird für jedes $x \in \mathbb{Z}_{11}$, $x^3 + 3x + 9$ berechnet. Da nicht für jedes $y^2$ eine *Quadratwurzel* in $\mathbb{Z}_{11}$ existiert, wird zunächst auf *quadratische Reste* geprüft. Ist dieser gleich 0, so lassen sich die beiden Quadratwurzeln $y_1$ und $y_2$ finden. Die letzte Spalte der Tabelle gibt die jeweiligen Punkte der elliptischen Kurve an.

| $x$ | $y^2$ | QR | $y_1; y_2$ | Punkte |
|---|---|---|---|---|
| 0 | 9 | OK | $3; 8$ | $(0,3), (0,8)$ |
| 1 | 2 | - | | |
| 2 | 1 | OK | $1; 10$ | $(2,1), (2,10)$ |
| 3 | 1 | OK | $1; 10$ | $(3,1), (2,10)$ |
| 4 | 8 | - | | |
| 5 | 6 | - | | |
| 6 | 1 | OK | $1; 10$ | $(6,1), (6,10)$ |
| 7 | 10 | - | | |
| 8 | 6 | - | | |
| 9 | 6 | - | | |
| 10 | 5 | OK | $4; 7$ | $(10,4), (10,7)$ |

Nachdem nun alle möglichen Punkte der Kurve berechnet wurden, müssen die Punkte in der Reihenfolge der Kurve berechnet werden. Dazu wählt man ein *erzeugendes Element* $g \in E(\mathbb{Z}_{11})$ und berechnet dann die Elemente $ig, 0 \leq i \leq ord(E(\mathbb{Z}_{11}))$. In diesem Fall ist die Ordnung der elliptischen Kurve zufällig gleich der Ordnung der Gruppe, was nicht immer der Fall ist.

$$\begin{array}{cccc} g = (2,1) & 2g = (0,3) & 3g = (10,7) & 4g = (3,1) \\ 5g = (6,10) & 6g = (6,1) & 7g = (3,10) & 8g = (10,4) \\ 9g = (0,8) & 10g = (2,10) & 11g = \infty & \end{array}$$

Zur Codierung der Nachricht stehen nun 10 Punkte auf der elliptischen Kurve zur Verfügung. Der Punkt im Unendlichen steht praktisch nicht zur Codierung zur Verfügung, da die binäre Repräsentation nicht trivial ist. In diesem Fall wäre der Klartextraum und der Chiffretextraum durch die Anzahl der Punkte der elliptischen Kurve auf 10 Elemente beschränkt.
Nachdem jedes Element des Klartextraumes in jeweils einen Punkt der elliptischen Kurve codiert ist, kann die Nachricht verschlüsselt werden. Auf die Grundlegende Idee der Codierungsfunktionen wird später genauer eingegangen.

**Verschlüsselung:**

$$\mathcal{K} = \left\{p, E(\mathbb{Z}_{11}) : x^3 + 3x + 9, m, \alpha, \beta\right\} = \{11, G, 7, (2,1), (3,10)\}$$

Bob möchte die Nachricht $x = (10,4)$ verschlüsseln. Er berechnet dazu:

$$k\alpha = 3 \cdot (2,1) = (10,7)$$

$$\begin{aligned} x + k\alpha &= (10,4) + 3 \cdot (3,10) \\ &= (10,4) + (2,10) \\ &= (3,10) \end{aligned}$$

Bob übermittelt $C = ((10,7);(3,10))$ an Alice.

**Entschlüsselung:**

Aus diesem Chiffretext berechnet Alice die ursprüngliche Nachricht $\mathcal{P}$:

$$\begin{aligned} \mathcal{P} &= y_2 - my \\ &= (3,10) - (7 \cdot (10,7)) \\ &= (3,10) - (2,10) \\ &= (10,4) \end{aligned}$$

Und erhält somit den ursprünglich von Bob verschlüsselten Punkt $(10,4)$. Durch Umkehrung der Codierungsfunktion kann Alice dann die Nachricht im Klartext lesen.

**3.3. Laufzeit von ElGamal bei elementaren Algorithmen.** Wie wir später sehen werden ist es nicht ganz einfach, effiziente Codierungsfunktionen zu finden. Somit sei die Laufzeit der Codierungsfunktion für die Laufzeitanalyse vernachlässigt. Die Laufzeit wird im folgenden in der Anzahl arithmetischen, bzw. Bit Operationen angegeben. Es ergeben sich folgende Berechnungen auf der elliptischen Kurve:

**Verschlüsseln:**
2× Skalare Multiplikation, 1× Punktaddition

**Entschlüsseln:**
je 1× Skalare Multiplikation, Punktaddition und Invertierung.

Aus den Grundlagen der elliptischen Kurven wissen wir, dass sich die Multiplikation mit dem Skalar $s$ als eine $s$-fache Punktaddition auffassen lässt. Somit ist es für die Laufzeitabschätzung ausreichend, die skalare Multiplikation zu betrachten, da jeweils nur noch eine Punktaddition hinzukommt.
Betrachten wir zunächst die Punktaddition.

**Definition Punktaddition:**
Gegeben seien zwei Punkte $P_1 = (x_1, y_1)$ und $P_2 = (x_2, y_2)$ der elliptischen Kurve mit $P_1 \neq -P_2$.

$$\begin{aligned} P_1 + P_2 &= P_3 = (x_3, y_3) \\ x_3 &= m^2 - x_1 - x_2 \\ y_3 &= m \cdot (x_1 - x_3) - y_1 \\ m &= \begin{cases} \dfrac{3x_1^2 + a}{2y_1}, & \text{falls } x_1 = x_2 \\ \dfrac{y_2 - y_1}{x_2 - x_1}, & \text{sonst} \end{cases} \end{aligned}$$

Für die Berechnung von $m$ muss das Inverse Element zu $2y_1$, bzw. $x_2 - x_1$ berechnet werden. Dies geschieht mit Hilfe des *erweiterten euklidischen Algorithmus*, der Laufzeit $\mathcal{O}(log_2(p))$ hat.
Somit benötigt jede Punktaddition Laufzeit $\mathcal{O}(log_2(p))$. Die Laufzeit der Multiplikation mit dem Skalar $s$ lässt sich mit $\mathcal{O}(log(s \cdot log^2(p)))$ abschätzen. Da $s$ im worstcase die Größe von $n$ hat, lässt sich die resultierende Laufzeit für die Verschlüsselung und die Entschlüsselung mit jeweils $\mathcal{O}(log^3(p))$ abschätzen.

Um die Laufzeit bei der Verschlüsselung zu verbessern, können Werte für

$k\alpha$ und $k\beta$ auf Vorrat berechnet und gespeichert werden. Da die Kryptosysteme basierend auf elliptischen Kurven besonders im Bereich der Chipkarten eingesetzt werden, können die vorausberechneten Werte auf der Karte gespeichert und bei Bedarf abgerufen werden.

**3.4. Idee der Codierungsfunktion.** Bei der Anwendung von ElGamal auf elliptischen Kurven wird jeder Klartext auf einen Punkt der elliptischen Kurve abgebildet. Diese Abbildung geschieht über die *Kodierungsfunktion.* Die Schwierigkeit liegt darin, dass die Punkte nicht explizit gegeben sind, sondern nur durch die Kurvengleichung beschrieben werden.
Idealerweise würde eine Kodierungsfunktion jedem Punkt der Kurve einen Block des Klartextes zuordnen. Beispielsweise den $i$-ten Klartextblock von $p-1$ Klartextblöcken auf das $i$-te Element der Gruppe für ein Verschlüsselungsschema, welches über $\mathbb{Z}_p$ definiert sei.
Für die Punkte einer elliptischen Kurve über $Z_p$ nimmt man an, dass diese zufällig gleichverteilt sind. Es existiert aber nicht zu jedem Element aus $\mathbb{Z}_p$ ein Punkt auf der Kurve. Daher ist die Idee der Kodierungsfunktion, dass man $\mathbb{Z}_p$ in Blöcke der Größe #Klartexte teilt. Setzt man nun voraus, dass die elliptische Kurve möglichst groß ist, d.h. möglichst viele Punkte besitzt, so kann man davon ausgehen, dass für jeden Klartext ein Punkt auf der Kurve gefunden werden kann. Eine Abschätzung der Punkte der elliptischen Kurve lässt sich mit der Hasse-Schranke [Lem03] angeben.
Dazu wird zunächst das erste, Element eines jeden Blockes geprüft, ob dazu ein Punkt auf der Kurve existiert. Wahlweise kann auch ein zufälliges Element gewählt werden, was den Algorithmus effizienter macht. Sollte kein Punkt gefunden werden, wird so lange ein weiterer Punkt gewählt, bis ein Element des Blockes auf der Kurve gefunden wird.
Sollte die Menge der Punkte der elliptischen Kurve nicht ausreichend sein, oder aber in einem der Blöcke kein Punkt gefunden werden, so wird eine andere elliptische Kurve gewählt.

Hier soll nur eine grobe Idee für die Codierungsfunktionen gegeben werden. Es sind verschiedene effiziente Algorithmen bekannt, die sich relativ leicht implementieren lassen. Generell ist zu sagen, dass kein deterministischer Algortihmus bekannt ist. Dies ist leicht ersichtlich, da nicht für jedes Element aus $\mathbb{Z}_p$ ein Punkt auf der Kurve gefunden werden kann. Nähere Informationen zu den Algorithmen waren nicht verfügbar, da die Firma *certicom (http://www.certicom.com)* ein Großteil der Patente inne hält und selbst ein Einsehen der Patente gebührenpflichtig ist.

# 4. Menezes-Vanstone Kryptosystem

Wir haben bei dem ElGamal Verschlüsselungssystem gesehen, dass die Codierung der Nachricht in einen Punkt der elliptischen Kurve einerseits nicht für alle Kurven möglich, und andererseits sehr aufwendig ist. Menezes, Okamoto und Vanstone [MV93] haben ein Verschlüsselungssystem entworfen, bei dem die Nachricht in einen beliebigen Punkt codiert werden kann. Diese Verfahren ist hinsichtlich der Codierung wesentlich effizienter.

Sei $p, q$ prim, $G = (E(\mathbb{Z}_p), \oplus)$ mit Ordnung $q$.

Klartextraum: $\mathcal{P} = \mathbb{Z}_p^* \times \mathbb{Z}_p^*$
Chiffretextraum: $\mathcal{C} = E(\mathbb{Z}_p) \times \mathbb{Z}_p^* \times \mathbb{Z}_p^*$
Schlüsselraum: $\alpha, \beta \in G$ öffentlich, $m$ geheim, $0 < m < q - 1$

$$\mathcal{K} = (p, E, m, \alpha, \beta) : \beta = m\alpha \mod p$$

Verschlüsselung:

Kodiere Nachricht $N$ als Element aus $\mathbb{Z}_p^* \times \mathbb{Z}_p^* : N = (x_1, x_2)$
Mit $k$ zufällig, $0 < k < p - 1$, berechne: $(c_y, c_2) = k\beta, c_1, c_2 \neq 0$

$$\begin{aligned} \gamma &= k\alpha \\ b_1 &= c_1 \cdot x_1 \mod p \\ b_2 &= c_2 \cdot x_2 \mod p \\ \mathcal{C} &= enc(N) = (\gamma, b_1, b_2) \end{aligned}$$

Entschlüsselung:

$$\begin{aligned} m\gamma &= (c_1, c_2) = mk\alpha = k\beta \\ dec(\mathcal{C}) &= N = (b_1 \cdot c_1^{-1} \mod p, b_2 \cdot c_2^{-1} \mod p) \end{aligned}$$

**4.1. Menezes-Vanstone Verschlüsselungsschema.** Der wesentliche Unterschied zum ElGamal Schema liegt darin, dass die Koordinaten des zu verschlüsselnden Punktes durch zwei Nachrichten bestimmt werden. In dem Menezes Vanstone Schema wird je eine Nachricht für die Kodierung der $x$ und der $y$ Koordinate verwendet. Zur Erinnerung, bei ElGamal bestimmt die Kurvengleichung die $y$ Koordinate eindeutig, während die Nachricht ausschließlich in die $x$ Koordinate kodiert wird. Ein nützlicher Nebeneffekt dieses

Verfahrens ist, das sich der Chiffretext nicht so stark vergrößert wie bei ElGamal.

Das Eingangs erwähnte ECIES Verschlüsselungsverfahren wählt die $x$ Koordinate analog zum Menezes Vanstone, wählt die $y$ Koordinate aber als einen Punkt der Kurve, ähnlich dem ElGamal Schema. Wie sich hier leicht erkennen lässt, funktioniert das ECIES nicht für jede beliebige Kurve, da nicht alle elliptischen Kurven jeden Punkt aus $\mathbb{Z}_p$ in $E(\mathbb{Z}_p)$ widerspiegeln. Die Verschlüsselung des Punktes erfolgt dann nach dem Menezes Vanstone Verfahren. Bei der Sicherheitsanalyse wird deutlich, warum es nicht immer sicher ist, zwei Nachrichten pro Chiffrierung zu benutzen.

**4.2. Sicherheit von Menezes-Vanstone.** Relativ einfach lässt sich zeigen, dass man das Menezes-Vanstone System einerseits mit einem bekannten Nachrichtenteil $x_i$, und andererseits bei Nutzung eines konstanten $k$s angreifen kann.

**Angriff mit bekanntem Nachrichtenteil:**

Aus dem oben angegebenen Kryptosystem ist Eve $b_1, b_2, \gamma$ und eines der beiden $x_i$, mit $i \in \{1, 2\}$ bekannt. Gehen wir davon aus, dass Eve den Nachrichtenteil $x_1$ kennt. Dann kann sie:

$$\begin{aligned} b_1 &= c_1 \cdot x_1 \mod p \\ c1 &= b_1 \cdot x_1^{-1} \end{aligned}$$

berechnen. Mit $c_1$, oder $c_2$ falls Eve $x_2$ bekannt war, kann sie den Punkt auf der elliptischen Kurve bestimmen. Sie setzt dazu entweder $c_1$ in die Kurvengleichung ein und erhält maximal zwei Punkte, oder sie kann anhand der zweiten Koordinate den Punkt auf maximal 3 Alternativen eingrenzen. Somit ist Eve der Punkt $(c_1, c_2)$ bis auf wenige Alternativen bekannt. Damit kann sie nun die zweite Nachricht $x_2 = b_2 \cdot c_2^{-1}$ bestimmen.

**Angriff bei konstantem $k$:**

Eve ist bekannt, dass Bob bei zwei Nachrichten $b$ und $b'$ das gleiche $k$ verwendet hat. Daraus resultiert, dass zweimal derselbe Punkt der elliptischen Kurve $k \cdot \beta = (c_1, c_2)$ benutzt wurde. Mit diesem Wissen kann Eve folgende

Gleichungen aufstellen und lösen:

$$\begin{aligned} b_1 &= c_1 \cdot x_1 \\ b_2 &= c_2 \cdot x_2 \\ b'_1 &= c_1 \cdot x'_1 \\ b'_2 &= c_2 \cdot x'_2 \end{aligned}$$

Daraus folgt:

$$\begin{aligned} c_1 &= b_1 \cdot x_1^{-1} = b'_1 \cdot x_1^{'-1} \\ \Leftrightarrow c_1 &= b_1 \cdot b_1^{'-1} = x'_1 \cdot x_1^{-1} \end{aligned}$$

Somit kann Eve den verwendeten Punkt der elliptischen Kurve ermitteln und alle Klartexte berechnen, bei denen sich das $k$ nicht ändert. Da für $k$ gilt: $0 < k < p - 1$ sieht man auch recht schnell, warum das $p$ für dieses Kryptosystem nicht vernachlässigbar klein werden darf. Andernfalls könnte Eve, wenn sie genügend Nachrichten abfängt, alle Nachrichten auf gleiches $k$ überprüfen. Sollte sie zwei Nachrichten finden, bei denen das gleiche $k$ benutzt wurde, könnte sie diese entschlüsseln.

Darüber hinaus hat sich Klaus Kiefer, s. [Kie97], mit der Eindeutigkeit von Chiffretexte beschäftigt. Hierbeit handelt es sich weniger um einen direkten Angriff auf das Kryptosystem, sondern mehr um ein analytisches Verfahren. Eve erzeugt sich eine beliebigee, zufällige Bitfolge. Diese Bitfolge gibt sie als Chiffretext an ein Orakel weiter, das entscheiden kann, ob dieser Chiffretext gültig ist, das heißt das es zu diesem Chiffretext einen gültigen Klartext gibt, oder ob die Bitfolge ungültig ist. Kiefer ist der Frage nachgegangen, ob es Eve möglich ist, immer gültige Chiffretexte zu erzeugen, bzw. ob sie die Wahrscheinlichkeit für gültige Chiffretexte zu ihren Gunsten verbessern kann. Der Verweis auf seinen Artikel soll hier weniger seine Idee im Detail beschreiben, sondern einfach nur verdeutlichen, dass zwar die Sicherheit des Menezes-Vanstone System analysiert wird, derzeit aber noch keine effizienten Algorithmen zum brechen dieses Systems bekannt sind.

# 5. Fazit

Dieses Paper ist die schriftliche Ausarbeit zum Proseminarvortrag *Kryptographie auf elliptischen Kurven*. Basierend auf dem vorherigen Vortrag *Einführung in elliptische Kurven* sollte es einen Einblick in die Funktionsweise von Kryptosystemen vermitteln, die auf elliptischen Kurven basieren. Dazu wurde das ElGamal System beispielhaft erläutert.

Geht man der Frage nach, warum man elliptische Kurven statt der traditionellen Gruppen verwendet, so wurde recht schnell klar, das Kryptosysteme auf elliptischen Kurven derzeit zwei Vorteile bieten. Sie benötigen im Vergleich zu anderen Gruppen eine kürzere Schlüssellänge und sind nach derzeitigem Kenntnisstand nur mit generischen Algorithmen angreifbar.

Auf Grund der kürzeren Schlüssellängen werden die ECCs besonders für den Einsatz mit SmartCards verwendet. Wie bei dem Menezes-Vanstone System zu sehen war, können die Laufzeit intensiven Berechnungen der Schlüssel bereits auf Vorrat im Vorfeld geschehen. Somit ergibt sich für die reine Verschlüsselung eine sehr effiziente Laufzeit. Der einzige Nachteil liegt in der vergrößerten Länge der verschlüsselten Nachricht. Je nach Effizienz der Implementierung besteht der Klartextes aus zweifach bis vierfach mehr Bits, als die ursprüngliche Nachricht.

Betrachtet man hingegen die Sicherheit, so ist der erhöhte Datentransfer durchaus akzeptabel. Derzeit sind keine effizienten Angriffe auf ECCs bekannt. Trotz dotierter Wettbewerbe und immer leistungsstärkerer Hardware sind bisher nur Verschlüsselungen mit relativ kleiner Schlüssellänge und diese in inakzeptablen Zeiträumen gebrochen worden.
Ursache hierfür ist die Komplexität und Struktur der Gruppe. Während sich für einige Kryptosysteme spezielle Algorithmen zum Brechen nutzen lassen, können derzeit auf elliptischen Kurven keine effizienten Algorithmen angewandt werden.

Ein Problem bei der Verwendung elliptischer Kurven basierender Krptosysteme wurde bei der Recherche jedoch deutlich. Fast jede effiziente Implementierung ist patentrechtlich Geschützt. Dadurch dürfte die Verwendung besonders durch kleinerer Unternehmen und Forschungseinrichtungen ausgeschlossen werden. Dennoch überwiegen derzeit die Vorteile der ECCs eindeutig, so dass eine Weiterentwicklung gesichert sein dürfte.

# 6. Anhang

In diesem Kapitel sind einige Algorithmen und Tabellen aufgeführt, die thematisch den vorherigen Kapiteln nicht eindeutig zugeordnet werden konnten. Zuerst möchte ich auf allgemeine Informationen zur Wahl geeigneter elliptischer Kurven eingehen.
Der *Punktkompressionsalgorithmus* kann das Datenaufkommen für die Verschlüsselung verringern. Hier wird die Grundidee wiedergegeben. Genauere Informationen, besonders im Hinblick auf die Implementierung unterliegen einem Patent des certicom.com Unternehmens. Abschließend möchte ich noch einige Tabellen anfügen, anhand derer sich die Laufzeiten der unterschiedlichen Verschlüsselungssysteme vergleichen lassen.

**6.1. Geeignete elliptische Kurven.** Es existieren elliptische Kurve mit verschiedenen Eigenschaften, je nachdem, wie die Parameter gewählt werden. Hier wurden nur elliptische Kurven der *Charakteristik* ungleich 2 und 3 betrachtet [Wer02].
Für die hier betrachteten kryptographischen Systeme gültige elliptische Kurven haben die Form:

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

Theoretisch können die Parameter $a$ und $b$ für jede Gruppe $\mathbb{Z}_p$ beliebig gewählt werden. Der Körper $\mathbb{Z}_p$ kann ebenfalls über beliebigem $p$ gewählt werden.
Für alle hier betrachteten Kryptosysteme sei $p$ prim. Die Parameter $a$ und $b$ können zwar beliebig zufällig gewählt werden, müssen aber folgenden Eigenschaften genügen:

$$4a^3 + 27b^2 \neq 0.$$

Diese Eigenschaft sichert, dass in jedem Punkt der elliptischen Kurve die Tangente eindeutig definiert ist. Dies ist für die Punktaddition wichtig, damit der Punkt eindeutig bestimmt werden kann.
Weiterhin sollte die elliptische Kurve möglichst viele Punkte besitzen, damit sich eine Codierungsfunktion finden lässt. Um nicht jede Kurve auf Korrektheit und die Anzahl der Punkte prüfen zu lassen, kann man sich entweder geeignete Kurven generieren lassen, s. *http://www.kurvenfabrik.de/*, oder man wählt von der *NIST*, bzw. *IEEE* angegebene Kurven. Hierzu sei gesagt, dass sich Verschlüsselungssysteme nur mit Kenntniss der Kurve nicht brechen lassen. Es ist also ratsam, von offizieller Stelle geprüfte Kurven zu nutzen.

Um einen kleinen Eindruck zu bekommen, wie verwendete elliptische Kurven aussehen, hier eine von der *NIST* in *RECOMMENDED ELLIPTIC CURVES FOR FEDERAL GOVERNMENT USE, July 1999* vorgeschlagene Kurve:

**Elliptische Kurve P-192:**

$$
\begin{aligned}
p &= 6277101735386680763835789423207666416083908700390324961279 \\
r &= 6277101735386680763835789423176059013767194773182842284081 \\
s &= 3045ae6fc8422f64ed579528d38120eae12196d5 \\
c &= 3099d2bbbfcb2538542dcd5fb078b6ef5f3d6fe2c745de65 \\
b &= 64210519e59c80e70fa7e9ab72243049feb8deecc146b9b1 \\
G_x &= 188da80eb03090f67cbf20eb43a18800f4ff0afd82ff1012 \\
G_y &= 07192b95ffc8da78631011ed6b24cdd573f977a11e794811
\end{aligned}
$$

Mit folgenden Parametern:
$p$: primer Modulus, Ordnung $r$ der Kurve, $s$ und $c$ werden als Parameter für den *SHA-1* bei Unterschriftenverfahren verwendet, Koeffeizient $b$ mit $b^2 \cdot c \equiv -27 \pmod{p}$ und den $(x, y)$-Koordinaten für einen erzeugenden Punkt $G$ der Kurve. Der Parameter $a$ der Kurve sei fest mit $a = -3$ gewählt. Wie sich leicht nachvollziehen lässt gilt, dass $4a^3 + 27b \neq 0$ ist und die Kurve somit singulär ist.

**6.2. Punktkompression.** *Punktkompression* beschreibt eine Möglichkeit, wie man das Datenvolumen für die Darstellung der Punkte einer elliptischen Kurve verkleinern kann. Betrachten wir die Punkte einer elliptischen Kurve als Zahlentupel $(x, y)$ aus $\mathbb{Z}_p \times \mathbb{Z}_p$. Dann existieren zu jedem $x$ genau zwei Punkte auf der Kurve: $(x, y)$ und $(x, -y)$. Da nach Voraussetzung $p$ prim und $p > 2$ gilt, ist es insbesondere auch ungerade. Somit gilt für jedes $x$ entweder $y$ oder aber $-y$ gerade. Ausserdem lassen sich die beiden $y$-Koordinaten eindeutig für jedes $x$ mit Hilfe der Kurvengleichung bestimmen. Rechnet man nun die $y$ Koordinate modulo 2, so lässt sie sich mit nur einem Bit darstellen. Auf diese Art und Weise kann der Speicheraufwand für Punkte der elliptischen Kurve nahezu halbiert werden, ohne Informationen zu verlieren.

**6.3. Übersicht.** Die beiden unten angeführten Tabellen sollen einen kleinen Einblick in die Effizienz EC basierender Kryptosystemen geben.
Quelle: Handbuch der Chipkarten [RE99]

| Jahr | Schlüssellänge symmetrischer Verfahren | Asymmetrische Schlüssellänge (z.B. RSA) | Schlüssellängen von ECC | Erforderliche MIPS-Jahre | Erforderliche Jahre auf 450 Mhz PC |
|---|---|---|---|---|---|
| 2000 | 70 | 952 | 132 | $7.13 * 10^{9}$ | $1.58 * 10^{7}$ |
| 2002 | 72 | 1028 | 137 | $2.06 * 10^{10}$ | $4.59 * 10^{7}$ |
| 2004 | 73 | 1108 | 141 | $5.98 * 10^{10}$ | $1.33 * 10^{8}$ |
| 2006 | 75 | 1191 | 145 | $1.73 * 10^{11}$ | $3.84 * 10^{8}$ |
| 2008 | 76 | 1279 | 149 | $5.01 * 10^{11}$ | $1.11 * 10^{9}$ |
| 2010 | 78 | 1369 | 153 | $1.45 * 10^{12}$ | $3.22 * 10^{9}$ |
| 2012 | 80 | 1464 | 157 | $4.19 * 10^{12}$ | $9.32 * 10^{9}$ |
| 2014 | 81 | 1562 | 162 | $1.21 * 10^{13}$ | $2.70 * 10^{10}$ |
| 2016 | 83 | 1664 | 166 | $3.51 * 10^{13}$ | $7.81 * 10^{10}$ |
| 2018 | 84 | 1771 | 170 | $1.02 * 10^{14}$ | $2.26 * 10^{11}$ |
| 2020 | 86 | 1881 | 175 | $2.94 * 10^{14}$ | $6.54 * 10^{11}$ |

Die Tabelle vergleicht die Schlüssellängen der derzeit verwendeten Kryptosysteme. Dabei werden die ECCs mit symmetrischen und asymmetrischen Kryptosystemen verglichen. Angegeben wird jeweils die Schlüssellänge, die bei den derzeit bekannten Verfahren zum brechen des Systems vergleichbare Laufzeiten benötigt. Die Laufzeiten werden sowohl in *MIPS*,Million Instructions Per Second, als auch in PC Jahren angegeben. Das hierbei nur ein 450 MHz PC als Referenzgerät benutzt wird liegt daran, dass die ECCs in der Regel für SmartCards eingesetzt werden, deren Rechenleistung derzeit bei maximal 25 MHz liegt. Für zukünftige Systeme ist die ständige Leistungssteigerung der Computer nach dem Moorschen Gesetz einbezogen worden. Hierbei wird jedoch nicht berücksichtigt, dass effizientere Methoden gefunden werden können, die das Brechen dieser Systeme vereinfachen und effizienter gestalten können.

| Jahr | Zahl | Bit | Durch | Methode | MIPS-Jahre | Operationen |
|---|---|---|---|---|---|---|
| 1997 | ECCp-79 | 79 | INRIA (Harley) | Birthday Par. | | $1{,}4 \cdot 10^{12}$ |
| 1997 | ECC2-79 | 79 | INRIA (Harley) | Birthday Par. | | $1{,}7 \cdot 10^{12}$ |
| 1998 | ECCp-89 | 89 | INRIA (Harley), BT | Birthday Par. | | $3{,}0 \cdot 10^{13}$ |
| 1998 | ECC2-89 | 89 | INRIA (Harley) | Birthday Par. | | $1{,}8 \cdot 10^{13}$ |
| 1998 | ECCp-97 | 97 | BT, INRIA (Harley) | Pollard Rho | | $2{,}0 \cdot 10^{14}$ |
| 1998 | ECC2k-95 | 95 | INRIA (Harley) | Birthday Par. | | $2{,}2 \cdot 10^{13}$ |
| 1999 | ECC2-97 | 97 | INRIA (Harley) | Birthday Par. | $16 \cdot 10^{3}$ | |
| 2001 | ECC2k-108 | 108 | INRIA (Harley) | Pollard Rho | | $2{,}8 \cdot 10^{15}$ |

Quelle: http://www.certicom.com
Die Tabelle zeigt die bisher gebrochenen Schlüssellängen der Certicom Challenge. Das in Kanada ansässige Unternehmen *Certicom* hält einen Großteil der Patente für effiziente Implementierungen für Algorithmen basierend auf elliptischen Kurven. Unter anderem das patent für den bereits erwähnten *Point Compression Algorithm.* Mitte der 90er Jahre hat das Unternehmen einen dotierten Wettkampf ausgeschrieben, in dem jeder versuchen kann, elliptische Kurven, bzw. Kryptosysteme auf diesen, zu brechen. Die Ergebnisse der Challange werden in regelmäßigen Abständen von dem Unternehmen aktualisiert. Bereits jetzt ist die angegebene Tabelle nicht mehr aktuell. Die letzten gebrochenen Herausforderungen sind *ECCp-109*, gelöst im November 2002, und *ECC2-109*, gelöst im April 2004.

## Literatur


[BK06] Balasubramaniam, P. und E. Karthikeyan: *Implementation Issues in Elliptic Curves based Cryptosystem.* Transaction of Cryptology, Vol. 3:21–30, 2006.

[Bon98] Boneh, Dan: *The Decision Diffie-Hellman Problem.* Lecture Notes in Computer Science, 1423:48–63, 1998.

[BuDF02] Bertsch, Andreas und Frank Bourseau und Dirk Fox: *Perspektive kryptographischer Verfahren auf elliptischen Kurven.* Datenschutz und Datensicherheit, Vol. 26:90–96, 2002.

[Cer03] Certicom: *Guide to Elliptic Curve Cryptography.* Code and Cipher, Vol. 1, 2003.

[CLRS01] Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest und Clifford Stein: *Introduction to Algorithms.* MIT Press, 2001.

[Ehl02] Ehli, Oliver: *ElGamal-Signaturen mit elliptischen Kurven.* Technischer Bericht, Fachbereich Informatik der Technischen Universität Darmstadt, 2002.

[Hen03] Henhapl, Birgit: *Zur Effizienz von Elliptischen Kurven Kryptographie, Dissertation.* Technischer Bericht, Fachbereich Informatik der Technischen Universität Darmstadt, 2003.

[Kie97] Kiefer, Klaus: *A Weakness of the Menezes-Vanstone Cryptosystem.* Security Protocols Workshop, Seiten 201–206, 1997.

[Kir05] Kirchhoff, Ralf: *ECC - Kryptographie auf Basis elliptischer Kurven.* Technischer Bericht, CV cryptovision GmbH, 2005.

[Kob84] Koblitz, Neal: *Introduction to Elliptic Curves and Modular Forms.* Graduate Texts in Mathematics, No. 97, 1984.

[Kob94] Koblitz, Neal: *ACourse in Number Theory and Cryptography*, Band Vol. 1. Springer, 1994.

[Lem03] Lemmermeyer, Franz: *Elliptische Kurven.* Technischer Bericht, Universität Heidelberg, http://www.rzuser.uni-heidelberg.de/ hb3/ellc.html, 2003.

[Mil85] Miller, Victor: *Use of Ellitic Curves in Cryptology.* Proceedings of Crypto, Vol. 39:417–426, 1985.

[MOV96] Menezes, Alfred J., Paul C. Van Oorschot und Scott A. Vanstone: *Handbook of Applied Cryptography.* CRC Press, 1996.

[MV93] Menezes, Alfred J. und Scott A. Vanstone: *Elliptic Curve Cryptosystems and Their Implementation.* Journal of Cryptography, 1993.

[P1303] P1363a, IEEE: *Draft Standard Specifications for Public Key Cryptography.* Vers. 2, 2003.

[Pie00] Pietiläinen, Henna: *Elliptic curve cryptography on smart cards.* Technischer Bericht, Helsinky University of Technologie, 2000.

[RE99] Rankl, Wolfgang und Wolfgang Effing: *Handbuch der Chipkarten.* Hanser, 1999.

[Rei01] Reinhardt, Dr. Kurt: *Vorlesung Sicherheitsprotokolle in Rechnernetzen.* Technischer Bericht, Fachhochschule Mannheim, 2001.

[Sti02] Stinson, Douglas R.: *Cryptography Theory and Practice, 2nd edition.* Chapman & Hall /CRC, 2002.

[Wer02] Werner, Anette: *Elliptische Kurven in der Kryptographie.* Springer Verlag, 2002.

[Wät04] Wätjen, Prof. Dr. Dietmar: *Kryptographie.* Spektrum Akademischer Verlag, 2004.