DuD Datenschutz und Datensicherheit

Perspektive kryptografischer Verfahren auf elliptischen Kurven

Andreas Bertsch, Frank Bourseau, Dirk Fox

Vor dem Hintergrund der großen Verbreitung des RSA-Verfahrens und der damit einher gehenden Abhängigkeit kryptografischer Lösungen von der Sicherheit dieses Verfahrens wird kryptografischen Verfahren auf der Basis elliptischer Kurven zunehmend Bedeutung beigemessen. Der vorliegende Beitrag diskutiert die Vor- und Nachteile dieser alternativen Verfahren im Vergleich mit RSA und bewertet deren aktuelle Perspektiven[1].

Dr. Andreas Bertsch

SIZ – Informatikzentrum der Sparkassenorganisation GmbH
Arbeitsschwerpunkt: Trust und Authentication Services
E-Mail: andreas.bertsch@siz.de

Dr. Frank Bourseau

dvg Hannover GmbH
Arbeitsschwerpunkt: Strategie IT-Sicherheit
E-Mail: frank.bourseau@siz.de

Dipl.-Inform. Dirk Fox

Security Consultant und Geschäftsführer der Secorvo Security Consulting GmbH.
Arbeitsschwerpunkt: Public Key Infrastrukturen, Digitale Signaturen, Sicherheit in Netzen.
E-Mail: fox@secorvo.de

1 Kryptoverfahren auf elliptischen Kurven

Die Verwendung elliptischer Kurven in der Kryptografie wurde erstmalig im Jahre 1985 unabhängig von Neil Koblitz [Kobl_84] und Victor Miller [Mill_85] vorgeschlagen.[2] Im Laufe der inzwischen vergangenen 15 Jahre wurden, aufbauend auf den Ideen von Koblitz und Miller, viele kryptografische Verfahren auf der Basis elliptischer Kurven zur Erzeugung digitaler Signaturen, zum sicheren Schlüsselaustausch und zur Verschlüsselung entwickelt.

Einige ausgewählte der vorgeschlagenen kryptografischen Verfahren auf der Basis elliptischer Kurven haben Eingang in die Standardisierung gefunden. Dabei konnten sich insbesondere der ECDSA als Signaturverfahren, ECIES als Verschlüsselungs- und EC-DH als Schlüsselaustauschverfahren durchsetzen. Bei der Spezifikation dieser Algorithmen in internationalen Standards wurden neueste Erkenntnisse aus der Zahlentheorie und der Kryptografie berücksichtigt, die vor allem die Wahl der Sicherheitsparameter betreffen.

1.1 Elliptic Curve DSA (ECDSA)

Unter den kryptografischen Verfahren auf der Grundlage elliptischer Kurven spielt die Variante des DSA, kurz ECDSA eine zentrale Rolle. Dieses Verfahren wurde Anfang der 90er Jahre für die Standardisierung vorgeschlagen. Im Januar 1999 wurde der ECDSA von ANSI als amerikanischer Banking Standard X9.62-1999 [ANSI_99] angenommen. Ein Jahr später, im Februar 2000, veröffentlichte das amerikanische NIST eine überarbeitete Version des Digital Signature Standard (DSS), der nun ebenfalls das ECDSA-Verfahren einschließt [NIST_00]. Dort wird auf den ANSI-Standard X9.62 verwiesen.

Funktionsweise

Die *Systemparameter* des ECDSA-Verfahrens setzen sich zusammen aus einer geeignet gewählten elliptischen Kurve E über einem endlichen Körper GF(q) mit q prim oder $q = 2^m$ sowie einem Basispunkt (Generator) $G = (x_G, y_G) \in E(GF(q))$ und $G \neq 0$ mit primer Ordnung $n = \mathrm{Ord}(G)$ mit

$$n > 2^{160} \text{ und } n > \sqrt[4]{q}$$

sowie dem Co-Faktor $h = \#E(GF(q))/n$, d.h. $\#E(GF(q))$ ist ein ganzzahliges Vielfaches der Ordnung des Basispunkts G).

Zu einer Nachricht m kann bezüglich gegebener Systemparameter nun mit einem geheimzuhaltenden Signierschlüssel d und einer zufällig oder pseudo-zufällig gewählten Zahl k (mit $0 < k < n$) die Signatur (r, s) wie folgt bestimmt werden:

$$r = x_1 \bmod n$$
$$(\text{mit } (x_1, y_1) = k \cdot G \text{ und } r \neq 0)$$
$$s = k^{-1} (\text{SHA-1}(m) + d \cdot r)$$
$$(\text{mit } s \neq 0)$$

Dabei ist d eine zufällig (oder pseudozufällig) gewählte Zahl mit $0 < d < n$.

Der zugehörige öffentliche Schlüssel ist der Kurvenpunkt Q mit $Q = d \cdot G$. Mit Q kann die digitale Signatur wie folgt geprüft werden:

$$x_1 = r \,?$$
$$\text{mit: } (x_1, x_2) = u_1 \cdot G + u_2 \cdot Q \text{ und}$$
$$u_1 = \text{SHA-1}(m) \cdot s^{-1} \bmod n,\ u_2 = r \cdot s^{-1} \bmod n$$

Dabei müssen r, s im Intervall [1..n-1] liegen und $(x_1, x_2) \neq \underline{0}$.

Sicherheit

Durch die Wahl der Systemparameter ist der ECDSA nicht anfällig für spezielle Algorithmen zur Lösung des ECDLP bzw. andere bekannte Angriffsmethoden (siehe z.B. [JoMV_99]):

[1] Der Beitrag basiert auf den Ergebnissen einer für das Informatikzentrum der Sparkassenorganisation (SIZ) durchgeführten Studie.

[2] Zu einer Einführung in elliptische Kurven in der Kryptografie siehe Paulus/Müller, DuD 9/1998, S. 496-499 [PaMü_98].

- Der Pohlig-Hellman-Algorithmus [PoHe_78] nutzt die Primfaktorzerlegung der Ordnung n von G zur schnelleren Lösung eines ECDLP. Da n als hinreichend große Primzahl ($n > 2^{160}$) gewählt wird, ist der ECDSA von Angriffen mit diesem Algorithmus nicht betroffen.
- Der ANSI-Standard X9.62-1999 schreibt die Wahl einer elliptischen Kurve mit einem Reduktionskoeffizienten von $r > 19$ vor [ANSI_99]. Damit ist der ECDSA nicht anfällig für effiziente Reduktionsalgorithmen nach [MeOV_93].

Diese Eigenschaften der Systemparameter lassen sich nicht nur von einem Schlüsseleinhaber prüfen, sondern sie können jederzeit von jedermann überprüft werden, denn alle für eine Prüfung erforderlichen Werte sind öffentliche Systemparameter.

Auch ein ECDSA-Public Key Q kann sehr leicht überprüft werden. Er muss den folgenden arithmetischen Anforderungen genügen:

- der Punkt Q muss auf der elliptischen Kurve liegen (Prüfung: Koordinaten in Kurvengleichung einsetzen),
- für Q muss gelten: $Q \neq \underline{0}$, und
- Q muss die Ordnung n besitzen, d.h. es muss gelten: $n \cdot Q = \underline{0}$

Bei der Implementierung des ECDSA ist vor allem die Korrektheit einer Reihe wichtiger Parameter-Tests zu beachten, um sicherzustellen, dass kein Angriff unter Ausnutzung von Implementierungsfehlern möglich ist:

- Prüfung, ob r und s in [1, n-1]: Ist diese Bedingung für eine Signatur (r, s) nicht erfüllt, kann ein Angreifer ECDSA-Signaturen fälschen. Sei beispielsweise der Basis-Punkt G einer elliptischen Kurve $y^2 = x^3 + ax + b$ über GF(p): $G = (0, \sqrt{b})$. Dann ist $(0, \text{SHA-1}(m))$ eine gültige ECDSA-Signatur.
- Prüfung, ob $n > 2^{160}$: Nur, wenn diese Bedingung erfüllt ist, ist der ECDSA nicht anfällig für den von Vaudenay 1996 vorgestellten existentiellen Fälschungsangriff auf den DSA [Vaud_96]: Wird das 160-bit-Ergebnis der Hashfunktion SHA-1 nicht modulo n reduziert, sind Kollisionen möglich.
- Prüfung, ob k einmalig: Wie auch beim DSA kann der ECDSA vollständig gebrochen (d.h. der geheime Schlüssel sk bestimmt) werden, wenn der je Signatur zufällig gewählte Wert k zweimal verwendet wird. Daher muss die Implementierung sicherstellen, dass k sowohl unvorhersagbar als auch einmalig ist.

Die Spezifikation des ECDSA legt (wie auch die des DSA) als Hashfunktion den SHA-1 fest [NIST_95]. Damit hängt die Sicherheit des ECDSA zugleich an der Kollisionsresistenz des SHA und ist durch die unveränderliche Länge der Ausgabe des SHA-1 auf Hashwerte von 160 bit Länge festgelegt.

Rechenaufwand

Die Zahl der arithmetischen Basis-Operationen, die für die Berechnung einer ECDSA-Signatur bzw. deren Prüfung durchgeführt werden müssen, sind deutlich geringer als die von RSA- (S: 384/P: 17) oder DSA-Signaturen (S: 240/P: 480). Allerdings liegt der Aufwand einer Basis-Operation bei ECDSA-Signaturen erheblich über denen einer Langzahlarithmetik, denn jede Punkt-Addition auf einer elliptischen Kurve setzt sich wiederum aus vielen Langzahloperationen im darunterliegenden Galois-Feld zusammen [FoRö_96].

ln(n) = 160 bit	Operationen	Pentium (200 MHz)[1]	Luna CA[3] ECC[1]	SLE66CX160S (5 MHz)[1]	SLE66CX320P (15 MHz)
Signieren	ca. 60	5 ms	5,8 ms	260 ms	87 ms
Prüfen	ca. 120	19 ms	9,7 ms	550 ms	183 ms

Tabelle 1: Rechenaufwand der Operationen für ECDSA-160

Tabelle 1 zeigt, dass der Aufwand der ECDSA-Operationen auf einem handelsüblichen PC bei etwa 70% des Aufwands der Operationen des DSA liegen. Auf einem Smartcard-Kryptochip liegt der Aufwand allerdings über dem der entsprechenden DSA-Operationen.

1.2 Elliptic Curve Integrated Encryption Scheme (ECIES)

Als Alternative zur RSA-Verschlüsselung wurde das ECIES spezifiziert. Das Verfahren ist ein hybrides Verschlüsselungsverfahren: Unter Verwendung des ECDH (oder eines ähnlichen EC-basierenden Schlüsselvereinbarungsverfahrens) wird aus dem Public Key des Empfängers ein symmetrischer Schlüssel abgeleitet. Mit diesem wird die zu verschlüsselnde Nachricht unter Verwendung einer Strom- oder Blockchiffre symmetrisch verschlüsselt. Der Nachricht wird ein kryptografischer MAC angefügt. Der Empfänger leitet den Nachrichtenschlüssel aus den ECDH-Parametern (bzw. nach einem ähnlichen Schlüsselvereinbarungsverfahren) ab und entschlüsselt damit die Nachricht.

Die Spezifikation des ECIES lässt dabei mehrere Optionen für die Ableitung des Nachrichtenschlüssels zu. Nicht festgelegt sind auch das Verfahren zur Nachrichtenverschlüsselung und der MAC-Algorithmus. Nur für die Schlüsselvereinbarung kommt ein EC-basierendes Verfahren zum Einsatz. Damit eignet sich das ECIES-Verfahren für einen Einsatz in Store-and-Forward-Protokollen wie z.B. die Verschlüsselung von E-Mail-Nachrichten.

1.3 Elliptic Curve Diffie-Hellman Scheme (ECDH)

Analog zur Diffie-Hellman-Schlüsselvereinbarung in GF(p) wurde ein einfaches Protokoll für eine Schlüsselvereinbarung auf der Basis elliptischer Kurven definiert. Das Protokoll besitzt dieselben Eigenschaften der DH-Schlüsselvereinbarung: Die verwendeten öffentlichen Schlüsselkomponenten müssen separat (z.B. unter Verwendung von Schlüssel-Zertifikaten o.ä.) authentisiert werden.

Die Erzeugung der Systemparameter erfolgt analog dem ECDSA (ausgenommen: Co-Faktor h). Die Ableitung eines gemeinsamen geheimen Schlüssels K durch zwei Teilnehmer A und B mit den Schlüsselpaaren (d_A, Q_A) und (d_B, Q_B) geschieht wie folgt:

A bestimmt $K = d_A \cdot Q_B$. B bestimmt $K = d_B \cdot Q_A$. Damit gilt: $K = d_B \cdot (d_A \cdot G) = d_A \cdot (d_B \cdot G)$. Die x-Koordinate des Kurvenpunktes K bildet den gemeinsamen geheimen Schlüssel.

Eine Implementierung sollte zusätzlich prüfen, ob das Ergebnis K der Punktmultiplikation ungleich dem Punkt $\underline{0}$ ist – dies wäre ein Fehlerfall, da G als Generator der elliptischen Kurve gewählt wurde.

Das Verfahren ist sehr effizient: Es benötigt lediglich eine Punktmultiplikation auf der elliptischen Kurve.

2 Sicherheit

Die Sicherheit der Implementierung der vorgestellten kryptografischen Verfahren auf der Basis elliptischer Kurven setzt sich vor allem aus zwei allgemeinen Aspekten zusammen:

- den zugrundeliegenden komplexitätstheoretischen Sicherheitsannahmen
- der Wahl der Sicherheitsparameter

Hinzu kommt natürlich die Korrektheit der Implementierung des jeweiligen Verfahrens. Auf diesen dritten Aspekt wurde im vorausgegangenen Kapitel kurz eingegangen.

2.1 Sicherheitsannahmen

Die Sicherheit aller kryptografischen Verfahren auf der Basis elliptischer Kurven basieren auf der Schwierigkeit, diskrete Logarithmen in einer elliptischen Kurve über einem endlichen Körper zu berechnen. So kann auf zyklischen Untergruppen einer elliptischen Kurve E analog Galois-Feldern GF(p) ein *Diskretes Logarithmusproblem für elliptische Kurven* (ECDLP) definiert werden:

> Sind $P, Q \in E$ Generatoren einer zyklischen Untergruppe einer elliptischen Kurve E, dann heißt $k \in \mathrm{GF}(n)$ mit $n = \mathrm{Ord}(P)$, prim und $Q = k \cdot P$ *Diskreter Logarithmus* von Q bezüglich P.

Für die Bestimmung Diskreter Logarithmen auf (allgemeinen) elliptischen Kurven gab es in den vergangenen zehn Jahren erhebliche Weiterentwicklungen. Ursprünglich wurde das Baby-Step-Giant-Step-Verfahren von Shanks verwendet [Odly_95], dessen asymptotischer Aufwand bei $o(\sqrt{n} \cdot \log \sqrt{n})$ Kurvenoperationen (Punktadditionen) liegt. Der beste heute bekannte allgemeine Algorithmus zur Lösung des ECDLP ist die Pollard-Rho-Methode [Poll_78]. Mit von Wiener und Zuccherato [WiZu_99] sowie Gallant, Lambert und Vanstone [GaLV_00] vorgeschlagenen Optimierungen hat das Verfahren einen asymptotischen Aufwand von

$$o(\sqrt{n \cdot \pi} / 2)$$

Kurvenoperationen. Damit gehört das ECDLP zur Komplexitätsklasse der Probleme mit exponentiellem Lösungsaufwand.

Zwar lässt sich die DLP-Berechnung durch Parallelisierung linear beschleunigen. Bis heute jedoch sind keine Verfahren mit subexponentiellem Aufwand zur Lösung des ECDLP bekannt. Bekannte Algorithmen die das DLP in GF(q) mit subexponentiellem Aufwand lösen, wie z.B. der Index-Calculus-Algorithmus, sind nicht auf das ECDLP anwendbar [Mill_85].

2.2 Wahl der Sicherheitsparameter

Die asymptotisch größere Komplexität des ECDLP erlaubt es, Digitale Signatursysteme auf elliptischen Kurven (mit genügend großem Reduktionskoeffizienten r, siehe unten) zu realisieren, die bei gleichem Sicherheitslevel mit wesentlich kleineren Schlüssellängen als Digitale Signatursysteme über GF(q) auskommen.

Kurvenordnung

Für die Wahl der Kurvenordnung, genauer der Ordnung n des Basispunktes G, sind im Vergleich mit dem RSA-Verfahren – bei gleichem Sicherheitsniveau, d.h. etwa gleichem erwarteten Aufwand für die Lösung des jeweils zugrundeliegenden Sicherheitsproblems (Faktorisierung vs. ECDLP) – deutlich geringere Schlüssellängen möglich.

RSA	EC	Verhältnis
430 bit	112 bit	26 %
760 bit	160 bit	21 %
1020 bit	192 bit	19 %
1620 bit	256 bit	16 %

Tabelle 2: Kostenbasierende Schätzung vergleichbar sicherer Schlüssellängen [Silv_00]

Silverman gibt auf der Grundlage einer detaillierten Kostenschätzung für die Faktorisierung großer Moduln und die Bestimmung eines ECDLP die in Tabelle 2 wiedergegebenen Vergleichswerte an [Silv_00].

Eine etwas ältere Abschätzung stammt von Lenstra und Verheul [LeVe_99]. Die Autoren kommen in ihrem aufwandsbasierten Modell auf deutlich kleinere Werte für n (siehe Tabelle 3).

RSA	EC	Verhältnis
488 bit	110 bit	22,5 %
777 bit	124 bit	21 %
1028 bit	135 bit	13 %
1613 bit	154 bit	9,5 %

Tabelle 3: Aufwandsbasierende Schätzung vergleichbar sicherer Schlüssellängen [LeVe_99]

Im von der *Standards for Efficient Cryptography Group* (SECG) im Jahr 2000 verabschiedeten SEC-Standard „Elliptic Curve Cryptography“ [SECG_00] wird die etwas „gerundete“ Abschätzung vergleichbarer Schlüssellängen von RSA/DSS zu EC-basierenden Verfahren aus Tabelle 4 gegeben.

RSA	EC	Verhältnis
512 bit	112 bit	22 %
1024 bit	160 bit	15,5 %
2048 bit	224 bit	11 %
3072 bit	256 bit	8,3 %
7680 bit	384 bit	5 %
15360 bit	512 bit	3,3 %

Tabelle 4: Schätzung vergleichbar sicherer Schlüssellängen [SECG_00]

Die Schätzungen zeigen, dass bereits bei heute empfohlenen Schlüssellängen für RSA und DSA die entsprechenden Schlüssellängen für EC-basierenden Verfahren erheblich kürzer ausfallen. Liegt das Verhältnis bei 1024 bit RSA-Modullänge noch bei etwa 1:7, verbessert es sich für 7680 bit RSA-Moduln sogar auf 1:20.

Reduktionskoeffizient

Von Menezes, Okamoto und Vanstone wurde 1993 ein Reduktionsalgorithmus vorgestellt, der das DLP in E(GF(q)) auf ein DLP in einem Erweiterungskörper GF(q^r) reduziert [MeOV_93]. Dabei ist der Reduktionskoeffizient r die kleinste natürliche Zahl, für die ein solcher Erweiterungskörper zu GF(q) existiert. Für kleine r ist der Reduktionsalgorithmus effizient.

In GF(q^r) kann das DLP dann mit dem asymptotisch subexponetiellen Index-Calculus-Verfahren gelöst werden. Damit ist das DLP in E(GF(p)) für $r < \log q$ subexponentiell. Durch geeignete Wahl von q und Ord(P) kann jedoch ein ausreichend großer Wert für den Reduktionskoeffizienten r erzwungen werden. Dazu ist folgende Bedingung für den größten Primteiler p von #E(GF(q)) zu erfüllen:

$$\forall s < r : p \nmid q^s - 1$$

Unsichere Kurven

Menezes, Okamoto und Vanstone zeigten 1993, dass bei den besonders effizient implementierbaren supersingulären Kurven – das sind elliptische Kurven, deren Punktezahl sich in der geschlossenen Formel #E(GF(p)) = p + 1 angeben lässt – für den Reduktionskoeffizienten r gilt: $r < 7$ [MeOV_93]. Supersinguläre Kurven gelten daher inzwischen als unsicher und werden

in den einschlägigen Standards ausgeschlossen.

Weiter sollten bei der Kurvengenerierung alle Kurven über GF(2^m) mit m nicht prim und $m < 160$ bit ausgeschlossen werden. Auch wenn heute kein allgemeiner Angriff auf solche „zusammengesetzten" Kurven bekannt ist, gibt es zumindest Indizien dafür, dass ein solcher systematischer Angriff mit dem Weil Descent-Ansatz möglich ist [GaSm_99]. Jacobson, Menezes und Stein gelang auf diesem Weg kürzlich die effiziente Reduktion des ECDLP auf das DLP in Hyperelliptischen Kurven und damit die Lösung ausgewählter ECDLP auf den elliptischen Kurven GF(2^{62}), GF(2^{93}) und GF(2^{124}) [JaMS_01].

Ein mögliches Risiko liegt auch in der Festlegung auf einige wenige feste Kurven, wie in den aktuellen Standards empfohlen. Dadurch lässt sich zwar die Schlüsselgenerierung erheblich beschleunigen. Sollte sich allerdings eine dieser Kurven als unsicher erweisen, wäre die Sicherheit der auf dieser Kurve basierenden Implementierung gefährdet und möglicherweise auch das Vertrauen in EC-basierende Verfahren erschüttert.

3 Stand Forschung und Normung

Unter den aktuellen Ergebnissen der Forschungsarbeit im Gebiet elliptischer Kurven sind vor allem die Entwicklungen bei der Lösung ausgewählter ECDLP von Bedeutung. Sie geben Hinweise auf die tatsächliche Berechnungskomplexität des ECDLP für die heute empfohlenen Schlüssellängen. Dabei zeigt sich, dass die Ergebnisse auffallend gut mit den asymptotischen Aufwandsschätzungen zusammenfallen.

Vor allem bei der internationalen Standardisierung wurden Verfahren auf der Basis elliptischer Kurven in den vergangenen Jahren in sehr vielen Normungsverfahren aufgegriffen. Die wichtigsten darunter werden in Abschnitt 3.2 zusammengefasst.

3.1 Berechnungskomplexität des ECDLP

Neben einer theoretischen Abschätzung der Sicherheit kryptografischer Verfahren auf der Basis elliptischer Kurven, die mit vereinfachten Aufwandsformeln (O-Kalkül) bestimmt werden, sind Erfahrungswerte mit der Bestimmung Diskreter Logarithmen auf elliptischen Kurven wichtig, um eine genauere Vorstellung von der tatsächlichen Berechnungskomplexität zu gewinnen. Damit lassen sich die mit einem Angriff auf EC-Verfahren verbundenen Aufwände und Kosten deutlich realistischer abschätzen.

Jahr	Zahl	Bit	Durch	Methode	MIPS-Jahre	Operationen
1997	ECCp-79	79	INRIA (Harley)	Birthday Par.		$1{,}4\cdot10^{12}$
1997	ECC2-79	79	INRIA (Harley)	Birthday Par.		$1{,}7\cdot10^{12}$
1998	ECCp-89	89	INRIA (Harley), BT	Birthday Par.		$3{,}0\cdot10^{13}$
1998	ECC2-89	89	INRIA (Harley)	Birthday Par.		$1{,}8\cdot10^{13}$
1998	ECCp-97	97	BT, INRIA (Harley)	Pollard Rho		$2{,}0\cdot10^{14}$
1998	ECC2k-95	95	INRIA (Harley)	Birthday Par.		$2{,}2\cdot10^{13}$
1999	ECC2-97	97	INRIA (Harley)	Birthday Par.	$16\cdot10^3$	
2001	ECC2k-108	108	INRIA (Harley)	Pollard Rho		$2{,}8\cdot10^{15}$

Tabelle 5: Gelöste Aufgaben der Certicom ECC-Challenge (bis Juli 2001)

Zu diesem Zweck schrieb die kanadische Firma Certicom, Corp. im November 1997 eine öffentliche „ECC-Challenge" aus. Diese setzt sich aus drei Aufgabengruppen zusammen, in denen jeweils der Private Key eines ECC aus den öffentlichen Parametern des Public Key zu bestimmen ist:[3]

- *Exercise Level*: Sieben vergleichsweise einfach zu lösende „Übungs"-Aufgaben mit elliptischen Kurven über GF(2^k) bzw. GF(p) mit k resp. ln(p) = 79, 89 und 97.
- *Level 1*: Sechs Aufgaben, die signifikant höhere Ressourcen zur Lösung benötigen, mit elliptischen Kurven über GF(2^k) bzw. GF(p) mit k resp. ln(p) = 109 und 131.
- *Level 2*: Zehn Aufgaben, die nach heutigem Kenntnisstand als unlösbar gelten, mit elliptischen Kurven über GF(2^k) bzw. GF(p) mit k resp. ln(p) = 163, 191, 239 und 359.

[3] Die Aufgaben finden sich im Internet unter http://www.certicom.com/resources/ecc_chall/curves.html

Auf die erfolgreiche Lösung der einzelnen Aufgaben wurden von der Firma Certicom Corp. Preise bis 5.000 USD für den Exercise Level, in Höhe von 10.000-20.000 USD für Level 1 und von 30.000 bis 100.000 USD für Level 2-Aufgaben ausgelobt.
Tabelle 5 gibt eine Übersicht der bislang gelösten Aufgaben dieser ECC-Challenge (Stand: Juli 2001):

Die Ergebnisse der bisher Berechnungen decken sich erstaunlich gut mit den Erwartungen aus der asymptotischen Aufwandsschätzung. So waren zur Lösung der ECCp-97-Challenge $2{,}0\cdot10^{14}$ Operationen erforderlich; die Abschätzung für die optimierte Pollard-Rho-Methode liegt bei $3{,}5\cdot10^{14}$ [ESST_99, JoMV_99].

ANSI-Standard X9.62 gibt die in Tabelle 6 zusammengefaßte Abschätzung für die Berechnungskomplexität des ECDLP [ANSI_99, IEEE_01, Silv_00].

ln(n) [bit]	Operationen[1]	MIPS-Jahre [ANSI_99]	MIPS-Jahre [IEEE_01]	MIPS-Jahre [Silv_00]
128 bit	$0{,}89\cdot2^{64}$		$4{,}0\cdot10^5$	
160 bit	$0{,}89\cdot2^{80}$	$8{,}5\cdot10^{11}$		$9{,}6\cdot10^{11}$
172 bit	$0{,}89\cdot2^{86}$		$3\cdot10^{12}$	
186 bit	$0{,}89\cdot2^{93}$	$7{,}0\cdot10^{15}$		$7{,}9\cdot10^{15}$
234 bit	$0{,}89\cdot2^{117}$	$1{,}2\cdot10^{23}$	$3\cdot10^{21}$	$1{,}6\cdot10^{23}$
314 bit	$0{,}89\cdot2^{157}$		$2\cdot10^{33}$	
354 bit	$0{,}89\cdot2^{177}$	$1{,}3\cdot10^{41}$		$1{,}5\cdot10^{41}$
426 bit	$0{,}89\cdot2^{213}$	$9{,}2\cdot10^{51}$		$1{,}0\cdot10^{51}$

Tabelle 6: Geschätzte Aufwände für die Lösung des ECDLP abhängig von der Modullänge

In der aktuellen Fassung der Empfehlung geeigneter Algorithmen nach dem deutschen Signaturgesetz des Bundesamtes für Sicherheit in der Informationstechnik (BSI) werden die folgenden Empfehlungen für die Wahl der Schlüssellänge bei Varianten des DSA auf elliptischen Kurven gegeben [BSI_01]:

- *Eignung bis Ende 2005*:
  E(GF(p): ln(p) = 192 bit,
  E(GF(2^n)): n = 191 bit, Ord(P) = 160 bit

- *Eignung bis Ende 2006*:
  E(GF(p)): ln(p) = 192 bit,
  E(GF(2^n)): n = 191 bit, Ord(P) = 180 bit.

3.2 Stand der Normung

Kryptografische Verfahren auf der Basis elliptischer Kurven wurden inzwischen in einer größeren Zahl von Standards spezifiziert. Die folgenden internationalen Standards zählen dazu (Tabelle 7).
Eine zentrale Rolle für den ECDSA spielt der im Januar 1999 verabschiedete ANSI-

Jahr	Standard	Verfahren
1998	ANSI X9.62	ECDSA
1998	ISO 14888-3	ECDSA
1999	ISO 15946 Part 1-3	ECDSA, ECDH, ECMQV
2000	FIPS 186-2	ECDSA (Verweis auf ANSI X9.62)
(2000)	ANSI X9.63	ECIES, ECDH, ECMQV
2000	IEEE P1363	ECDSA, ECDH, ECMQV
2000	SEC-1, SEC-2	ECDSA, ECDH, ECMQV
2001	ATM-Forum	ECDSA
(2001)	IEEE P1363a[1]	ECIES

Tabelle 7: Übersicht der internationalen Standards zu elliptischen Kurven

Standard X9.62 – er ist die Referenzspezifikation für die im Februar 2000 publizierte aktualisierte und erweiterte Fassung des NIST-Standards FIPS 186-2 und bildet auch die Grundlage der ECDSA-Spezifikation des IEEE-Standard P1363, verabschiedet im August 2000. Die allgemeine Beschreibung von Signaturalgorithmen im ISO-Standard 14888-3 schließt ebenfalls den ECDSA ein und stimmt mit der Spezifikation des ANSI-Standards überein. Auch die SECG-Spezifikationen SEC-1 und SEC-2 schließen diese ECDSA-Spezifikation ein [SECG_00, SECG2_00].

Für die Verfahren ECIES, ECDH und ECMQV ist der IEEE-Standard P1363 einschlägig [IEEE_00].

4 Bewertung

4.1 Vorzüge

Komplexität des ECDLP
Das der Sicherheit der EC-basierenden kryptografischen Verfahren zugrundeliegende Problem, die Lösung des ECDLP, gilt nach heutigem Erkenntnisstand als ein Problem mit asymptotisch exponentieller Komplexität.

Das bedeutet, dass die Sicherheit dieser Verfahren zu einem wesentlich geringeren Grad abhängig ist von der technischen Entwicklung, d.h. insbesondere der Zunahme der einem potentiellen Angreifer realistischerweise maximal verfügbaren Rechenleistung. Schon eine geringfügige Vergrößerung der Schlüssellänge um wenige Bit sorgt für eine Zunahme des Aufwands eines potentiellen Angreifers um einige Größenordnungen.

Die Mindestanforderungen an die Schlüssellänge EC-basierender Verfahren müssen daher wesentlich seltener an die technischen Entwicklungen angepasst werden. Bei den derzeit empfohlenen Schlüssellängen kann sogar davon ausgegangen werden, dass sie mindestens für die kommenden 20 Jahre genügen – sofern es keine durchschlagend neuen Erkenntnisse bei der Entwicklung von Lösungsalgorithmen für das ECDLP gibt.

Schlüsselgenerierung
Die sicherheitsrelevanten Parameter von EC-basierenden Verfahren, deren Wahl einen vergleichsweise großen initialen Berechnungsaufwand erfordert, sind ausschließlich Systemparameter. Sie werden einmal erzeugt und können anschließend von vielen Anwendungen eingesetzt werden. Außerdem lässt sich die kryptografische Stärke dieser Systemparameter – anders als bei RSA – jederzeit überprüfen, da dafür die öffentlich bekannten Kurvenparameter ausreichen.

Die geheimen und öffentlichen Schlüssel für Nutzer eines EC-basierenden kryptografischen Verfahrens lassen sich hingegen sehr einfach erzeugen: Es genügen die (nichtvorhersagbare, pseudozufällige) Wahl eines geheimen Schlüssels und eine einzige Kurven-Multiplikation zur Bestimmung des zugehörigen öffentlichen Schlüssels.

Damit eignen sich EC-basierende kryptografische Verfahren besonders gut für die Nutzung auf Smartcards oder Komponenten mit eingeschränkter Speicher- und Rechenkapazität: Die Erzeugung des geheimen Schlüssels kann – ohne Auswirkungen auf die Güte des Schlüssels – direkt in der gesicherten Komponente erfolgen.

Signaturlänge
Die Länge einer ECDSA-Signatur liegt mit 320 bit deutlich unter der Länge vergleichbar sicherer Signaturverfahren (RSA: mind. 1024 bit). Das kann in transaktionsbasierenden Verfahren die erforderliche Bandbreite erheblich reduzieren.

4.2 Nachteile

Verlässlichkeit der Sicherheitsannahme
Die der Sicherheit der EC-basierenden kryptografischen Verfahren zugrundeliegende Komplexitätsannahme, dass der Aufwand allgemeiner Algorithmen zur Lösung des ECDLP mindestens exponentiell in der Ordnung n des Basispunktes ist, ist – ebenso wie die Sicherheitsannahmen von RSA, DH und DSA – nicht bewiesen.

Ein wichtiger Unterschied ist allerdings, dass – nicht zuletzt wegen der hohen Komplexität der mathematischen Zusammenhänge – elliptische Kurven im Zusammenhang mit kryptografischen Anwendungen bis heute wesentlich weniger gut untersucht sind als beispielsweise das Faktorisierungsproblem. Daher kann nicht ausgeschlossen werden, dass überraschend effiziente Verfahren zur Lösung des ECDLP entwickelt werden, die ebenfalls subexponentiellen Aufwand besitzen, und damit die Vorzüge der EC-basierenden Verfahren aufheben. So war auch die Entwicklung des MOV-Reduktionsalgorithmus (s.o.) im Jahr 1993 eine überraschende Erkenntnis, die zu einem Ausschluss der besonders effizienten supersingulären elliptischen Kurven führte [MeOV_93].

Außerdem ist zu erwarten, dass viele Implementierungen EC-basierender Verfahren feste, in Standards empfohlene Kurven verwenden werden. Dies vereinfacht die Implementierung und beschleunigt die Schlüsselgenerierung. Allerdings hängt damit die Sicherheit der EC-basierenden Verfahren auch an der Güte dieser vergleichsweise kleinen Auswahl an Kurven. Sollten sich einzelne dieser Kurven durch neue Erkenntnisse als unsicher erweisen, wären davon möglicherweise viele Implementierungen betroffen.

Patentsituation
EC-basierende Verfahren (ECDSA, ECDH) unterliegen als kryptografisches Verfahren keinem Patentschutz. Wohl aber ist die Firma Certicom Corp., Ontario (Kanada) Inhaberin von mehr als 40 Patenten auf Algorithmen im Zusammenhang mit der Arithmetik elliptischen Kurven. Diese Patente umfassen insbesondere die folgenden Aspekte von Implementierungen kryptografischer Verfahren auf elliptischen

Kurven, die erhebliche Bedeutung für eine effiziente Realisierung EC-basierender Kryptoverfahren haben:

- die effiziente Implementierung einer Arithmetik auf elliptischen Kurven über endlichen Körpern, einschließlich der Berechnung von Inversen
- Verfahren zur Punkte-Kompression
- Methoden zur Beschleunigung von Private-Key-Operationen
- verschiedene Versionen der MQV-Schlüsselvereinbarungsprotokolle
- Verfahren zur Verhinderung von Angriffen auf Kurven mit kleinen Untergruppen
- schnelle und effiziente Verfahren für die Punkt-Multiplikation
- Techniken zur Beschleunigung von Multiplikationen in endlichen Körpern
- effiziente Algorithmen für die Arithmetik modulo n
- Verfahren zur Validierung öffentlicher Schlüssel von Kryptoverfahren auf der Basis elliptischer Kurven
- Algorithmen zur effizienten Basis-Umrechnung

Zu den wichtigsten (US-) Patenten von Certicom zählen die folgenden:

- Patent No. 4.745.568 (17.05.1988): *Computational Method and Apparatus for Finite Field Multiplication*. Dieses Patent beinhaltet die effiziente Implementierung von endlichen Körpern über einer Normalbasis-Repräsentation.
- Patent No. 5.600.725 (04.02.1997): *Digital Signature Method and Key Agreement Method*. Dieses Patent umfasst das digitale Signatursystem nach Nyberg-Rueppel (NR).
- Patent No. 5.761.305 (02.06.1998): *Key Agreement and Transport Protocol with Implicit Signatures*. Dieses Patent schließt Versionen der MQV Protokolle ein.
- Patent No. 5.787.028 (28.07.1998): *Multiple Bit Multiplier*.
- Patent No. 5.889.865 (30.03.1999): *Key Agreement and Transport Protocol with Implicit Signatures*. Dieses Patent umfasst ebenfalls Versionen der MQV Protokolle.
- Patent No. 5.896.455 (20.04.1999): *Key Agreement and Transport Protocol with Implicit Signatures*. Auch dieses Patent umfasst Versionen der MQV Protokolle.

Zu diesen Patenten wurden – soweit jeweils möglich – korrespondierende internationale Patente angemeldet.

Da eine Patentierung von Algorithmen bislang nach deutschem Recht nicht möglich war, gelten alle Certicom-Patente nicht für deutsche Hersteller – sofern sie ihre Produkte nicht in Länder mit einem entsprechenden Patentschutz exportieren. In diesem Fall müssen auch europäische Hersteller Lizenzgebühren an Certicom abführen, wenn sie die in den angeführten Patenten geschützten Algorithmen in ihren Produkten einsetzen.

Existierende Patente – wiewohl grundsätzlich ein natürlich legitimer Schutz von Entwicklungen eines Unternehmens – haben in der Vergangenheit häufig die Entstehung und Verbreitung von Standards verhindert oder zumindest erschwert. Denn ein Standard, der patentierte Verfahren vorschreibt, begünstigt einseitig den Patentinhaber. So dürfte die Existenz vieler Certicom-Patente im Zusammenhang mit elliptischen Kurven erheblich mit dazu beigetragen haben, dass kryptografische Verfahren auf der Basis elliptischer Kurven – immerhin seit über 15 Jahren bekannt – nach wie vor in der Praxis eine untergeordnete Rolle spielen.

Integration in Anwendungen

Ein zentrales Manko von Verfahren auf der Basis elliptischer Kurven ist, dass sie bei der Integration in Anwendungen weit hinter dem RSA- und auch hinter dem DSA-Verfahren zurückliegen. Das hat insbesondere damit zu tun, dass – anders als beispielsweise der DSA – auch die standardisierten kryptografischen EC-basierenden Verfahren wie der ECDSA noch nicht in Anwendungsstandards aufgenommen wurden. Insbesondere so wichtige, da bereits in vielen Anwendungen integrierte Internet-Spezifikationen wie S/MIMEv3 oder SSL/TLS berücksichtigen EC-basierende Verfahren nicht.

4 Fazit und Empfehlungen

Kryptografische Verfahren auf der Basis elliptischer Kurven sind heute die einzige ernsthafte Alternative zum RSA-Verfahren. Sie besitzen einige sehr schöne Eigenschaften, wie deutlich kürzere Schlüssel- und Signaturlängen bei gleicher kryptografischer Sicherheit. Durch die kürzere Schlüssellänge verringern sich nicht nur Speicher- und Bandbreitebedarf, sondern auch die Anforderungen an die kryptografischen Operationen. Diese Eigenschaft spielt für Implementierungen auf embedded systems, Smartcards oder mobilen Systemen mit begrenzten Speicher- und Rechenressourcen eine wichtige Rolle. Hinzu kommt, daß EC-basierende Verfahren über endlichen Körpern der Charakteristik 2 ($GF(2^k)$) ohne eine spezielle Arithmetikeinheit oder kryptografische Koprozessoren, sondern mit Standard-Hardware realisiert werden können. Schließlich ist die Schlüsselgenerierung wesentlich effizienter und leichter zu realisieren als beim RSA-Verfahren. Auch lassen sich – anders als beim RSA-Verfahren – die Systemparameter anhand der öffentlichen Parameter auf ihre kryptographische Stärke überprüfen.

Gegen die Verwendung kryptographischer Verfahren auf der Basis elliptischer Kurven spricht derzeit noch ihre geringe Verbreitung. Auch zukünftig werden der umfassende Patentschutz der Firma Certicom auf effiziente Implementierungen sowie das Fehlen einer Integration der Verfahren in wichtige technische Standards (wie S/MIME oder TLS/SSL) die Verbreitung dieser Verfahren noch über viele Jahre behindern.

Daher sind kryptografische Verfahren auf der Basis elliptischer Kurven heute sinnvoll nur in geschlossenen Benutzergruppen mit proprietären Herstellerlösungen einsetzbar. So kommen derzeit der Schlüsselaustausch in VPNs und die interne Dateiverschlüsselung und –signierung als einzige für die Verwendung von kryptografischen Verfahren auf der Basis elliptischer Kurven in Frage. Allerdings ist hier auch der Nutzen begrenzt.

Für den Schutz externer Kommunikation und Kundenanwendungen (E-Mail, Internet-Anwendungen) mit Verfahren auf der Basis elliptischer Kurven sollte die weitere Entwicklung der Standardisierung abgewartet werden. Möglicherweise werden zukünftige Versionen des S/MIME-Standards und des TLS-Protokolls EC-basierende Verfahren berücksichtigen.

Eine wichtige Rolle könnten EC-basierende Verfahren zukünftig in mobilen Anwendungen spielen. Da mobile Endgeräte (PDAs, Handys etc.) in der Regel über sehr wenig Speicherplatz und Rechenkapazität verfügen und auch die Kommunikationsbandbreite zumindest derzeit noch sehr gering ist, würden hier einige der Vorzüge von EC-basierenden Verfahren voll zur Geltung kommen. Auch ist die Leistungsaufnahme von EC-basierenden Krypto-Coprozessoren sehr gering. Zudem gibt es

hier hinsichtlich der verwendeten kryptografischen Verfahren noch wenige Festlegungen in Standards; EC-basierende Verfahren haben daher große Chance, bereits in den ersten Versionen der Standards Berücksichtigung zu finden.

Literatur

ANSI_99 American National Standards Institute (ANSI): *Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*. ANSI X9.62, 1999.

BSI_01 Bundesamt für Sicherheit in der Informationstechnik (BSI): *Geeignete Kryptoalgorithmen in Erfüllung der Anforderungen nach § 17 (1) SigG vom 22. Mai 2001 in Verbindung mit § 17 (2) SigV vom 22. Oktober 1997*, vom 05.07.2001.

ESST_99 Escott, Adrian E.; Sager, John C.; Selkirk, Alexander P.L.; Tsapakidis, Dimitrios: *Attacking Elliptic Curve Cryptosystems Using the Parallel Pollard rho Method*. Cryptobytes, Vol. 4, No. 2, RSA Laboratories, S. 15-19.

FoRö_96 Fox, Dirk; Röhm, Alexander W.: *Effiziente Digitale Signatursysteme auf der Basis elliptischer Kurven*. In: Horster, P. (Hrsg.): Digitale Signaturen. Proceedings der Arbeitskonferenz Digitale Signaturen 96, vieweg-Verlag, 1996, S. 201-220.

GaLV_00 Gallant, Robert; Lambert, Robert; Vanstone, Scott: *Improving the Parallelized Pollard Lambda Search on Binary Anomalous Curves*. Mathematics of Computation, Vol. 69 (2000), S. 1699-1705.

IEEE_00 Institute of Electrical and Electronics Engineers: *Standard Specifications for Public Key Cryptography.* IEEE Standard P1363, 2000.

IEEE_01 Institute of Electrical and Electronics Engineers: *Standard Specifications for Public Key Cryptography: Additional Techniques*. IEEE Standard P1363a, Draft 9, 13.06.2001.

JaMS_01 Jacobson, Michael; Menezes, Alfred J.; Stein, Andreas: *Solving Elliptic Curve Discrete Logarithm Problems using Weil Descent*. Technical Report CORR 2001-31, University of Waterloo, 16.05.2001.

JoMV_99 Johnson, Don B.; Menezes, Alfred J.; Vanstone, Scott: *The Elliptic Curve Digital Signature Algorithm (ECDSA)*. Univ. of Waterloo, 1999 http://cacr.math.waterloo.ca

Kobl_84 Koblitz, Neal: *Introduction to Elliptic Curves and Modular Forms*. Graduate Texts in Mathematics, No. 97, Springer, New York 1984.

LeVe_99 Lenstra, Arjen K.; Verheul, Eric: *Selecting Cryptographic Key Sizes*. November 24, 1999; http://www.cryptosavvy.com.

Mill_85 Miller, Victor: *Use of Elliptic Curves in Cryptology*. In: Williams, H.C. (Hrsg.): Proceedings of Crypto ‘85, LNCS 218, Springer, Berlin 1986, S. 417-426.

MeOV_93 Menezes, Alfred J.; Okamoto, Tatsuaki; Vanstone, Scott A.: *Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field*. IEEE Transactions on Information Theory, Vol. 39, No. 5, Sept. 1993, S. 1639-1646.

NIST_95 National Institute of Standards and Technology (NIST): *Secure Hash Standard (SHS-1)*. Federal Information Processing Standards Publication 180-1 (FIPS-PUB), 17.04.1995.

NIST_00 National Institute of Standards and Technology (NIST): *Digital Signature Standard (DSS)*. Federal Information Processing Standards Publication 186-2 (FIPS-PUB), 27.01.2000.

Odly_95 Odlyzko, A.: *Discrete logarithms in finite fields and their cryptographic significance*. In: Beth, T.; Cot, N.; Ingemarsson, J. (Hrsg.): Proceedings of Eurocrypt ‘84, LNCS 209, Springer, Berlin 1995, S. 224-314.

PaMü_98 Paulus, Sachar; Müller, Volker: *Elliptische Kurven und Public Key-Kryptographie*. Datenschutz und Datensicherheit (DuD), 9/1998, S. 496-499.

PoHe_78 Pohlig, Stephen C.; Hellman, Martin E.: *An Improved Algorithm for Computing Logarithms over GF(p) and Its Cryptographic Significance*. IEEE Trans. on Inf. Theory, Vol IT-24, No.1, January 1978, S. 106-110.

Poll_78 Pollard, J. M.: *Monte Carlo methods for index computation (mod p)*. Mathematics of Computation, 32, 1978, S. 918-924.

SECG_00 Standards for Efficient Cryptography Group: *Elliptic Curve Cryptography*. SEC-1, Version 1.0, 20.09.2000.

SECG2_00 Standards for Efficient Cryptography Group: *Recommended Elliptic Curve Domain Parameters*. SEC-2, Version 1.0, 20.09.2000.

Silv_00 Silverman, Robert D.: *A Cost-Based Security Analysis of Symmetric and Asymmetric Key Lengths*. In: RSA Laboratories Bulletin, No. 13, April 2000, S. 1-22.

Vaud_96 Vaudenay, Serge: *Hidden Collisions on DSS*. In: Koblitz, N. (Hrsg.): Proceedings of Crypto ‘96, LNCS 1109, Springer, Berlin 1996, S. 83-88.

WiZu_99 Wiener, Michael J.; Zucckerato, Robert J.: *Faster Attacks on Elliptic Curve Cryptosystems*. Selected Areas in Cryptography. LNCS 1556, Springer-Verlag, Berlin 1999, S. 190-200.