

# Elliptische Kurven – Basis für ein alternatives Public Key Kryptosystem

Uwe Krieger, cv cryptovision

Im Bereich der Kryptographie sind in letzter Zeit deutliche Veränderungen spürbar: Nicht nur, daß die Vereinigten Staaten ihre Exportbeschränkungen in diesem Bereich lockern, auch auf der Ebene der Algorithmen ist der Umschwung in vollem Gange. Bei den symmetrischen Verfahren ist mit dem AES ein Nachfolger für den DES in Sicht, im Public Key Bereich finden Algorithmen auf Basis von ECC (*Elliptic Curve Cryptography*) immer mehr Verbreitung. Es handelt es dabei um eine Klasse von Verfahren, welche eine Alternative zu dem am weitesten verbreiteten Public Key Algorithmus, dem RSA, darstellen.

## Kryptographie im Wandel

War die Kryptographie früher hauptsächlich ein Gegenstand von militärischem Interesse, so hat sich dies mit zunehmender Verbreitung von Computern im Laufe letzten Jahrzehnte entscheidend verändert. Mehr und mehr wurden Einsatzmöglichkeiten auch außerhalb dieses Umfeldes entdeckt. Zum einen wurde dadurch die Forschung auf diesem Gebiet im Laufe der Zeit immer intensiver, außerdem wurden aber auch die erzielten Ergebnisse für einen immer größer werdenden Anwenderkreis erreichbar. Im weiteren Verlauf dieses Beitrages wird eine Klasse von neuen Verfahren, welche durch diese Öffnung profitierten, genauer vorgestellt wird.

Im Bereich der symmetrischen Chiffren ist der bekannteste Vertreter der bei IBM entwickelte und Mitte der siebziger Jahre standardisierte DES (*Data Encryption Standard*). Auch wenn dieser vom Design immer noch als wegweisend gilt, kann er heute auf Grund seiner zu geringen Schlüssellänge von 56 Bit nicht mehr als sicher angesehen werden. Inzwischen sind spezielle Rechner vorhanden, die in der Lage sind den verwendeten Schlüssel durch reines Ausprobieren aller möglichen Kombinationen innerhalb weniger Stunden zu bestimmen.

Schon vor einiger Zeit hat deswegen das NIST (*National Institute of Standards and Technology*) in den Vereinigten Staaten den Prozess zur Bestimmung eines Nachfolgers angestoßen. Die Eckdaten für diesen, unter dem Stichwort AES (*Advanced Encryption Standard*) geführten Algorithmus setzen neue Maßstäbe für die Sicherheit kryptographischer Verfahren. Anforderungen an Kandidaten für den kommenden Standard sind unter anderem, mit Schlüsseln von 128, 192 und 256 Bit Länge arbeiten zu können.

Symmetrische Chiffren sind aber nur einer der Mechanismen, die in der Kryptographie verwendet werden. Der Name deutet bereits die Hauptschwierigkeit beim Einsatz dieser Verfahren an: Für Ver- und Entschlüsselung müssen bei dieser Klasse von Algorithmen auf beiden Seiten der Kommunikation dieselbe geheime Information (der kryptographische Schlüssel) vorhanden sein. Dieser muß also vorab über einen hinreichend sicheren Kanal - welcher natürlich häufig nicht vorhanden ist - übertragen werden.

Man macht sich außerdem leicht klar, welche Schwierigkeiten diese Verfahren beim Einsatz in einem Netz mit einer großen Anzahl an Teilnehmern bedeutet: Für jede mögliche Kombination von Kommunikationspartnern ist ein separater Schlüssel zu erzeugen und vor allem zu verwalten. Da damit die Anzahl der Schlüssel quadratisch wächst, stellt dies beträchtliche Anforderungen an die Infrastruktur wenn man an Benutzerkreise von mehreren Tausend Personen denkt.

## Public Key Verfahren

Sogenannte asymmetrische (oder auch Public Key) Verfahren helfen bei der Bewältigung einiger der zuletzt angedeuteten Probleme. Bei diesen Algorithmen wird der eigentliche Schlüssel eines Benutzers in zwei Teile aufgespalten, aus dem Schlüssel wird ein Schlüsselpaar. Der öffentliche Schlüssel (auch Public Key genannt) darf oder sollte sogar allgemein bekannt sein, lediglich der geheime Schlüssel, der Private Key, ist nur dem rechtmäßigen Besitzer bekannt.

Die beiden Bestandteile des Schlüssels hängen üblicherweise über eine mathematische Beziehung zusammen die in der einen Richtung leicht auszuwerten, in der anderen Richtung aber praktisch nicht berechenbar ist. Es muß natürlich vermieden werden, dass der private Teil des Schlüssels aus dem Public Key berechnet werden kann. Funktionen, die diese Eigenschaften gewährleisten nennt man in der Kryptographie bezeichnenderweise Einwegfunktionen.

Hat man solch ein Verfahren erst einmal gefunden, so sind die Vorteile offensichtlich: Statt einem Schlüssel pro möglicher Verbindung ist nur noch einer pro Benutzer erforderlich; dies reduziert die Anzahl der Schlüsselpaare gewaltig. Al-

lerdings ist darauf zu achten, daß die Verwendung findenden Schlüssel authentisch sind, also wirklich vom gewünschten Kommunikationspartner stammen. Im Idealfall sind sämtliche öffentliche Schlüssel (gesichert durch sogenannte Zertifikate, welche die Authentizität sichern) in einer für alle zugänglichen Datenbank gespeichert.

Die unterschiedlichen Rollen der beiden Schlüsselbestandteile werden bei der Schilderung der Einsatzmöglichkeiten klar; die beiden bekanntesten sind die Verschlüsselung von Daten sowie die Erstellung von Digitalen Signaturen. Bei der Verschlüsselung mit Hilfe eines asymmetrischen Verfahren geht es darum, dass der Absender einer Nachricht diese mit Hilfe des öffentlichen Schlüssels des Empfängers so chiffriert, daß nur dieser die Operation unter Verwendung seines zugehörigen privaten Schlüssels rückgängig machen kann.

Das zweite Einsatzfeld sind sogenannte Digitale Signaturen; das Ziel ist, bei einem in digitaler Form vorliegenden Dokument das Äquivalent einer handschriftlichen Unterschrift zu erzeugen. Der Absender einer Nachricht berechnet dabei unter Verwendung seines privaten Schlüssels eine zusätzliche Information, die eigentliche Signatur. Mit Hilfe dieser kann sich der Empfänger Gewissheit über den Absender verschaffen, indem er überprüft, ob eine gewisse Gleichung - in welche die Nachricht, die Signatur und der öffentliche Schlüssel des Absenders eingehen - Gültigkeit besitzt.

Der bekannteste Vertreter dieser Klasse von Verfahren ist der nach seinen Entdeckern Ron Rivest, Adi Shamir und Leonard Adleman benannte RSA-Algorithmus; mit seiner Vorstellung im Jahre 1977 begann der Siegeszug der Public Key Kryptographie. Das zugrundeliegende mathematische Problem bei diesem Verfahren ist die Faktorisierung von zusammen gesetzten Zahl; die bereits erwähnte Einwegfunktion ist hierbei dadurch gegeben, daß es einfach ist, für zwei gegebene Primzahlen  $p$  und  $q$  das Produkt  $n = p \cdot q$  zu berechnen; die Umkehrung jedoch, d.h. die Berechnung von  $p$  und  $q$  bei gegebenem  $n$  ist ungleich schwieriger.

Wäre man in der Lage, dieses Problem bei den in der Praxis eingesetzten Parametergrößen zu lösen, so wäre der RSA-Algorithmus geknackt. Um momentan Sicherheit zu gewährleisten, müssen die Werte für  $p$  und  $q$  im Bereich von jeweils ungefähr 150 Dezimalstellen Länge gewählt werden. Damit liegt der Wert von  $n$  in der Größenordnung von etwa 300 Dezimalstellen, dies entspricht den so zitierten 1.024 Bit.

Da asymmetrische Verfahren von der notwendigen Rechenzeit aber üblicherweise um Größenordnungen langsamer sind als ihre symmetrischen Konkurrenten, werden im Praxiseinsatz meist so genannte hybride Methoden implementiert. Bei diesen wird für die eigentliche Verschlüsselung eine schnelle, symmetrische Chiffre verwendet; ein asymmetrischer Mechanismus ist für den Schlüsselaustausch bzw. die Schlüsselübermittlung verantwortlich. Diese Verbindung zweier Verfahren führt aber zurück auf die anfangs erwähnten Sicherheitsanforderungen für die kommenden symmetrischen Chiffren: Eine Kette ist immer nur so stark wie ihr schwächstes Glied, für den geschilderten hybriden Ansatz bedeutet dies, dass die asymmetrischen Algorithmen, wollen Sie eine adäquate Sicherheit bieten, mit den Anforderungen wie sie z.B. an den AES gestellt werden, gleichziehen müssen.

Rechnet man dazu die bekannten Angriffsmethoden auf die für Schlüsselübermittlung verwendeten asymmetrischen Verfahren hoch, so kommt man zu nachdenklich stimmenden Ergebnissen. Heute übliche RSA-Schlüssel müssen dann auf einmal mit 4.096 bis weit über 10.000 Bit Schlüssellänge kalkuliert werden, um eine angemessene Sicherheitsstufe zu erreichen. Die Größe der Parameter stößt damit in nicht mehr praktikable Bereich vor. Die Rechenzeit der notwendigen arithmetischen Operationen wären selbst auf einem modernen PC-Prozessoren deutlich spürbar, in Umgebungen mit beschränkten Rechenkapazitäten (wie z.B. Smartcards) sogar nicht mehr durchführbar.

## Elliptische Kurven

Es gibt aber Alternativen, neue Verfahren bieten Möglichkeiten, diesen Problemen zu begegnen. Die bekannteste und sicherlich attraktivste dieser Methoden ist die Kryptographie auf Basis elliptischer Kurven, kurz ECC (*Elliptic Curve Cryptography*) genannt. Besonderes Merkmal dieser Klasse von Verfahren, ist, dass die notwendigen Berechnungen in diesem Fall nicht direkt mit Zahlen, sondern mit anderen Objekten, den Punkten auf so genannten Elliptischen Kurven durchgeführt werden. Dadurch, dass in der damit zugrunde liegenden mathematischen Struktur einem potenziellen Angreifer bestimmte Angriffsmethoden nicht zur Verfügung stehen, lassen sich die verwendeten Schlüssel- und Parameterlängen ohne Sicherheitseinbuße deutlich reduzieren. Dies macht die Algorithmen insbesondere interessant für den Einsatz innerhalb von Umgebungen mit beschränkten Rechen- oder Speicherkapazitäten wie z.B. Smartcards.

Wie alle anderen Ansätze zur Konstruktion von asymmetrischen Algorithmen basieren auch diese Verfahren auf einem mathematischen Problem, dessen Lösung als schwer angesehen wird. Im Falle des RSA-Algorithmus war dieses Problem die Faktorisierung, also das Zerlegen einer Zahl in ihre Primfaktoren. In der Geschichte der Public Key Kryptographie hat aber ein anderes Problem eine längere Historie: Das Problem des Diskreten Logarithmus ist die Basis des Diffie-Hellman Schlüsselaustausches und war damit die eigentliche Grundlage für den Beginn der asymmetrischen Kryptogra-

phie im Jahre 1976. Das zugrunde liegende Problem ist dabei vergleichsweise einfach zu schildern: Die verwendete Einwegfunktion ist in diesem Fall die Potenzierung einer Zahl, im Folgenden  $g$  genannt.

Die Sicherheit dieses Protokolls beruht auf der Tatsache, dass man für beliebiges  $n$  stets einen effizienten Algorithmus zur Berechnung der Potenz  $g^n$  finden kann, die Umkehrung dieser Operation, d.h. die Bestimmung des Wertes von  $n$  ist aber wieder ab einer gewissen Größenordnung der Parameter nicht mehr praktikabel. Abbildung 1 zeigt das durchzuführende, denkbar einfache Protokoll: Beide Parteien (aus historischen Gründen in der Kryptographie oft mit Alice und Bob benannt) bilden die Potenz des allgemein bekannten Wertes von  $g$  mit ihren temporären, privaten Schlüsseln und sind nach Austausch der berechneten Werte in der Lage, einen nur ihnen bekannten Sitzungsschlüssel, den *session key*, zu berechnen.

Die darauf aufbauende, für die Verwendung von ECC-Algorithmen maßgebliche Idee ist eigentlich nicht neu: Bereits 1985 kamen zwei Mathematiker, Neal Koblitz und Victor Miller, unabhängig voneinander zu der Erkenntnis, daß sich auch andere mathematische Strukturen für den Einsatz innerhalb dieses Diffie-Hellman Protokolls eignen. In der Punktegruppe elliptischer Kurven läßt sich ein Gruppengesetz formulieren, welches es ermöglicht, daß man mit diesen Objekten im Prinzip rechnen kann wie mit normalen Zahlen. Für die dazu auf einem Computer durchzuführenden Berechnungen wird ein Punkt auf solch einer Kurve (s. Graphik) dargestellt durch zwei Werte, die  $x$ - und die  $y$ -Koordinate. Die Länge dieser Zahlenwerte hängt dabei natürlich wieder von der gewünschten Sicherheit ab: Orientiert man sich z.B. an der Sicherheit eines 1024 Bit RSA Schlüssels, so sind diese Parameter im Bereich von 160 Bit zu wählen. Diese deutlich kleineren Zahlen sind auch der Grund für sich einstellenden Performance-Gewinn; auch wenn die durchzuführenden Berechnungen im Falle elliptischer Kurven etwas komplizierter ausfallen als beim Einsatz des RSA, so ist der Unterschied in der Rechenzeit doch deutlich spürbar.

Für eine anschaulichere Darstellung läßt sich dieses Gruppengesetz aber auch geometrisch formulieren, dies ist der zweiten Abbildung zu entnehmen. Für die Addition zweier Werte (im Beispiel  $P$  und  $Q$  genannt) zieht man einfach die Verbindungsgerade durch die beiden Punkte; man erhält dabei einen eindeutigen dritten Schnittpunkt dieser Gerade mit der Kurven. Dies ist allerdings noch nicht das endgültige Ergebnis der Operation, man muß diesen Wert noch einmal wie dargestellt an der  $x$ -Achse spiegeln um die „Summe“ der beiden ursprünglichen Punkte zu erhalten.

Hat man somit erst einmal erklärt, was die Summe von zwei verschiedenen Punkten bedeuten, so läßt sich natürlich dementsprechend einfach erklären, was das Vielfache eines Punktes bedeutet, dieses ergibt sich einfach durch entsprechend häufiges Aufaddieren eines Wertes. Diese Berechnung von  $P = n \cdot G$  für einen Punkt  $G$  auf der elliptischen Kurve und eine Zahl  $n$  im Bereich von 160 Bit ist die wesentliche Operation im Falle von ECC. Wie im Falle des originalen Diffie-Hellman läßt sich auch hier stets ein effizienter Algorithmus für diese Berechnung finden, grundlegend für die Sicherheit des Verfahrens ist wieder, daß für die Berechnung von  $n$  bei bekanntem  $P$  und  $G$  aber kein praktikabler Algorithmus bekannt ist.

Der besondere Reiz von Verfahren auf Basis elliptischer Kurven liegt deswegen darin begründet, daß sich aufgrund dieser nahen Verwandtschaft der beiden Berechnungen im Wesentlichen alle Protokolle, welche für das DL-Problem entwickelt wurden, relativ problemlos auf diese neue Struktur übertragen lassen. Das bekannte Diffie-Hellman Protokoll, üblicherweise mit DH bezeichnet, wird zum EC-DH, die entsprechende Variante eines Verfahrens zur Erstellung Digitaler Signaturen wie dem DSA heißt in diesem Falle EC-DSA. Der entscheidende Vorteil des neuen Ansatzes besteht dabei darin, daß die bekannten schnellen Algorithmen zur Lösung des DL-Problems in endlichen Körpern (wie sie z.B. beim normalen Diffie-Hellman-Protokoll oder dem DSA verwendet werden) in diesem Falle nicht anwendbar sind. Da für das DL-Problem in der Punktegruppe elliptischer Kurven nur sehr allgemeine Verfahren vorhanden sind, kommt man mit deutlich geringeren Schlüssel- und Parameterlängen aus, ohne Abstriche an die Sicherheit in Kauf nehmen zu müssen.

Das es sich bei diesen Verfahren um keine Nischenlösung oder um ein proprietäres Verfahren handelt sieht man bei einem Blick auf die Standardisierungsbemühungen der letzten Jahren: In allen Standards, welche sich mit der Public Key Kryptographie auseinandersetzen, fanden Algorithmen auf Basis elliptischer Kurven entsprechende Berücksichtigung. Dies umfaßt sowohl Dokumente von ANSI, IEEE und ISO als auch zum Beispiel das deutsche Signaturgesetz, welches seit 1997 die mögliche Verwendung einer elektronisch erzeugten Äquivalenz zu einer handschriftlichen Unterschrift regelt.

## **RSA vs. ECC**

Aber ein neuer, alternativer Algorithmus muß heutzutage deutliche Vorteile aufweisen, um sich gegen etablierte Verfahren durchsetzen zu können. Noch ist der Einsatz wirklich starker Kryptographie nicht überall selbstverständlich; oft genug besteht Erklärungsbedarf für die Verwendung asymmetrischer Chiffren. Diese Vorteile sind allerdings im Falle von ECC vorhanden, dies bezieht sich vor allen Dingen auf deutlich verringerten Speicherverbrauch sowie auf kürzere Rechenzeiten. Ein Vergleichswert wurde bereits genannt: Ein 160 Bit langer ECC-Schlüssel entspricht von der Sicherheit einem

1.024 Bit langen RSA-Schlüssel. Die Vergleichbarkeit dieser Werte ist dabei ebenfalls den zuletzt zitierten Standards zu entnehmen.

Die absoluten Zahlen sind aber nicht der eigentliche Reiz dieser Algorithmen; dieser offenbart sich vielmehr beim Blick auf steigende Sicherheitsanforderungen. Da wie erwähnt im Falle von ECC einem potenziellen Angreifer bestimmte Möglichkeiten zum Angriff auf das Verfahren nicht zur Verfügung stehen, erreicht man deutlich erhöhte Sicherheit bei moderatem Anstieg der Schlüssellängen. Beispielsweise erhöht man den für einen Angreifer notwendigen Aufwand auf das Doppelte, indem man um 2 Bit vergrößerte Parameter wählt. Natürlich ist eine Verdoppelung der notwendigen Arbeit - gegeben durch den rasanten Fortschritt bei der Leistungsfähigkeit moderner Rechner - heutzutage für einen Angreifer keine wirkliche Abschreckung. Wo man aber beim RSA-Algorithmus schon an eine Verdoppelung der Bitlänge denken muß, reichen beim Einsatz elliptischer Kurven wenige zusätzliche Bits um wirklich erhöhte Sicherheit zu gewährleisten.

Entsprechende Vergleiche für die gängigen eingesetzten Schlüssellängen machen die Unterschiede deutlich: Ein 512 Bit langer RSA Schlüssel entspricht der Sicherheit von ungefähr 112 im Falle von ECC, der momentane Standard von 1.024 Bit ist vergleichbar mit den bereits erwähnten 160 Bit bei der Verwendung elliptischer Kurven. Die kommenden Schlüssellängen von 2.048 und 4.096 Bit im Falle von RSA werden durch ECC-Schlüssel im Bereich von 200 bis 300 Bit abgedeckt. Wählt man als Extrem einmal ein ECC-System mit 512 Bit Parametergröße (was auf einem modernen PC immer noch auf durchaus akzeptable Rechenzeiten hinausläuft), so werden die Vergleichswerte utopisch: Der RSA hätte, um vergleichbare Sicherheit zu gewährleisten mit weit über 10.000 Bit Schlüssellänge zu operieren und würde durch die notwendigen Rechenzeiten die Geduld eines Anwenders schon gewaltig strapazieren.

Das solche Anpassungen an größere Schlüssellängen notwendig sind, war in den letzten Jahren deutlich zu sehen. Noch vor nicht allzu langer Zeit galten 512 Bit RSA-Schlüssel für praktische Anwendungen als hinreichend sicher, auch wenn schon frühzeitig auf die prinzipielle Angreifbarkeit solcher Werte hingewiesen wurde. Seit einiger Zeit weiß man aber nun, daß diese Parametergrößen nicht nur theoretisch angreifbar sind, die entsprechenden Attacken führten auch in der Praxis zum Erfolg. Im August letzten Jahres ist es einem Forscherteam gelungen, die Aktivitäten einer großen Anzahl von weltweit im Netz verteilten Rechnern zu koordinieren und so die notwendigen Berechnungen zur Lösung des Problems durchzuführen.

Trotzdem werden Schlüssel mit zu geringer Länge heutzutage noch vielfach eingesetzt, z.B. in vielen WWW-Browsern. Auch Anwendungen wie etwa HBCI (*Home Banking Computer Interface*) verwenden mit 768 Bit Schlüssellänge noch Parametergrößen, welche nicht mehr als wirklich sicher bezeichnet werden können. Zwar werden als Standard momentan Schlüssellängen von 1024 Bit angesehen, doch auch diese sind bereits mit einem „Verfallsdatum“ versehen; Trustcenter-Schlüssel, eingesetzt zur Beglaubigung von Schlüsseln der Anwender, verwenden üblicherweise bereits jetzt 2048 Bit. Ab dem Jahre 2004 schreibt das Signaturgesetz dies auch als Mindestschlüssellänge im Falle des Einsatzes von RSA vor.

## Key Management

Eine Besonderheit beim Einsatz von Verfahren auf Basis elliptischer Kurven darf aber nicht unerwähnt bleiben: Beide Kommunikationspartner müssen Kenntnis darüber haben, in welcher Struktur die notwendigen Berechnung durchgeführt werden müssen. Was im Falle des RSA unproblematisch ist (da diese Information direkt im Schlüssel enthalten ist) läuft hier darauf hinaus, daß beide Partner sich auf die zu verwendende Kurve einigen müssen. Ein naheliegender Ansatz wäre natürlich, diese Informationen direkt als Teil des kryptographischen Schlüssels zu betrachten. Dies würde allerdings bedeuten, daß die ECC-Algorithmen auf einen Schlag viel von ihrem Charme verlieren, die damit entstehende Schlüssellänge läge wieder in Bereichen, wie man sie von Algorithmen wie dem RSA gewohnt ist.

Üblicherweise wird deswegen ein anderer Weg beschritten. Schaut man noch einmal in die gängigen Standards aus dem Public Key Bereich, so sieht man, dass dort im Falle von ECC nicht nur die eigentlichen Protokolle fest geschrieben werden. Zusätzlich werden auch Vorschläge für einzusetzende Kurvenparameter, die sogenannten *domain parameter*, gegeben, so dass man dort für alle gängigen Sicherheitsanforderungen entsprechende Parameter mit passender Bitlänge nachschlagen kann. Dadurch ergibt sich die Möglichkeit, beim konkreten Einsatz lediglich auf diese Daten zu verweisen, etwa über einen Namen für eine bestimmte Kurve oder über eine OID (*object identifier*, eine eindeutige Nummer für ein bestimmtes Objekt).

Es würde auch auf ein Performance-Problem hinaus laufen, würde man diese Daten als Teil des Schlüssels betrachten. Abgesehen davon, daß es gerade nicht das Ziel ist, dass jeder Benutzer seine eigene elliptische Kurve verwendet, ist die Bestimmung einer für den Einsatz in der Kryptographie geeigneten Kurve keinesfalls eine einfache Aufgabe. Dies ist auch einer der Gründe, warum solche Daten mit in die Standardisierung aufgenommen wurden. Es gibt zwar verschiedene Möglichkeiten für die Berechnung der notwendigen Parameter welche die Kurve charakterisieren, eines ist ihnen aber

allen gemeinsam: Es ist ein relativ zeitaufwendiger Prozess, zu überprüfen ob eine bestimmte Kurve wirklich sicher und damit für den Einsatz innerhalb der Kryptographie geeignet ist.

Die eigentliche Schlüsselerzeugung ist dagegen in idealer Weise zu realisieren: Für den Einsatz als privaten Schlüssel eignet sich bei dieser Klasse von Algorithmen jeder beliebige, zufällig gewählte Zahlenwert im Bereich der zugrundeliegenden Bitlänge. Es sind keine weiteren Voraussetzungen zu erfüllen, keinerlei zeitaufwendige Berechnungen wie z.B. die im Falle des RSA notwendigen Primzahltests. Damit ist ein guter Zufallszahlengenerator das einzige Hilfsmittel, was für die eigentliche Schlüsselgenerierung benötigt wird. Seine Ausgabe kann im Wesentlichen direkt als privater Schlüssel verwendet werden, der zugehörige öffentliche Schlüssel ergibt sich durch eine schnell durchführbare Operation auf der Kurve.

Der dadurch entstehende Vorteil wird vielleicht noch deutlicher, wenn man sich eine mögliche Anwendung überlegt: Jede zufällige Bitfolge ist als Schlüssel geeignet ist, man ist aber nicht unbedingt auf einen Zufallszahlengenerator angewiesen um diese zu erhalten. Es ist genauso gut möglich, dass ein Benutzer sich einen längeren, leicht zu merkenden Text als Passphrase wählt, aus der sich der eigentliche kryptographische Schlüssel zum Beispiel durch Anwendung einer Hashfunktion ergibt. Dies ist ein wesentlicher Unterschied zu anderen Verfahren, hier ist man üblicherweise dazu gezwungen, den eigentlichen Schlüssel auf der Festplatte des Rechners oder auf einer Diskette zu speichern. Einerseits ist nämlich niemand in der Lage, sich einen Zahlenwert von mehr als 300 Dezimalstellen Länge zu merken, andererseits ist aber auch kein Verfahren bekannt, wie man solch eine Zahl auf einen leicht zu merkenden Text abbilden könnte. Die erforderliche Struktur des Schlüssels macht es in diesem Fall unmöglich, auf effiziente Weise den umgekehrten Weg zu gehen, d.h. erst solch eine Information festzulegen und daraus den Schlüssel zu generieren.

## Smartcards

Eine ideale Einsatzumgebung für ECC-basierte Verfahren bieten die immer weiter verbreiteten Smartcards; diese intelligenten Chipkarten haben die Aufgabe, den für die kryptographische Berechnungen notwendigen Schlüssel sicher aufzubewahren. Es soll vermieden werden, dass diese sensitiven Daten im Prinzip für jedermann zugänglich auf dem PC gespeichert sind. Anfangs als reine Speicherkarten eingesetzt, ergibt sich durch die gesteigerte Leistungsfähigkeit der eingesetzten Prozessoren nunmehr auch die Möglichkeit, die eigentlichen Berechnungen direkt auf der Karte durchzuführen. Der Vorteil ist offensichtlich: Der Schlüssel verläßt bei dieser Vorgehensweise nie die speziell gesicherte Umgebung in der Karte, er gelangt nicht in den (potenziell durch Viren oder Trojanische Pferd angreifbaren) Adressraum des Rechners.

Allerdings darf man nicht zu viele Erwartungen an die Rechenkapazität solcher Geräte stellen; diese sind auch bei den High-End-Prozessoren in diesem Bereich mit den üblichen 1024 Bit für den RSA am Rande ihrer Leistungsfähigkeit. Bei dem zur Verfügungen stehenden Speicherplatz denkt man üblicherweise heute an Werte zwischen 8 und 32 KByte; Eckdaten für Rechenzeiten auf solchen Chips sind ungefähr 500 – 1000 ms für die Erstellung einer Signatur mit dem RSA-Algorithmus, bei Verfahren auf Basis elliptischer Kurven ist solch eine Operation dagegen in ungefähr 100 – 150 ms durchgeführt.

Aber nicht nur dieser deutliche Geschwindigkeitsvorteil ist ein Argument für den Einsatz dieser Algorithmen. Ein weiteres ergibt sich bei der Betrachtung der Sicherheit: Maßgeblich für den Einsatz von Smartcards war wie schon angemerkt, daß der geheim zu haltende private Schlüssel zu keinem Zeitpunkt außerhalb der Karte verfügbar ist. Dadurch, dass die Schlüsselerzeugung im Falle von ECC ohne zeitaufwendige Tests durchgeführt werden kann, ist die Erzeugung der Schlüssel auf der Karte problemlos möglich, in der Regel stellen die verwendeten Prozessoren den dazu notwendigen Zufallszahlengenerator zur Verfügung. Es ist damit nicht - wie z.B. im Falle des RSA üblich - nötig, das Schlüsselpaar außerhalb der Karte zu erzeugen und erst anschließend in einem aufwendig gesicherten Prozess auf diese zu übertragen. Für die Personalisierung von Karten ergeben sich damit zahlreiche Vereinfachungen und zum Teil vollkommen neue Möglichkeiten.

All dies sind überzeugende Argumente für die Verwendung dieser Algorithmen; in Zukunft werden sich ein Großteil von Anwendungen im Bereich solcher Prozessoren auf diese Verfahren stützen. Schon jetzt bietet z.B. die Firma Orga für ihre Smartcards die ECC-Algorithmen mit der neuesten Version ihres Betriebssystems MICARDO an, zahlreiche andere Hersteller arbeiten ebenfalls an entsprechenden Implementierungen. Es ist damit absehbar, daß solche Funktionalität in absehbarer Zeit auf gängigen SIM-Karten für den GSM-Bereich verfügbar sein werden.

Smartcards sind aber nur eine Spielform von so genannten PSEs (*Personal Security Environments*); andere mögliche Geräte zum Schutz von so vertraulichen Daten wie kryptographischen Schlüsseln sind z.B. PDAs wie ein Palm Pilot oder aber eben auch Mobilfunkgeräte. Wesentlich ist bei all diesen Geräten, dass sie mobil eingesetzt werden können, und dass es in diesem Zusammenhang wesentlich ist, effiziente Algorithmen zur Verfügung zu haben welche auch in solchen Einsatzumgebungen hohe Sicherheitsstandards gewährleisten können.

## **Zusammenfassung**

Algorithmen auf Basis elliptischer Kurven sind inzwischen seit einiger Zeit etabliert. In allen Standardisierungsbemühungen der letzten Jahre, welche sich mit Public Key Verfahren beschäftigen, bilden sie einen wesentlichen Bestandteil. Es ist für einen alternativen Ansatz allerdings nicht einfach, weite Verbreitung zu finden solange ein einzelner Algorithmus die Szene beherrscht. Die Vorteile sprechen in diesem Falle aber für sich, speziell neue Anwendungen wie z.B. im Bereich M-Commerce sowie die in nächster Zeit notwendigen Anpassungen der Schlüssellängen werden dabei ihre Wirkung zeigen. Nicht ohne Grund sind diese Verfahren bekanntermaßen der vom BSI (Bundesamt für Sicherheit in der Informationstechnik) bevorzugte Public Key Algorithmus, entscheidend dürfte dabei der durch die vorhandenen Sicherheitsreserven gegebene „Investitionsschutz“ sein. Auch im Bereich der Anwendungen werden sie immer sichtbarer, beispielsweise erfolgt die Absicherung im Informationsverbund Bonn – Berlin durch diese Verfahren.

## **Zum Unternehmen**

Die cv cryptovision GmbH engagiert sich speziell im Bereich der geschilderten Verfahren und liefert Lösungen für die verschiedensten Plattformen von Smartcards bis hin zu Großrechnersystemen. Weitere Informationen sind erhältlich unter <http://www.cryptovision.com> oder durch Nachfrage per Email an [info@cryptovision.com](mailto:info@cryptovision.com).