Verschlüsselung und elektronische Unterschriften mit elliptischen Kurven

Beginnen wir mit einem Verschlüsselungsverfahren, das – genau wie der Schlüsselaustausch nach DIFFIE-HELLMAN – auch dann funktioniert, wenn die beiden Teilnehmer anfänglich keinerlei Information übereinander haben, das Verfahren von MASSEY und OMURA.

Das Verfahren läßt sich am einfachsten verstehen, wenn wir mit einem nichtmathematischen Analogon beginnen: Angenommen, A möchte einen Container mit wichtigen Unterlagen an B schicken, traut aber dem Transporteur nicht. Wenn er B vorher treffen kann, kauft er einfach ein gutes Vorhängeschloß und gibt B einen der beiden Schlüssel. Später kann er dann den Container mit dem Schloß und seinem Schlüssel verschließen, und B kann mit seinem Schlüssel das Schloß wieder entfernen, um den Container zu öffnen.

Wenn A und B keine Möglichkeit zu einem vorherigen Treffen haben, müssen sie umständlicher vorgehen: Jetzt kauft sich jeder der beiden ein Schloß, dessen Schlüssel dann nur er hat. A verschließt den Container mit seinem Schloß und schickt ihn an B. Der kann ihn natürlich nicht öffnen und schickt ihn deshalb ungeöffnet wieder zurück, verschließt ihn aber vorher noch zusätzlich mit *seinem* Schloß. A kann nun *sein* Schloß entfernen und schickt den Container, nun nur noch mit Bs Schloß gesichert, an B. Dieser kann *sein* Schloß entfernen und dann den Container öffnen.

In der digitalen Welten sieht das mit elliptischen Kurven so aus:

A und B einigen sich auf eine elliptische Kurve E über einem endlichen Körper. Sie bestehe aus n Punkten. Diese Kurve kann auch für ein ganzes Netzwerk, dem A und B angehören, vorgegeben sein.

Wenn A eine Nachricht an B schicken will, kodiert er sie durch einen Punkt $M \in E$. Sodann wählt er eine zu n teilerfremde natürliche Zahl m_A und schickt den Punkt $M_1 = m_A M$ an B. Der kann damit natürlich nichts anfangen, genauso wenig wie ein etwaiger Angreifer: Für beide ist M_1 einfach irgendein Vielfaches irgendeines Punktes. Selbst ein Angreifer mit unbegrenzter Rechenkraft, der für jeden Punkt von E alle

Vielfachen berechnen kann, wird viele verschiedene Paare (c, P) aus einer natürlichen Zahl und einem Punkt bekommen, für die $cP = M_1$ ist, und je nach Kodierung werden möglicherweise eine ganze Reihe der Punkte P sinnvollen Nachrichten entsprechen.

Daher wählt B als nächstes eine zu n teilerfremde natürliche Zahl m_B und schickt $M_2 = m_B M_1$ an A. Da m_A teilerfremd zu n ist, konnte A zwischenzeitlich mit dem erweiterten EUKLIDischen Algorithmus eine Zahl n_A berechnen mit $m_A n_A \equiv 1 \bmod n$; er schickt $M_3 = n_A M_2$ zurück an B. Der hat inzwischen via EUKLID eine Zahl n_B gefunden mit $m_B n_B \equiv 1 \bmod n$ und berechnet $M_4 = n_B M_3$.

E ist eine Gruppe der Ordnung n; nach LAGRANGE ist daher $nM = O$ für jeden Punkt $M \in E$ und somit auch $n_A m_A M = M$ und $n_B m_B M = M$. Daher ist

$$M_4 = n_B\Big(n_A\big(m_B(m_A M)\big)\Big) = (n_B m_B)(n_A m_A)M = M\,,$$

womit B die Nachricht kennt.

Auch die Sicherheit dieses Verfahrens hängt an diskreten Logarithmen: Ein etwaiger Lauscher kennt die Punkte M_1, M_2 und $M_3 = n_A M_2$; falls er das diskrete Logarithmenproblem auf E lösen kann, kennt er also n_A und kann $M = n_A M_1$ berechnen.

Ein Nachteil dieses Verfahrens ist, daß die (jeweils modifizierte) Nachricht dreimal versendet werden muß und daß, wie bei DIFFIE-HELLMAN, natürlich auch hier eine *man in the middle attack* möglich ist – es sei denn, A und B haben die Möglichkeit, eine der dort angedeuteten Gegenmaßnahmen zu ergreifen.

Ein weiteres, leichter zu lösendes Problem, ist die Frage, wie man eine Nachricht durch einen Punkt einer elliptischen Kurve kodieren kann. Ein Vorschlag davon stammt von einem der Pioniere der Kryptographie mit elliptischen Kurven, dem amerikanischen Mathematiker NEAL KOBLITZ von der University of Washington in Seattle.

Er wählt eine elliptische Kurve E mit einer WEIERSTRASS-Gleichung $y^2 = x^3 + ax + b$ über einem Körper $\mathbb{F}_p$. Für die Nachricht m sei $0 \le m < \frac{p}{100}$. KOBLITZ betrachtet für $j = 0, 1, 2, \ldots$ die Elemente

$x_j = 100m + j \in \mathbb{F}_p$ und berechne dazu $x_j^3 + ax_j + b$, bis eine dieser Zahlen das Quadrat eines Elements $y_j \in \mathbb{F}_p$ ist. Die Nachricht wird dann kodiert als der Punkt (x_j, y_j). Der Empfänger kann m aus diesem Punkt bestimmen, indem er einfach die letzten beiden Stellen von x_j streicht.

Dies wirft natürlich die Frage auf, ob es ein $j < 100$ gibt, für das $x_j^3 + ax_j + b$ in $\mathbb{F}_p$ ein Quadrat ist. Dazu betrachten wir den Homomorphismus multiplikativer Gruppen

$$\begin{cases} \mathbb{F}_p^\times \to \mathbb{F}_p^\times \\ \quad x \mapsto x^2 \end{cases} .$$

Sein Kern besteht aus 1 und $p-1$, hat also zwei Elemente. Nach dem Homomorphiesatz hat das Bild daher $(p-1)/2$ Elemente. Da Null ein Quadrat ist, sind unter den p Elementen des Körpers $\mathbb{F}_p$ daher $(p+1)/2$ Quadrate, also etwas mehr als die Hälfte. Falls wir davon ausgehen, daß die Elemente $x_j^3 + ax_j + b$ für $j = 0, \ldots, 99$ genau wie auch die Quadrate in $\mathbb{F}_p$ zufällig verteilt sind (wofür es keinerlei Grund gibt), ist also die Wahrscheinlichkeit, daß unter diesen hundert Elementen kein Quadrat ist kleiner als

$$2^{-100} = \left(2^{-10}\right)^{10} = \frac{1}{1024^{10}} < \frac{1}{1000^{10}} = 10^{-30} .$$

Auch wenn die hier verwendeten Annahmen sehr heuristisch sind, kann man doch davon ausgehen, daß sich in der Praxis immer ein Punkt $(x_j, y_j) \in E$ mit $j < 100$ finden läßt, meist mit einem j im niedrigen einstelligen Bereich.

Für die praktische Anwendung stellt sich als nächstes die Frage, wie man erkennt, ob $x_j^3 + ax_j + b$ ein Quadrat in $\mathbb{F}_p$ ist, und wie man dann ein y_j findet mit $y_j^2 = x_j^3 + ax_j + b$.

Grundsätzlich stellt die Zahlentheorie Verfahren bereit, mit denen dieses Problem gelöst werden kann: Mit dem sogenannten quadratischen Reziprozitätsgesetz läßt sich für jedes $c \in \mathbb{F}_p$ schnell entscheiden, ob es ein $y \in \mathbb{F}_p$ gibt mit $y^2 = c$, und ein Algorithmus von SHANKS erlaubt es auch, ein solches y zu berechnen.

Am einfachsten und elementarsten ist die Situation, wenn $p \equiv 3 \bmod 4$ ist, so daß wir uns hier auf diesen Fall beschränken wollen. Für solche Primzahlen ist $p+1$ durch vier teilbar, was wir im Folgenden wesentlich ausnutzen.

Falls es ein $y \in \mathbb{F}_p$ gibt mit $y^2 = c \neq 0$, ist nach dem kleinen Satz von FERMAT

$$\left(c^{(p+1)/4}\right)^2 = c^{(p+1)/2} = y^{p+1} = y^{p-1} \cdot y^2 = 1 \cdot y^2 = c\,,$$

also ist $c^{(p+1)/4}$ in $\mathbb{F}_p$ eine Quadratwurzel von c. Um zu testen, ob ein vorgegebenes Element $c \in \mathbb{F}_p^\times$ ein Quadrat ist, kann man daher $y = c^{(p+1)/4}$ berechnen; falls $y^2 = c$ ist, ist c ein Quadrat und y ein Quadratwurzel; andernfalls kann c kein Quadrat sein.

Ein anderer Ansatz zur Kryptographie mit elliptischen Kurven geht zurück auf ELGAMAL. Er entwarf ein Kryptoverfahren mit öffentlichen Schlüsseln auf der Grundlage von DIFFIE-HELLMAN. Dabei ging er natürlich von deren ursprünglichem Verfahren in $\mathbb{F}_p^\times$ aus, aber nachdem KOBLITZ und MILLER auf die Idee gekommen waren, diskrete Logarithmenprobleme auch auf elliptischen Kurven zu betrachten, war die Übertragung eine reine Formalität.

Beim Schlüsselaustausch nach DIFFIE-HELLMAN ist die Beziehung zwischen den beiden Teilnehmern A und B völlig symmetrisch; bei einem Verfahren mit öffentlichen Schlüsseln ist sie das natürlich nicht mehr. A sei der Teilnehmer, der einen öffentlichen Schlüssel publizieren will, und B sei einer der Vielen, die eine Nachricht an A senden möchten.

Zunächst muß eine elliptische Kurve E über einem endlichen Körper $\mathbb{F}_p$ gewählt werden sowie ein Punkt $P \in E$ mit möglichst hoher Ordnung, die idealerweise eine Primzahl sein sollte, zumindest aber einen großen Primteiler haben muß. E und P können für ein ganzes Netzwerk global festgelegt sein, sie können aber auch von A selbst gewählt werden. In der Praxis gibt es Empfehlungen geeigneter Paare (E, P) von Sicherheitsbehörden, wobei sich allerdings inzwischen auch eine der von der amerikanischen *National Security Agency* empfohlenen Kurven als (wahrscheinlich gewollt) problematisch erwiesen hat.

Auf jeden Fall selbst wählen muß A seine Zufallszahl x, die gleichzeitig sein geheimer Schlüssel ist; als öffentlichen Schlüssel publiziert er den Punkt $U = xP$.

Wenn nun B eine Nachricht an A schicken möchte, kodiert er diese (z.B. nach dem Verfahren von KOBLITZ) als einen Punkt $M \in E$ und wählt eine Zufallszahl y. Damit berechnet er die Punkte $V = yP$ und $W = M + yU$. Das Paar (V, W) geht als verschlüsselte Nachricht an A.

Dieser entschlüsselt die Nachricht als

$$W - xV = M + yU - xyP = M + yU - yU = M\,.$$

Für eine längere Nachricht wird ein Punkt M meist nicht ausreichen; die Nachricht muß dann in mehrere Blöcke $m_1, \ldots, m_r$ zerlegt werden, die zu entsprechenden Punkten $M_1, \ldots, M_r$ führen. Nun ist es extrem wichtig, daß B für jeden Punkt M_i eine neue Zufallszahl y_i wählt: Angenommen, er verwendet für alle Blöcke die gleiche Zufallszahl y, und ein Angreifer errät für einen Block den Klartext. Solche Angriffe mit bekanntem Klartext darf man in der Kryptographie nie ausschließen, denn viele Nachrichten haben eine feste Struktur: Manche beginnen mit einem immer gleichen Briefkopf mit leicht erratbarem Datum, gefolgt von Information über den Adressaten; andere, beispielsweise im Bankenbereich, sind nicht für menschliche Empfänger, sondern für Computer geschrieben und daher sehr stark formalisiert. Ein Angreifer, der eine Überweisung an eine kleinere Bank in Auftrag gibt und deren verschlüsselte Übermittlung abfängt, kann, sofern er das System kennt, den Klartext wohl problemlos rekonstruieren. Auch Schlußformeln in Nachrichten sind oft sehr vorhersehbar.

Wir können also nicht ausschließen, daß ein Angreifer zumindest einen der Nachrichtenblöcke m_i kennt und damit auch den Punk M_i. Zusätzlich kennt er aus seinem Lauschangriff die übermittelten Daten $V = yP$ und $W_i = M_i + yU$. Damit kann er $yU = W_i - M_i$ berechnen, und damit für jeden anderen Nachrichtenblock m_j auch $M_j = W_j - yU$.

Somit muß für jeden Nachrichtenblock m_i eine neue Zufallszahl y_i gewählt werden, und als chiffrierter Block muß das Paar aus $V_i = y_iP$ und $W_i = M_i + y_iU$ übertragen werden. Der Chiffretext ist damit mehr

als doppelt so lang wie der Klartext, was das Verfahren für lange Nachrichten eher unattraktiv macht.

Andererseits sind die bekannten Verfahren mit öffentlichen Schlüsseln allesamt deutlich langsamer als klassische symmetrische Kryptoverfahren, so daß sie für lange Nachrichten ohnehin nicht benutzt werden: Ihre Hauptanwendung ist die sichere Vereinbarung eines Schlüssels für ein symmetrisches Kryptoverfahren, wobei zur Sicherheit auch insbesondere die gegen eine *man in the middle attack* gehört.

Diese Sicherheit wird in der Praxis garantiert durch elektronische Unterschriften, und die sind die Hauptanwendung von ELGAMALs Verfahren.

Eine elektronische Unterschrift unter eine Nachricht m ist eine Information, die nur der Inhaber eines geheimen Schlüssels aus m berechnen kann, von der aber jedermann anhand des öffentlichen Schlüssels überprüfen kann, ob sie korrekt ist.

Ein einfaches Beispiel dafür liefert das RSA-Verfahren: Angenommen, der öffentliche Schlüssel ist das Paar (N, e), und der geheime Schlüssel ist d. Die Verschlüsselungsfunktion ist somit die Abbildung von $\mathbb{Z}/N$ nach $\mathbb{Z}/N$, die m auf $c = m^e \bmod N$ abbildet; die Entschlüsselungsfunkition bildet $c \in \mathbb{Z}/N$ ab auf $c^d \bmod N$. Hier kann man für eine Nachricht $m \in \mathbb{Z}/N$ die Potenz $u = m^d \bmod N$ als elektronische Unterschrift nehmen: Nur der Inhaber des geheimen Schlüssels d kann sie berechnen, aber da $u^e = m^{de} = m^{ed} \equiv m \bmod N$ ist, kann jeder mit dem öffentlichen Schlüssel (N, e) überprüfen, ob die Unterschrift zur Nachricht paßt. (Bei längeren Nachrichten wird üblicherweise ein kryptographisch sicherer Hashwert unterschrieben; da bislang meines Wissens noch niemand Hashverfahren auf der Grundlage elliptischer Kurven vorgeschlagen hat, braucht uns das im Rahmen dieser Vorlesung nicht weiter interessieren.)

ELGAMALs Verfahren für elektronische Unterschriften funktioniert auf elliptischen Kurven folgendermaßen:

Der Unterschreibende legt folgende Daten fest:

- Eine elliptische Kurve E über einem Körper $\mathbb{F}_p$ zusammen mit einen Punkt $P \in E$, dessen Ordnung q möglichst groß ist, idealerweise eine Primzahl, auf jeden Fall aber eine Zahl mit einem großen Primteiler. Dazu kommt noch eine Abbildung $f\colon E \to \mathbb{Z}$, die beispielsweise einen Punkt $P \in E$ abbildet auf seine x-Koordinate, aufgefaßt als Element der Menge $\{0, \ldots, p-1\}$. Das Tripel (E, P, f) kann auch global für ein ganzes Netzwerk gelten.
- Eine natürliche Zahl x als seinen geheimen Schlüssel
- Den Punkt $U = xP$ als seinen öffentlichen Schlüssel.

Unterschrieben werden Nachricht $m < q$. Dazu wählt der Unterschreibende für jede Nachricht eine neue zu q teilerfremde Zufallszahl $y < q$ und berechnet $V = yP$. Wegen der Teilerfremdheit hat y modulo N ein mit dem erweiterten EUKLIDischen Algorithmus leicht berechenbares Inverses y^{-1}, mit dem er $s = y^{-1}\big(m - xf(V)\big) \bmod q$ berechnen kann. Die Unterschrift unter die Nachricht m besteht aus dem Paar (V, s).

Falls die Unterschrift korrekt ist, gilt

$$\begin{aligned} f(V)U + sV &= f(V)xP + syP \\ &\equiv f(V)xP + \big(m - xf(V)\big)P = mP\,, \end{aligned}$$

und das kann jeder anhand der öffentlich bekannten Daten verifizieren. Wesentlich ist dabei, daß P ein Punkt der Ordnung q ist, so daß es bei Vielfachen von P nur auf die Kongruenzklasse des Faktors modulo q ankommt.

Bei Kryptosystemen auf der Basis klassischer diskreter Logarithmen muß man nach derzeitigen Sicherheitsstandards bekanntlich modulo einer Primzahl mit mindestens 2000 oder besser 3000 Bit arbeiten. Das macht einerseits die Rechnungen umständlich, aber damit haben heutige Computer und selbst Chipkarten keine großen Probleme. Problematischer ist die große Länge der Unterschriften: Unterschrieben wird typischerweise ein Hashwert, der bei den heute empfohlenen Verfahren meist 256 Bit lang ist, und verglichen damit erscheint eine Unterschrift der Länge 3000 doch übertrieben.

1991 wurde daher eine Variante entwickelt, bei der zwar die wesentlichen Rechnungen modulo einer großen Primzahl p ausgeführt werden,

so daß die Sicherheit auf dem diskreten Logarithmenproblem modulo p beruht; die Unterschriften liegen aber in einer Untergruppe von $\mathbb{F}_p^\times$, deren Ordnung q vergleichbar mit der Länge der zu unterschreibenden Hashwerte ist, so daß die Unterschrift etwa die gleiche Länge wie der Hashwert hat. Das Verfahren wurde standardisiert als DSA = Digital Signature Algorithm.

Bei der Verwendung elliptischer Kurven reichen nach heutigen Standards Kurven über Körpern $\mathbb{F}_p$ mit 256-Bit-Primzahlen p; trotzdem wurde die Idee, statt in der ganzen Gruppe E nur in einer Untergruppe zu rechnen, auch auf elliptische Kurven übertragen und führte zu ECDSA, dem Elliptic Curve Digital Signature Algorithm.

Die verwendete Untergruppe ist, wie bei ELGAMAL, die von einem festen Punkt erzeugte zyklische Untergruppe; wir gehen also aus von einer elliptischen Kurve E über einem Körper $\mathbb{F}_p$ und wählen dort einen Punkt P, dessen Ordnung eine Primzahl $q \neq p$ sei. Sie sollte nach derzeitigen Sicherheitsanforderungen mindestens 256 Bit lang sein.

Als geheimen Schlüssel wählt jeder Teilnehmer zufällig eine Zahl x aus $\{1, 2, \ldots q-1\}$; sein öffentlicher Schlüssel ist (abgesehen von E und P) der Punkt $U = xP$.

Um eine Nachricht $m \in \{1, \ldots, q-1\}$ zu unterschreiben, wählt er zunächst eine Zufallszahl $y \in \{1, \ldots, q-1\}$ und berechnet $V = yP$; dieser Punkt habe die Koordinaten (u, v). Sodann berechnet er

$$s = k^{-1}(m + xu) \bmod q\,.$$

Die Unterschrift ist das Paar (V, s).

Zur Verifikation der Unterschrift werden zunächst die beiden Zahlen $a = s^{-1}m \bmod q$ und $b = s^{-1}u \bmod q$ berechnet. Mit diesen ist

$$aP + bU = s^{-1}mP + s^{-1}uU = s^{-1}(mP + uxP) = s^{-1}(m + ux)P\,.$$

Aus $s = y^{-1}(m + xu) \bmod q$ folgt $s^{-1} \equiv y(m + xu)^{-1} \bmod q$, also

$$aP + bU = y(m + xu)^{-1}(m + xu)P = yP = V\,,$$

denn P ist ja ein Punkt der Ordnung q, so daß es beim Vorfaktor nur auf dessen Kongruenzklasse modulo q ankommt.

Die Bedingung $aP + bU = V$ kann anhand der öffentlichen Information überprüft werden; ist sie erfüllt, wird die Unterschrift akzeptiert.

Die teuerste Operation beim Unterschreiben ist die Inversenbildung modulo q: Dazu muß der erweiterte EUKLIDische Algorithmus auf die zu invertierende Zahl und q angewandt werden, was erheblich aufwendiger ist als bloße modulare Additionen und Multiplikationen und Gruppenoperationen auf E. Die Variante ECGDSA (Elliptic Curve German Digital Signature Algorithm) verlegt die Inversenbildung von der Unterschrift in die Schlüsselwahl, Da Unterschriften oft mit Chipkarten erzeugt werden, deren Rechenkraft deutlich kleiner ist als die eines Computers, macht dies das Unterschreiben schneller und einfacher.

Ausgangspunkt ist wieder eine elliptische Kurve E über einem Körper $\mathbb{F}_p$ zusammen mit einem Punkt $P \in E$ der Ordnung $q < p$, und der geheime Schlüssel x wird zufällig aus der Menge $\{1, \ldots, q-1\}$ gewählt. Der öffentliche Schlüssel ist jetzt aber nicht mehr der Punkt xP, sondern $Q = x'P$, wobei $x' = x^{-1} \bmod q$ ist, aufgefaßt als Element der Menge $\{1, \ldots, q-1\}$.

Zum Unterschreiben einer Nachricht $m \in \{1, \ldots, q-1\}$ wird wieder eine Zufallszahl $y \in \{1, \ldots, q-1\}$ gewählt, und $V = yP$ berechnet; die Koordinaten dieses Punktes seien (u, v). Da E eine Kurve über $\mathbb{F}_p$ ist, liegen u und v in $\mathbb{F}_p$, was wir mit der Menge $\{0, \ldots, p-1\}$ identifizieren. Dann können wir $r = u \bmod q$ bilden; falls r verschwindet, wählen wir ein neues y und wiederholen die Berechnung für den neuen Punkt yP. Andernfalls wird $s = (yr - m)x \bmod q$ berechnet. Auch wenn s verschwindet, fangen wir mit einem neuen y von vorne an; andernfalls ist (r, s) die Unterschrift.

Zur Verifikation der Unterschrift ist eine Inversion notwendig, und zwar muß $r' = r^{-1} \bmod q$ berechnet werden, aufgefaßt als natürliche Zahl aus $\{1, \ldots, q-1\}$. Damit können dann zwei Zahlen $a = r'm$ und $b = r's$ berechnet werden sowie der Punkt $aP + bQ$.

Falls die Unterschrift korrekt zu Stande kam, ist

$$aP + bQ = r'mP + r'sx^{-1}P = (r'm + r'sx^{-1})P\,,$$

und

$$\begin{aligned} r'm + r'sx^{-1} &= r'm + r'(yr - m)xx^{-1} = r'm + r'ryxx' - r'm \\ &= r'ryxx' \equiv y \bmod q\,. \end{aligned}$$

Somit ist $aP + bQ = yP$, wobei wir wieder ausgenutzt haben, daß da der Punkt P die Ordnung q hat. Der Punkt yP hat die dem Verifizierer unbekannten Koordinaten (u, v), aber $r = u \bmod q$ ist Teil der Unterschrift. Die Unterschrift wird daher akzeptiert, wenn die x-Koordinate von $aP + bQ$ kongruent r modulo q ist.

Man beachte, daß wir es bei allen DSA-Varianten mit zwei Primzahlen p und q zu tun haben. Für die Gruppenoperation auf der elliptischen Kurve müssen wir mit den Koordinaten aus $\mathbb{F}_p$ rechnen, aber mit den hier als Zahlen kodierten Nachrichten rechnen wir modulo q. Bei ECGDSA werden beide Restklassenbildungen teilweise kombiniert; deshalb ist es hier wichtig, daß $q < p$ ist und daß wir $\mathbb{F}_p$ mit der Menge $\{0, \ldots, p-1\} \subset \mathbb{N}_0$ identifizieren; nur durch diese Festlegung ist es möglich, sinnvoll vom Wert einer Koordinaten modulo q zu reden.