Kapitel 8
Kryptographische Protokolle

Bislang hatten wir Kryptologie nur im Zusammenhang mit Verschlüsselung und mit elektronischen Unterschriften betrachtet; in diesem Kapitel sollen einige darüber hinausgehende Aspekte betrachtet werden.

Eine wichtige solche Anwendung ist beispielsweise die Identitätsfeststellung: Der Gebrauch einer Geldkarte (oder auch eines Mobiltelephons) hängt wesentlich davon ab, daß die Auszahlung *bzw.* die Gesprächsgebühren dem richtigen Konto belastet werden können.

Bei Geldkarten wird dies heute so realisiert, daß der Benutzer eine Geheimzahl eingeben muß, die in verschlüsselter Form im Magnetstreifen der Karte kodiert ist. Die Verschlüsselung hängt nicht nur ab von der Geheimzahl, sondern auch von den Kontodaten des Inhabers, so daß keine Bijektion zwischen den nur knapp zehn Tausend verschiedenen Geheimzahlen und Feldern auf dem Magnetstreifen besteht. Zur Verschlüsselung wird ein Triple DES benutzt, dessen Schlüssel im gesamten System konstant ist und der daher sehr sorgfältig geheimgehalten werden muß; er ist nur den Computern der Clearingstellen bekannt.

Geldautomaten oder *point of sale* Terminals müssen daher sowohl die eingetippte Geheimzahl als auch die Information auf dem Magnetstreifen an so eine Clearingstelle übermitteln; dort wird beides verglichen und die Zahlung entweder autorisiert oder auch nicht.

Die dabei verwendeten Terminals funktionieren so, daß ein Händler die übermittelte Kundendaten nicht zu Gesicht bekommt; allerdings ist natürlich denkbar, daß ein betrügerischer Händler Geräte so manipuliert, daß sie sowohl eine Kopie des Magnetstreifens als auch die eingetippte Geheimzahl in einer ihm zugänglichen Weise speichern. Mit diesen

Informationen kann er sich dann gegenüber Dritten als Karteninhaber ausgeben und beliebig über dessen Konto verfügen. Sicherer wäre ein Verfahren, das dem Händler zwar garantiert, daß er den legitimen Karteninhaber vor sich hat, bei dem er aber keine Chance hat, in betrügerischer Absicht an dessen Daten zu gelangen.

Andere nichtklassische Anwendungen der Kryptologie sind etwa das Werfen von Münzen (für eine zufällige Entscheidung „Kopf oder Zahl“) via Telephon oder auch ein Pokerspiel per Internet. Die dafür eingesetzten Protokolle können durchaus auch ernste Anwendungen haben, beispielsweise beim verteilten Rechnen, wenn die Zuweisung von Aufgaben an die einzelnen Rechner nach einem Zufallsverfahren erfolgt. Falls die Kosten für die Inanspruchnahme der verschiedenen Rechner von verschiedenen Personen getragen werden, sollten diese definitiv daran interessiert sein, daß niemand dem Zufall auf ihre Kosten nachhilft.

§ 1: Werfen einer Münze per Telephon

Beim Werfen einer Münze geht es darum, daß zwei Partner A und B eine Entscheidung herbeiführen, die für beide als zufällig erkennbar ist. Bei einer Telephonverbindung ohne Videokanal sind echte Münzen natürlich nutzlos.

Die Zahlentheorie liefert eine praktikable Alternative: A wählt zwei große Primzahlen p und q und schickt deren Produkt $N = pq$ an B. Dieser wählt eine Zufallszahl x zwischen $\sqrt{N} + 1$ und $N - 1 - \sqrt{N}$ und schickt $y = x^2$ an A.

Da y modulo N ein Quadrat ist, ist es erst recht ein Quadrat modulo p und modulo q. Der wesentliche Punkt ist nun, daß sich Quadratwurzeln modulo einer Primzahl leicht berechnen lassen: Im einfachsten Fall, wenn $p \equiv 3 \bmod 4$ ist gilt für ein Quadrat $y \equiv x^2 \bmod p$ nach dem kleinen Satz von FERMAT die Gleichung

$$\left(y^{\frac{p+1}{4}}\right)^2 \equiv y^{\frac{p+1}{2}} \equiv x^{p+1} = x^p \cdot x \equiv x \cdot x \equiv y \bmod p\,,$$

die beiden Quadratwurzeln von y modulo p sind also $\pm y^{\frac{p+1}{4}} \bmod p$.

Für Primzahlen $p \equiv 1 \bmod 4$ ist die Berechnung der Quadratwurzel etwas aufwendiger, aber wie bereits in Zusammenhang mit dem quadratischen Sieb in Kap. 4, §8*b)* diskutiert, gibt es entsprechende Algorithmen. Falls A damit nicht vertraut ist, kann er sich natürlich auch einfach auf Primzahlen $p, q \equiv 3 \bmod 4$ beschränken.

A berechnet nun die Quadratwurzeln $\pm w_1, \pm w_2$ von y modulo p und modulo q; zwei davon setzt er nach dem chinesischen Restesatz zusammen zu einer Quadratwurzel w modulo N. Dabei hat er *vier* Möglichkeiten: Je nach Wahl der Vorzeichen bekommt er entweder den (ihm unbekannten) Wert x oder dessen Negatives, oder aber einen neuen Wert $\pm u$, der modulo der einen Primzahl kongruent x, modulo der anderen aber kongruent $-x$ ist. Er muß sich für eine dieser vier Möglichkeiten entscheiden und schickt die betreffende Zahl an B. Diese Entscheidung von A simuliert den Münzwurf.

Falls die geschickte Zahl gleich $\pm x$ ist, erhält B keine neuen Informationen und hat verloren; dies geschieht offenbar in 50% aller Fälle.

In den übrigen 50% der Fälle schickt A eine Zahl u, die modulo genau einer der beiden Primzahlen kongruent x ist. Somit ist ggT$(x-u, N)$ gleich dieser Primzahl, und auf diese Weise kann B die Zahl N faktorisieren. In diesem Fall hat B gewonnen und schickt zum Beweis einen der beiden Primfaktoren an A.

Falls p und q hinreichend groß und verschieden sind, hat B keine realistische Möglichkeit, N auf andere Weise zu faktorisieren, insbesondere nicht in den wenigen Sekunden, mit denen er bei korrekter Durchführung des Protokolls auskommen muß. Auch sonst kann er den Ausgang nicht zu seinen Gunsten zu beeinflussen: Er könnte zwar versuchen, zwei Zahlen x und $u \neq \pm x$ mit gleichem Quadrat zu finden und eine davon an A schicken, aber wie wir bei der Diskussion des quadratischen Siebs gesehen haben, ist genau das die derzeit effizienteste Methode zur Faktorisierung von N und damit nicht leichter als diese Faktorisierung.

Auch A hat keine Möglichkeit, das Ergebnis zu seinen Gunsten zu beeinflussen, denn er kann zwar alle vier Quadratwurzeln von y modulo N berechnen, weiß aber nicht, welche davon die Zahl x ist, deren Quadrat ihm B übermittelte.

§2: Poker per Telephon

Poker ist ein Kartenspiel, das traditionellerweise in verrauchten Hinterzimmern von Restaurants gespielt wurde, wobei meist auch viel Alkohol im Spiel war. Der moderne Internetuser allerdings möchte sich nicht einen ganzen Pokerabend lang von seinem Computer trennen und muß daher einen anderen Weg finden. Das kryptographische Problem besteht darin, daß bei einem traditionellen Pokerspiel die Karten jeweils gemischt, abgehoben und verteilt werden müssen und am Ende niemand die Karten seiner Mitspieler kennen darf.

Der Ansatz ist ähnlich wie bei den blinden Unterschriften, die für elektronisches Bargeld benutzt werden, allerdings wird statt RSA ein einfacheres Verfahren verwendet, das Verfahren von POHLIG und HELLMANN. Es funktioniert so ähnlich wie RSA, jedoch wird anstelle des Produkts zweier Primzahlen nur eine Primzahl p als Modul verwendet. Zur Verschlüsselung dient ein zu $p-1$ teilerfremder Exponent e, mit dem eine Nachricht m verschlüsselt wird als $m^e \bmod p$. Zur Entschlüsselung dient ein zweiter Exponent d mit der Eigenschaft, daß $de \equiv 1 \bmod p-1$ ist, denn nach dem kleinen Satz von FERMAT ist dann $m^{de} \equiv m \bmod p$. Ähnlich wie bei RSA kann d mit dem erweiterten EUKLIDischen Algorithmus (angewandt auf e und $p-1$) bestimmt werden.

Die Berechnung von d kann jeder ausführen, der die zur Anwendung des Algorithmus notwendigen Zahlen p und e kennt; der Algorithmus von POHLIG und HELLMANN ist also kein asymmetrisches Verfahren, sondern ein symmetrisches Kryptoverfahren, dessen Schlüssel geheim bleiben muß. Man kann wahlweise das Paar (p, e) als Schlüssel betrachten oder aber die Primzahl p innerhalb eines Netzwerks ein für alle man fest wählen und öffentlich bekanntgeben und dann nur den Exponenten e als geheimen Schlüssel betrachten.

Auf den ersten Blick vereint das Verfahren von POHLIG und HELLMANN alle Nachteile der symmetrischen und der asymmetrischen Kryptographie: Wie bei allen symmetrischen Verfahren hat man das Problem des Schlüsselaustauschs, und das bei einem Rechenaufwand, der dem des RSA-Verfahrens entspricht!

Tatsächlich muß die Sicherheit des Verfahrens von POHLIG/HELLMANN nach völlig anderen Kriterien beurteilt werden als die des RSA-Verfahrens: Die empfohlene Schlüssellänge von 2048 Bit bei RSA erklärt sich aus dem Stand und dem zu erwartenden Fortschritt bei Faktorisierungsalgorithmen; diese aber spielen für die Sicherheit von POHLIG/HELLMANN keinerlei Rolle. Hier muß ein Angreifer versuchen, den bei RSA öffentlich bekannten Exponenten e zu ermitteln; falls er die möglichen Exponenten einfach durchprobiert, ist er ungefähr in derselben Situation wie bei einem Angriff auf DES oder AES, so daß man vielleicht argumentieren könnte, daß ungefähr dieselben Sicherheitsparameter wie für Algorithmen dieser Art gewählt werden sollten, d.h. nach heutigem Stand mindestens 128 Bit für Primzahl und Exponent.

Dies ist aber zu optimistisch: Wie wir schon bei der Diskussion der Sicherheit von DES gesehen haben, müssen sich bei einer guten Blockchiffre die Transformationen wie eine Zufallsauswahl aus der vollen Permutationsgruppe über der Menge aller möglicher Blöcke verhalten; insbesondere dürfen sie keine zu kleine Untergruppe dieser symmetrischen Gruppe erzeugen.

Diese Bedingung ist hier klar verletzt: Die Transformationen von POHLIG und HELLMANN bilden sogar bereits eine (zyklische) Untergruppe der vollen Permutationsgruppe. Dies gibt einem Angreifer eine ganze Reihe zusätzlicher Möglichkeiten; insbesondere muß er bei einer Attacke mit bekanntem Klartext nur ein diskretes Logarithmenproblem lösen, wozu es, wie wir aus §4 von Kapitel fünf wissen, deutlich schnellere Algorithmen gibt als das vollständige Durchsuchen des Schlüsselraums.

Wenn wir trotzdem oft mit deutlich kürzeren Schlüssellängen arbeiten als bei Verfahren mit diskreten Logarithmen, rechtfertigt sich das vor allem aus der Art der Anwendungen: Das Verfahren von POHLIG und HELLMANN wird in erster Linie eingesetzt für Protokolle, die in Echtzeit ablaufen; falls man dazu dann auch noch *ad hoc*-Schlüssel einsetzt, nützt eine Kryptanalyse dem Gegner nur dann, wenn er sie innerhalb weniger Sekunden oder höchstens Minuten durchführen kann. In solchen Situationen sind die Sicherheitsanforderungen natürlich erheblich geringer als etwa bei elektronischen Unterschriften, die oft jahrelang sicher sein müssen.

Speziell beim Kartenspiel per Telephon (oder Internet) wird das Verfahren folgendermaßen eingesetzt:

Die n Teilnehmer einigen sich auf eine Primzahl p, die hinreichend groß sein muß, daß niemand in der zur Verfügung stehenden Zeit einen nennenswerten Teil aller möglicher Exponenten durchprobieren kann. Eine Größenordnung von 100 Bit oder dreißig Dezimalstellen sollte dazu mehr als ausreichend sein. Außerdem wird jeder der n Spielkarten eine natürliche Zahl $1 < x_i < p-1$ zugeordnet. Dabei kann es um eine fortlaufende Nummerierung handeln oder aber auch um eine strukturierte, bei der beispielsweise Farbe und Wert der Karte in den Ziffern von x_i kodiert sind.

Sodann wählt jeder der r Spieler zwei Exponenten d_k, e_k derart, daß $d_k e_k \equiv 1 \bmod p-1$ ist; nach dem kleinen Satz von FERMAT ist also $\left(x^{e_k}\right)^{d_k} \equiv x \bmod p$ für alle natürlichen Zahlen x.

Vor dem Start des eigentlichen Spiels müssen die Karten gemischt werden. Üblicherweise ist dies Aufgabe eines der Spieler; die anderen sehen nur zu, daß alles seine Richtigkeit hat, und vielleicht hebt einer von ihnen auch noch ab.

Beim Spiel per Telephon kann niemand beim Mischen zusehen; deshalb werden *alle* Spieler daran beteiligt. Der Kartenstapel wird simuliert durch die Folge $(i, x_i)_{i=1,\dots,n}$ der Karten, wobei die erste Komponente eines jeden Paars die Position der Karte im Stapel angibt.

Jeder Spieler wählt eine Permutation π_k der Menge der Zahlen von eins bis n. Mit dieser mischt er den Stapel, wobei er gleichzeitig die Karten mit seinem Exponenten e_k verschlüsselt.

Der erste Spieler ersetzt also jedes Paar (i, x_i) auf dem Stapel durch $(\pi_1(i), x_i^{e_1} \bmod p)$, sortiert die entstehende Liste wieder nach ihren ersten Komponenten und gibt die so entstandene Folge $(i, y_i)_{i=1,\dots,n}$ weiter an den zweiten. Der ersetzt jedes Paar (i, y_i) durch $(\pi_2(i), y_i^{e_2} \bmod p)$, sortiert wieder nach der ersten Komponente *usw.*, bis jeder Spieler seine Permutation angewandt hat. Der „gemischte“ Stapel besteht also aus den Paaren $\left(\pi_r \circ \cdots \circ \pi_1(i),\ x_i^{e_1 \cdots e_r} \bmod p\right)$, sortiert nach ihren ersten Komponenten.

Zum Spielen müssen die Karten wieder entschlüsselt werden, allerdings so, daß nur der Empfänger den Klartext x_i erkennen kann. Wenn daher einer der Spieler eine Karte vom Stapel erhalten soll, nimmt sein rechter Nachbar die oberste Karte vom Stapel und entschlüsselt sie mit seinem Exponenten d_k. Dann reicht er sie weiter an *seinen* rechten Nachbarn, der seinen Exponenten d_{k+1} (oder d_1, falls $k = r$) anwendet *usw.*, bis die Karte ihren Empfänger erreicht hat. Nachdem auch dieser noch mit seinem d-Exponenten potenziert hat, wurde die Karte x_i zu

$$x_i^{e_1\cdots e_r d_1\cdots d_r} = x_i^{(e_1d_1)(e_2d_2)\cdots(e_rd_r)} \equiv x_i \bmod p\,,$$

ist also wieder erkennbar.

Ausgespielt werden die Karten nun unverschlüsselt nach den üblichen Regeln des jeweiligen Kartenspiels; wer also die Karte x_i ausspielen möchte, schickt die Zahl x_i per E-Mail oder sonstige Software an alle Spielteilnehmer.

Im Gegensatz zur Situation bei einem realen Kartenspiel können diese nun allerdings nicht sicher sein, daß jeder nur Karten ausspielt, die ihm auch wirklich ausgeteilt wurden: Beim Skat etwa könnte der Alleinspieler während des Spiels unbemerkt eine „gedrückte“ Karte ausspielen. Um dies zu verhindern, werden nach Spielende alle Exponenten d_k, e_k bekanntgegeben, so daß sich jeder vergewissern kann, daß regelgerecht gespielt wurde.

Speziell beim Poker, wo Bluffen ein wesentlicher Teil des Spiels ist, sollte allerdings auch nach Spielende nicht bekannt werden, ob jemand wirklich Karten auf der Hand hatte, die soviel wert sind, wie er suggerierte. Mit etwas komplizierteren Verfahren kann auch hier die Einhaltung der Spielregeln kontrolliert werden.

§3: Zero Knowledge Protokolle

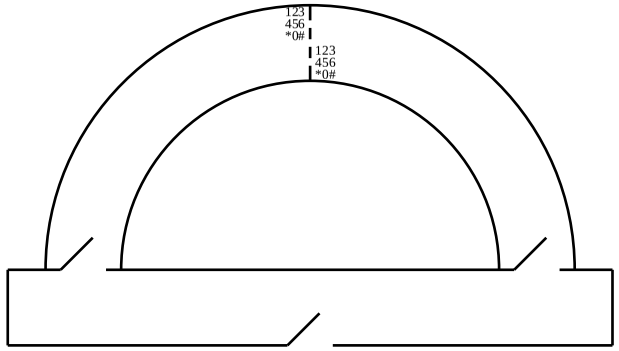
Wer heute mit einer Kreditkarte oder Bankkarte bezahlt, gibt dem Zahlungsempfänger recht viel Information in die Hand: Außer der Karten- oder Kontonummer sind auch sein Name, seine Bankverbindung und ähnliches auf der Karte kodiert. Darüber hinaus muß er beim Bezahlen entweder eine Unterschrift hinterlassen (die oft nicht allzu schwer

nachzumachen ist) oder eine PIN eintippen, die von einem manipulierten Gerät aufgezeichnet werden kann. Der Karteninhaber muß also ziemliches Vertrauen in den Zahlungsempfänger haben, kann dessen Vertrauenswürdigkeit aber oft nicht überprüfen: Schließlich hat nicht jedes von der Mafia geführte Restaurant ein Bronzeschild an der Tür mit der Aufschrift A PROUD MEMBER OF MOB ENTERPRISES CENTRAL EUROPE LTD.

Eine Alternative zur Kartenzahlung wäre das im Zusammenhang mit RSA behandelte elektronische Bargeld, jedoch scheint sich dieses zumindest derzeit kommerziell nicht durchzusetzen.

Eine andere Möglichkeit bestünde darin, daß der Karteninhaber keine PIN eintippt, sondern nur beweist, daß er die PIN kennt. Verfahren, bei denen jemand beweist, daß er über eine Information verfügt *ohne* irgendeinen Teil dieser Information preiszugeben, bezeichnet man in der Kryptologie als Zero Knowledge Protokolle oder, falls nur sehr wenig Information preisgegeben wird, als Minimal Disclosure Protokolle.

Zur Illustration der grundsätzlichen Idee betrachten wir zunächst eine durch ein Nummernschloß gesicherte Tür. Diese sei in einem runden Gang, dessen beide Enden durch Türen mit einem Vorraum verbunden sind:



Wenn B sieht, daß A durch die eine Tür in den halbkreisförmigen Gang eintritt und durch die andere herauskommt, kann er sicher sein, daß A

durch die mit Nummernschloß gesicherte Tür gegangen ist, erfährt aber nicht die dort einzugebende Geheimzahl.

Kann umgekehrt auch A sicher sein, daß B *überhaupt nichts* erfahren hat außer der Tatsache, daß A die Nummer kennt? Offensichtlich nicht, denn auf jeden Fall weiß B, daß A durch die Tür gegangen ist. Das ist nicht weiter schlimm, aber könnte B noch mehr erfahren haben?

Die Antwort auf diese Frage ist vor allem deshalb sehr schwierig, weil „etwas“ nur schwer zu definieren ist. Mit „nichts“ haben wir weniger Schwierigkeiten:

Angenommen, wir filmen den gesamten Ablauf der Verifikation. Falls dabei ein Film entsteht, den B auch ohne Teilnahme von A zusammen mit einem Statisten C auch drehen könnte, hat er offensichtlich nichts erfahren, was er nicht schon vorher wußte. Im vorliegenden Fall ist dieses Kriterium nicht erfüllt, denn C kann ohne Kenntnis der Geheimzahl den halbkreisförmigen Gang nicht durchqueren.

Mit einer kleinen Änderung des Protokolls geht das: Nun geht A zunächst allein in den Vorraum und verschwindet dort nach seiner Wahl hinter einer der beiden Türen. Danach erst kommt B mit seiner Kamera in den Vorraum und kann bestimmen, durch welche der beiden Türen A zu ihm kommen soll.

Damit kann B allerdings nicht wissen, ob A wirklich durch die gesicherte Tür gegangen ist: Falls er zu Beginn durch die Türe verschwunden ist, aus der ihn B kommen sehen möchte, kann er einfach herauskommen ohne seine Geheimzahl anwendenden zu müssen. B wird daher erst dann glauben, daß A diese wirklich kennt, wenn das Protokoll mehrfach wiederholt wird, und selbst dann gibt es immer noch ein Restrisiko von 2^{-n} bei n Versuchen.

A kann nun aber wirklich sicher sein, daß B keine neuen Informationen bekommen hat: Nun könnte B auch mit einem Statisten C ein Video drehen, in dem alles so abläuft wie beim echten Protokoll mit A: B muß nur C vorher informieren, durch welche Tür er kommen soll, oder aber er schneidet nachträglich alle Szenen heraus, in denen C durch die falsche Tür kommt.

In einer digitalen Version könnte man die Geheimzahl beispielsweise ersetzen durch die beiden Primteiler eines Produkts $N = pq$ zweier großer Primzahlen. A könnte die Kenntnis dieser Primteiler beweisen, indem er zu einer von B vorgegebenen Quadratzahl y modulo N eine Quadratwurzel produziert. Hierbei erfährt allerdings B, wie wir beim Münzwurf per Telephon gesehen haben, mit einer Wahrscheinlichkeit von 50% die beiden Primzahlen, was natürlich inakzeptabel ist.

Praktikabel ist dagegen die folgende Version, die SHAMIR (den wir vom RSA-Verfahren her kennen) und sein damaiger Doktorand AMOS FIAT 1986 vorgeschlagen haben: A wählt als Geheimzahl irgendein x mit $\sqrt{N} < x < N - \sqrt{N}$ und veröffentlicht $y = x^2 \bmod N$. Soll er nun gegenüber A nachweisen, daß er x kennt, erzeugt er zunächst eine nur für diesen einen Austausch gültige Zahl $\sqrt{N} < u < N - \sqrt{N}$ und schickt $v = u^2 \bmod N$ an B. Dieser kann nun entscheiden, ob er eine Quadratwurzel (modulo N) aus v oder aus yv sehen möchte. Falls er sich für v entscheidet, schickt ihm A entweder u oder $-u$, ansonsten $\pm xu$. A kann also beide Anfragen beantworten, braucht aber nur bei der zweiten Alternative seine Geheimzahl x.

Trotzdem sollte B stets zufällig zwischen seinen beiden Möglichkeiten wählen, denn wenn er sich stets oder überwiegend für die zweite entscheidet, könnte sich ein Betrüger C für A ausgeben, indem er y über den EUKLIDischen Algorithmus invertiert, eine Zufallszahl u erzeugt, und $v = y^{-1}u^2 \bmod N$ an B schickt. Verlangt B nun eine Wurzel aus yv, so kann er einfach u schicken. Würde B jedoch eine Wurzel aus v verlangen, müßte C passen, denn so etwas kann er nur dann berechnen, wenn er eine Wurzel aus y kennt. C kann sich also immer so vorbereiten, daß er *eine* der beiden möglichen Fragen von B korrekt beantworten kann; beide Fragen kann aber nur beantworten, wer eine Wurzel aus y modulo N kennt.

Die Faktorisierung von N spielt hier offensichtlich keinerlei Rolle, denn A muß nie Wurzeln modulo p oder modulo q ziehen. Daher muß N auch nicht unbedingt ein Produkt zweier Primzahlen sein, allerdings muß die Primzerlegung von N so schwierig sein, daß sie niemand finden kann, denn wer immer ein z finden kann mit $z^2 \equiv y \bmod N$ kann sich bei diesem Protokoll als A ausgeben.

Die Simulation mit einem Statisten ist auch hier wieder problemlos: Entweder der Statist erfährt vorher, welche Frage B stellen wird und kann sich darauf vorbereiten, oder aber alle Szenen, in denen er sich für die falsche Alternative entschieden hat, werden anschließend herausgeschnitten.

§4: Schlußbemerkung

Münzwurf oder Kartenspielen per Telephon gehören sicherlich nicht zu den praktisch relevantesten Anwendungen der Kryptologie, sie sind jedoch relativ elementare Beispiele aus einem Problemkreis, der durchaus auch ernstzunehmende Themen behandelt wie etwa das Rechnen mit verdeckten Daten:

Hinter Daten aus geologischen Explorationen oder Sensordaten von Satelliten steckt meist ein gewaltiger (auch finanzieller) Aufwand, so daß diese Daten einen großen Wert haben und nicht ohne Bezahlung an andere weitergegeben werden. Sie müssen allerdings oft auch mit sehr spezialisierten Verfahren ausgewertet und aufbereitet werden, und auch das kann nicht jeder.

Falls der Besitzer der Daten eine Auswertung wünscht, die er selbst nicht durchführen kann, braucht er also die Hilfe eines Spezialisten. Er möchte diesem jedoch nicht seine Daten anvertrauen, denn der Spezialist weiß schließlich, was man damit machen kann und wird sich möglicherweise über Strohmänner dann Schürfrechte, Optionen und ähnliches verschaffen. Umgekehrt möchte aber auch der Spezialist seine Programme nicht weitergeben, denn wer darüber verfügt, braucht ihn künftig nicht mehr.

Auch beim sogenannten *cloud computing* wäre ein Rechnen mit verschlüsselten Daten nützlich: Hier werden die einzelnen Schritte eines komplexen Algorithmus mehr oder weniger zufällig auf Rechner mit freier Kapazität verteilt in einem großen Cluster, das durchaus nicht nur vertrauenswürdige Computer enthalten muß.

Benötigt werden hierzu vollständig homomorphe Verschlüsselungsverfahren, d.h. Verschlüsselungsfunktionen φ, die nicht nur (wie etwa RSA und POHLIG/HELLMAN) mit der Multiplikation kompatibel sind, sondern

mit *allen* Rechenoperationen. Erste solche Verfahren gibt es, allerdings sind sie bislang so aufwendig, daß es kaum eine Rechnung geben dürfte, bei der sich ihr Einsatz lohnt. Die entsprechende Forschung steht aber noch ganz am Anfang; vielleicht wird es schon in naher Zukunft auch praktikable Verfahren geben.

Um zu sehen, daß Rechnen mit verdeckten Daten zumindest grundsätzlich möglich ist, betrachten wir ein extrem einfaches Beispiel: Angenommen, n Personen, die sich gegenseitig vertrauen, wollen ihr Durchschnittsgehalt berechnen, allerdings möchte (oder darf) keiner den anderen sein Gehalt nennen.

In diesem Fall reicht zur „Verschlüsselung“ der Daten die Addition einer zufälligen Zahl: Der erste wählt eine Zufallszahl z und addiert dazu sein Gehalt g_1; das Ergebnis $z_1 = z + g_1$ gibt er weiter an den zweiten. Dieser addiert sein Gehalt g_2 und gibt $z_2 = z_1 + g_2$ weiter an den dritten *usw.* Der letzte schließlich gibt $z_n = z_{n-1} + g_n$ weiter an den ersten. Da

$$z_n = z_{n-1} + g_n = z_{n-2} + g_{n-1} + g_n = \cdots = z + g_1 + \cdots + g_n$$

ist, kann dieser die Summe der Gehälter berechnen als $z_n - z$, und damit ist auch der Durchschnitt bekannt.

§5: Literatur

Kryptographische Protokolle werden zunehmend auch in Standardlehrbüchern behandelt; ein einfach lesbares relativ dünnes Buch, das sich ausschließlich darauf spezialisiert, ist

ALBRECHT BEUTELSPACHER, JÖRG SCHWENK, KLAUS-DIETER WOLFENSTETTER: *Moderne Verfahren der Kryptographie – Von RSA zu Zero-Knowledge*, Vieweg, ^{7}2010

Dort findet man auch einige Literaturhinweise zur ausführlicheren Beschäftigung mit weitergehenden Verfahren.