folgende Verschärfung einer etwa vierzig jahre älteren Ungleichung von TSCHEBYSCHEFF:

$$0{,}95695\frac{x}{\ln x} < \pi(x) < 1{,}04423\frac{x}{\ln x}\,,$$

$\pi(x)$ verhält sich also asymptotisch ungefähr wie $x/\ln x$. Nach dem Primzahlsatz, den GAUSS vermutete und den HADAMARD und DE LA VALLÉE POUSSIN 1896 bewiesen, ist die Asymptotik sogar exakt, d.h.

$$\lim_{x\to\infty}\frac{\pi(x)}{x/\ln x} = 1\,.$$

Numerisch besser ist die Approximation durch den Integrallogarithmus:

$$\pi(x) = \int_2^x \frac{d\xi}{\ln\xi} + O(xe^{-c\sqrt{\ln x}})$$

mit einer (im Prinzip berechenbaren) Konstanten $c > 0$.

Für uns wichtig ist die Folgerung: Unter den Zahlen der Größenordnung N haben der Primzahlen die Dichte $1/\ln N$, d.h. der Abstand zwischen zwei Primzahlen liegt im Durchschnitt bei etwa $\ln N$. Für $N = 10^n$ ist dies

$$\ln 10^n = n\ln 10 \approx 2{,}3n\,;$$

bei hundertstelligen Zahlen ist also etwa jede 230ste prim.

c) Das Sieb des Eratosthenes

Da ein FERMAT-Test für hundertfünfzigstellige Zahlen einiges an Zeit kostet, verzichtet man meist auf die Erzeugung wirklich zufälliger Primzahlen sondern nimmt stattdessen zu einer Zufallszahl n die kleinste Primzahl $p \geq n$. Damit sind nicht mehr alle Primzahlen gleich wahrscheinlich, sondern eine Primzahl wird mit umso größerer Wahrscheinlichkeit gewählt, je weiter sie von der nächstkleineren Primzahl entfernt ist. Bislang wurden keine Ansätze bekannt, wie ein Gegner diese Ungleichbehandlung der Primzahlen ausnutzen kann.

Diese Zahl p kann man wie folgt suchen: Man definiert ein Suchintervall $[n, n+a]$, in dem Primzahlen zu erwarten sind; beispielsweise könnte man $a \approx 5\ln n$ oder $a \approx 10\ln n$ setzen, so daß etwa fünf oder zehn Primzahlen zu erwarten sind.

In diesem Intervall sind alle geraden Zahlen uninteressant; diese können also gleich gestrichen werden. Sodann sind auch alle Dreierzahlen uninteressant; wir bestimmen also n mod 3; sodann kennen wir die erste auf n folgende Dreierzahl und streichen von dort ausgehend jede dritte Zahl. Genauso geht es weiter mit fünft, sieben und so weiter, bis zu einer Primzahl, die etwa die Gröse von a hat. (Bei größeren Primzahlen gibt es nur noch selten etwas zu streichen.) Die übriggebliebenen Zahlen haben dann zumindest keine kleinen Primteiler, und wir beschränken uns darauf, den FERMAT-Test für diese Zahlen durchzuführen.

§6: Weitere Aspekte zum RSA-Verfahren

Wir haben die Sicherheit des RSA-Verfahrens bislang nur unter dem Aspekt der Faktorisierung des Moduls N betrachtet. Tatsächlich muß man ein Kryptoverfahren aber immer als ganzes betrachten; viele im Prinzip gute Verfahren wurden in der Praxis schon durch unsachgemäße Anwendung angreifbar und konnten geknackt werden. In diesem Paragraphen sollen einige im Zusammenhang mit RSA wesentliche Aspekte diskutiert werden

a) Das Problem der Schlüsselübergabe

Der große Vorteil asymmetrischer Kryptoverfahren besteht darin, daß die beiden Teilnehmer keinen gemeinsamen Schlüssel zu vereinbaren brauchen; jeder verwendet einfach den öffentlichen Schlüssel des Empfängers. Wie bekommt er den aber?

Falls es einem Gegner gelingt, dem Absender einen selbst konstruierten Schlüssel als Schlüssel des Empfängers zu unterschieben, kann nur der *Gegner* die so verschlüsselte Nachricht lesen, nicht aber der intendierte Empfänger.

Für die Sicherheit von RSA ist daher von entscheidender Wichtigkeit, daß die benutzten Schlüssel die korrekten sind. Da E-Mails und Webseiten gefälscht werden können, empfiehlt sich bei unbekannten

Empfängern, den Schlüssel von eienr Zentrale zu holen, die Name und Schlüssel des Empfängers digital unterschreibt und dann übermittelt. Auf diese Weise muß nur der öffentliche Schlüssel der Zentrale allgmeinen bekannt sein, und der kann leicht über die verschiedensten elektronischen und klassischen Medien häufig bekanntgegeben werden. Die Regulierungsbehörde für Telekommunikation und Post zertifiziert in Deutschland solche Zentralen, sofern sie die Gewähr für eine sichere und zuverlässige Schlüsselübermittlung bieten.

b) Primzahlen sind Wegwerfartikel

Die Suche nach einer Primzahl mit 1024 Bit kann auf einem nicht sonderlich leistungsfähigen PC rund eine Minute dauern; wenn man viele Schlüssel erzeugen muß, bietet sich also an, mit den teuren Primzahlen sparsam umzugehen.

Genau das darf man aber aus Sicherheitsgründen auf keinen Fall tun: Wenn jemals zwei unterschiedliche Moduln M, N dieselbe Primzahl p enthalten, kann p leicht als ggT von M und N berechnet werden, so daß beide Moduln faktorisiert sind. Der EUKLIDische Algorithmus erfordert auch für 2048-Bit-Zahlen auf einem Standard-PC einen Aufwand, der höchstens im unteren Sekundenbereich liegt; es ist also durchaus möglich, auch bei einer Liste von mehreren Tausend Moduln für jedes Paar den ggT zu berechnen.

Der sichere Umgang mit Primzahlen besteht darin, die Primzahlen sofort nach Berechnung des öffentlichen und des privaten Schlüssels zu vernichten. Die Wahrscheinlichkeit, daß später wieder einmal eine dieser Primzahlen als Zufallsprimzahl auftaucht, liegt bei vernünftiger Implementierung des „Zufalls“ deutlich unter 10^{-100} und kann daher für alle praktischen Zwecke ignoriert werden.

c) Kenntnis des privaten Exponenten führt zur Faktorisierung

Zu zwei RSA-Moduln, die eine Primzahl gemeinsam haben, liefert der EUKLIDische Algorithmus diese Primzahl und damit die Faktorisierung der beiden Moduln. Falls die beiden Moduln allerdings gleich sind, läßt sich mit dem EUKLIDischen Algorithmus natürlich nichts anfangen-.

Innerhalb einer Organisation, in der RSA-Schlüssel zentral vergeben werden, könnte es sich daher anbieten, den Modul N für alle Teilnehmer gleich zu wählen und lediglich für jeden Teilnehmer einen eigenen öffentlichen Exponenten e und zugehörigen geheimen Exponenten d zu erzeugen.

Bei einem solchen System könnte dann natürlich die Zentrale alle Nachrichten aller Teilnehmer entschlüsseln, aber daß kann in gewissen Fällen durchaus erwünscht sein: Falls es sich etwa im Bankenbereich um Nachrichten handelt, die Geld transferieren, muß die Zentrale alles kontrollieren können.

Schlimm wäre allerdings, wenn auch einzelne Teilnehmer alles entschlüsseln könnten, denn dann wäre es möglich, Überweisungsaufträge mit falschem Namen zu unterschreiben, womit jede Art von Betrug möglich würde.

Wir wollen uns überlegen, wie ein Teilnehmer mit seinem öffentlichen und privaten Exponenten die Zahl N faktorisieren kann und damit die privaten Exponenten aller anderer Teilnehmer aus den öffentlichen Exponenten berechnen kann.

Dazu sei $N = pq$ Produkt zweier Primzahlen, e sei der öffentliche Exponent und d der private. Dann ist für alle $a \in \mathbb{Z}$

$$(a^e)^d = a^{ed} \equiv a \bmod N\,;$$

für ein zu N teilerfremdes a ist also

$$a^{de-1} \equiv 1 \bmod N\,.$$

Wir schreiben

$$de - 1 = 2^r \cdot u \quad \text{mit einer ungeraden Zahl } u$$

und betrachten die Folge der Zahlen

$$a^u \bmod N, \quad a^{2u} \bmod N, \quad \dots, \quad a^{2^r u} \bmod N\,.$$

Die letzte dieser Zahlen ist stets gleich eins; wenn wir Pech haben, ist für das gerade betrachtete a auch schon die erste gleich eins. Wenn nicht, gibt es einen kleinsten Exponenten s, so daß

$$a^{2^s u} \not\equiv 1 \bmod N \qquad \text{und} \qquad a^{2^{s+1} u} \equiv 1 \bmod N\,.$$

Es könnte sein, daß dann $a^{2^s u} \equiv -1 \bmod N$ ist, aber da N eine zusammengesetzte Zahl ist, muß das nicht der Fall sein: Modulo 15 ist beispielsweise auch vier eine Zahl mit Quadrat eins.

Modulo einer Primzahl hat eins natürlich nur die beiden Quadratwurzeln ± 1, denn die natürlichen Zahlen modulo einer Primzahl bilden einen Körper, und in einem Körper kann das quadratische Polynom $x^2 - 1$ höchstens zwei Nullstellen haben.

Ist aber $x^2 \equiv 1 \bmod N = pq$, so ist auch $x^2 \equiv 1 \bmod p$ und mod$q$, also $x \equiv \pm 1 \bmod p$ und $x \equiv \pm 1 \bmod q$, wobei nicht in beiden Fällen das gleiche Vorzeichen stehen muß. In der Hälfte aller Fälle werden beide Vorzeichen gleich sein; dann ist $x \equiv \pm 1 \bmod N$ mit demselben Vorzeichen.

Ist aber etwa $x \equiv 1 \bmod p$ und $x \equiv -1 \bmod q$, so ist

$$p = \mathrm{ggT}(x-1, N) \qquad \text{und} \qquad q = \mathrm{ggT}(x+1, N)\,,$$

sobald wir eine solche Zahl kennen, haben wir N also faktorisiert.

Wenn wir mit einem zufällig gewählten a so wie oben verfahren, werden wir in etwa der Hälfte aller Fälle eine von ± 1 verschiedene Quadratwurzel der Eins erhalten. Mit nur wenigen Werten für a bekommen wir fast sicher eine Faktorisierung von N.

d) Der chinesische Restesatz

Das Argument im vorigen Abschnitt kann auch zu einer schnelleren Durchführung der RSA-Entschlüsselung und damit zu schnelleren elektronischen Unterschriften führen: Die Berechnung von $m^d \bmod N$ besteht aus Quadrierungen und Multiplikationen modulo N, und deren Aufwand steigt quadratisch mit der Ziffernlänge von N. Kennt man also die Faktorisierung $N = pq$, kann man die beiden Werte $a^d \bmod p$ und $m^d \bmod q$ in jeweils einem Viertel der Zeit für $m^d \bmod N$ berechnen, beide zusammen also in der halben Zeit. Das Gesamtergebnis ist dadurch eindeutig bestimmt, denn da p und q teilerfremd sind, ist jede Zahl die modulo p und modulo q bekannt ist, auch modulo N eindeutig bestimmt.

Dies läßt sich leicht konstruktiv durchführen: Mit dem erweiterten Euklidischen Algorithmus findet man Zahlen α, β, so daß $\alpha p + \beta q = 1$ ist; dann ist

$$\beta q \equiv 1 \bmod p \qquad \text{und} \qquad \alpha p \equiv 1 \bmod q\,.$$

Für zwei beliebig vorgegebene Zahlen a, b ist entsprechend

$$a\beta q \equiv a \bmod p \qquad \text{und} \qquad b\alpha p \equiv b \bmod p$$

eine Lösung der Kongruenz

$$x \equiv a \bmod p \qquad \text{und} \qquad x \equiv b \pmod{q}\,.$$

Da α und β nur einmal für p und q berechnet werden müssen, ist der Aufwand für das Zusammensetzen der Restklassen modulo p und modulo q zu einer Restklasse modulo N sehr gering.

Mit diesem Verfahren läßt sich der Aufwand für eine elektronische Unterschrift also praktisch halbieren, was vor allem dann von Bedeutung ist, wenn die Unterschrift mit einer Smartcard geleistet wird.

Leider ist das Verfahren aber mit einem potentiell katastrophalen Risiko verbunden: Fall bei einer der beiden Rechnungen

$$a \leftarrow m^d \bmod p \qquad \text{und} \qquad b \leftarrow m^d \bmod q$$

ein Fehler auftritt, ist das Ergebnis für $m^d \bmod N$ korrekt modulo der einen, nicht aber modulo der anderen Primzahl. Nehmen wir etwa an, das Ergebnis modulo q sei falsch; dann wird also eine Unterschrift u berechnet, die modulo p kongruent ist zu m^d, nicht aber modulo q.

Zum Überprüfen der Unterschrift berechnet der Empfänger $u^e \bmod N$ und vergleicht dies mit m; im vorliegenden Beispiel stimmen die beiden Zahlen nur modulo p überein, nicht aber modulo q, sie sind also insbesondere verschieden, so daß die Unterschrift nicht akzeptiert wird.

Da $u^e - m$ aber durch p teilbar ist und nicht durch q, kann der Prüfer nun

$$p = \mathrm{ggT}(u^e - m, N)$$

berechnen und dadurch N faktorisieren; er kann also künftig die Unterschrift des anderen nachmachen.

Hardwarefehler, die bei der Berechnung von einem der beiden Teilergebnisse zu einem falschen Ergebnis führen sind sicherlich sehr seltene Ereignisse, allerdings kann man dem nachhelfen: Durch Magnetfelder, Mikrowelle, Hitze, mechanische Belastung und ähnliches kann man die Karte durchaus so beeinflussen, daß Fehler wahrscheinlich, aber nicht zu wahrscheinlich werden, so daß man eine gute Chance hat, daß genau eines der beiden Zwischenergebnisse falsch berechnet wird.

Sicherer ist es also auf jeden Fall, wenn man trotz des rund doppelt so hohen Aufwands direkt modulo N rechnet; die oben aufgestellte Regel, wonach man die Primzahlen so schnell wie möglich vergessen sollte, hat durchaus ihren Sinn.

Die Methode, eine Zahl modulo N aus ihren Restklassen modulo von Teilern von N zu bestimmen, hat aber noch viele andere Anwendungen, so daß wir den Satz hier allgemein formulieren wollen. Er wurde angeblich früher von chinesischen Generälen benutzt, um Truppenstärken zu berechnen, indem sie die Soldaten in Reihen verschiedener Breite antreten ließen und dann die Anzahl der Soldaten in der letzten Reihe zählten.

Chinesischer Restesatz: Die natürlichen Zahlen $d_1, \ldots, d_r$ seien paarweise teilerfremd und $N = d_1 \cdots d_r$ sei ihr Produkt. Dann hat das Gleichungssystem

$$x \equiv a_1 \bmod d_1, \quad \ldots, \quad x \equiv a_r \bmod d_r$$

für jede Wahl der a_i eine modulo N eindeutig bestimmte Lösung; diese kann mit Hilfe des erweiterten EUKLIDischen Algorithmus berechnet werden.

Beweis: Für nur zwei Zahlen $d_1 = p$ und $d_2 = q$ haben wir das oben nachgerechnet, wobei offensichtlich nur eine Rolle spielte, daß p und q teilerfremd sind, nicht aber, daß sie Primzahlen sind. Der allgemeine Fall folgt durch vollständige Induktion, denn auch $d_1 \cdots d_s$ und $d_1 \cdots d_s d_{s+1}$ sind teilerfremd. ■

e) Kleine öffentliche Exponenten

Da der Verschlüsselungsaufwand bei RSA linear in der Ziffernzahl des Exponenten steigt, wird gelegentlich vorgeschlagen, einen kleinen Verschlüsselungsexponenten e zu verwenden; populär sind etwa $e = 3$ und $e = 2^{16} + 1$.

Im allgemeinen ist das unproblematisch, insbesondere wenn es um den öffentlichen Exponenten zur Überprüfung einer digitalen Unterschrift geht, in einem Fall kann es allerdings Probleme geben: Wenn nämlich dieselbe Nachricht (oder derselbe Nachrichtenteil, wie z.B. eine Anlage oder ein Block ASCII-Kunst am Ende) an mehrere Empfänger geht.

Nehmen wir etwa an, die Nachricht m solle an drei Empfänger geschickt werden, deren öffentliche Schlüssel $(N_1, 3)$, $(N_2, 3)$ und $(N_3, 3)$ seien. Verschickt werden also die drei Blöcke

$$m^3 \bmod N_1, \quad m^3 \bmod N_2 \qquad \text{und} \qquad m^3 \bmod N_3 .$$

Ein Gegner, der alle drei abfängt, kann dann nach dem chinesischen Restesatz

$$m^3 \bmod N_1 N_2 N_3$$

berechnen, und da m kleiner als jedes N_i sein muß, kennt er somit m^3. Die Berechnung der Kubikwurzel auch einer sehr großen Zahl ist vom Aufwand her mit einer Division vergleichbar, liegt also höchstens im unteren Sekundenbereich.

(Falls N_1, N_2, N_3 nicht paarweise teilerfremd sein sollten, merkt man das bei der Anwendung des erweiterten EUKLIDischen Algorithmus und hat dann sogar eine Faktorisierung von mindestens zwei Moduln, was dann über die privaten Exponenten insbesondere auf die Nachricht führt.)

f) Kleine private Exponenten

Da elektronische Unterschriften häufig mit Smartcards oder in Zukunft vielleicht auch Mobiltelephonen und ähnlichen Geräte mit vergleichsweise schwacher Rechenleistung erzeugt werden, bietet sich an, nicht den öffentlichen, sondern den privaten Exponenten möglichst klein zu wählen. Ein privater Exponent drei wäre natürlich unmöglich, denn der private Exponent ist schließlich geheim und darf nicht durch systematisches Durchprobieren kleiner Zahlen gefunden werden.

Systematisches Durchprobieren ist aber, wie wir bei der Diskussion symmetrischer Kryptoverfahren gesehen haben, mit heutiger Technologie nur bis zu etwa 2^{64} Fällen möglich; 2^{128} Möglichkeiten, wie sie etwa der AES bietet, gelten heute als sicher. Ein privater Exponent mit 128 statt 1024 Bit führt zu einer Reduktion des Rechenaufwands um den Faktor acht, was gerade bei Smartcards spürbar sein sollte.

Leider gilt aber auch hier, wie in so vielen Fällen bei RSA, das Rechenerleichterungen zu Sicherheitsmängeln führen; ein privater Exponent mit 128 Bit würde bei einem Modul von 1024 oder 2048 Bit innerhalb von Sekunden zu dessen Primfaktorzerlegung führen.

Der Grund ist folgender: Der öffentliche Exponent e und der private Exponent d erfüllen die Gleichung

$$de - k(p-1)(q-1) = 1$$

mit einer natürlichen Zahl k. Division durch $d(p-1)(q-1)$ macht daraus

$$\frac{e}{(p-1)(q-1)} - \frac{k}{d} = \frac{1}{d(p-1)(q-1)} .$$

Der linke Bruch hat einen Nenner, der ungefähr die gleiche Größenprdnung hat wie der Moduk $N = pq$; davon subtrahiert wird ein Bruch mit Nenner d, und die rechte Seite der Gleichung sagt uns, daß die Differenz sehr klein ist.

Ist also der private Exponent d klein, so kann der linksstehende Bruch durch einen Bruch mit sehr viel kleinerem Nenner sehr gut approximiert werden. Damit kann ein Gegner noch nichts anfangen, denn er kennt den Nenner $(p-1)(q-1)$ nicht; andererseits kennt er $N = pq$, und die

Differenz ändert sich nicht sehr, falls man den Nenner durch N ersetzt:

$$\begin{aligned}\left|\frac{e}{N} - \frac{k}{d}\right| &= \left|\frac{e}{N} - \frac{e}{(p-1)(q-1)} + \frac{e}{(p-1)(q-1)} - \frac{k}{d}\right| \\ &\le \left|\frac{e(p-1)(q-1) - epq}{N(p-1)(q-1)}\right| + \frac{1}{d(p-1)(q-1)} \\ &= \frac{e(p+q-1)}{N(p-1)(q-1)} + \frac{1}{d(p-1)(q-1)} .\end{aligned}$$

Da p und q in der Größenordnung von $\sqrt{N}$ liegen, ist auch das noch eine recht kleine Zahl.

Bei kleinem d kann sich das ein Gegner mittels des folgenden Satzes zunutze machen:

Satz: Für die reelle Zahl $x > 0$ gebe es teilerfremde natürliche Zahlen a, b derart, daß

$$\left|x - \frac{a}{b}\right| < \frac{1}{2b^2} .$$

Dann ist a/b eine Konvergente der Kettenbruchentwicklung von x.

Um damit etwas anfangen zu können, müssen wir zunächst wissen, was die Kettenbruchentwicklung einer reellen Zahl ist. Diese berechnet sich nach folgendem Algorithmus:

1. Schritt: Setze $x_0 = x$ und $a_0 = [x]$.

n-ter Schritt, $n \ge 1$: Falls $x_{n-1} = a_{n-1}$ bricht der Algorithmus an dieser Stelle ab; andernfalls setze

$$x_n = \frac{1}{x_{n-1} - a_{n-1}} \qquad \text{und} \qquad a_n = [x_n] .$$

Die n-te Konvergente dieser Kettenbruchentwicklung ist die rationale Zahl

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{\cdots}{a_{n-1} + \cfrac{1}{a_n}}}}} .$$

Der Beweis des obigen Satzes ist elementar, aber langwierig; wir wollen daher darauf verzichten.

Falls man diesen Satz anwenden kann, läßt sich d also bestimmen, indem man die Nenner der Konvergenten der Kettenbruchentwicklung von e/N bestimmt und jeweils durch Ausprobieren nachprüft, ob für einen Zufallsblock a die gewünschte Beziehung $a^{de} \equiv a \bmod N$ gilt.

Eine einfache Abschätzung zeigt, daß er für p und q von etwa gleicher Größe anwendbar ist, sofern d höchstens die Größenordnung von etwa $\sqrt[4]{N}$ hat; neuere, etwas aufwendigere Untersuchungen zeigen, daß auch man d auch noch für $d < N^{0,289}$ rekonstruieren kann; Fachleute erwarten, daß möglicherweise sogar alle $d < \sqrt{N}$ unsicher sind.

Private Exponenten müssen also immer groß sein. Falls man von einem vorgegebenen öffentlichen Exponenten ausgeht, ist das für realistische N mit an Sicherheit grenzender Wahrscheinlichkeit erfüllt; Vorsicht ist nur geboten, wenn man mit dem privaten Exponenten startet.

h) Blinde Unterschriften und digitales Bargeld

Einer der erfolgversprechendsten Ansätze zum Aushebeln eines Kryptosystems besteht darin, sich auf die Dummheit seiner Mitmenschen zu verlassen.

So sollte es durch gutes Zureden nicht schwer sein, jemanden zu Demonstrationszwecken zum Unterschreiben einer sinnlosen Nachricht zu bewegen: Eine Folge von Nullen und Einsen ohne sinnvolle Interpretation hat schließlich keine rechtliche Wirkung.

Nun muß eine sinnlose Nachricht aber nicht unbedingt eine Zufallszahl sein: Sie kann sorgfältig präpariert sein. Sei dazu etwa m eine Nachricht, die ein Zahlungsversprechen enthält, (N, e) der öffentliche Schlüssel des Opfers und r eine Zufallszahl zwischen 2 und $N - 2$. Dann wird

$$x = m \cdot r^e \bmod N$$

wie eine Zufallsfolge aussehen, für die man eine Unterschrift

$$u = x^d \bmod N = (mr^e)^d \bmod N = m^d r \bmod N$$

bekommt. Multiplikation mit r^{-1} macht daraus eine Unterschrift unter die Zahlungsverpflichtung m.

Das angegebene Verfahren kann nicht nur von Trickbetrügern benutzt werden; blinde Unterschriften sind auch die Grundlage von *digitalem Bargeld.*

Zahlungen im Internet erfolgen meist über Kreditkarten; die Kreditkartengesellschaften haben also einen recht guten Überblick über die Ausgaben ihrer Kunden und machen teilweise auch recht gute Geschäfte mit Kundenprofilen.

Digitales Bargeld will die Anonymität von Geldscheinen mit elektronischer Übertragbarkeit kombinieren und so ein anonymes Zahlungssystem z.B. für das Internet bieten.

Es wir ausgegeben von einer Bank, die für jede angebotene Stückelung einen öffentlichen Schlüssel (N, e) bekanntgibt. Eine Banknote ist eine mit dem zugehörigen geheimen Schlüssel unterschriebene Seriennummer.

Die Seriennummer kann natürlich nicht einfach *jede* Zahl sein; sonst wäre jede Zahl kleiner N eine Banknote. Andererseit dürfen die Seriennummern aber auch nicht von der Bank vergeben werden, denn sonst wüßte diese, welcher Kunde Scheine mit welchem Seriennummern hat. Als Ausweg wählt man Seriennummern einer sehr speziellen Form: Ist $N > 10^{150}$, kann man etwa als Seriennummer eine 150-stellige Zahl wählen, deren Ziffern spiegelsymmetrisch zur Mitte sind, d.h. ab der 76. Ziffer werden die vorherigen Ziffern rückwärts wiederholt. Die Wahrscheinlichkeit, daß eine zufällige Zahl x nach Anwendung des öffentlichen Exponenten auf so eine Zahl führt, ist 10^{-75} und damit vernachlässigbar.

Seriennummern werden von den Kunden zufällig erzeugt. Für jede solche Seriennummer m erzeugt der Kunde eine Zufallszahl r, schickt $mr^e \bmod N$ an die Bank und erhält (nach Belastung seines Kontos) eine Unterschrift u für diese Nachricht zurück. Wie oben berechnet er daraus durch Multiplikation mit r^{-1} die Unterschrift $v = m^d \bmod N$ für die Seriennummer N, und mit diesem Block kann er bezahlen.

Der Zahlungsempfänger berechnet $v^e \bmod N$; falls dies die Form einer gültigen Seriennummer hat, kann er sicher sein, einen von der Bank unterschriebenen Geldschein vor sich zu haben. Er kann allerdings noch nicht sicher sein, daß dieser Geldschein nicht schon einmal ausgegeben wurde.

Deshalb muß er die Seriennummer an die Bank melden, die mit ihrer Datenbank bereits ausbezahlter Seriennummern vergleicht. Falls sie darin noch nicht vorkommt, wird sie eingetragen und der Händler bekommt sein Geld; andernfalls verweigert sie die Zahlung.

Bei 10^{75} möglichen Nummern ist die Wahrscheinlichkeit dafür, daß zwei Kunden dieselbe Nummer erzeugen vernachlässigbar klein, es ist aber nicht völlig auszuschließen. Das System kann nur funktionieren, wenn in diesem Fall der zweite Geldschein mit derselben Seriennummer nicht anerkannt wird, so daß der zweite Kunde sein Geld verliert. Dies muß als eine zusätzliche Gebühr gesehen werden, die zwar praktisch nie fällig wird, aber eben doch nie ausgeschlossen werden kann. Da digitales Bargeld in kleinen Stückelungen ausgegeben wird, kann man damit leben; bei Geldscheinen im Millionenwert gäbe es hier ein Problem.