KES – Zeitschrift für Kommunikations- und EDV-Sicherheit 7/5 (1991) 321-326.

SICHERHEIT VON HERKÖMMLICHEN UND FAIL-STOP-SIGNATUREN

von Birgit Pfitzmann, Institut für Informatik, Universität Hildesheim

Es wird ein informeller Überblick über die Eigenschaften von herkömmlichen digitalen Signatursystemen gegeben, insbesondere deren Vor- und Nachteile bzgl. Sicherheit. Anschließend werden die Eigenschaften der neuartigen, sichereren Fail-stop-Signaturen dagegengestellt.

Dipl.-Inform. Birgit Pfitzmann ist wissenschaftliche Mitarbeiterin am Institut für Informatik der Universität Hildesheim. Ihre Hauptinteressen sind Sicherheit und Datenschutz in öffentlichen Kommunikationsnetzen und Zahlungssystemen, mathematische Kryptographie, Softwareimplementierungen von Kryptosystemen und fehlertolerante verteilte Algorithmen.

Einordnung

Dieser Artikel nimmt Bezug auf einen Abschnitt über Fail-stop-Signaturen im Bericht über die Securicom 1991, KES 91/3, der in seiner Kürze evtl. einen falschen Eindruck, vor allem über Vor- und Nachteile herkömmlicher Signaturen, erweckte. (Nicht der Empfänger, sondern der Unterzeichner trägt bei sinnvoller Verwendung das Risiko, und auch dieser ist nur unter bestimmten Voraussetzungen „hilflos ausgeliefert".)

Damit die Begriffe klar sind, vorweg eine kurze Erklärung der gebräuchlichsten Verwendung: Die meisten kryptographischen Systeme zur Datensicherung gehören zu genau einer der zwei folgenden Klassen:

- *Konzelationssysteme*, d.h. Systeme zur Geheimhaltung von Nachrichteninhalten.
- *Authentikationssysteme*, d.h. Systeme zur Sicherung der Unverfälschtheit von Nachrichteninhalten und des Absenders.

(„Kryptosystem" meint meist kryptographisches System, manchmal Konzelationssystem, „Verschlüsselungssystem" meist Konzelationssystem. Man darf sich auch nicht davon verwirren lassen, daß Grundfunktionen wie RSA und DES zu Konzelation und Authentikation dienen können: Sie müssen für die beiden Fälle im Gesamtsystem verschieden eingesetzt werden.)

Innerhalb jeder der zwei Klassen unterscheidet man nach der Schlüsselverteilung

- *symmetrische*, bei denen Sender und Empfänger einer Nachricht denselben (geheimen) Schlüssel haben müssen, und
- *asymmetrische*, bei denen Sender und Empfänger verschiedene Schlüssel haben, von denen einer geheim, der andere öffentlich ist.

Digitale Signatursysteme

Digitale Signatursysteme, herkömmliche wie Fail-stop-, sind asymmetrische Authentikationssysteme. Hier ist es der Sender, Unterzeichner genannt, der seinen Schlüssel, Signierschlüssel S genannt, geheimhalten muß. Alle anderen haben nur einen öffentlichen Testschlüssel T. Diesen hat der Unterzeichner zusammen mit S erzeugt und dann veröffentlicht; man sollte mit T aber nicht signieren können. Wer eine signierte Nachricht erhält (d.h. eine, die den Test mit dem Testschlüssel besteht), ist dann sicher, daß nur der Besitzer des Signierschlüssels sie erzeugen konnte.

(Schlüsselverteilung ist auch bei digitalen Signaturen nicht trivial, aber einfacher als bei symmetrischen Systemen: Am besten gibt der Unterzeichner sein T einer Zentrale. Diese muß unterschreiben, welches T sie verteilen soll. Wer T bei der Zentrale erfragt, läßt sich die Antwort ebenfalls unterschreiben. Somit ist die Richtigkeit der Verteilung überprüfbar.)

Eine digitale Signatur kann also für digitale Nachrichten das leisten, was die eigenhändige Unterschrift im derzeitigen Rechtsverkehr leisten soll: Sie garantiert, daß der angebliche Sender einer Nachricht auch wirklich der Sender genau dieser Nachricht ist (bzw. sie autorisiert hat). Der Empfänger kann dies auch Dritten nachweisen, z.B. im Streitfall einem Gericht: Auch dieses kennt ja den öffentlichen Testschlüssel T und weiß, daß nur dessen Besitzer der Sender der Nachricht sein kann.

(Anders ist es bei symmetrischen Authentikationssystemen, z.B. mit DES (vgl. z.B. [4]): Dort haben Sender und Empfänger einen gemeinsamen Schlüssel, den sie ansonsten möglichst geheimhalten. Nun kann jeder dem anderen authentisierte Nachrichten schicken. Selbst wenn aber ein Gericht wüßte, welchen Schlüssel sie verwen-

den, kann es nicht entscheiden, wer von beiden eine Nachricht erzeugte.)

Zu beachten ist auch, daß eine digitale Signatur stark nachrichtenabhängig sein muß, so daß, sobald jemand an einer signierten Nachricht auch nur ein bißchen ändert, die Signatur nicht mehr paßt.

Problem

Nicht beweisbar

Das einzige Problem mit herkömmlichen digitalen Signatursystemen wie RSA [11] und ElGamal [5] ist, daß es in einem strengen Sinne keine beweisbar sicheren gibt. Bei allen derzeit bekannten Signatursystemen muß man eine „kryptographische Annahme“ machen, z.B. die „Faktorisierungsannahme“, daß das Zerlegen von großen Zahlen in ihre Primfaktoren schwierig sei, wenn auch die Primfaktoren groß sind. Im Prinzip ist aber die Primfaktorzerlegung eindeutig durchführbar. Die Sicherheit beruht also nur darauf, daß ein Möchtegernfälscher nicht genug Zeit und keinen hinreichend guten Algorithmus hat, die Faktoren zu finden. (Gleiches gilt für die „Diskreter-Logarithmus-Annahme“ des ElGamal-Systems.)

Zudem ist bei RSA und ElGamal nicht einmal klar, ob sie unter der jeweiligen Annahme sicher sind. Insbesondere ist die Sicherheit schneller Hashfunktionen, wie sie oft verwendet werden, um das Signieren zu beschleunigen, noch wesentlich unklarer als die der Systeme selbst.

Allerdings gibt es hier eine Verbesserung: Das Signatursystem GMR ist wenigstens unter der Faktorisierungsannahme bewiesenermaßen sicher [7, 6].

Bewertung kryptographischer Annahmen

Natürlich ist zu hoffen, daß diese kryptographischen Annahmen gelten. Sie sind umso vertrauenswürdiger, je intensiver sie untersucht wurden, d.h. Wissenschaftler vergeblich versuchten, schnellere Algorithmen zu finden, um sie zu brechen. Insbesondere die Faktorisierungsannahme, aber auch die Diskreter-Logarithmus-Annahme, interessiert Mathematiker sehr, so daß viele Untersuchungen stattfinden. (In dieser Hinsicht ähneln kryptographische Systeme Geigen: Wenn sie überhaupt noch heil sind, dann sind die ältesten die besten. Allerdings muß man wegen steigender Rechenleistung und verbesserten Brechalgorithmen genügend oft den Sicherheitsparameter erhöhen, also z.B. die Größe der benutzten Zahlen, die man nicht faktorisieren können soll.)

Auswirkung im Fälschungsfall

Wenn ein digitales Signatursystem aber doch gebrochen wird, ist der angebliche Unterzeichner einer Nachricht mit einer gefälschten Unterschrift schutzlos: Die gefälschte digitale Signatur sieht ganz genau wie die echte aus. Wird sie einem Gericht vorgelegt, so entscheidet dies (immer noch mit dem öffentlichen Testschlüssel T), daß es sich um eine gültige Unterschrift handelt. Damit wird der angebliche Unterzeichner haften müssen. (Der Empfänger einer signierten Nachricht ist hingegen sicher: Sobald er geprüft hat, daß die Signatur den Test mit dem Testschlüssel besteht, weiß er, daß sie das auch vor Gericht tut, sei sie nun echt oder gefälscht.)

Dies ist jedenfalls die technische Sicht dessen, was ein Gericht tun sollte. Ein wirkliches Gericht ist natürlich nicht verpflichtet, sich daran zu halten. (Auch eine eigenhändige Unterschrift muß ein deutsches Gericht nicht unbedingt anerkennen!) Es könnte also den Beteuerungen des angeblichen Unterzeichners, die Unterschrift sei gefälscht, glauben, zumal dieser anführen kann, daß das Signatursystem nur auf einer unbewiesenen Annahme beruht, wie lang auch immer diese untersucht ist.

Wäre dies so, wären aber die Empfänger überhaupt nicht mehr sicher: Selbst wenn die Annahme vollkommen stimmt und nie eine Unterschrift gefälscht wird, könnten nun wirkliche Unterzeichner vor Gericht ihre Unterschrift ableugnen, genauso wie oben der angebliche Unterzeichner. Das Gericht hat keinerlei (technische) Möglichkeit, die beiden Fälle zu unterscheiden. Müßten Empfänger also mit diesem Verhalten der Gerichte rechnen, so würde niemand mehr digitale Signaturen von jemandem akzeptieren, dem er nicht vertraut. Gerade das Sichern von Sender und Empfänger voreinander, z.B. bei einem Kaufvertrag, war aber der Hauptvorteil von digitalen Signaturen über symmetrische Authentikationssysteme (neben der Vereinfachung der Schlüsselverteilung).

Fail-stop-Signaturen

Kennzeichen

Auch bei Fail-stop-Signaturen beruht die Unfälschbarkeit auf einer Annahme. Im konkret vorgeschlagenen System, Versteckssystem genannt, ist es die Faktorisierungsannahme; und, wie oben nur bei GMR, ist bewiesen, daß unter dieser Annahme Fälschungen unmöglich sind [10].

Ihr neues Kennzeichen ist: Wenn aber doch eine Signatur gefälscht wird, kann der angebliche Un-

terzeichner beweisen, daß es sich um eine Fälschung handelt. Genauer: Er kann beweisen, daß die zugrundeliegende Annahme gebrochen wurde, z.B. indem er nun Faktoren einer Zahl vorzeigen kann, die er unter der Annahme nicht hätte faktorisieren können. Für Einzelheiten, wie dies gelingt, siehe die beiden Kästen und [10].)

Fehlerwahrscheinlichkeit gegen kryptographische Annahme
Man sollte erwähnen, daß mit einer (beliebig wählbaren) winzigen Wahrscheinlichkeit, z.B. 2^{-100}, der Beweis der Fälschung mißlingt. Dies schadet aber nichts:

Erstens ist eine winzige Fehlerwahrscheinlichkeit eine prinzipielle Eigenschaft digitaler Authentikationssysteme (auch symmetrischer): Der Unterzeichner benutzt nur eine endliche Anzahl Bits geheimer Information. Wenn ein Fälscher sie alle rät, ist nichts zu machen. „Unbedingt sicher“ nennt man ein Authentikationssystem, wenn bewiesen ist, daß man

- prinzipiell nicht besser fälschen kann als durch völlig ungezieltes Raten, und
- nicht selbst prüfen kann, ob man richtig geraten hat (außer indem man vor Gericht geht oder den Signierer selbst fragt).

In diesem Sinn ist das Erkennen von Fälschungen unbedingt sicher.

(Ähnliches bei eigenhändigen Unterschriften wäre, daß ein Fälscher eine Horde Affen anstellt, die Kritzeleien machen. Davon muß er welche als Unterschriften auswählen, ohne schauen zu dürfen, ob sie der Unterschrift auch ähnlich sind.)

Zweitens ist z.B. schon die Wahrscheinlichkeit größer, daß wegen eines Hardwarefehlers eine falsche Unterschrift akzeptiert wird, was die gleichen Folgen hat, und überhaupt die Wahrscheinlichkeiten fast aller Risiken des täglichen Lebens.

Der Unterschied eines Systems mit winziger Fehlerwahrscheinlichkeit zu einem, das wie herkömmliche Signaturen auf einer Annahme beruht, ist, daß man bei letzterem nicht *weiß*, ob die Fälschungswahrscheinlichkeit so klein ist; statt eine Horde Affen einzusetzen, *könnte* der Fälscher auch eine automatische Signiermaschine konstruiert haben.

Ebenfalls erwähnen sollte man, daß die Fail-stop-Eigenschaft nicht bei physischem Schlüsseldiebstahl hilft. Auch dies ist eine prinzipielle Eigenschaft jedes digitalen Authentikationssystems: Hat ein Fälscher alle Information des Signierers, so ist er auf digitalem Wege vom Signierer nicht mehr zu unterscheiden. Hiergegen sind also weiterhin physische Maßnahmen nötig.

Vorteile
Bei Fail-stop-Signaturen ist also (jedenfalls bzgl. des digitalen Teils) einem Gericht in jeder Situation klar, ob eine Unterschrift gefälscht ist oder nicht, im Gegensatz zu herkömmlichen Signaturen.

Auch ist bereits nach der ersten Fälschung allen weiteren Teilnehmern und dem Systembetreiber (z.B. einer Bank, die die Signaturen in einem Zahlungssystem einsetzt) klar, daß das Signatursystem gebrochen ist. Man kann es somit stoppen, bevor viel weiterer Schaden entsteht, oder zumindest den Sicherheitsparameter erhöhen. Dies erklärt den Namen „Failstop“: Er wird in der Fehlertoleranz für Systeme verwendet, die bei Ausfall wenigstens in einen festen Zustand gelangen, z.B. Eisenbahnsignale, die, wenn ein Zugseil reißt, in die Halt!-Stellung klappen.

Man muß sich natürlich klarmachen, daß das sichere Erkennen von Fälschungen zwar schön ist, aber nicht sämtliche Probleme löst: Wenn der Empfänger einer gefälschten Unterschrift nicht selbst der Fälscher ist, so hat er sich auf die Unterschrift verlassen. Gilt sie nun, als gefälscht erkannt, nicht mehr, so entsteht ihm, je nach Anwendung, Schaden. Sofern dieser finanziell ist, könnte ihn aber eine Versicherung tragen: Sie kann ja, wie ein Gericht, das Eintreten des Schadensfalls jetzt einwandfrei überprüfen, im Gegensatz zu herkömmlichen Signaturen; und der Schaden tritt nur auf, falls die Annahme gebrochen wird.

Anwendung in Zahlungssystemen

Einsatz
Am interessantesten dürften Failstop-Signaturen in digitalen Zahlungssystemen werden. Dort könnten die Kunden mit Failstop-Signaturen unterschreiben, die Bank mit herkömmlichen.

Dann sind die Kunden unbedingt sicher, d.h. sie müssen sich nicht mehr auf die Gültigkeit der Faktorisierungsannahme verlassen (denn sie sind Unterzeichner von Fail-stop-Signaturen und Empfänger von herkömmlichen Signaturen, und in beiden Rollen ist man, wie geschildert, unbedingt sicher).

Warum so?
Warum macht man nicht umgekehrt die Bank unbedingt sicher?

Erstens, weil sie der stärkere Partner ist: Sie kann das herkömmliche Signatursystem und

den Sicherheitsparameter auswählen und so für ihre eigene Sicherheit sorgen. Auch kann sie sich über die Vertrauenswürdigkeit der Annahme informieren, während manchen Bürger an einer Faktorisierungs- oder Diskreter-Logarithmus-Annahme schon der Name abschrecken dürfte.

Zweitens benötigt die Bank, wenn ihre eigene Unterschrift gefälscht würde, gar keinen Fälschungsbeweis: Als Systembetreiber kann sie selbst den Systemstop veranlassen.

Drittens könnte die Bank sogar einen Vorteil gegenüber einem digitalen Zahlungssystem nur mit herkömmlichen digitalen Signaturen haben: Die Möglichkeit, Fälschungen zweifelsfrei zu erkennen, könnte die juristische Akzeptanz erhöhen, d.h. wie sicher die Bank davon ausgehen kann, daß im Einzelfall die Signaturen der Kunden vor Gericht anerkannt werden.

Praktisch einsetzbar?

Wie schon in der o.g. Tagungsbesprechung gesagt, liegt keine Implementierung des Systems vor. Es ist auch nicht die Art von System, die sich z.B. jeder für die Sicherung seines lokalen Netzes kaufen würde, sondern für Systeme rechtlicher Relevanz und hoher Sicherheitsanforderungen wie Zahlungssysteme gedacht. Speziell hierfür ist die theoretische Entwicklung so weit, daß man an Implementierungen denken kann (siehe zweiter Kasten).

Empfehlung

Mit der Darstellung, daß mehr Sicherheit als bei herkömmlichen digitalen Signaturen erreicht werden kann, soll nicht der Eindruck erweckt werden, diese seien zu nichts nütze. Beim derzeitigen Stand der Photokopiertechnik sind sie vermutlich sogar wesentlich sicherer als eigenhändige Unterschriften.

Natürlich sollte an der Entwicklung sicherer Signatursysteme weitergearbeitet werden. (Es wurde sogar ein System mit ähnlichen Eigenschaften wie Signaturen vorgeschlagen, in dem man unbedingt sicher vor Fälschungen ist; es ist aber zur Zeit nicht praktikabel [3].) Anwendungen der Größenordnung von Zahlungssystemen haben aber erfahrungsgemäß eine lange Planungsphase, vgl. die Entwicklung des Einsatzes von RSA. Einstweilen ist es also auf jeden Fall besser, z.B. RSA-basierte Anwendungen, die nahezu einsatzbereit sind, zügig einzusetzen, als völlig ungeschützt zu sein oder nur symmetrische Authentikation zu verwenden.

Dank

Für viele gute Hinweise zu diesem Text danke ich Andreas Ort, Andreas Pfitzmann und Michael Waidner.

Literatur

[1] G. Bleumer, B. Pfitzmann, M. Waidner: A remark on a signature scheme where forgery can be proved; Eurocrypt '90, LNCS 473, Springer-Verlag, Berlin 1991, 441-445.

[2] D. Chaum, E. van Heijst, B. Pfitzmann: Cryptographically Strong Undeniable Signatures, Unconditionally Secure for the Signer; Interner Bericht 1/91, Fakultät für Informatik, Universität Karlsruhe 1991 (Überarbeitung erscheint auf Crypto '91).

[3] D. Chaum, S. Roijakkers: Unconditionally Secure Digital Signatures; Crypto '90, Santa Barbara 1990, Abstracts, 209-217.

[4] D. Davies, W. Price: Security for Computer Networks, An Introduction to Data Security in Teleprocessing and Electronic Funds Transfer; (2nd ed.) John Wiley & Sons, New York 1989.

[5] T. ElGamal: A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms; IEEE Transactions on Information Theory 31/4 (1985) 469-472.

[6] D. Fox, B. Pfitzmann: Effiziente Software-Implementierung des GMR-Signatursystems; Verläßliche Informationssysteme (VIS'91), Informatik-Fachberichte 271, Springer-Verlag, Heidelberg 1991, 329-345.

[7] S. Goldwasser, S. Micali, R. Rivest: A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks; SIAM J. Comput. 17/2 (1988) 281-308.

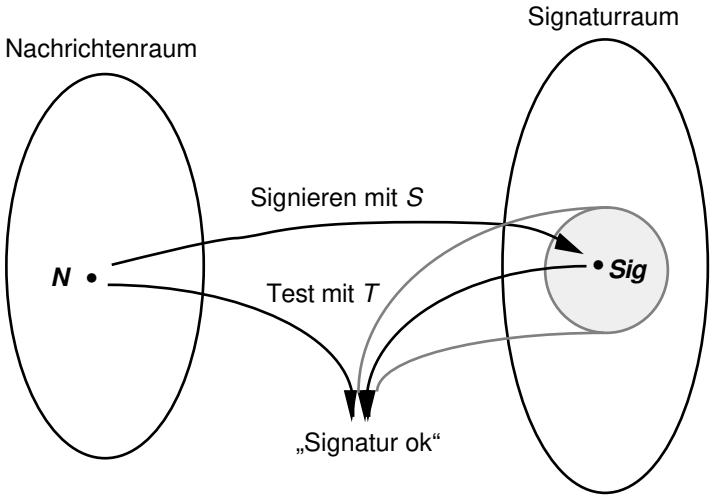
[8] P. Horster: Kryptologie; Reihe Informatik/47, BI, Mannheim 1985.

[9] B. Pfitzmann, M. Waidner: Formal Aspects of Fail-stop Signatures; Interner Bericht 22/90, Fakultät für Informatik, Universität Karlsruhe 1990.

[10] B. Pfitzmann, M. Waidner: Fail-stop-Signaturen und ihre Anwendung; Verläßliche Informationssysteme (VIS'91), Informatik-Fachberichte 271, Springer-Verlag, Heidelberg 1991, 289-301.

[11] R. Rivest, A. Shamir, L. Adleman: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems; Communications of the ACM 21/2 (1978) 120-126.

Grundidee zur Konstruktion eines Fail-stop-Signatursystems:



Wie bei den meisten herkömmlichen Signatursystemen erhält der Unterzeichner mit dem Signierschlüssel S zu jeder Nachricht N genau eine Signatur Sig, und der Test mit T, angewandt auf N und Sig, liefert "ok" als Ergebnis.

Neu ist, daß es außer Sig noch eine Menge weiterer Werte (im Bild der graue Kreis) geben muß, die denselben Test bestehen, also als Signaturen unter N akzeptabel sind. Die Konstruktion garantiert: Wenn die kryptographische Annahme gilt, kann niemand außer dem Unterzeichner auch nur eine akzeptable Signatur finden und der Unterzeichner nur diese einzige. (Insbesondere ist der graue Kreis im Vergleich zum ganzen Signaturraum immer noch sehr klein)

Falls aber ein Fälscher die Annahme bricht, so kann er anhand der Testfunktion zwar den grauen Kreis berechnen, aber nicht herausfinden, welcher der Werte darin die echte Signatur Sig ist. (Dabei müssen weitere Bedingungen gelten: Damit der Fälscher nicht zuerst S rät, gibt es genausoviele weitere mögliche Signierschlüssel, die zu T passen, und diese führen gerade zu den anderen akzeptablen Signaturen. Außerdem dürfen andere signierte Nachrichten keine Information über die neue Signatur geben.) Um für seine Fälschung genau die echte Signatur zu verwenden, muß der Fälscher also unter all diesen Werten richtig raten.

Ist man z.B. bereit, eine Fehlerwahrscheinlichkeit von 2^{-100} zu tolerieren, so wählt man als Größe des grauen Kreises also 2^{100}.

Rät der Fälscher falsch, so kann der Unterzeichner seine echte Signatur vorlegen und zeigen, daß sie anders ist. Das Vorliegen von zwei Signaturen zur selben Nachricht beweist, daß die kryptographische Annahme gebrochen ist, denn solange sie galt, konnte selbst der Unterzeichner nur eine einzige Signatur erzeugen. Im konkret vorgeschlagenen Verstecksystem (siehe zweiter Kasten) kann der Unterzeichner aus den zwei Signaturen auch die Faktorisierung einer Zahl berechnen, deren Faktoren er vorher nicht kannte. Damit ist der Zusammenhang mit der Faktorisierungsannahme noch klarer.

Rahmenkonstruktion für Zahlungssysteme: 3-Phasen-Protokoll

Das Problem mit derzeitigen Fail-stop-Signaturen im allgemeinen Fall ist, daß sie stark mit der Länge der Nachricht wachsen. Speziell für Zahlungssysteme gibt es aber ein Rahmenprotokoll, in dem die Kunden mit ihren Fail-stop-Signaturen immer nur ein Bit lange Nachrichten unterschreiben müssen. Dadurch wird das System praktikabel.

Will ein Kunde *K* der Bank seinen *i*-ten Auftrag geben, etwa eine Überweisung, so geschieht dies in 3 Phasen:

1. *Mitteilung*: Der Kunde teilt der Bank seinen *i*-ten Auftrag mit, etwa „Überweise 24 DM an Kunden *L*."
2. *Bestätigung*: Die Bank bestätigt diesen Auftrag in einer (mit ihrer herkömmlichen Signatur) unterschriebenen Nachricht, etwa: „Kunde *K*'s *i*-ter Auftrag ist: Überweise 24 DM an *L*."
3. *Einverständnis*: Ist die Bestätigung richtig, so unterschreibt der Kunde ein einziges Bit, das „ok" bedeutet, mit seiner *i*-ten Fail-stop-Signatur.

Zu Phase 3 ist anzumerken, daß derzeitige Fail-stop-Signaturen automatisch numeriert sind.

Ein Gericht kann ggf. einen Streit, ob die Bank den *i*-ten Auftrag falsch ausgeführt hat, so entscheiden: Zuerst muß die Bank das Einverständnis des Kunden vorlegen, daß sie den *i*-ten Auftrag überhaupt ausführen durfte. Der Kunde erhält Gelegenheit zu beweisen, daß dies eine Fälschung ist. Andernfalls muß er die unterschriebene Bestätigung der Bank vorlegen. Deren Inhalt gilt als richtige Form des Auftrags.

Wie versprochen, sind die Kunden unbedingt sicher: Ihre Sicherheit hängt nur von ihren Fail-stop-Signaturen ab und davon, daß sie die richtige Bestätigung der Bank gut aufbewahren. Die Bank ist unter der von ihr gewählten kryptographischen Annahme sicher: Um sie zu betrügen, muß jemand entweder ihre Bestätigung fälschen oder einen Fälschungsbeweis für eine Fail-stop-Signatur vorlegen.

Einfache Fail-stop-Signaturen: Verstecksystem für „ok"-Nachrichten

Das Verstecksystem beruht auf einer sogenannten *bündelnden Funktion g*. Dies ist eine Einwegfunktion (also eine Funktion, die man effizient berechnen kann, aber nicht ihre Umkehrfunktion) mit zwei Zusatzeigenschaften:

A. Es wird immer eine gewisse Mindestanzahl von Werten (2^{100} im bisherigen Beispiel) auf das gleiche Bild abgebildet.

B. Unter der kryptographischen Annahme ist es aber unmöglich, zwei Werte mit dem gleichen Bild zu finden.

Wollte der Kunde nur einmal ein Bit „ok" unterschreiben, so ginge dies mit einer bündelnden Funktion so (man beachte, daß im obigen 3-Phasen-Protokoll nicht einmal mehr 0 und 1 zu unterscheiden sind): Als Unterschrift und zugleich geheimen Schlüssel S wählt der Kunde vorweg einen beliebigen Wert Sig. Dessen Bild $T = g(Sig)$ gibt er als öffentlichen Schlüssel bekannt. Damit erreicht man genau die im ersten Kasten beschriebenen Eigenschaften: Solange die kryptographische Annahme gilt, kann wegen der Einwegeigenschaft von g niemand eine Signatur fälschen, und der Kunde kennt wegen Eigenschaft B nur die eine Signatur Sig. Und selbst wenn die Annahme gebrochen ist, muß ein Fälscher wegen Eigenschaft A Sig unter 2^{100} Werten raten.

Beim Einsatz im oben beschriebenen 3-Phasen-Protokoll kann der Austausch eines neuen Schlüssels T_i für den i-ten Auftrag jeweils mit der vorigen Signatur erfolgen: Der Kunde wählt Sig_i vor dem $(i-1)$-ten Auftrag und teilt T_i als Teil des $(i-1)$-ten Auftrags der Bank mit und erhält es von ihr bestätigt.

Konkrete bündelnde Funktionen

Konkrete bündelnde Funktionen für die Faktorisierungsannahme und die Fehlerwahrscheinlichkeit 2^{-100} ergeben sich so: Die Bank wählt zwei große Primzahlen p, q mit $p \equiv 3$, $q \equiv 7 \bmod 8$ und gibt $n = p \cdot q$ bekannt.

Der Definitionsbereich besteht aus Paaren (z, d). Dabei ist z eine natürliche Zahl kleiner 2^{100} und d eine Zahl modulo n mit zwei Zusatzeigenschaften:

(1) $d < n/2$.

(2) d hat das Jacobisymbol 1 (s. z.B. [8]).

Eine Hilfsfunktion g'_n wird definiert durch

$$g'_n(z, d) = 4^z \cdot d^{2^{100}} \bmod n.$$

Falls $g'_n(z, d) < n/2$, so ist $g_n(z, d) = g'_n(z, d)$, sonst $-g'_n(z, d) \bmod n$.

Für eine ausführlichere Beschreibung sei noch einmal auf [10] verwiesen, für formale Beweise auf [1] und [9].

(Kenner mögen bemerkt haben, daß die Bank, die p und q kennt, die Funktion g invertieren kann. Das stört nicht, denn sie könnte damit zwar Kundenunterschriften fälschen, der Kunde aber die Fälschung beweisen. Somit würde gemäß dem Rahmenprotokoll die Bank nur sich selbst betrügen.)

Auch für die Diskreter-Logarithmus-Annahme gibt es vergleichbar effiziente bündelnde Funktionen [2].

Die Effizienz hängt davon ab, welchen herkömmlichen Signaturen die Bank vertraut. Selbst wenn sie RSA mit einer schnellen Hashfunktion verwendet, ist ein Kundenauftrag im beschriebenen System nur etwa um den Faktor 3 ineffizienter, als wenn auch die Kunden RSA verwenden würden.