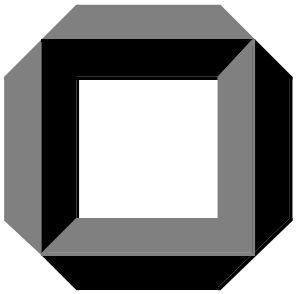


UNIVERSITÄT KARLSRUHE

FAKULTÄT FÜR INFORMATIK

Postfach 6980, D 7500 Karlsruhe

Vergleich der algebraischen und kryptographischen Modellierung von Kryptoprotokollen

Birgit Pfitzmann

Studienarbeit am
Institut für Rechnerentwurf und Fehlertoleranz

Betreuer:
Michael Waidner
Prof. Dr. Winfried Görke

Dezember 1988

Ich versichere hiermit, daß ich die vorliegende Arbeit selbständig und ohne unzulässige fremde Hilfe angefertigt habe. Die verwendeten Literaturquellen sind im Literaturverzeichnis vollständig aufgeführt.

Gliederung

1 Motivation 1

1.1 Einsatzbereich der betrachteten Protokolle 1
1.2 Warum kryptographisch ? 1
1.3 Zweck der Modellierung 4
1.4 Ziele dieser Arbeit 5

2 Algebraische Modellierung 6

2.1 Digitale Signaturen 6
2.2 Nachrichtenunkenntnis und Ungleichungen 7
2.3 Algebraische Modellierung für andere kryptographische Systeme 9

3 Kryptographische Modellierung 11

3.1 Komplexität 11
3.1.1 Schaltnetzkomplexität 12
3.1.2 Komplexität für fast alle Fälle 14
3.2 „Brechen" speziell bei digitalen Signaturen 17
3.3 Existierende „kryptographisch sichere" digitale Signatursysteme 18
3.4 Ungleichungen und Nachrichtenunkenntnis 20
3.5 Andere Anwendungen der kryptographischen Modellierung 26

4 Vergleich der Modellierungsarten 26

4.1 Entsprechungen zwischen den Modellwelten 26
4.1.1 Kryptographische Modelle als Interpretationen von Spezifikationen 27
4.1.2 Korrektheit der kryptographischen Interpretationen 29
4.1.3 Entsprechung zwischen Termen und kryptographischen Bauarten 30
4.1.4 Entsprechungen zwischen algebraischem und kryptographischem Modell 32
4.2 Einfache Vergleichsergebnisse 33
4.2.1 Zusammenhang von Nachrichtenunkenntnis und Ungleichungen 34
4.2.2 Anmerkungen zu einem rein algebraischen Lemma 39
4.2.3 Nichtkorrektheit bei Funktionen mit kleinen Bildbereichen 42
4.2.4 Nichterfassung möglicher innerer Struktur der Unterschriften 45
4.2.5 Diskussion des Ergebnisses 46

5 Ausblick 48

6 Literatur 49

Bilder

Bild 1 Grobe Skizze eines Protokollablaufs 3
Bild 2 Beispiel für die Nachrichtenkenntnis eines Angreifers 7
Bild 3 Darstellung von Schlüssel- und Zufallszahlenanteilen zu Anmerkung 1 16
Bild 4 Schaltnetz zum Brechen eines digitalen Signatursystems 18
Bild 5 Schaltnetz für das Kennen einer Nachricht der Bauart B 23
Bild 6 Beispiel zu Lemma 5 37
Bild 7 Sicherheitsbeweis für Signatursysteme mit Anhängsel 45

1 Motivation

Bevor die genaue Aufgabenstellung dieser Arbeit erläutert werden kann (1.4), ist es sinnvoll, zunächst den praktischen Zweck der zu modellierenden Protokolle zu betrachten (1.1). Davon ausgehend wird erklärt, weshalb die Protokolle kryptographisch sein müssen und welche Funktion in ihnen die wichtigsten kryptographischen Hilfsmittel erfüllen (1.2). Daraufhin wird der Zweck der hier betrachteten Modellierungen dieser Protokolle genannt und die wichtigsten Teilaufgaben, die in beiden Modellierungen gelöst werden müssen und anhand derer der Vergleich durchgeführt werden soll, angegeben (1.3).

1.1 Einsatzbereich der betrachteten Protokolle

Die im folgenden zu betrachtenden Protokolle sollen im wesentlichen für Zahlungssysteme und zum Informationskauf in offenen digitalen Systemen dienen.

Gesucht ist also eine Menge von Regeln, wie eine nicht geschlossene Gruppe von Teilnehmern miteinander in digitaler Form, also nur über ein Kommunikationsnetz, Geschäfte durchführen kann. Als physische Grundlage, die die Einführung eines solchen Systems nahelegt, wird das von der Deutschen Bundespost geplante ISDN gesehen.

Für ein digitales System geeignet sind natürlich vor allem solche Geschäfte, die sich von vornherein auf Information beziehen, etwa kostenpflichtige Datenbankanfragen oder das Beziehen von Zeitungen oder Ausschnitten davon in digitaler Form. Zur Unterstützung hiervon muß es ein digitales Zahlungssystem geben (oder mehrere), das aber auch für andere Geldtransfers, etwa Überweisungen, verwendet werden kann.

Im engeren Sinne „offen" heißt solch ein System dann, wenn jedermann in jeder Rolle daran teilnehmen kann, im allgemeinen allerdings erst nach einer geeigneten Anmeldungsprozedur. Praktisch wird dies etwa heißen, daß man Kunde etwa so leicht werden kann wie derzeit Telefonbesitzer, Anbieter wie derzeit Gewerbetreibender, und daß alle Banken digitale Zahlungen ihrer Kunden unterstützen dürfen.

Es ist bereits umstritten, ob solche Systeme ihrer Zielsetzung nach gesellschaftlich wünschenswert sind [SteW_85, KuRo_86]; hier soll jedoch nur betrachtet werden, wie sie realisiert werden können, ohne irgendwelche Rechte der Teilnehmer, zumindest im Vergleich zu heutigen nicht digitalen Systemen, einzuschränken.

1.2 Warum kryptographisch ?

Im Grunde genügt schon die Tatsache allein, daß es bei Zahlungen und Kauf um Rechtsgeschäfte geht, als Begründung, daß die dazu verwendeten Protokolle Rechtssicherheit garantieren müssen, also vor allem sichern müssen, daß niemand ohne Zustimmung eines anderen in dessen Namen ungestraft Geschäfte tätigen kann (vgl. [PWP_87], das auch ausführlich die Konstruktion von geeigneten Protokollen behandelt und größere Beispiele enthält).

Verstärkt wird diese Notwendigkeit durch die beabsichtigte Offenheit der Systeme, da man dann mit Sicherheit nicht mehr erwarten kann, daß jeder, der sich zur Teilnahme am System entschließt, allen anderen möglichen Teilnehmern vertraut, nicht einmal all denjenigen, mit denen er Geschäfte tätigen wird. Am einleuchtendsten ist, daß kein Anbieter und keine Bank sich auf die Ehrlichkeit all ihrer potentiellen Kunden verlassen wollen wird, aber auch die Kunden werden meist die

Vertrauenswürdigkeit der Anbieter, deren Angebote ihnen interessant scheinen, nicht prüfen können. Die Banken oder die Post verdienen zwar sicher mehr Vertrauen, aber auch bei den Beziehungen zu diesen sollte auf Rechtssicherheit nicht verzichtet werden, wie z.B. auch derzeit das eigenhändige Unterschreiben von Überweisungen sichern soll, daß Streit darüber, ob ein Kunde einen Überweisungauftrag gab, möglichst gar nicht erst entsteht (zumal Vertrauen zur Bank als solcher sich nicht unbedingt auch auf deren einzelne Angestellte erstreckt).

Daß ein Protokoll Rechtssicherheit garantiert, bedeutet etwas konkreter, daß jedem Teilnehmer, der sich daran hält, gewisse Garantien gemacht werden, auch für den Fall, daß die übrigen vom Protokoll abweichen.

Eine wichtige derartige Garantie wird meist sein, daß dieser Teilnehmer kein Geld verliert, ohne einen Gegenwert dafür zu erhalten. Nicht garantiert werden kann (und soll) dagegen im allgemeinen, daß das Ergebnis eines versuchten Geschäfts immer dasselbe ist, solange sich auch nur einer der daran Beteiligten ans Protokoll hält, z.B. wird nicht unbedingt garantiert sein, daß jeder Anbieter bereit ist, mit jedem Kunden ein Geschäft abzuschließen.

Wie bei den derzeitigen Geschäftsformen kann es natürlich vorkommen, daß gewisse Garantien erst mittels eines Gerichts durchgesetzt werden können, etwa wenn eine Bank erhaltenes Geld veruntreut haben sollte.

Um diese Sicherheit zu verwirklichen, muß jeder Teilnehmer Möglichkeiten haben, seinen Willen im digitalen System in einer Weise auszudrücken, die kein anderer imitieren kann, wozu kryptographische Hilfsmittel nötig sind. Die wichtigsten sind für diesen Zweck digitale Signaturen, d.h. kryptographische Funktionen, die das leisten, was zur Zeit die eigenhändige Unterschrift im Idealfall leisten sollte:

- Jeder sollte testen können, ob eine empfangene Nachricht von einem bestimmten anderen Teilnehmer unterschrieben wurde.
- Dennoch sollte man, auch nachdem man bereits einige unterschriebene Nachrichten dieses anderen Teilnehmers empfangen hat, keine davon ändern können (ohne daß die Unterschrift ungültig wird) und keine weiteren hinzuerfinden können.

Digitale Signaturen werden im folgenden bei beiden Modellierungsarten noch genauer behandelt (2.1, 3.1).

Unter Verwendung digitaler Signaturen anstelle eigenhändiger Unterschriften können Protokolle recht ähnlich zu bisherigen nicht digitalen Geschäftsabläufen gestaltet werden. Neben der Forderung nach Rechtssicherheit besteht in offenen digitalen Systemen wegen der gegenüber nicht digitalen Systemen stark erweiterten Beobachtungsmöglichkeiten auch die nach überprüfbarem Datenschutz. Dieser wird im folgenden kaum behandelt. In [PWP_87] kann aber nachgelesen werden, daß überprüfbarer Datenschutz, vor allem durch Verschlüsselung von Nachrichten und Zuordnung der digitalen Unterschriften zu Pseudonymen statt zu den Namen der Teilnehmer, erreicht werden kann, ohne die Rechtssicherheit zu zerstören.

Um diese Forderung zu wissen, ist aber auch bei ausschließlicher Betrachtung der Rechtssicherheit interessant, weil man sich ihretwegen nicht darauf beschränken kann, nur Protokolle zu behandeln, die bis auf die Ersetzung eigenhändiger Unterschriften durch digitale Signaturen genau wie bisher im nicht digitalen Fall ablaufen.

Als Beispiel ist in Bild 1 eine grobe Skizze des beabsichtigten Ablaufs eines (besonders einfachen, nicht besonders datenschützenden) Protokolls zum Informationskauf zu sehen (stark vereinfacht nach [Waid_85]). Dabei sendet ein Kunde eine Anfrage mit dem maximalen Preis, den er für deren Beantwortung zahlen würde, zunächst zur Bank, bei der er ein Konto besitzt. Diese fügt eine Garantie

hinzu, daß er soviel Geld besitzt und leitet das Ganze an den Informationsanbieter weiter. Dieser sendet, sofern ihm das garantierte Geld genügt, die Antwort mit dem genauen Preis über die Bank zurück. Diese transferiert dabei das Geld, und beide erhalten einen Kontoauszug. Im Bild sind alle Unterschriften durch Siegel mit dem Anfangsbuchstaben des Unterschreibenden dargestellt.

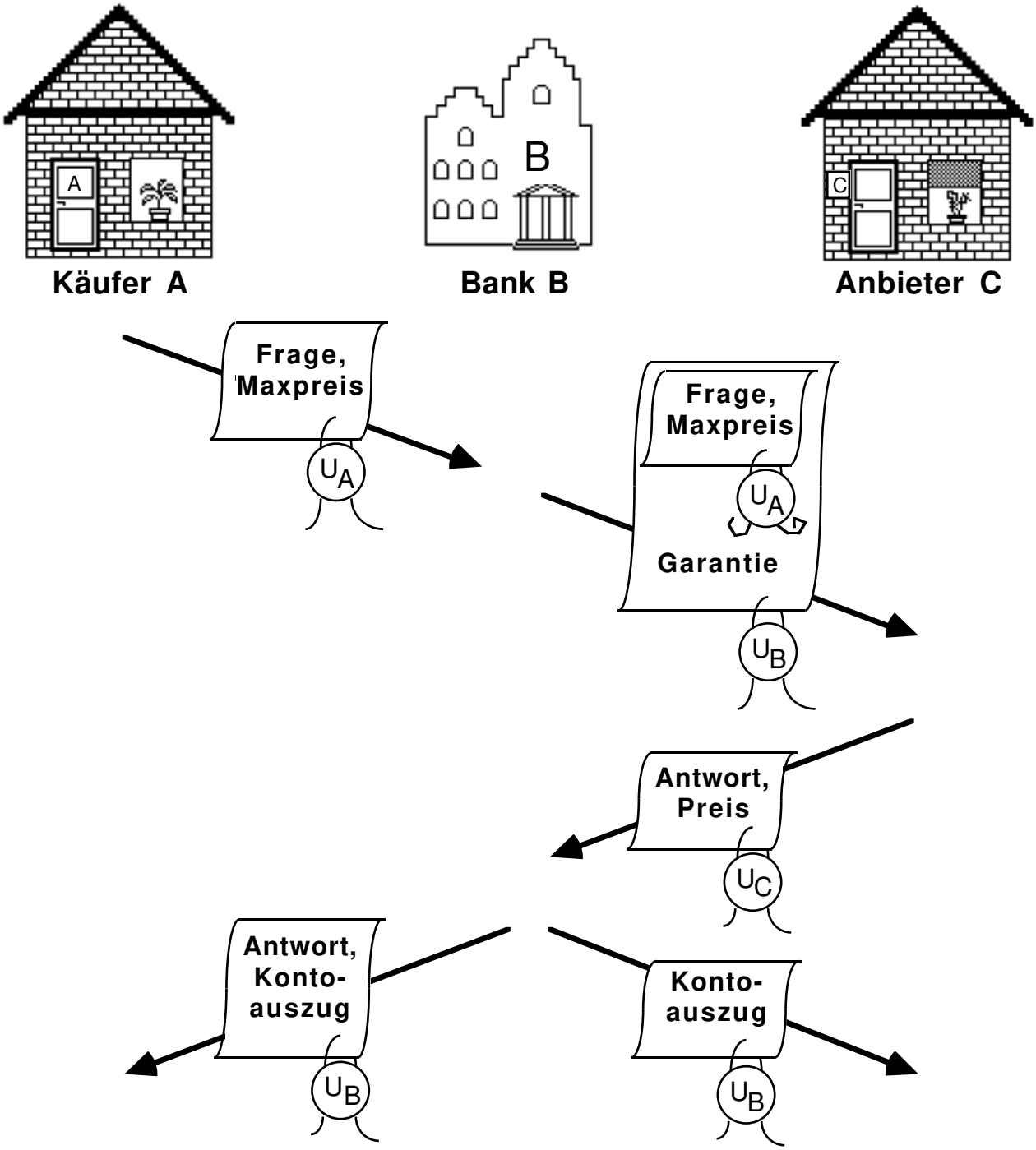


Bild 1 Grobe Skizze eines Protokollablaufs

Das eigentliche Protokoll wird allerdings nicht durch diesen Ablauf angegeben, sondern durch Regeln für die einzelnen Teilnehmer (bzw. die Rollen, die sie im System spielen, da z.B. für alle Anbieter die gleichen Regeln gelten werden). Beispielsweise besagt die Regel für den Anbieter, wie er die

Korrektheit der Garantie prüft und in welcher Form er gegebenenfalls seine Antwort senden soll.

Diese Darstellung durch Regeln ist nötig, weil das Protokoll lokal durch die einzelnen Teilnehmer ausgeführt werden muß, und insbesondere, damit auch klar ist, wie sie sich zu verhalten haben, wenn andere Teilnehmer vom Protokoll abweichen.

1.3 Zweck der Modellierung

Als Zweck der zwei zu vergleichenden Modellierungsarten wird im folgenden fast ausschließlich der Beweis der Rechtssicherheit von Protokollen betrachtet, die als einzige kryptographische Hilfsmittel digitale Signaturen verwenden (obwohl beide im Prinzip auch anderes erlauben, insbesondere das Betrachten von Verschlüsselung, vgl. 2.3 und 3.5).

Die Sicherheit des verwendeten Signatursystems wird dabei vorausgesetzt. Was „Sicherheit" dabei genau heißen soll, ist ein wesentliches Charakteristikum der beiden Modellierungsarten und wird dort jeweils genauer beschrieben.

Die Betrachtung vollständiger Protokollbeweise scheint allerdings für den Vergleich, zumindest für den Anfang, unnötig aufwendig. Statt dessen werden nur die Beweise gewisser einfacherer Aussagen betrachtet, die jedem dieser Protokollbeweise zugrundezuliegen scheinen. (Allgemeines zu Beweisen ganzer Protokolle findet sich in [WaPf_85, Waid_85, Bürk_87], in den beiden letzten auch Beweise im algebraischen Modell unter Voraussetzung solcher einfacheren Aussagen).

Den ersten Typ solcher Aussagen bilden solche über die Nachrichtenunkenntnis eines Angreifers (d.h. jemandes, der das Protokoll zu stören versucht) zu einem bestimmten Zeitpunkt. Sie haben die Gestalt:

> „Ein Teilnehmer, der bisher gewisse Nachrichten $N_1,...,N_m$ erhalten hat, kann trotzdem keine Nachricht N einer gewissen Gestalt erzeugen."

Im Beispiel aus Bild 1 könnten $N_1,...,N_m$ einige bisher erhaltene Kontoauszüge und Bankgarantien sein, N eine weitere Bankgarantie. Drastischer wird das Beispiel bei einem Überweisungsprotokoll, wo $N_1,...,N_m$ die bisher erhaltenen Überweisungen sein könnten und N eine zusätzliche, möglichst über einen hohen Betrag.

Allgemeiner werden im Beweis eines Gesamtprotokolls $N_1,...,N_m$ die Nachrichten sein, die der Angreifer in allen bisherigen Protokollabläufen erhalten hat, also sowohl den korrekt oder fehlerhaft beendeten als auch dem aktuellen. N kann

- entweder für eine Nachricht stehen, deren Erzeugung durch den Angreifer bereits eine Verletzung einer der Garantien, die das Protokoll den anderen Teilnehmern geben soll, bedeuten würde (z.B. die Überweisung)
- oder (wie die Bankgarantie) für eine Nachricht, deren Gestalt ein anderer Teilnehmer in einem folgenden Protokollschritt prüft, und bei deren Fälschung er so handeln würde, daß in der Folge seine Garantien verletzt werden können.

Beim Beweis von Aussagen über die Nachrichtenunkenntnis stößt man bei beiden Modellierungsarten als zweiten Typ auf Unteraussagen in der Form von Ungleichungen:

> „Für gewisse Nachrichten N_i' und N' gilt: $N_i' \neq N'$."

Dabei steht N_i' vor allem für die Nachrichten N_i aus der ersten Art von Aussagen oder solche, die man daraus herleiten kann, und N' für N oder seine Bestandteile.

1.4 Ziele dieser Arbeit

Der Rest der Arbeit behandelt also im wesentlichen den Beweis von Nachrichtenunkenntnis und Ungleichungen in Systemen, deren einzige kryptographische Funktionen digitale Signaturen sind. Einige Anmerkungen werden auch über die Änderungen bei Hinzunahme von Verschlüsselung gemacht.

Zunächst wird die ältere algebraische Modellierung, die von einem idealisierten Signatursystem ausgeht, skizziert und soweit präzisiert, daß ein exakter Vergleich möglich wird (Kapitel 2).

Dann (Kapitel 3) wird die neuere kryptographische Modellierung, die sich mit dem Beweis der, möglichst optimalen, Sicherheit realer kryptographischer Funktionen beschäftigt, am Beispiel digitaler Signaturen dargestellt und dabei der Versuch unternommen, zu begründen, weshalb die Modellierung nicht anders lautet. Es wird untersucht, ob Nachrichtenunkenntnis analog definiert werden kann und ob mit den derzeitigen digitalen Signatursystemen Chancen bestehen, Nachrichtenunkenntnis in diesem Sinne für die in Protokollen vorkommenden Nachrichtenkombinationen zu beweisen; diese Fragen können positiv beantwortet werden.

Zuletzt (Kapitel 4) kann der Vergleich vorgenommen werden. Da aus der Art der Idealisierung des Signatursystems in der algebraischen Modellierung klar sein wird, daß aus der Unsicherheit eines Protokolls im algebraischen Modell (speziell hier: aus der Fähigkeit des Angreifers, die betrachtete Nachricht N zu erzeugen) die Unsicherheit auch im kryptographischen Modell folgt, ist die Hauptfrage:

> „Folgt aus der Sicherheit im algebraischen Modell die im kryptographischen?"

bzw.

> „Folgt daraus, daß ein Angreifer im algebraischen Modell N aus $N_1,...,N_m$ nicht erzeugen kann, dasselbe im kryptographischen Modell, wenn man an die Stelle des idealisierten Signatursystems ein kryptographisch sicheres setzt ?"

Die praktische Bedeutung dieser Frage ist folgende: Da ein Protokoll in der Realität nur mit realen, nicht mit idealisierten Signatursystemen implementiert werden kann, ist die kryptographische Modellierung diejenige, deren Sicherheit maßgebend ist. Daher sollte für Sicherheitsbeweise die algebraische Modellierung nur an solchen Stellen weiterhin verwendet werden, wo aus der darin bewiesenen Sicherheit eines Protokoll(teil)s auch die kryptographische folgt, wenn man in dasselbe Protokoll ein im kryptographischen Sinne sicheres Signatursystem einsetzt, d.h. wenn der algebraische Beweis eine (hoffentlich einfachere) Variante des kryptographischen ist. (Diese Ansicht setzt natürlich voraus, daß man die Definitionen der Sicherheit des Protokolls bzw. der Nachrichtenunkenntnis im kryptographischen Modell nicht für zu stark hält und die der Signatursysteme nicht für zu schwach.)

Die obige Frage muß negativ beantwortet werden. Im Ausblick werden mögliche Konsequenzen aus dieser Antwort für zukünftige Protokollbeweise kurz angedeutet.

Der Stil der Kapitel, die völlig neu sind, also vor allem 3.4 und 4, ist formaler als der der übrigen, die (in der Hoffnung, daß diese Arbeit ohne vorherige Kenntnis der Modelle verständlich wird) vor allem anderweitig in der Literatur verstreute Sachverhalte zusammenfassen. Bei ersteren sollte ja die Korrektheit im einzelnen überprüfbar sein, während die Einzelheiten bei letzteren bei Bedarf an den angegebenen Literaturquellen nachzuschlagen sind. Die Kernaussagen der Arbeit müßten jedoch auch verständlich sein, wenn die wirklich formalen Teile, die als „Lemma", „Beweis", „Anmerkung" o. ä. gekennzeichnet und mit „◊" abgeschlossen sind, übersprungen werden.

2 Algebraische Modellierung

Zunächst wird in 2.1 der Begriff eines (realen) digitalen Signatursystems soweit präzisiert, wie das für diese Modellierung notwendig ist. Die Idealisierung dieses Systems, die die Grundlage der algebraischen Modellierung bildet, wird in 2.2 anhand der Regeln für die Nachrichtenunkenntnis, die in diesem Modell gelten sollen, erläutert. In 2.3 wird noch kurz auf die Darstellung anderer kryptographischer Systeme als der Signatursysteme aus 2.1 in diesem Modell eingegangen, insbesondere auf Verschlüsselung.

2.1 Digitale Signaturen

Ein digitales Signatursystem muß drei Tätigkeiten gestatten [DiHe_76]: Schlüsselerzeugung, Unterschreiben von Nachrichten und Testen von Unterschriften. Ergebnis der Schlüsselerzeugung sind dabei immer zwei Schlüssel, ein Signier- und ein Testschlüssel. Der erste dient dem Erzeuger der beiden Schlüssel im folgenden zum Signieren und wird demgemäß geheimgehalten, der zweite hingegen soll das Testen dieser Unterschriften gestatten und wird dazu veröffentlicht.

Der zur Schlüsselerzeugung verwendete Algorithmus wählt dabei einen Signierschlüssel innerhalb einer großen Menge $\mathcal{S}$ möglicher Schlüssel zufällig aus, so daß man ihn als eine Funktion auffassen kann, die die Folge der im Algorithmus verwendeten Zufallszahlen auf einen gültigen Schlüssel abbildet. Ist $\mathcal{Z}$ die Menge der endlichen Zufallszahlenfolgen, so läßt sich die Erzeugung des Signierschlüssels darstellen als

$$\text{Erz:} \quad \mathcal{Z} \to \mathcal{S} .$$

Ohne Beschränkung der Allgemeinheit kann man annehmen, daß zu jedem Signierschlüssel s genau ein Testschlüssel s* gehört, den man damit als t(s) schreiben kann („oBdA" deswegen, weil man andernfalls das Paar (s, s*) insgesamt als neuen Signierschlüssel verwenden kann). Ist $\mathcal{T}$ die Menge der möglichen Testschlüssel, so ist also t eine Funktion

$$\text{t:} \quad \mathcal{S} \to \mathcal{T} .$$

Oft wird dabei $\mathcal{T}=\mathcal{S}$ sein.

Das Signieren ist im einfachsten Fall eine Funktion U (Unterschrift), die einem Signierschlüssel s und einer Nachricht N eine neue Nachricht U(s,N), die Signatur, zuordnet. Ist $\mathcal{N}$ der Nachrichtenraum, so ist also

$$\text{U:} \quad \mathcal{S} \times \mathcal{N} \to \mathcal{N} .$$

Das Testen der Unterschriften wird durch eine Funktion Test dargestellt, die einem Testschlüssel t(s) und zwei Nachrichten N, N' genau dann „wahr" zuordnet, wenn N' die Signatur zu N mit s ist, also

$$\text{Test:} \quad \mathcal{S} \times \mathcal{N} \times \mathcal{N} \to \{\text{wahr, falsch}\}$$

mit

$$\text{Test} (t(s), N, N') = \text{wahr} \quad \Leftrightarrow \quad N' = U (s, N) .$$

Neben dieser funktionalen muß das Signatursystem natürlich noch eine zweite wichtige Eigenschaft haben:

U(s, •) darf ohne Kenntnis von s nicht zu berechnen sein.

Die genauere Interpretation dieses Satzes, nämlich in einer vereinfachenden, idealisierenden oder einer

realen Weise, unterscheidet nun die beiden Modellierungsarten.

2.2 Nachrichtenunkenntnis und Ungleichungen

Die erste Idealisierung der algebraischen Modellierung besagt, daß man das, was man nicht berechnen können sollte, tatsächlich in überhaupt keiner Weise berechnen kann. Genauer werden die Fähigkeiten eines Angreifers zu jedem Zeitpunkt in diesem Modell darauf beschränkt, eine gewisse Menge von Funktionen („die er berechnen kann") auf eine gewisse Menge von Nachrichten („die ihm bis zu diesem Zeitpunkt bekannten") anzuwenden.

Formal werden seine Fähigkeiten also angegeben durch eine Menge $\mathcal{E}$ von ihm explizit bekannten Nachrichten und eine Menge $\mathcal{F}$ von Funktionen. $\mathcal{E}$ ist dabei immer eine Teilmenge von $\mathcal{N}$und wird bei Modellierung eines konkreten Protokolls und Betrachtung eines bestimmten Zeitpunkts i. allg. drei Sorten von Nachrichten enthalten:

- Solche, die jeder Teilnehmer von vornherein kannte, also etwa alle Zahlen (wie 17 im Beispiel aus Bild 2) und gewisse für das Protokoll nötige Klartextnachrichten (in Bild 2 repräsentiert durch „ok" und eine interessante Nachricht N, die der Angreifer gern von einem anderen Teilnehmer unterschrieben bekäme, etwa eine Überweisung auf sein Konto),
- die eigenen Schlüssel des Angreifers (s in Bild 2), und
- die Nachrichten, die er in vorangegangenen Protokollschritten von anderen Teilnehmern erhalten hat (U(s', 6) in Bild 2).

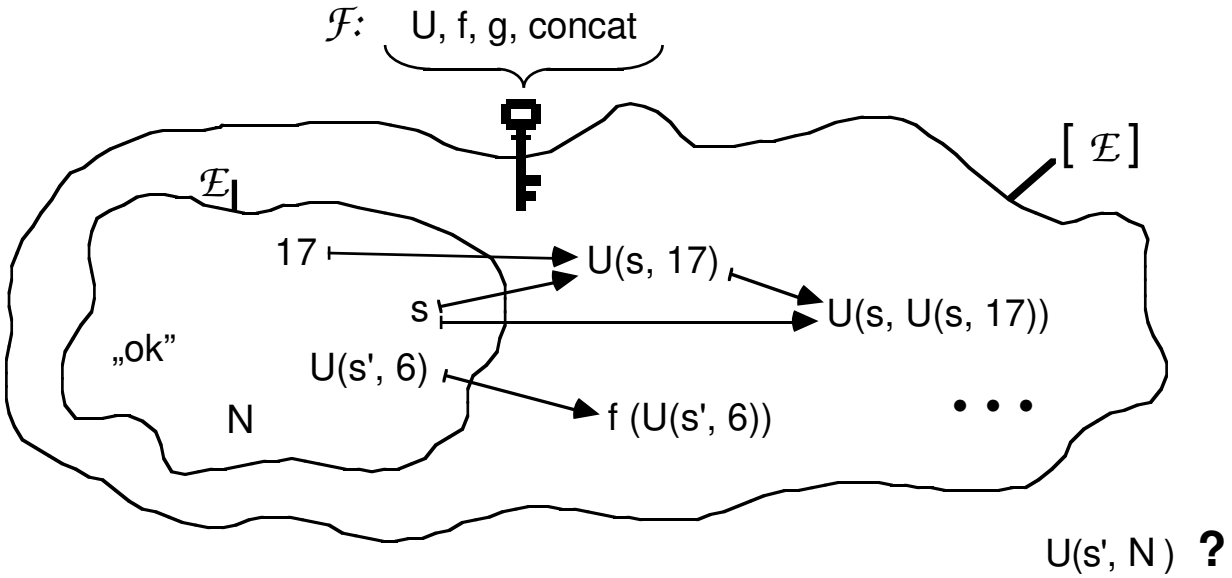


Bild 2 Beispiel für die Nachrichtenkenntnis eines Angreifers

$\mathcal{F}$ wird die Funktionen U und Test enthalten und daneben alle sonstigen für das Protokoll nötigen Funktionen (z.B. in Bild 2 „concat" für die Konkatenation von Nachrichten und zwei weitere Funktionen f und g, die für arithmetische Funktionen wie Inkrement und + stehen könnten).

Die gesamte Menge an Nachrichten, die der Angreifer kennt, wird nun durch $[\mathcal{E}]$, die abgeschlossene Hülle von $\mathcal{E}$ unter den Funktionen aus $\mathcal{F}$, modelliert.

Als Beispiel sind in Bild 2 in $[\mathcal{E}]$ außer $\mathcal{E}$ die Nachricht U(s,17), die durch Signieren mit U aus

dem eigenen Schlüssel s und 17 erhalten wird, gezeichnet, dann U(s,U(s,17)), das durch Signieren aus der vorigen Nachricht hervorgeht, aber auch f(U(s',6)), das durch Anwenden von f auf eine von einem anderen Teilnehmer unterschrieben erhaltene Nachricht entsteht.

Dem Zweck der Modellierung entsprechend sollte nun im Beispiel aus Bild 2 gelten, daß U(s',N), also die mit dem Schlüssel s' eines anderen Teilnehmers unterschriebene Überweisung an den Angreifer, nicht in [$\mathcal{E}$] liegt. Für reale Mengen $\mathcal{N}$ und $\mathcal{E}$ wird dies aber i.allg. nicht stimmen, da in gewisser Weise jeder den ganzen Nachrichtenraum kennt. (Beispielsweise ist einerseits $\mathcal{N}$ eine Menge von Zeichenreihen, die sich auch als natürliche Zahlen auffassen lassen, also gilt U(s',N) $\in$ |N. Andererseits sollte |N in $\mathcal{E}$ liegen, also erst recht in [$\mathcal{E}$].)

Diese Tatsache nützt allerdings dem Angreifer nicht viel, da er wissen müßte, welches n $\in$ |N gerade gleich U(s',N) ist, und die Bestimmung hiervon genauso aufwendig ist wie die ursprüngliche Berechnung von U(s',N), die man ausgeschlossen hat.

Diese Beobachtung führt zur zweiten Idealisierung des Modells: Man möchte im Modell anstelle des realen Nachrichtenraumes einen anderen, $\mathcal{N}^*$ verwenden, in dem solche „unnützen" Gleichungen, die den Teilnehmern nicht bekannt sind, erst gar nicht gelten. Dementsprechend sollen dann auch statt der realen Funktionen auf $\mathcal{N}$ entsprechende mit gleichen Namen und möglichst ähnlichen Eigenschaften auf $\mathcal{N}^*$ verwendet werden.

Beispielsweise sollte also in $\mathcal{N}^*$ für alle Nachrichten M, N, N' mit N≠M, alle Schlüssel s, s' mit s≠s' und alle einstelligen Funktionen f gelten:

$$U(s,N) \neq U(s',N'),$$
$$U(s,N) \neq f(U(s,M))$$

und

$$U(s,N) \neq N', \text{ falls } N' \in \mathcal{N} \text{ Grundnachricht ist.}$$

Dies vom Modell zu verlangen, hat natürlich nur dann Sinn, wenn es einen solchen Nachrichtenraum $\mathcal{N}^*$ gibt, denn wenn die bisherigen Gleichungen und Ungleichungen bereits widersprüchlich wären, so könnte man aus ihnen beliebiges folgern, so daß ein Beweis in diesem Modell nicht die geringste Aussagekraft besäße.

Naheliegend ist es, zur Konstruktion eines solchen $\mathcal{N}^*$ die Terme zu verwenden, die aus den zu von vornherein in irgendwelchen $\mathcal{E}$ liegenden Nachrichten und den Funktionen in $\mathcal{F}$ gebildet werden können, also z.B. die Ausdrücke „U(s,N)", „f(U(s',M))" usw., die ja in $\mathcal{N}^*$ alle verschieden sein sollten. Dabei müssen allerdings diejenigen, deren Gleichheit von vornherein und für jedes reale Signatursystem bekannt ist, zu jeweils einer Klasse zusammengefaßt werden, z.B. „Test (t(s), N, U(s,N))" mit „wahr" und „-((+(U(s,N),1),1)" mit „U(s,N)". Diese Klassen bilden die Elemente von $\mathcal{N}^*$. Das Modell auf $\mathcal{N}^*$ einer Funktion f auf $\mathcal{N}$ ordnet der Klasse eines Terms tr die Klasse des Terms f(tr) zu.

Zur endgültigen Präzisierung dieses Modells (d.h. der genauen Bedingung, wann Terme zur selben Klasse gehören), kann man auf die Theorie der abstrakten Datentypen zurückgreifen, die dann auch die Existenz von $\mathcal{N}^*$ garantiert (vgl. [EhMa_85], wo auch die genauen Definitionen von „Term" usw. nachzulesen sind): Für $\mathcal{N}^*$ wird eine Spezifikation *Spez* angegeben [EhMa_85, Definition 1.14]; diese besteht aus

- den drei Mengen von Namen für
  - Sorten (z.B. {$\mathcal{S}$, $\mathcal{T}$, $\mathcal{N}$}),
  - Grundnachrichten (diese entsprechen der Vereinigung der Mengen $\mathcal{E}$ aller Teilnehmer zu Systembeginn) mit Angabe ihrer Sorte und

- Funktionen ($\mathcal{F}$) mit Angabe ihrer Stelligkeit und der Sorten der Definitions- und Bildbereiche

(dieser Teil einer Spezifikation heißt „Signatur", dieses Wort soll hier aber wegen der Verwechslungsmöglichkeit mit digitalen Signaturen so weit wie möglich vermieden werden)

- und aus einigen vorzugebenden allquantifizierten Gleichungen, nämlich „Test(t(s), N, U(s,N)) =wahr" für das Signatursystem und solchen, die die nicht kryptographischen Funktionen charakterisieren.

Dabei soll es mindestens eine Grundnachricht s der Sorte $\mathcal{S}$ geben und U und Test in keiner außer der eben genannten Gleichung vorkommen.

$\mathcal{N}^*$ selbst wird definiert als initiale Semantik dieser Spezifikation [EhMa_85, Definition 2.10].

In zwei von mehreren äquivalenten Formulierungen heißt dies gerade, daß $\mathcal{N}^*$ diejenige Quotientenstruktur der freien Termalgebra über $\mathcal{S}pez$ ist (oder dazu isomorph), bei der genau diejenigen Terme in einer gemeinsamen Klasse liegen,

1. deren Auswertungen in jedem Modell von $\mathcal{S}pez$ gleich sind [EhMa_85, Definition 2.10 mit 2.5 und 2.4] (d.h. die für jedes Unterschriftensystem und alle diese Spezifikation erfüllenden weiteren Funktionen gleich sind) bzw.
2. deren Gleichheit sich mit Hilfe eines gewissen Gleichungskalküls (das im wesentlichen aus Regeln für das Einsetzen von Gleichungen ineinander besteht) aus den vorgegebenen Gleichungen herleiten läßt [EhMa_85, Definition 5.3 und Satz 5.5].

(Dies präzisiert den obigen Begriff von „nützlichen" bzw. „bekannten" Gleichungen, die gelten sollen, während zwischen allen übrigen Termen Ungleichheit gilt).

Genaueres Eingehen darauf, wie man im allgemeinen Fall in einem so spezifizierten Modell die eigentlich interessierenden Ungleichungen beweist und die Tatsache, daß gewisse Nachrichten nicht in gewissen [$\mathcal{E}$] liegen, erübrigt sich, da in Kapitel 4 einige Beispiele reichen werden, zu zeigen, daß dieses Modell zu stark idealisiert und folglich abgewandelt werden muß.

Zur Abkürzung der Schreibweise werden im folgenden in den Mengen $\mathcal{E}$ die Grundnachrichten, die keine Schlüssel sind, nicht mehr erwähnt, da sie sowieso allen Teilnehmern bekannt sind. Im in Bild 2 dargestellten Beispiel würde also die Nachrichtenkenntnis zu [{s, U(s', 6)}] abgekürzt.

2.3 Algebraische Modellierung für andere kryptographische Systeme

Zunächst ist zu erwähnen, daß auch digitale Signaturen bisher manchmal anders modelliert wurden als gemäß 2.1 [Waid_85], nämlich nach Art des aus dem Kryptosystem RSA abgeleiteten Signatursystems. Dort gibt es eine Funktion Extr (für „Extraktion" der Nachricht aus einer Unterschrift)

$$\text{Extr:} \quad \mathcal{T} \times \mathcal{N} \rightarrow \mathcal{N}$$

mit der Eigenschaft

$$\text{Extr}\,(\,t(s),\,N'\,) \;=\; N \quad \Leftrightarrow \quad N' \;=\; U\,(s,\,N)\,. \qquad (*)$$

für alle s, N und N'. Die Funktion „Test" kann damit durch

$$\text{Test}\,(\,t(s),\,N,\,N'\,) \;=\; \text{wahr} \quad \Leftrightarrow \quad \text{Extr}\,(\,t(s),\,N'\,) \;=\; N\,.$$

definiert werden. Demgemäß wurden zur algebraischen Modellierung dieses Systems U und Extr statt

U und Test verwendet.

Da aber eine Funktion „Extr" mit dieser Eigenschaft von der Idee eines Signatursystems her, im Gegensatz zu „Test", nicht existieren muß und bei neueren, kryptographisch sichereren (vgl. Kap. 3) Unterschriftenssystemen auch nicht existiert (zumindest nicht offensichtlich), war es sinnvoll, die Modellierung zu wechseln.

Angemerkt sei, daß man natürlich aus jedem realen digitalen Signatursystem eines machen kann, in dem man der Unterschrift die unterschriebene Nachricht ansieht, indem man U durch U' ersetzt mit U'(s,N) := (N, U(s,N)). Dann existiert eine Funktion „Extr", für die stets Extr(t(s),U'(s,N))=N gilt. Diese ist aber einfach die Projektion auf die erste Komponente der Nachricht und kann nicht zum Testen der Korrektheit der Unterschrift verwendet werden, d.h. sie erfüllt von der obigen Äquivalenz (*) nur den Teil „⇐".

Ursprünglich entwickelt wurde die algebraische Modellierung von Kryptoprotokollen für Kryptosysteme mit öffentlichen Schlüsseln (und ihre Anwendung in besonders einfachen, sogenannten Ping-Pong-Protokollen) [DoYa_81]. Solche enthalten wie digitale Signatursysteme einen Algorithmus zur Erzeugung zweier Schlüssel s und s*, von denen einer geheim bleibt und einer veröffentlicht wird. Ersterer dient zum Ent-, letzterer zum Verschlüsseln, und s* kann wieder als Funktion v(s) von s aufgefaßt werden. Geht man noch davon aus, daß s und s* aus derselben Menge $\mathcal{S}$ stammen, so würde man Ver- und Entschlüsselungsfunktion als

$$V: \quad \mathcal{S} \times \mathcal{N} \to \mathcal{N}$$

$$E: \quad \mathcal{S} \times \mathcal{N} \to \mathcal{N}$$

mit

$$E(s, V(v(s),N)) = N$$

schreiben. Hiermit kann man wie in 2.2 idealisierte Nachrichtenmengen definieren.

Allerdings wurde bisher bei dieser Modellierung oft ([DoYa_81, DoYa_83, DoEk_82], nicht jedoch [Merr_83]) zusätzlich angenommen, daß das Entschlüsseln nicht nur links-, sondern auch rechtsinvers zum Verschlüsseln sei, d.h. daß zusätzlich die Gleichung

$$V(s, E(v(s), N)) = N$$

gelte, was wieder eine spezielle Eigenschaft von RSA ist, aber nicht notwendigerweise für jedes Kryptosystem gelten muß und bei neueren, kryptographisch sicheren (vgl. Kap. 3) tatsächlich nicht gilt. In den Fällen können also die Ergebnisse sicher nicht im Sinne von 1.4 auf das kryptographische Modell übertragen werden.

Des weiteren ist die Modellierung der Nachrichtenkenntnis eines Angreifers hier komplizierter als für digitale Signaturen, weil es nun darum geht, daß der Angreifer, der etwa V(s, 17) erhält und s nicht kennt, darin die Nachricht „17", die ihm real bekannt ist, nicht erkennt. Deshalb werden in diesen Modellen bereits die zu verschlüsselnden Klartextnachrichten so aufgeteilt, daß die verschiedenen Teilnehmer disjunkte Teilmengen davon „kennen", d.h. in $\mathcal{E}$ haben; damit kann das Entschlüsselnkönnen einer Nachricht wieder dadurch modelliert werden, daß die zugrundeliegende Klartextnachricht in $[\mathcal{E}]$ liegt.

[Merr_83] verallgemeinert diese Modellierung noch so, daß zusätzlich zur Nachrichtenunkenntnis z.B. auch Anonymität modelliert werden kann, worauf hier nicht gesondert eingegangen werden soll. Außerdem gibt es Modelle, die absichtlich speziell auf RSA zugeschnitten sind [EvGS_85, ChEv_87], d.h. noch weitere Gleichungen enthalten, die wohl ausschließlich bei RSA gelten, teils um zu untersuchen, ob diese Eigenschaften übliche Protokolle prinzipiell stören müssen, teils um

Protokolle zu modellieren, die von diesen Eigenschaften Gebrauch machen. Diese Modelle in den Vergleich einzubeziehen, erscheint nicht sinnvoll, da es bisher bzgl. der kryptographischen Modellierung nicht nur keine sicheren Kryptosysteme mit diesen Zusatzeigenschaften gibt, sondern nicht einmal eine Definition, was darunter genau zu verstehen wäre.

3 Kryptographische Modellierung

Im Gegensatz zur algebraischen Modellierung beschäftigt sich die kryptographische mit den realen statt mit idealisierten kryptographischen Systemen, nimmt also etwa bei digitalen Signatursystemen nicht an, daß man U(s, •) ohne s überhaupt nicht erhalten könnte, sondern betrachtet, mit welchen Schwierigkeiten dies real verbunden ist.

Bezüglich der verwendeten Nachrichten- und Schlüsselräume, der Funktionen und der Beschreibung der Protokolle handelt es sich also (innerhalb der digitalen Welt) eigentlich nicht um eine Modellierung, sondern um die Realität. Modellierung ist aber natürlich die Überlegung, wann diese Systeme als sicher gelten sollen.

Dies darf allerdings nicht dahingehend mißverstanden werden, daß die kryptographischen Systeme bereits die gesamte Realität bildeten. Das wäre nur dann der Fall, wenn alle Teilnehmer sämtliche Rechnungen im Kopf ausführen könnten und ihre Schlüssel diesen nie verließen. Andernfalls ist zusätzlich die Schnittstelle zwischen den Teilnehmern und ihren Rechnern, die die kryptographischen Funktionen für sie berechnen, zu betrachten, etwa mit der Möglichkeit, daß der Rechner samt Schlüssel gestohlen wird (was wiederum durch PINs u.ä. zu vermeiden versucht wird.)

In 3.1, 3.2 wird zunächst eine möglichst starke Definition für die Sicherheit von digitalen Signatursystemen (nach [GoMR_84, GoMR_88, GoMi_84]) vorgestellt. Dabei wird in 3.1 betrachtet, welche Komplexität man für das Brechen eines solchen Systems verlangen kann, und in 3.2 präzisiert, wann ein digitales Signatursystem als gebrochen gilt. In 3.3 werden kurz die existierenden kryptographisch sicheren digitalen Signatursysteme vorgestellt. In 3.4 wird die Modellierung auf Protokollbeweisteile, also Nachrichtenunkenntnis und Ungleichungen, erweitert. In 3.5 wird mit wenigen Sätzen auf andere Anwendungen der kryptographischen Modellierung verwiesen.

3.1 Komplexität

Die einzig perfekte Definition für die Sicherheit irgendeines kryptographischen Systems ist die informationstheoretische [Shan_49] für Kryptosysteme mit privaten Schlüsseln, die besagt, daß ein Angreifer über die vor ihm zu verbergende Nachricht überhaupt keine Information erhält.

Etwas Vergleichbares ist bei digitalen Signatursystemen prinzipiell nicht möglich, denn durch die Möglichkeit, Unterschriften zu testen, hat ein Angreifer im Grunde immer die vollständige Information darüber, was korrekte Unterschriften sind. Mit beliebig viel Zeit könnte er dies auch ausnützen, indem er alle möglichen Unterschriften durchprobiert. Man kann also die Sicherheit höchstens komplexitätstheoretisch definieren, indem man verlangt, daß

- Schlüsselerzeugung, U(s, •) bei Kenntnis von s und Test „schnell”,
- U(s, •) bei Kenntnis nur von t(s) aber „langsam”

seien. (Insbesondere muß das Durchprobieren wegen der Länge der Unterschriften und Schlüssel so

lange dauern, daß innerhalb vernünftiger Zeiträume die Erfolgschance völlig zu vernachlässigen ist).

Im Grunde interessiert man sich natürlich dafür, wie langsam das Brechen des Systems bei bestimmten Schlüssel- und Unterschriftenlängen ist, da man sich in der Praxis auf solche festlegen muß. Da aber zur Zeit in der Komplexitätstheorie passende untere Schranken nicht bewiesen werden können, betrachtet man den Aufwand, wie bei fast allen Algorithmen, asymptotisch und präzisiert „langsam" durch „nicht polynomial".

Die Länge der Eingabe von U, in der ja der Aufwand gemessen wird, hängt dabei im wesentlichen von der Schlüssellänge n ab, die daher auch als Sicherheitsparameter bezeichnet wird, denn die Länge l des anderen Eingabeparameters, der zu unterschreibenden Nachricht, kann nicht unbedingt vom Entwerfer des digitalen Signatursystems variiert werden.

Anzumerken ist, daß die Möglichkeit zum Testen der Unterschriften das Brechen auf jeden Fall NP-leicht (vgl. [GaJo_79]) macht: Ein passender Algorithmus besteht darin, nichtdeterministisch eine Nachricht zu wählen und zu testen, ob sie zufällig die gewünschte Unterschrift ist. Dies dauert bei der akzeptierenden Rechnung, d.h. derjenigen, wo tatsächlich die richtige Unterschrift geraten wird, nur polynomial lang (in $n+l$, bei kurzen Nachrichten somit auch in n), weil die Existenz eines „schnellen" Algorithmus zum Unterschreiben bei Kenntnis von s impliziert, daß die Länge der Unterschrift polynomial in $n+l$ ist, und der Aufwand des Tests wiederum polynomial hierin ist. Der Beweis, daß das Brechen eines digitalen Signatursystems nicht in P wäre, würde also $P \neq NP$ implizieren.

Umgekehrt würde ein Beweis von NP=P nicht unbedingt die Idee eines digitalen Signatursystems zum Scheitern verurteilen: Könnte man z.B. von einem System beweisen, daß der Aufwand der notwendigen Algorithmen nur n^2 wäre, der des Brechens hingegen mindestens n^{10}, so ließe sich sicher auch ein passender Sicherheitsparameter für den praktischen Einsatz wählen.

Daß man trotzdem die Definition verwendet, daß das Brechen nicht polynomial sein dürfe, ist vor allem durch zweierlei motiviert: Einerseits kann man bei dieser Definition, wenn schon nicht auf einen vollständigen Sicherheitsbeweis, so doch wenigstens auf einen Reduktionsbeweis hoffen, daß das Brechen das Lösen eines wohluntersuchten, als schwierig geltenden Problems impliziert (wie sie beim Beweis von NP-Vollständigkeit verwendet werden). Andererseits ist diese Definition ziemlich maschinenunabhängig.

Nach dem bisherigen könnte man erwarten, daß die Definition der Sicherheit lauten würde, daß das Berechnen von U(s, N) bei Eingabe von t(s) und N nicht in P läge. Die Anforderungen an die Sicherheit sind aber in zwei Punkten schärfer, die nun in 3.1.1 bzw. 3.1.2 erläutert werden sollen. Auf die erste Verschärfung wird in manchen Papieren [GoMR_84, GoMR_88] verzichtet, die zweite ist hingegen essentiell.

3.1.1 Schaltnetzkomplexität

Der Sicherheitsparameter n, also die Schlüssellänge, wird im allgemeinen für lange Zeit festliegen, insbesondere, wenn die Algorithmen in Hardware implementiert werden. Daher kann der Angreifer sich seinen Brechalgorithmus in Abhängigkeit von n wählen, sofern er sich davon Vorteile verspricht, statt einen festen Algorithmus für Schlüssel aller Längen zu verwenden [GoMi_84]. Damit verfügt er insgesamt über eine Folge von Algorithmen für jeweils einen Sicherheitsparameter.

Den Aufwand einer solchen Algorithmenfolge („non-uniform complexity") wird am einfachsten durch seine Schaltnetzkomplexität [Sch4_85, Kapitel 2.1] gemessen. Dabei werden die einzelnen Algorithmen durch boolesche Schaltnetze C_n repräsentiert, die nur aus Gattern mit ein oder zwei Eingängen bestehen dürfen. Es sind dabei alle möglichen Gatter und beliebiges fan-out zugelassen.

Als Aufwand eines Schaltnetzes C wird seine Größe, d.h. die Anzahl seiner Gatter und Ausgänge verwendet, die im folgenden als

$$Gr(C)$$

bezeichnet wird.

(In [Sch4_85] werden ursprünglich nur boolesche Funktionen, d.h. solche mit Bildbereich {0,1} bzw. genau einem Ausgang, betrachtet, und bei der Größe werden folglich keine Ausgänge gezählt. Eine Erweiterung auf mehrere Ausgänge erscheint mir sinnvoll; damit aber nicht z.B. eine Funktion, die bei Eingabe n eine Folge von e^n Nullen ausgibt, als mit polynomialem Aufwand berechenbar gilt, muß man dann die Ausgänge wohl mitzählen.)

Es läßt sich relativ leicht zeigen, daß alles, was sich mit einer Turingmaschine mit polynomialem Zeitaufwand berechnen läßt, auch von einer polynomial wachsenden Schaltnetzfolge berechnet werden kann, die Umkehrung hiervon jedoch nicht gilt [Sch4_85, Seite 22 bzw. 18]. Daraus folgt, daß die Definition der Sicherheit dadurch, daß man dem Angreifer eine polynomial wachsende Schaltnetzfolge statt einer Turingmaschine mit polynomialem Zeitaufwand zugesteht, tatsächlich stärker wird.

Genauer gilt, daß eine Turingmaschine der Zeitkomplexität t(n) und Speicherkomplexität s(n) durch eine Schaltnetzfolge der Größe $t(n) \bullet s(n)$ simuliert werden kann. (Dazu wird zunächst ein kleines Grundschaltnetz konstruiert, das einen Schritt der Turingmaschine auf einem einzigen Feld ausführt. Aus diesen werden Reihen zu s(n) Stück gebildet, also mit einem Grundschaltnetz für jedes evtl. benutzte Feld. t(n) solche Reihen, also eine für jeden Zeitpunkt, zu dem die Turingmaschine evtl. noch läuft, werden hintereinandergeschaltet.) Allerdings wird in 3.2 sichtbar, daß das Brechen eines Systems der Verwendung von Maschinen mit Orakeln entspricht, so daß man eigentlich diese Sätze auf Orakelmaschinen erweitern müßte, was aber nicht schwierig erscheint.

Verwendet man Schaltnetzkomplexität in der Definition der Sicherheit, so pflegt man sie auch bei der Definition des Systems zu verwenden, d.h. man erlaubt, daß auch die notwendigen Algorithmen nicht durch polynomiale Turingmaschinen, sondern durch polynomial wachsende Schaltnetzfolgen durchgeführt werden. Bei allen bisher veröffentlichten konkreten kryptographisch sicheren Systemen (nicht nur digitalen Signaturen) genügen aber polynomiale Turingmaschinen; das dürfte auch so bleiben, da man als „ein System" i. allg. nur einen uniformen Algorithmus bezeichnen wird.

Praktisch ist es derzeit ziemlich gleichgültig, welche Komplexitätsdefinition man verwendet, da für Schaltnetzkomplexität genausowenig wie für Turingkomplexität gezeigt werden kann, daß das Brechen eines Systems wirklich nicht polynomial ist (vgl. 3.1), und die verwendeten Reduktionen von als schwierig geltenden Problemen auf das Brechen der bisher vorgeschlagenen Systeme in beiden Fällen gleichermaßen möglich sind (wobei die Bedeutung von „Brechen" und „Lösen" natürlich in beiden Fällen verschieden sind).

Theoretisch könnte es sogar geschickter sein, mit Turingkomplexität zu arbeiten, da vermutlich aus einer Reduktion im Turingmodell die Gültigkeit der entsprechenden Reduktion im Schaltnetzmodell folgt, nicht jedoch umgekehrt.

Ein weiterer Grund, warum speziell die Papiere über digitale Signaturen bisher mit Turingkomplexität arbeiten, könnte sein, daß man durch ein Schaltnetz nicht nur die Schlüssellänge, sondern auch die Nachrichten(maximal)länge festlegt, wogegen die bisherigen Unterschriftensysteme (vgl. 3.3) das Unterschreiben beliebig langer Nachrichten als einen einzigen Block gestatten. Daher muß die Folge der Schaltnetze nicht nur vom Sicherheitsparameter, sondern von der Summe aus diesem und der Nachrichtenlänge abhängen, was zu häßlichen Formulierungen führen könnte. Da allerdings die Länge der Unterschrift nicht von der Nachricht, sondern nur vom Sicherheitsparameter

abhängt, betrifft dieses Argument nur das Unterschreiben und Testen, nicht jedoch das Brechen.

Im folgenden wird Schaltnetzkomplexität verwendet, weil sie sich etwas anschaulicher darstellen läßt und es dank der größeren Publizität der Turingkomplexität leichter sein dürfte, sich vorzustellen, daß alles folgende auch mit Turingkomplexität möglich ist, als das umgekehrt für Schaltnetzkomplexität der Fall wäre.

3.1.2 Komplexität für fast alle Fälle

Wichtiger noch als die Beobachtung, daß der Angreifer seinen Algorithmus abhängig von n wählen könnte, ist die, daß für das Brechen eines digitalen Signatursystems die übliche Komplexität des schlimmsten Falles nicht genügt, sondern gleichmäßige Schwierigkeit gefordert werden muß. Wäre das Brechen beispielsweise NP-vollständig, gäbe es aber einen schnellen Algorithmus, der für ein hundertstel aller Schlüssel das System brechen könnte, so hieße dies, daß ein Angreifer im Mittel für jeden hundertsten Teilnehmer Unterschriften fälschen könnte, was natürlich nicht toleriert werden kann. (Daß man in diesem Fall evtl. selbst testen könnte, ob man einen solchen „schwachen" Schlüssel erwischt hat und ggf. einen anderen wählen, ist kein Gegenargument, da dies eine Erweiterung des Schlüsselerzeugungsalgorithmus, also eine Änderung des Systems ist; man muß nun die Sicherheit des neuen Systems beweisen.)

Das andere Extrem, daß es keinen Algorithmus gebe, der auch nur für einen Schlüssel Unterschriften erzeugt, läßt sich natürlich auch nicht realisieren, da bereits die Kombination aus Schlüsselerzeugung und anschließendem Unterschreiben mit dem erzeugten Schlüssel einen solchen bildet.

Daher verlangt man, daß es keine „schnelle" Möglichkeit gibt, für einen signifikanten Anteil ε der Schlüssel das System zu brechen. ε muß dabei so klein gewählt werden, daß die Wahrscheinlichkeit, daß sich jemals ein von einem anderen Teilnehmer gewählter Schlüssel in diesem Anteil befindet, zu vernachlässigen ist. Die Schnelligkeit der Algorithmen und ε werden dabei i.allg. voneinander abhängen. Es erscheint sinnvoll, die Sicherheit so zu definieren, daß mit wachsendem Sicherheitsparameter das noch mögliche ε schneller sinkt als $1/P(n)$ für jedes Polynom P.

Damit gelangt man zu folgender Beinahe-Definition („Beinahe-", weil „Brechen" noch nicht definiert ist, was in 3.2 nachgeholt wird):

Beinahe-Definition 1: Ein digitales Signatursystem heiße sicher, wenn
für alle Polynome P und Q (* P für das Wachstum von $1/\varepsilon$, Q für das der Schaltnetze *)
ein $N \in \mathbb{N}$ existiert,
so daß für alle $n \geq N$ gilt: (* für alle Sicherheitsparameter ab einer gewissen Größe *)

$$\min \{ Gr(C_n) \mid C_n \text{ bricht das System für einen Anteil} \geq 1/P(n) \text{ der Schlüssel der Länge n} \} \geq Q(n). \lozenge$$

Man könnte versuchen, die Sicherheitsdefinition weiter zu verschärfen, indem man probabilistische Algorithmen zuläßt (die im folgenden tatsächlich benötigt werden).

Als Schaltnetze stellt man diese dadurch dar, daß das Schaltnetz eine Zufallszahl als zusätzliche Eingabe erhält. Eine möglichst starke Definition für Sicherheit gegen diese Schaltnetze erhält man, wenn man in der obigen „C_n bricht das System für einen Anteil $\geq 1/P(n)$ der Schlüssel der Länge n" durch

„C_n bricht

für einen Anteil ≥ 1/P(n) der Schlüssel der Länge n
das System mit einer Wahrscheinlichkeit ≥ 1/P(n) ”

ersetzt. (Für die Wahrscheinlichkeit ein drittes Polynom R einzuführen, würde sicherlich nichts ändern, da man immer ein Polynom bilden könnte, das größer als P und R ist, und dies als P im bisherigen Sinne verwenden.) Gemäß dem folgenden Lemma kann man die Sicherheitsdefinition aber in ihrer einfacheren Gestalt lassen. Man beachte jedoch, daß dieser Beweis nicht ohne weiteres auf Turing-Maschinen übertragbar ist, da für jeden Sicherheitsparameter einzeln argumentiert wird. Wenn man also Turing-Maschinen statt Schaltnetzen verwendet, muß man probabilistische Definitionen verwenden, damit der Rest der Arbeit gilt.

Lemma 1: Diese Definition ist zur vorigen äquivalent. ◊

Beweis: Sei das System unsicher gegen probabilistische Schaltnetze.
Dann gibt es Polynome P und Q, so daß
für alle $N \in \mathbb{N}$ ein $n \geq N$ existiert, (* d.h. es existieren beliebig große Sicherheitsparameter *)
für das es ein Schaltnetz C_n mit $Gr(C_n) \leq Q(n)$ gibt, das
für einen Anteil ≥ 1/P(n) der Schlüssel der Länge n
das System mit einer Wahrscheinlichkeit ≥ 1/P(n) bricht.

Es genügt zu zeigen, daß es für dieselben n jeweils ein deterministisches Schaltnetz D_n derselben Größe wie C_n gibt, das das System für einen Anteil $\geq 1/P^2(n)$ der Schlüssel der Länge n bricht.

D_n konstruiert man, indem man an den probabilistischen Eingang von C_n diejenige Zahl z* fest anlegt, die für das Brechen am geeignetsten ist: Sei

- $\mathcal{Z}_n$ die Menge dieser Zufallszahlen,
- $\mathcal{S}_n$ die Menge der Schlüssel der Länge n,
- $C_n(z, \bullet)$ bezeichne für jedes $z \in \mathcal{Z}_n$ das Schaltnetz, das aus C_n durch festes Anlegen von z an den probabilistischen Eingang entsteht,
- Qua(z) für $z \in \mathcal{Z}_n$ die „Qualität” von z (aus Angreifersicht), d.h. die Anzahl der $s \in \mathcal{S}_n$, für die $C_n(z, \bullet)$ das System bricht, und
- z* eine der Zahlen aus $\mathcal{Z}_n$ mit $Qua(z^*) = \max \{Qua(z) \mid z \in \mathcal{Z}_n\}$. (Da $\mathcal{S}_n$ endlich ist, existiert solch ein z*.)

Es gilt einerseits nach den Voraussetzungen über C_n:

$$\begin{aligned} & |\{ (z, s) \in \mathcal{Z}_n \times \mathcal{S}_n \mid C_n(z, \bullet) \text{ bricht das System für } s\}| \\ = & \sum_{s \in \mathcal{S}_n} |\{z \in \mathcal{Z}_n \mid C_n(z, \bullet) \text{ bricht das System für } s\}| \\ \geq & \ 1/P(n) \bullet |\mathcal{S}_n| \bullet 1/P(n) \bullet |\mathcal{Z}_n| \end{aligned}$$

andererseits

$$\begin{aligned} & |\{ (z, s) \in \mathcal{Z}_n \times \mathcal{S}_n \mid C_n(z, \bullet) \text{ bricht das System für } s\}| \\ = & \sum_{z \in \mathcal{Z}_n} |\{s \in \mathcal{S}_n \mid C_n(z, \bullet) \text{ bricht das System für } s\}| \\ = & \sum_{z \in \mathcal{Z}_n} Qua(z) \\ \leq & \ |\mathcal{Z}_n| \bullet Qua(z^*), \end{aligned}$$

also

$$Qua(z^*) \geq 1/P^2(n) \bullet |\mathcal{S}_n|,$$

was noch zu zeigen war. ◊

Anmerkung 1: Am Inhalt der Definition ändert sich auch nichts, wenn man die Wahrscheinlichkeit über die beiden Zufallsauswahlen der Schlüssel, statt einzeln, zusammen bildet, d.h. den Satz

„C_n bricht das System
für einen Anteil ≥ 1/P(n) der Paare
aus einem Schlüssel der Länge n und einer Zufallszahl"

anstelle des bisherigen verwendet.

Klar ist, daß ein im vorigen Sinne unsicheres System auch in diesem Sinne unsicher ist, denn die dortigen Schaltnetze C_n brechen das System für einen Anteil ≥ $1/P(n)^2$ der Paare aus einem Schlüssel der Länge n und einer Zufallszahl, und P^2 ist auch ein Polynom.

Sei nun ein System unsicher im neuen Sinn und C_n so ein Schaltnetz, das das System für einen Anteil ≥ 1/P(n) der Paare bricht. Die Situation ist auch in Bild 3 dargestellt.

Sei a der Anteil an Schlüsseln der Länge n, für die das System mit einer Wahrscheinlichkeit ≥ 1/(2•P(n)) gebrochen wird. Für diesen Anteil a wird das System höchstens mit der Wahrscheinlichkeit 1 gebrochen, für den restlichen Anteil 1-a höchstens mit der Wahrscheinlichkeit 1/(2•P(n)). Für die Gesamtwahrscheinlichkeit w (d.h. den Anteil der Paare von Schlüssel und Zufallszahl), mit der C_n das System bricht, gilt also

$$1/P(n) \leq w \leq a\bullet 1 + (1-a)\bullet 1/(2\bullet P(n)) = a\bullet(1-1/(2\bullet P(n))) + 1/(2\bullet P(n)),$$

was umgeformt werden kann als

$$1/(2\bullet P(n)) \leq a \bullet (1-1/(2\bullet P(n))) \Leftrightarrow 1 \leq a \bullet (2\bullet P(n)-1) \Leftrightarrow 1 / (2\bullet P(n)-1) \leq a.$$

Dabei ist

$$1 / ((2\bullet P(n)-1)) \geq 1 / (2\bullet P(n)),$$

also bricht C_n für einen Anteil ≥ 1/(2•P(n)) der Schlüssel der Länge n das System mit einer Wahrscheinlichkeit ≥ 1/(2•P(n)). ◊

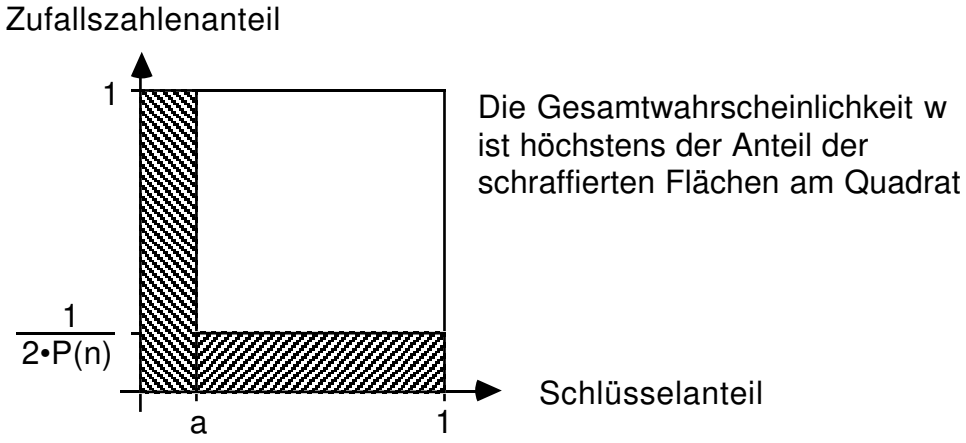


Bild 3 Darstellung von Schlüssel- und Zufallszahlenanteilen zu Anmerkung 1

Die Anmerkung 1 gilt übrigens im Gegensatz zum Lemma in analoger Form auch wieder für Turing-

Maschinen, d.h. man muß zwar probabilistische Turing-Maschinen betrachten, kann aber die Wahrscheinlichkeiten auf beide Arten definieren.

3.2 „Brechen" speziell bei digitalen Signaturen

Die Sicherheitsdefinition für digitale Signatursysteme macht man so sicher wie möglich, indem man davon ausgeht, daß der Angreifer die für ihn günstigsten denkbaren Bedingungen hat, und bereits sagt, das System sei gebrochen, wenn er sehr schwache Erfolge aufweisen kann.

Die günstigsten Bedingungen für einen Angreifer liegen dann vor, wenn der Besitzer eines Schlüssels s ihm jederzeit jede beliebige Nachricht N_i unterschreibt, und der schwächstmögliche Erfolg besteht darin, daß der Angreifer daraus zu irgendeinem Zeitpunkt irgendeine zusätzliche Unterschrift, genauer ein Paar (N, U(s, N)) aus Nachricht und zugehöriger Unterschrift, erzeugen kann (die Nachricht muß also nicht von ihm vorherbestimmbar sein und auch nicht unbedingt sinnvoll). Ersteres wird als ein Angriff mit adaptiv gewählten Nachrichten (adaptive chosen message attack) bezeichnet, letzteres als beliebige Fälschung (existential forgery) [GoMR_84, GoMR_88].

Bei Einsatz innerhalb von Protokollen sind diese Bedingungen nicht einmal sehr weit von der Realität entfernt: Im Beispiel von Bild 1 (in 1.2) unterschreibt die Bank eine Garantie, die zumindest als einen Teil die Frage enthält, die aus ihrer Sicht ziemlich (und wenn man sie aus Datenschutzgründen verschlüsselt, völlig) beliebig ist. Auch könnte je nach Protokoll der Anteil der Nachrichten, die einem Angreifer nützen, an der Gesamtnachrichtenmenge recht groß sein, z.B. würde ein Angreifer nicht unbedingt eine bestimmte Überweisung unterschrieben bekommen wollen, sondern mit jeder beliebigen ab einem bestimmten Betrag zufrieden sein.

Für die Folge (C_n) von Schaltnetzen, durch die der Angreifer repräsentiert werden sollte (3.1.1), heißt dies folgendes (vgl. Bild 4): C_n hat

- eine Testschlüsseleingabe te,
- einige Zwischenausgaben za_i (i=1,...,m), an denen zwischendurch die Nachrichten ausgegeben werden, die der Schlüsselbesitzer unterschreiben soll,
- zu den za_i korrespondierende Zwischeneingaben ze_i (i=1,...,m), an denen die Eingabe der Unterschriften erwartet wird,
- eine Nachrichtenausgabe na, an der die Nachricht ausgegeben wird, deren Unterschrift der Angreifer erzeugt hat
- eine Unterschriftenausgabe ua für die erzeugte Unterschrift.

Alle Ein- und Ausgaben sind natürlich Leitungsbündel. Dabei hat te eine feste Breite (oBdA n, da man das System so abändern kann, daß s mindestens so lang wie t(s) ist), ebenso die Leitungsbündel für Unterschriften; die Beschränkung der Nachrichtenlängen ergibt sich durch die Größe von C_n. Bei den Zwischenausgaben kann zusätzlich eine Leitung vorgesehen sein, die anzeigt, ob an dieser Stelle überhaupt eine Anfrage gestellt wird, da ein Angreifer die Möglichkeit hat, bei verschiedenen Eingaben verschieden viele Anfragen zu stellen.

Analog zur Notation für Turingmaschinen, die zwischendurch Anfragen stellen dürfen, werden solche Schaltnetze als Schaltnetze mit Orakeln bezeichnet.

In Bild 4 ist ein C_n als Platine mit angedeutetem Schaltnetz gezeichnet, auf der za_i und ze_i jeweils so angeordnet sind, daß man dort einen Chip einsetzen kann, der das Unterschreiben mit dem Schlüssel s, bzgl. dessen das System gebrochen werden soll, durchführt.

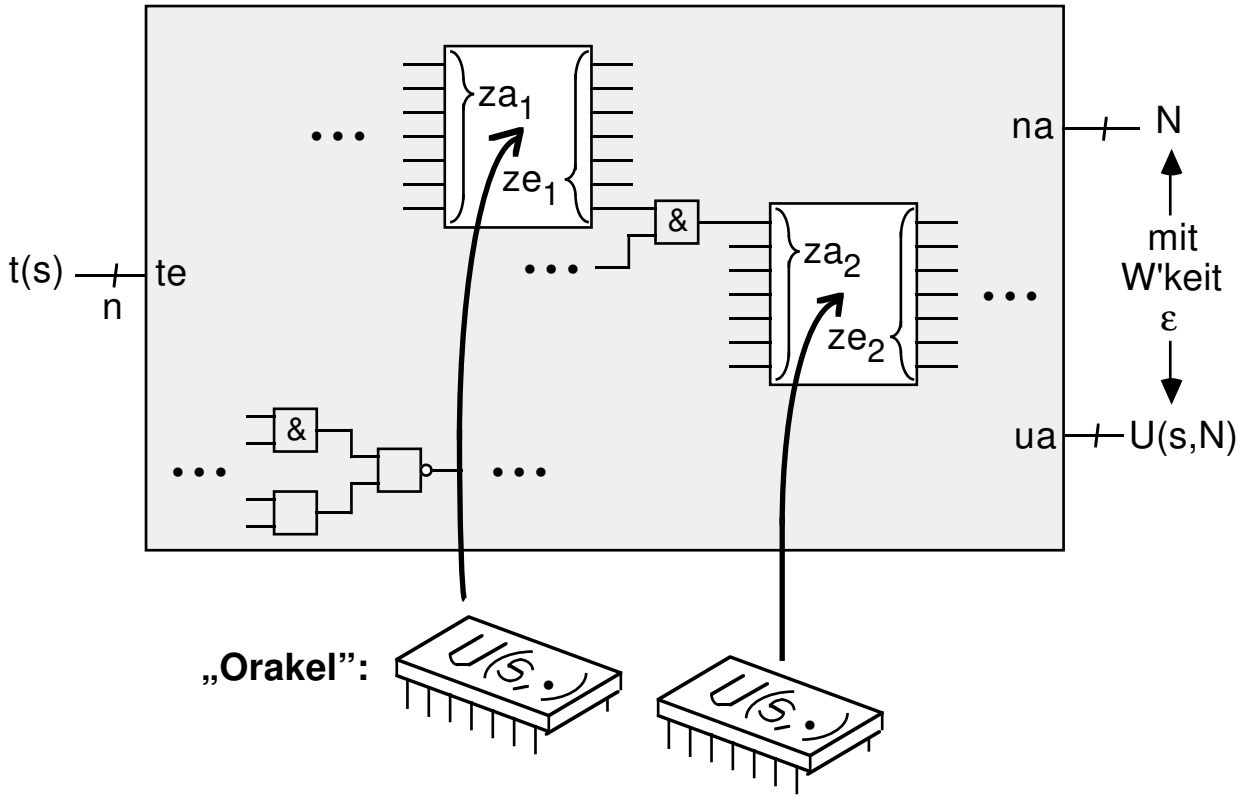


Bild 4 Schaltnetz zum Brechen eines digitalen Signatursystems

Als Ergänzung zur Beinahe-Definition 1, die damit zur Definition wird, kann man nun definieren, wann C_n das System für einen bestimmten Schlüssel gebrochen hat:

Definition 2: Ein Schaltnetz C_n gemäß Bild 4 bricht ein digitales Signatursystem für einen Schlüssel s genau dann, wenn

$$te = t(s) \wedge \forall i: [\, ze_i = U(s, za_i)] \;\Rightarrow\; na \notin \{za_i \mid i=1,...,m\} \wedge \text{Test}\,(te, na, ua) = \text{wahr}$$

(d.h. daß unter der Voraussetzung korrekter Eingaben die ausgegebene Nachricht neu und die Unterschrift korrekt ist). ◊

Man kann sich noch fragen, weshalb in Definition 1 und 2 überhaupt die Möglichkeit vorgesehen wird, daß der Angreifer manchmal eine falsche Unterschrift ausgibt, denn er könnte ja selbst feststellen, daß sie falsch ist und z.B. weitersuchen. Hier handelt es sich nur um die einfachste Definition dafür, was das Schaltnetz ausgeben soll, wenn wegen der Aufwandsbeschränkung überhaupt keine Unterschrift gefunden wird.

3.3 Existierende „kryptographisch sichere" digitale Signatursysteme

Wie schon mehrfach erwähnt, würde der vollständige Beweis der Sicherheit eines digitalen Signatursystems im Sinne von Definition 1 und 2 implizieren, daß P≠NP gilt. Da ein Beweis hiervon bisher nicht möglich war, muß man sich zur Zeit auf Reduktionen beschränken.

„Kryptographisch stark" („cryptographically strong") nennt man daher heutzutage ein System, für

das gezeigt wurde, daß sich aus jedem Algorithmus, der es bricht, ein Algorithmus zur schnellen Lösung eines wesentlich überschaubareren, länger bekannten, wohluntersuchteren Problems konstruieren läßt, für das ein solcher noch nicht gefunden wurde.

Das Problem, das dafür üblicherweise herangezogen wird, ist die Faktorisierung großer Zahlen, die das Produkt zweier ebenfalls großer Primfaktoren sind. Die genaue Formulierung der Schwierigkeit des Problems muß dabei, damit Reduktionen möglich sind, genau zur gewählten Sicherheitsdefinition passen; bei der hier gewählten lautet die entsprechende

Faktorisierungsannahme:
Für alle Polynome P und Q (* P für das Wachstum von 1/ε, Q für das der Schaltnetze *)
existiert $N \in \mathbb{N}$,
so daß für alle $n \geq N$ gilt: (* für alle Sicherheitsparameter ab einer gewissen Größe *)
$\min \{ Gr(C_n) \mid C_n$ faktorisiert einen Anteil $\geq 1/P(n)$ der Zahlen,
die Produkte aus zwei Primfaktoren von n/2 Dualstellen sind$\}$
$\geq Q(n)$. ◊

Die schnellsten derzeit bekannten Faktorisierungsalgorithmen, die nicht nur für bestimmte Zahlentypen funktionieren, scheinen alle einen asymptotischen Aufwand von etwa

$$e^{\sqrt{n \bullet \ln(n)}}$$

zu haben (vgl. [Pome_84] für einen Überblick, und für den darin noch nicht enthaltenen mittels elliptischer Kurven [Lens_86]). Sie sind also zwar nicht mehr rein exponentiell in n, wie die naive Faktorisierung mittels Division durch alle ungeraden Zahlen bis zur Wurzel der zu faktorisierenden Zahl, aber auch nicht annähernd polynomial.

Praktisch können zur Zeit damit bei großem Hardware- und Programmieraufwand noch Produkte von zwei etwa 50-dezimalstelligen Primzahlen faktorisiert werden. (Beispielsweise berichtet [DaHo_88] über die konkrete Faktorisierung von Zahlen der Größenordnung 10^{70}, [PoST_88] über ein Konzept für Maschinen, mit denen bei Einsatz von 50 000 $ Zahlen der Größenordnung 10^{100} in 2 Wochen, bei Einsatz von 10 000 000 $ solche der Größenordnung 10^{144} in einem Jahr faktorisiert werden können müßten.) Zur Erzielung von Sicherheit muß n also wesentlich größer als $\log_2(10) \bullet 100$, möglichst sogar größer als 500 gewählt werden.

Angemerkt sei, daß Faktorisierung nicht als NP-vollständig bekannt ist, und daß auch bisherige Versuche mit auf NP-vollständigen Problemen basierenden kryptographischen Systemen scheiterten, da die Schwierigkeit der Probleme nicht gleichmäßig genug war (auf dem Knapsack-Problem basierende Kryptosysteme mit öffentlichen Schlüsseln, vgl. z.B. [Sham_84, Bric_85]).

In [GoMR_84, GoMR_88] wurde erstmals ein kryptographisch starkes digitales Signatursystem vorgestellt, allerdings unter Verwendung von Turingkomplexität (vgl. 3.1.1), basierend auf der entsprechenden Faktorisierungsannahme, eingeschränkt auf sogenannte Blumzahlen, d.h. Zahlen von der Bauart $n=p \bullet q$, wobei p und q Primzahlen gleicher Länge mit $p \equiv 3$ und $p \equiv 7 \bmod 4$ sind.

Es ist allerdings nicht rein funktional, sondern sowohl probabilistisch als auch gedächtnisbehaftet, d.h. eine Unterschrift hängt von den bisherigen Unterschriften und einigen Zufallszahlen ab. Außerdem gehört bei diesem System zum Schlüssel eine Zahl 2^b, die angibt, wieviele Nachrichten mit diesem Schlüssel maximal unterschrieben werden können (weil die Unterschriften Blättern eines Authentikationsbaumes entsprechen, dessen Tiefe bekannt sein muß). Letzteres schränkt die Brauchbarkeit des Systems nicht wesentlich ein, da z.B. für b=n schon 2^n Unterschriften gebildet werden dürfen, was vom Aufwand her völlig undurchführbar ist (da genauso aufwendig wie das

Durchprobieren aller Schlüssel). In der Praxis kann b wesentlich kleiner als n sein, etwa 10 bis 20. Mit diesem b und für Nachrichten der Länge *l* ist der Aufwand des Systems etwa folgender:

- Länge einer Unterschrift:

  $$3 \bullet b \bullet n$$

  (unabhängig von der Länge der Nachricht, die man i. allg. aber noch dazuschreiben muß).
- Aufwand zum Unterschreiben (mit der in [GoMR_88] schon berücksichtigten Verbesserung aus [Gol1_87]) laut letzterem Artikel:

  $$O(n^3 + l \bullet n)$$
- Aufwand zum Testen:

  $$O(b \bullet n^3 + l \bullet n^2)$$
- Der Speicherbedarf ist zu vernachlässigen, da nur die Anzahl der bisher unterschriebenen Nachrichten und die letzte Unterschrift aufgehoben werden müssen.

Es gibt eine gedächtnislose und deterministische Variante dieses Systems (vgl. [GoMR_88, Abschnitt 10.3]), allerdings mit erhöhtem Aufwand zum Unterschreiben. Daher kann man für theoretische Zwecke von einem in der Art von Abschnitt 2.1 dargestellten System ausgehen, was im folgenden zur Vereinfachung der Schreibweise stets getan wird. Insbesondere genügt zum Beweis, daß die algebraische Modellierung nicht korrekt ist, ein Gegenbeispiel mit dem funktionalen System. Zumindest das zweite Gegenbeispiel (4.2.4) ließe sich aber auch mühelos auf das schnellere, nicht funktionale System übertragen. Hingegen muß beim Protokollentwurf der Unterschied beachtet werden; z.B. könnte bei einem funktionalen System jemand seine eigene Unterschrift U(s,N) prüfen, indem er N nochmal unterschreibt und vergleicht, in einem probabilistischen System geht dies nicht.

Außerdem wird stets nach möglichst schwachen Bedingungen gesucht, die die Existenz gewisser kryptographischer Systeme garantieren. Für digitale Signaturen war das in [GoMR_88] die Existenz von Familien von „clawfree" (deutsch etwa „konfluenzfreien") Paaren von Permutationen, mittlerweile ist es die Existenz von Falltüreinwegfunktionen [BeMi_88]. Auf die Darstellung dieses Aspekts soll hier verzichtet werden.

3.4 Ungleichungen und Nachrichtenunkenntnis

Nun sollen Unkenntnis von und Ungleichungen zwischen Nachrichten bestimmter Bauart im kryptographischen Modell definiert werden. Dies wird ein bißchen allgemeiner durchgeführt, als es im Rest der Arbeit benötigt wird, da so eine eventuelle zukünftige Weiterverwendung bei kryptographischer Modellierung von Kryptoprotokollen erleichtert wird.

Die Modellierung der Nachrichtenunkenntnis soll eine Verallgemeinerung der Unkenntnis fremder Unterschriften sein, d.h. es muß sich um eine Definition ähnlich Beinahe-Definition 1 (vgl. 3.1.2) handeln. Da hier die Kenntnis des Angreifers bei einem einzelnen Protokollschritt betrachtet werden soll, braucht kein aktiver Angriff modelliert zu werden, die für den Angreifer verwendeten Schaltnetze benötigen also keine Orakel. Festzulegen ist, wie man die Bauart einer Nachricht angibt und die Nachrichten, die der Angreifer bisher erhalten hat.

Der erste Unterschied zwischen einer allgemeinen Nachricht und einer einfachen Unterschrift ist, daß erstere von mehreren Schlüsseln (oft, aber nicht immer, von lauter verschiedenen Teilnehmern) abhängen kann, von denen manche dem Angreifer bekannt sind, andere nicht. Bei der Verwendung eines Signatursystems innerhalb von Protokollen muß also der Zusammenhang dieser Schlüssel betrachtet werden.

Dazu wird zunächst festgelegt, daß Teilnehmer, die sich an das Protokoll halten, jeden Schlüssel unabhängig von allen anderen gemäß einer Gleichverteilung aus dem Schlüsselraum wählen müssen. Im Normalfall, wenn jeder Schlüssel einem anderen Teilnehmer gehört, ist dies die einzig praktisch sinnvolle Weise. Die Bedingung würde aber wahrscheinlich auch in späteren Sicherheitsbeweisen benötigt; folglich wären auch in einem ansonsten als sicher bewiesenen Protokoll Teilnehmer, die aus mnemotechnischen Gründen verschiedene Schlüssel ähnlich wählen, nicht unbedingt geschützt (bzw. man müßte für diesen Fall einen gesonderten Beweis führen).

Angreifer hingegen dürfen, weil man in einem Sicherheitsbeweis immer vom schlimmstmöglichen Fall ausgehen muß, ihre Schlüssel nach beliebigen Kriterien wählen, also insbesondere auch mehrere in Abhängigkeit voneinander oder von den schon veröffentlichten Testschlüsseln braver Teilnehmer. Damit spielen Schlüssel des Angreifers keine Sonderrolle gegenüber anderen Nachrichten, die der Angreifer wählen kann, weshalb sie in der folgenden Definition nicht eigens genannt werden.

Während die Festlegung der Zusammenhänge zwischen den Schlüsseln ziemlich kanonisch war, hat man bei der Definition einer Bauart mehrere Möglichkeiten. In jedem Fall kann man aber die sich ergebende Bauart bei festen Schlüsseln als Teilmenge der Nachrichtenmenge $\mathcal{N}$ auffassen (nämlich als Menge aller Nachrichten dieser Bauart). Berücksichtigt man noch die Abhängigkeit der Nachrichten von den im konkreten Fall gewählten Schlüsseln, so kann man im folgenden durch die Schreibweise

$$N \in \mathcal{B}(s_1,\ldots,s_k)$$

ausdrücken, daß N die Bauart $\mathcal{B}$ hat, wenn als Schlüssel $s_1,\ldots,s_k$ gewählt wurden. Ganz formal ist also jede Bauart $\mathcal{B}$ eine Abbildung

$$\mathcal{B}: \quad \mathcal{S}^k \to \mathsf{P}(\mathcal{N})$$

für irgendein k; $\mathcal{B}$ heißt dann auch Bauart mit k Schlüsseln. Diese Definition ist allgemein genug, obwohl ein Protokoll im Prinzip das Vorkommen von beliebig vielen Schlüsseln gestattet (z.B. indem ein Grundprotokoll wie die Bestellung einer Ware, beliebig oft, auch ineinander verschachtelt, mit neuen Schlüsseln ausgeführt werden kann), da jede Nachricht zu einem bestimmten Zeitpunkt gewählt wird und nur von einer festen Anzahl von Schlüsseln abhängen kann.

Bei der Festlegung eines bestimmten $\mathcal{B}$ kann man zwischen expliziten und impliziten Definitionen unterscheiden. Explizite legen die Bauart durch die innere Struktur der Nachrichten, d.h. eine Konstruktionsvorschrift, fest, implizite hingegen durch eine Bedingung an die Nachrichten. Als einfaches Beispiel kann die Bauart der Unterschrift mit Schlüssel s unter die Nachricht N explizit als $\mathcal{B}(s) := \{U(s, N)\}$, implizit als $\mathcal{B}(s) := \{N \mid \text{Test}(t(s), N, U(s, N)) = \text{wahr}\}$ definiert werden.

Dieser Unterschied wäre z.B. dann wesentlich, wenn man von einem nicht funktionalen U ausginge (wodurch U(s, N) keine eindeutige Schreibweise mehr wäre und man zwischen der Unterschrift unter Verwendung einer bestimmten Zufallszahl und einer beliebigen Unterschrift unterscheiden müßte) oder wenn Test (t(s), N, M) auch von manchen Nachrichten M erfüllt würde, die gar nicht als U(s, N) konstruiert wurden (was impliziert, daß solche Nachrichten genauso schwer zu finden sein müssen wie eigentliche Unterschriften).

Praktisch erscheint es am sinnvollsten, beide Sorten von Definitionen zu verwenden. Für die Nachrichten, die der Angreifer schon kennt, bietet sich eine explizite Definition an, weil diese in den vorangegangenen Protokollschritten explizit konstruiert werden mußten. Bezüglich der Bauart, zu der er keine Nachricht kennen soll, wirkt hingegen eine implizite Definition angemessener, weil die Gefährlichkeit dieser Nachricht (vgl. 1.3) darin liegt, welchen Bedingungen sie genügt (wie sie konstruiert wurde, sehen ihr ja die anderen Teilnehmer bzw. ein Gericht nicht mehr an).

Im folgenden wird nur der Fall betrachtet, daß die Nachrichten, die der Angreifer kennt, durch die

Schlüsselwahl eindeutig festliegen, da dies die Definitionen vereinfacht und für den Rest der Arbeit genügt. (Beim Beweis praktischer Protokolle, die i. allg. auch bei den zu sendenden Nachrichten Wahlmöglichkeiten, d.h. Indeterminismus, zulassen, könnte hier eine Erweiterung notwendig werden.) Dazu heiße $\mathcal{B}$ eine einelementige Bauart, wenn $\mathcal{B}(s_1,\ldots,s_k)$ für alle Tupel $(s_1,\ldots,s_k)$ einelementig ist.

Bevor solche Bauarten in Definitionen verwendet werden, zur Verdeutlichung noch einige Beispiele:

Beispiel: Alle vier folgenden Bauarten gehen von der Situation aus, daß es eine Zentrale gibt, die Schlüssel normaler Benutzer authentisieren soll. Dazu hat sie einen eigenen Schlüssel s_Z, und die Authentikation eines Benutzerschlüssels s_B bestehe in der Nachricht

$$\text{Auth}(s_Z, s_B) := s_B \,\&\, U(s_Z, s_B \& \text{„ist gültig”})),$$

wobei & die Konkatenation bezeichnet.

1) Hat ein Angreifer sich einen Schlüssel s_A gewählt, den er gern authentisiert hätte, so interessiert er sich für die Bauart

$$\mathcal{B}_1: \quad \mathcal{S}^1 \to \mathsf{P}(\mathcal{N})$$
$$s_1 \to \{\text{Auth}(s_1, s_A)\}$$

2) Wählt er seinen Schlüssel s_A in Abhängigkeit von $t(s_Z)$, z.B. als $3 \bullet t(s_Z)$, so geht es um die Bauart

$$\mathcal{B}_2: \quad \mathcal{S}^1 \to \mathsf{P}(\mathcal{N})$$
$$s_1 \to \{\text{Auth}(s_1, 3 \bullet t(s_1))\}$$

3) Ist ihm hingegen ganz gleichgültig, was er für ein s_A authentisiert bekommt, so geht es um die Bauart

$$\mathcal{B}_3: \quad \mathcal{S}^1 \to \mathsf{P}(\mathcal{N})$$
$$s_1 \to \{\text{Auth}(s_1, s_A) \mid s_A \in \mathcal{S}\}$$

4) Möchte er zuletzt die Authentikation des Schlüssels s_B eines anderen Benutzers kennenlernen, so interessiert er sich für

$$\mathcal{B}_4: \quad \mathcal{S}^2 \to \mathsf{P}(\mathcal{N})$$
$$(s_1, s_2) \to \{\text{Auth}(s_1, s_2)\}$$

Hierbei waren alle Bauarten außer $\mathcal{B}_3$ einelementig. ◊

Nun folgt die Definition für Nachrichtenunkenntnis:

Definition 3: Seien Bauarten $\mathcal{B}, \mathcal{A}_1,\ldots,\mathcal{A}_j: \mathcal{S}^k \to \mathsf{P}(\mathcal{N})$ gegeben, davon $\mathcal{A}_1,\ldots,\mathcal{A}_j$ einelementig. Man sagt, ein Angreifer, der Nachrichten der Bauarten $\mathcal{A}_1,\ldots,\mathcal{A}_j$ kennt, könne keine Nachricht der Bauart $\mathcal{B}$ erzeugen, wenn

für alle Polynome P und Q (* P für das Wachstum von 1/ε, Q für das der Schaltnetze *)
  ein $N \in \mathbb{N}$ existiert,
  so daß für alle $n \geq N$ gilt: (* für alle Sicherheitsparameter ab einer gewissen Größe *)

$$\min \{ \text{Gr}(C_n) \mid C_n \text{ erzeugt für einen Anteil} \geq 1/P(n) \text{ der Tupel } (s_1,\ldots,s_k) \text{ von Schlüsseln der Länge } n \text{ bei Eingabe von } t(s_1),\ldots,t(s_k)$$

und $N_1 \in \mathcal{A}_1(s_1,\ldots,s_k)$, … , $N_j \in \mathcal{A}_j(s_1,\ldots,s_k)$
eine Nachricht aus $\mathcal{B}(s_1,\ldots,s_k)$ }
$\geq Q(n)$. ◊

Analog zu Bild 4 ist in Bild 5 auch ein Schaltnetz skizziert, das dem Angreifer in Definition 3 entspricht.

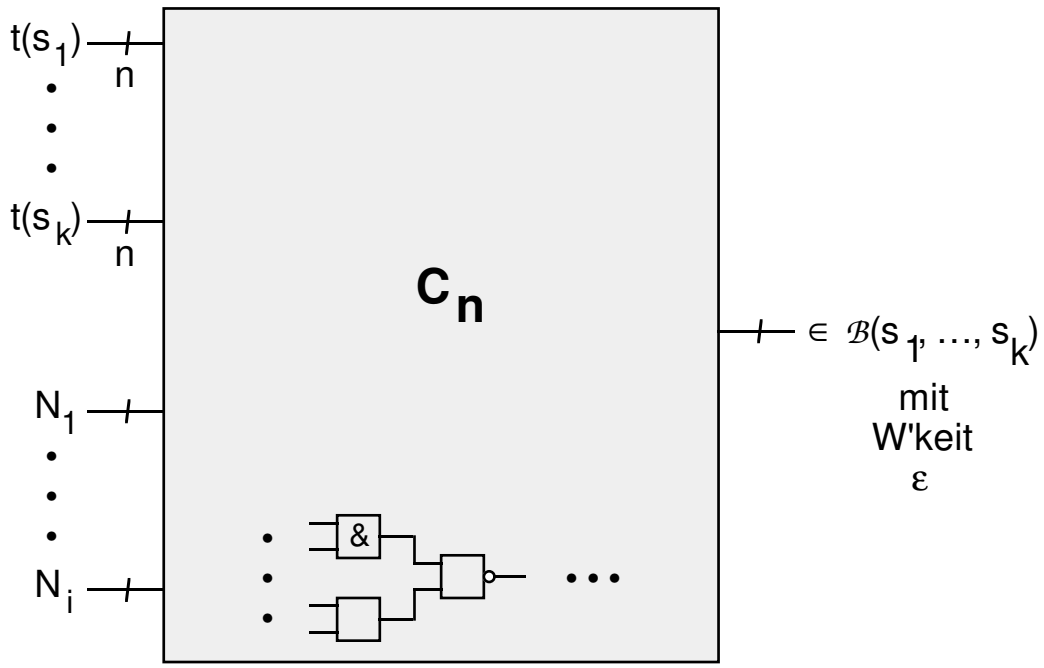


Bild 5 Schaltnetz für das Kennen einer Nachricht der Bauart $\mathcal{B}$

Auch die Ungleichheit von Nachrichten soll nicht für konkrete Exemplare (wo sie, im Gegensatz zur algebraischen Modellierung, die ganz normale Ungleichheit im realen Nachrichtenraum ist), sondern zwischen Bauarten definiert werden. Innerhalb von Protokollbeweisen wird diese Art der Ungleichheit benötigt, um zu zeigen, daß ein Angreifer Nachrichten einer Bauart $\mathcal{B}$ nicht mit vernünftigen Erfolgsaussichten erzeugen kann, indem er einfach Nachrichten einer Bauart $\mathcal{C}$ verwendet. (Dementsprechend wird häufig $\mathcal{B}$ implizit, $\mathcal{C}$ explizit definiert sein.) Zu diesem Zwecke erscheint in Bezug auf Definition 3 folgende Definition sinnvoll (bis auf die hier zur Vereinfachung vorausgesetzte Einelementigkeit):

Definition 4: Seien einelementige Bauarten $\mathcal{B}, \mathcal{C}: \mathcal{S}^k \rightarrow \mathsf{P}(\mathcal{N})$ gegeben. $\mathcal{B}$ und $\mathcal{C}$ heißen kryptographisch verschieden, wenn
für alle Polynome P (* für das Wachstum von 1/ε *)
ein $N \in \mathbb{N}$ existiert,

so daß für alle $n \geq N$ gilt: (* für alle Sicherheitsparameter ab einer gewissen Größe *)

Der Anteil der Tupel $(s_1,\ldots,s_k)$ von Schlüsseln der Länge n mit

$$\mathcal{B}(s_1,\ldots,s_k) = \mathcal{C}(s_1,\ldots,s_k)$$

ist kleiner als 1/P(n) ◊

Bei der Anwendung von Definition 3 oder 4 wird benötigt, daß alle beteiligten Bauarten die gleiche Anzahl an Schlüsseln, die der Angreifer nicht kennt, aufweisen. Dazu müssen sicher in vielen Fällen manche Bauarten um „überflüssige" Schlüssel erweitert werden, d.h. solche, von denen die Bauart nicht abhängt.

Zum Beispiel kann man sich fragen, ob der Angreifer eine Nachricht der Bauart $\mathcal{B}_4$ aus dem obigen Beispiel anhand der Testschlüssel $t(s_Z)$ und $t(s_B)$ konstruieren kann. Dazu muß man die Bauarten der Testschlüssel so definieren, daß beide Schlüssel vorkommen, also für $t(s_Z)$

$$\begin{aligned}\mathcal{B}_5: \quad & \mathcal{S}^2 \to \mathsf{P}(\mathcal{N}) \\ & (s_1, s_2) \to \{t(s_1)\}\end{aligned}$$

und für $t(s_B)$

$$\begin{aligned}\mathcal{B}_6: \quad & \mathcal{S}^2 \to \mathsf{P}(\mathcal{N}) \\ & (s_1, s_2) \to \{t(s_2)\}\end{aligned}$$

Welcher der Schlüssel als s_1 und welcher als s_2 modelliert wird, liegt dabei schon durch die Definition von $\mathcal{B}_4$ fest. Hier soll nun noch gezeigt werden, daß die Definitionen 3 und 4 von der Festlegung der Schlüsselreihenfolge und dem Hinzunehmen von zu vielen Schlüsseln unabhängig sind, wie man das erwarten sollte. Benötigt wird dies später insbesondere, wenn man die Bauarten aus Termen des algebraischen Modells gewinnt.

Definition 5: Eine Bauart $\mathcal{B}'$: $\mathcal{S}^{k'} \to \mathsf{P}(\mathcal{N})$ heißt Erweiterung von $\mathcal{B}$: $\mathcal{S}^k \to \mathsf{P}(\mathcal{N})$ mit $k<k'$, wenn $\mathcal{B}'(s_1,\ldots,s_{k'}) = \mathcal{B}(s_1, \ldots,s_k)$ für alle $s_1,\ldots,s_{k'} \in \mathcal{S}$.

Eine Bauart $\mathcal{B}'$: $\mathcal{S}^k \to \mathsf{P}(\mathcal{N})$ geht durch eine Schlüsselpermutation π aus $\mathcal{B}$: $\mathcal{S}^k \to \mathsf{P}(\mathcal{N})$ hervor (wobei π eine Permutation von $\{1,\ldots,k\}$ sein muß), wenn $\mathcal{B}'(s_1,\ldots,s_k) = \mathcal{B}(s_{\pi(1)},\ldots,s_{\pi(k)})$ für alle $s_1,\ldots,s_k \in \mathcal{S}$. ◊

Lemma 2: Einelementigkeit von Bauarten und die Definitionen 3 und 4 sind unter Erweiterungen und Schlüsselpermutationen invariant, wobei natürlich alle in einer Definition vorkommenden Bauarten gleich erweitert bzw. ihre Schlüssel gleich permutiert werden müssen. ◊

Beweis: Es sind sechs Aussagen zu beweisen.

Einelementigkeit bei Erweiterung: Ist $\mathcal{B}'$ Erweiterung von $\mathcal{B}$, so ist $\mathcal{B}'$ einelementig genau dann, wenn alle $\mathcal{B}'(s_1,\ldots,s_{k'})$ einelementig sind, also genau dann, wenn alle $\mathcal{B}(s_1,\ldots,s_k)$ einelementig sind.

Einelementigkeit bei Schlüsselpermutation: Geht $\mathcal{B}'$ durch die Schlüsselpermutation π aus $\mathcal{B}$ hervor, so ist $\mathcal{B}'$ einelementig genau dann, wenn alle $\mathcal{B}'(s_1,\ldots,s_{k'})$ einelementig sind, also genau dann, wenn alle $\mathcal{B}(s_{\pi(1)},\ldots,s_{\pi(k)})$ einelementig sind. Da π eine Permutation ist, bilden die Tupel $(s_{\pi(1)},\ldots,s_{\pi(k)})$ wieder den ganzen Raum $\mathcal{S}^k$, d.h. letzteres gilt genau dann, wenn $\mathcal{B}$ einelementig ist.

Definition 4 bei Erweiterungen: Sind $\mathcal{B}'$, $\mathcal{C}'$ Erweiterungen von $\mathcal{B}$ bzw. $\mathcal{C}$, so ist $\mathcal{B}'(s_1,\ldots,s_{k'}) = \mathcal{C}'(s_1,\ldots,s_{k'})$ genau dann, wenn $\mathcal{B}(s_1,\ldots,s_k) = \mathcal{C}(s_1,\ldots,s_k)$, und der Anteil an Tupeln $(s_1,\ldots,s_{k'})$, für die letzteres gilt, ist gleich dem an Tupeln $(s_1,\ldots,s_k)$.

Definition 4 bei Schlüsselpermutation: Gehen $\mathcal{B}'$, $\mathcal{C}'$ durch die Schlüsselpermutation π aus $\mathcal{B}$ bzw. $\mathcal{C}$ hervor, so ist $\mathcal{B}'(s_1,\ldots,s_k) = \mathcal{C}'(s_1,\ldots,s_k)$ genau dann, wenn $\mathcal{B}(s_{\pi(1)},\ldots,s_{\pi(k)}) =$

$\mathcal{C}(s_{\pi(1)},\ldots,s_{\pi(k)})$. Da die Tupel $(s_{\pi(1)},\ldots,s_{\pi(k)})$ wieder genau den Raum $\mathcal{S}^k$ ausschöpfen, ist der Anteil an Tupeln $(s_1,\ldots,s_k)$, für die letzteres gilt, gleich dem, für den $\mathcal{B}(s_1,\ldots,s_k) = \mathcal{C}(s_1,\ldots,s_k)$ gilt.

Definition 3 bei Schlüsselpermutation: Der Beweis der Invarianz von Definition 3 sei zuerst für den einfacheren Fall geführt, daß $\mathcal{B}'$ und $\mathcal{A}_1',\ldots, \mathcal{A}_j'$ durch die Schlüsselpermutation π aus $\mathcal{B}, \mathcal{A}_1,\ldots, \mathcal{A}_j$ hervorgehen. Da umgekehrt auch $\mathcal{B}, \mathcal{A}_1,\ldots, \mathcal{A}_j$ durch π^{-1} aus $\mathcal{B}', \mathcal{A}_1',\ldots, \mathcal{A}_j'$ hervorgehen, genügt es, eine Richtung zu beweisen, z.B. die, daß der Angreifer, wenn er (unter den passenden Voraussetzungen) Nachrichten der Bauart $\mathcal{B}$ erzeugen kann, auch solche aus $\mathcal{B}'$ erzeugen kann.

Sei dazu eines seiner Schaltnetze C_n zur Erzeugung von Nachrichten der Bauart $\mathcal{B}$ gegeben. Daraus soll ein Schaltnetz C'_n gebildet werden, das aus Eingaben $N_i' \in \mathcal{A}_i'(s_1,\ldots,s_k) = \mathcal{A}_i(s_{\pi(1)},\ldots,s_{\pi(k)})$ und $t(s_1),\ldots,t(s_k)$ ein $B' \in \mathcal{B}'(s_1,\ldots,s_k) = \mathcal{B}(s_{\pi(1)},\ldots,s_{\pi(k)})$ erzeugt.

Gibt man die N_i' an C_n, so hat dies die Funktion, ein solches B_i' zu erzeugen, sofern es als Testschlüsseleingaben $t(s_{\pi(1)}),\ldots,t(s_{\pi(k)})$ erhält, d.h. einfach eine Permutation von $t(s_1),\ldots,t(s_k)$. C'_n unterscheidet sich damit von C_n nur durch die Permutation der Eingaben und hat folglich dieselbe Größe, und es ist für einen gleichgroßen Anteil an Schlüsseln korrekt wie C_n (mit der üblichen Begründung, daß die Tupel $(s_{\pi(1)},\ldots,s_{\pi(k)})$ genau $\mathcal{S}^k$ ausschöpfen).

Definition 3 bei Erweiterungen: Es bleibt die Invarianz von Definition 3 für den Fall zu zeigen, daß $\mathcal{B}'$ und $\mathcal{A}_1',\ldots, \mathcal{A}_j'$ durch eine Erweiterung von k auf k' Schlüssel aus $\mathcal{B}, \mathcal{A}_1,\ldots$ bzw. $\mathcal{A}_j$ hervorgehen. Auch hiervon sei die leichtere Hälfte zuerst bewiesen, nämlich daß der Angreifer, wenn er $\mathcal{B}$ erzeugen kann, auch $\mathcal{B}'$ erzeugen kann.

Wieder sei C_n gegeben. Diesmal soll C'_n aus Eingaben $N_i' \in \mathcal{A}_i'(s_1,\ldots,s_{k'}) = \mathcal{A}_i(s_1,\ldots,s_k)$ und $t(s_1),\ldots,t(s_{k'})$ ein $B' \in \mathcal{B}'(s_1,\ldots,s_{k'}) = \mathcal{B}(s_1,\ldots,s_k)$ erzeugen.

Selbstverständlich ist dies gerade die Funktion von C_n, außer daß diesem die Testschlüssel $t(s_1),\ldots,t(s_k)$ genügen, d.h. C'_n soll die letzten Testschlüssel ignorieren und ansonsten identisch zu C_n sein. Damit ist es gleich groß. C'_n ist für $(s_1,\ldots,s_{k'})$ genau dann korrekt (d.h. es erzeugt eine Nachricht der gewünschten Bauart), wenn C_n für $(s_1,\ldots,s_k)$ korrekt ist, und der Anteil dieser Tupel in $\mathcal{S}^{k'}$ ist gleich dem der $(s_1,\ldots,s_k)$ in $\mathcal{S}^k$.

Umgekehrt sei nun C'_n gegeben und C_n zu bilden. Es soll aus Eingaben $N_i \in \mathcal{A}_i(s_1,\ldots,s_k)$ und $t(s_1),\ldots,t(s_k)$ ein $B \in \mathcal{B}(s_1,\ldots,s_k)$ erzeugen. Dabei gilt $\mathcal{A}_i(s_1,\ldots,s_k) = \mathcal{A}_i'(s_1,\ldots,s_{k'})$ und $\mathcal{B}(s_1,\ldots,s_k) = \mathcal{B}'(s_1,\ldots,s_{k'})$ für beliebige $s_{k+1},\ldots,s_{k'}$.

Es liegt also nahe, C_n im wesentlichen wie C'_n zu bauen und nur die zusätzlichen Testschlüssel $t(s_{k+1}),\ldots,t(s_{k'})$, die letzteres noch benötigt, vorweg selbst zu erzeugen, indem $s_{k+1},\ldots,s_{k'}$ gewählt und t angewendet wird. Da t mit polynomialem Aufwand berechnet werden kann, wächst auch die Größe der so entstehenden C_n nur polynomial im Sicherheitsparameter.

Damit sie auch für einen genügenden Anteil an Schlüsseln korrekt sind, muß allerdings die Wahl von $s_{k+1},\ldots,s_{k'}$ geeignet sein. Dies geschieht zunächst am einfachsten, indem man das Schaltnetz probabilistisch macht und $s_{k+1},\ldots,s_{k'}$ unabhängig voneinander gleichverteilt aus $\mathcal{S}$ wählt. Dies heißt, daß der Angreifer in C_n mit den Schlüsseln, von denen die Bauart nicht abhängt, dasselbe tut, was bisher der brave Benutzer mit ihnen außerhalb von C'_n tat. Insofern ist es klar, daß die Gesamtwahrscheinlichkeit für den Erfolg von C_n gleich der von C'_n ist, d.h. der Anteil an Schlüsseltupeln $(s_1,\ldots,s_{k'})$ in $\mathcal{S}^{k'}$, für die C_n korrekt ist, ist $\geq 1/P(n)$.

Diese Tupel kann man als Paare $((s_1,\ldots,s_k), (s_{k+1},\ldots,s_{k'}))$ aus den eigentlichen Schlüsseln und den Zufallszahlen auffassen. Völlig analog zu Anmerkung 1 aus 3.1.2 kann man folgern, daß für einen Anteil $\geq 1/(2\bullet P(n))$ an den Tupeln $(s_1,\ldots,s_k)$ der eigentlichen Schlüssel C_n mit einer Wahrscheinlichkeit $\geq 1/(2\bullet P(n))$ korrekt ist; und analog zu Lemma 1 folgt, daß man ein hinreichend

gutes deterministisches Schaltnetz erhält, wenn man die beste dieser Zufallszahlen fest anlegt. ◊

3.5 Andere Anwendungen der kryptographischen Modellierung

Auf eine Darstellung der weiteren Anwendungen der kryptographischen Modellierung, z.B. auf Kryptosysteme mit öffentlichen Schlüsseln oder Zufallszahlenerzeugung, wird im Gegensatz zur algebraischen Modellierung (2.3) verzichtet, da sie zum einen sehr viel vielfältiger sind, zum anderen bereits gut lesbare Überblicke existieren, z.B. [Bras_88].

Angemerkt sei jedoch, daß für die Anwendung in Protokollen Verschlüsselungssysteme zur Zeit noch größere Probleme machen als digitale Signatursysteme, da bei den ursprünglich, z.B. in [GoMi_82, GoMi_84] und noch in [MiRS_88] als kryptographisch sicher bezeichneten Kryptosystemen mit öffentlichen Schlüsseln keine Sicherheit gegen aktive Angriffe gefordert wurde und auch nicht existiert (vgl. z.B. [GoMT_82]).

Innerhalb von Protokollen ist jedoch aus ähnlichen Gründen wie den am Anfang von 3.2 für digitale Signaturen genannten mit Möglichkeiten für aktive Angriffe zu rechnen, da Teilnehmer verpflichtet sind, auf Anfragen von anderen zu reagieren. Bei Verwendung solcher Kryptosysteme ist also nicht zu erwarten, daß die geforderte Protokollsicherheit erreicht wird, im Gegensatz zu digitalen Signatursystemen, wo die in Protokollen möglichen Angriffe gegen ein sicheres Signatursystem tatsächlich erlaubt sind. Dies war einer der Gründe, in dieser Arbeit nur Signatursysteme zu betrachten.

Mittlerweile wird allerdings im Artikel [BlFM_88] ein Entwurf für ein gegen aktive Angriffe sicheres Kryptosystem mit öffentlichen Schlüsseln angegeben, aber ohne genaue Definition der Sicherheit und ohne genaue Verfahrensbeschreibung.

4 Vergleich der Modellierungsarten

Um die Modelle im in 1.4 beschriebenen Sinne vergleichen zu können, müssen zunächst Bezüge zwischen ihnen hergestellt werden, d.h. es muß formal festgelegt werden, was zu einer Aussage im algebraischen Modell, über deren Korrektheit entschieden werden soll, die entsprechende Aussage im kryptographischen Modell sein soll. Dies wird in 4.1 vorgenommen, bevor in 4.2 Vergleiche angestellt werden. Dabei wird o.B.d.A. das Signatursystem als einzige kryptographische Funktion des Modells angenommen; da das Ergebnis lauten wird, daß die algebraische Modellierung nicht korrekt ist, genügt ja ein einziges sinnvolles Gegenbeispiel.

Obwohl die Formalisierung der Entsprechungen in 4.1 notwendig ist, um sich der Korrektheit des Vergleichsergebnisses einigermaßen sicher zu sein, müßte es möglich sein, 4.2 direkt zu lesen und nur bei Bedarf auf die weniger formalen Einleitungen der Abschnitte in 4.1 zurückzugreifen.

4.1 Entsprechungen zwischen den Modellwelten

Alle im algebraischen Modell auftretenden Nachrichten und Funktionen sollten Modelle für solche aus der kryptographischen Realität sein (während auf der kryptographischen Seite viele weitere Funktionen denkbar sind, die ins algebraische Modell nicht aufgenommen wurden, da sie für das betrachtete Protokoll irrelevant sind). Somit erscheint es sinnvoll, die Zuordnung zwischen den Modellelementen über eine Abbildung von den Sorten, Grundnachrichten und Funktionen aus der Spezifikation $Spez$ des algebraischen Modells (vgl. 2.2) auf genau diejenigen Dinge im kryptographischen Modell zu definieren, die durch sie modelliert werden sollten (4.1.1).

In 4.1.2 wird die notwendige Zusatzforderung, daß wenigstens die Gleichungen aus der Spezifikation, die das algebraische Modell definieren, die kryptographische Realität korrekt widerspiegeln, formalisiert.

Da die Aussagen über Nachrichtenunkenntnis und Ungleichungen kryptographisch nicht von einzelnen Nachrichten, sondern von Bauarten handeln, wird die Zuordnung in 4.1.3 auf diese ausgedehnt. Auf der algebraischen Seite werden dazu zunächst Terme verwendet. In 4.1.4 wird von den Termen zu deren Klassen, also den Nachrichten im algebraischen Modell, übergegangen.

4.1.1 Kryptographische Modelle als Interpretationen von Spezifikationen

Eine solche Abbildung, über die die Entsprechung formalisiert werden soll, ist beinahe (siehe nächsten Absatz) das, was in der Logik als „Interpretation" einer Spezifikation bezeichnet wird [Sper_84, Definition 2.12]. Speziell für abstrakte Datentypen wird auch der Begriff „Algebra" verwendet ([EhMa_85], Definition 1.4), der hier wegen möglicher Verwechslungen vermieden werden soll.

Eine Sonderrolle unter den Nachrichten spielen allerdings die Schlüssel des Signatursystems: Bei Sicherheitsaussagen im algebraischen Modell ist ein Schlüssel s ein einziges Objekt (formal: die Termklasse der Grundnachricht s). In kryptographischen Aussagen kommt jedoch noch die Auswahl (gleichverteilt, sofern der Schlüssel einem braven Teilnehmer gehört) unter sämtlichen möglichen Schlüsselwerten vor (vgl. Definitionen 1, 3 und 4).

Letzteres ließe sich mit üblichen Interpretationen nur ausdrücken, wenn s im algebraischen Modell eine Variable statt einer Grundnachricht wäre. Dies wiederum ist nicht möglich, da die für das algebraische Modell verwendete Bildung von Termklassen (d.h. die initiale Semantik) nur für Terme ohne Variable definiert ist, und gerade die Schlüssel eine wichtige Rolle in den Termen spielen.

Daher scheint eine eigene Definition nötig zu sein, obwohl sie stark an der üblichen Definition von Interpretationen eines abstrakten Datentyps orientiert bleibt.

Dies kann man nutzen, um außer der Sonderrolle der Schlüssel noch einige kryptographische Spezialeigenschaften zu berücksichtigen, insbesondere, daß die Entsprechungen von U, Test usw. tatsächlich ein digitales Signatursystem bilden.

Hierbei wird verlangt, daß der kryptographische Nachrichtenraum die Menge sämtlicher Wörter über einem endlichen Alphabet ist. Dies wurde in Kapitel 3 noch nicht gefordert, tatsächlich verlangen die in 3.3 genannten Systeme eine präfixfreie Codierung. Der Anlaß für die neue Forderung ist, daß sie sich für die folgenden Gegenbeispiele als günstig erweist (vgl. Lemma 3 in 4.1.2). Es läßt sich jedoch auch allgemeiner rechtfertigen.

Zunächst stört diese Forderung nicht, da sie kein sinnvolles kryptographisch sicheres digitales Signatursystem ausschließt, denn ursprünglich sollen wohl mit jedem Signatursystem beliebige Nachrichten unterschreibbar sein. Daher muß es immer eine Codierung vom ursprünglichen Nachrichtenraum $\mathcal{N}_{ur} = A^*$ (für ein endliches Alphabet A) in den für das Signatursystem verwendeten $\mathcal{N}_{Sig}$ geben, also eine mit polynomialem Aufwand zu berechnende, injektive Funktion $C: \mathcal{N}_{ur} \to \mathcal{N}_{Sig}$. Es beeinträchtigt die Sicherheit des Signatursystems nicht, wenn man diese Funktion als zum Unterschreiben gehörig betrachtet, d.h. statt des bisherigen U_{alt} nun $U_{neu} := U_{alt} \circ C$ verwendet.

Zudem ist sie auch allgemein sinnvoll, wenn man ein algebraisches (oder anderweitig vom speziell gewählten digitalen Signatursystem abstrahierendes) Modell verwenden will. Die Codierung müßte

andernfalls nämlich in das Modell aufgenommen werden, da man Unterschriften als Teile zu unterschreibender Nachrichten gestattet (weil zumindest ein Angreifer immer davon Gebrauch machen kann): Wären allein konkrete Nachrichten wie 6 zu unterschreiben, so könnte man dies noch als U(s, 6) modellieren und davon ausgehen, daß die kryptographische Entsprechung der 6 bereits die (z.B. präfixfrei) codierte Form ist. Betrachtet man in dem Fall aber die Unterschrift unter U(s, 6), so kann man nicht analog U(s, U(s, 6)) schreiben, weil die kryptographische Unterschrift i. allg. nicht zufällig passend codiert ist. Die Darstellung U(s, C(U(s, 6))) erscheint andererseits unnötig umständlich.

Definition 6: Seien eine algebraische Spezifikation *Spez* gegeben und darin Sorten $\mathcal{S}$, $\mathcal{T}$ und $\mathcal{N}$ als Schlüssel- Testschlüssel- und Nachrichtensorte und Funktionen t, U und Test (mit passenden Stelligkeiten) ausgezeichnet.

Eine kryptographische Interpretation für *Spez* ist gegeben durch drei Abbildungen („kr" steht jeweils für „kryptographisch") :

- kr_1 ordnet jeder Sorte *Sort* aus *Spez* eine Nachrichtenmenge, also eine Menge von Wörtern über einem endlichen Alphabet A, zu, wobei speziell $kr_1(\mathcal{N}) = A^*$ ist,
- kr_2 ordnet jeder Grundnachricht N aus einer Sorte $Sort \neq \mathcal{S}$ eine Nachricht aus $kr_1(Sort)$ zu, und
- kr_3 ordnet jedem Funktionsnamen f aus *Spez* eine Funktion mit den gemäß kr_1 passenden Definitions- und Bildbereichen zu.

Zur Vereinfachung darf der Index, der meist aus dem Kontext klar ist, weggelassen und alle drei Abbildungen durch nachgestelltes „^{kr}" geschrieben werden, also $Sort^{kr}$, N^{kr} und f^{kr} statt $kr_1(Sort)$, $kr_2(N)$ bzw. $kr_3(f)$.

Alle Funktionen f^{kr} im kryptographischen Modell sollen effizient, d.h. durch eine polynomial wachsende Folge von Schaltnetzen zu berechnen sein.

Außerdem müssen t^{kr}, U^{kr} und $Test^{kr}$ ein gemäß Definition 1 und 2 sicheres digitales Signatursystem bilden. ◊

Bemerkungen:

- Bei den hier behandelten digitalen Signatursystemen können unabhängig von der Schlüssellänge Nachrichten beliebiger Länge unterschrieben werden. Deshalb konnte man problemlos kr_2 unabhängig vom Sicherheitsparameter wählen.
- $\mathcal{T}$ wird i.allg. keine Grundnachrichten enthalten, da sinnvolle Testschlüssel nur als t(s) für Signierschlüssel s erzeugt werden können. Deswegen erhielt $\mathcal{T}$ bei kr_2 im Gegensatz zu $\mathcal{S}$ keine Sonderbehandlung. Im folgenden genügt dies sicherlich. Sollte aber in einem konkreten Fall ein Angreifer betrachtet werden, der Testschlüssel direkt wählt, könnte eine solche nötig werden.

Wie bei allen abstrakten Datentypen (und allgemeiner allen logischen Spezifikationen) kann man nun auch Termen über *Spez* einen Wert im kryptographischen Modell zuordnen.

Enthält ein Term Schlüssel oder Variablen, so benötigt man zunächst noch deren aktuelle Belegung. Diese wird durch eine Abbildung B beschrieben, die wieder zur Vereinfachung durch nachgestelltes ^{B} ausgedrückt wird. Dabei ist es möglich, die Längen der vorkommenden Schlüssel verschieden zu wählen. Oftmals wird jedoch ein Sicherheitsparameter n festgelegt sein; B heißt dann eine Belegung zu dem Sicherheitsparameter n, wenn s^B für jedes $s \in \mathcal{S}$ die Länge n hat.

Auch die Abbildung, die nun jedem Term seinen Wert bei dieser Belegung zuordnet (und natürlich speziell für Schlüssel und Variablen gleich B, für Grundnachrichten gleich kr_2 sein muß), wird mit B bezeichnet. Es ist also z.B.

$$(U(s, N))^B := U^{kr}\,(s^B, N^{kr})$$

oder ausführlich $B(U(s, N)) := kr_3(U)\,(B(s), kr_2(N))$.

Für Terme T ohne Schlüssel und Variable wird auch wieder die Schreibweise T^{kr} statt T^B verwendet, da ihr Wert nicht von der Belegung abhängt.

Ebenso einfach kann man nun bei fester Belegung Gleichungen zwischen Termen und daraus aufgebauten Prädikaten auf kryptographischer Seite den Wert „wahr" oder „falsch" zuordnen, z.B. ist für die Formel $F :\Leftrightarrow (L = R \wedge R = T)$ mit Termen L, R und T

$$F^B = \text{wahr} :\Leftrightarrow L^B = R^B \wedge R^B = T^B$$

Da die Definition von B anhand dieser Beispiele klar sein dürfte, sei für die formalen Einzelheiten wieder auf [EhMa_85, Definitionen 1.10, 1.13] verwiesen.

4.1.2 Korrektheit der kryptographischen Interpretationen

Wie in der Einleitung zu 4.1 erwähnt, sollten wenigstens die Gleichungen des algebraischen Modells die kryptographische Realität korrekt widerspiegeln.

Eine solche Forderung ist unabdingbar, da andernfalls nicht einmal garantiert wäre, daß das algebraische Modell den Ablauf eines Protokolls ganz ohne Angreifer korrekt modelliert. Erfüllen z.B. concat und π_1 im algebraischen Modell $\pi_1(\text{concat}\,(N, M)) = N$ und werden zu einem Protokoll benutzt, in dem, informell, Teilnehmer A zwei Nachrichten konkateniert und Teilnehmer B daraus die erste wiedergewinnt, so müssen natürlich auch die Implementierungen concat^{kr} und $\pi_1{}^{kr}$ dieser Gleichung genügen, damit B seine Nachricht erhält (also muß ggf. bei der Konkatenation ein Trennzeichen eingefügt werden).

Formal heißt dies, daß man verlangen muß, daß kr_1, kr_2 und kr_3 die Spezifikation *Spez* erfüllen, d.h. daß die in *Spez* angegebenen Gleichungen für alle Belegungen den Wert „wahr" haben.

Solche Interpretationen, bei denen alle in einer Spezifikation angegebenen Formeln gelten, bezeichnet man in der Logik als „Modell" der Spezifikation, aber auch dieser Begriff soll hier wegen seiner Vieldeutigkeit vermieden werden. Statt dessen wird von einer korrekten kryptographischen Interpretation gesprochen. (Der Name legt nahe, daß die Korrektheit allein eine Sache des kryptographischen Modells sei. Formal ist dies sinnvoll, weil für die Abbildungen kr die Spezifikation bereits erstellt sein muß. Bei der Modellierung eines konkreten Protokolls ist die Korrektheit aber natürlich mindestens ebensosehr eine Forderung nach einer geeigneten Auswahl der Spezifikation *Spez* wie nach einer geeigneten kryptographischen Implementierung.)

Allerdings könnten bzgl. der Schlüssel, bzgl. derer im kryptographischen Modell vor Gleichungen wie bei Variablen Allquantoren angenommen wurden, Abstriche gemacht werden. Speziell das Erfülltsein der Gleichung „Test (t(s), N, U(s,N))=wahr" impliziert beispielsweise, daß korrekt geleistete Unterschriften nicht einmal mit einer winzigen ε-Wahrscheinlichkeit durch den Test fallen, was bei den bisherigen digitalen Signatursystemen auch stimmt. Wenn ein solches ε jedoch genauso schnell fallen würde wie in den Sicherheitsdefinitionen, könnte es wohl akzeptiert werden. Die folgenden Gegenbeispiele würden aber durch eine solche Erweiterung der Definition einer korrekten kryptographischen Interpretation nicht entkräftet.

Man kann sich noch überlegen, inwiefern man bei einer Spezifikation *Spez*, zu der es überhaupt eine korrekte kryptographische Interpretation gibt, ein beliebiges digitales Signatursystem verwenden kann. Dies wird in 4.2.3 und 4.2.4 für Gegenbeispiele benötigt, in denen man es bei der algebraischen Modellierung nicht allen denkbaren Interpretationen recht machen kann. Es wäre jedoch

auch sonst für die Anwendbarkeit des algebraischen Modells von Interesse, da sie für die Betrachtung von Protokollen ohne genaue Kenntnis des verwendeten Signatursystems gedacht ist.

Weitgehend unabhängig ist die Implementierung der kryptographischen Funktionen t, U und Test von der der übrigen Funktionen tatsächlich, weil sie nicht gemeinsam in einer Gleichung aus *Spez* auftreten (vgl. 2.2). Allerdings wird ein Teil der übrigen Funktionen auf denselben Sorten $\mathcal{S}$ und vor allem $\mathcal{N}$ operieren wie das Signatursystem. Haben zwei digitale Signatursysteme also verschiedene Schlüsselräume, so ist nicht garantiert, daß man bei einem Wechsel zwischen ihnen die evtl. für die Schlüssel existierenden übrigen Funktionen unverändert lassen kann. Für den Nachrichtenraum wurde dieses Problem im wesentlichen umgangen, indem $N^{kr} = A^*$ gefordert wurde. (Ähnliches könnte auch für den Schlüsselraum möglich sein, da die Schlüssel gleichverteilt wählbar sein müssen, wird aber nicht benötigt). Es gilt also:

Lemma 3: Sei *Spez* eine Spezifikation, die eine korrekte kryptographische Interpretation kr besitzt. Bilden U' und Test' ebenfalls zusammen mit t^{kr} ein kryptographisch sicheres digitales Signatursystem mit derselben Schlüsselmenge $\mathcal{S}^{kr}$ und demselben Alphabet A, so bildet auch kr', das mit kr bis auf $U^{kr'} := U'$ und $Test^{kr'} := Test'$ übereinstimmt, eine korrekte kryptographische Interpretation. ◊

Beweis: Zunächst ist zu zeigen, daß kr' überhaupt eine kryptographische Interpretation für *Spez* bildet. Dazu müssen $\mathcal{S}^{kr}$ und $\mathcal{N}^{kr}$ passende Definitions- und Bildbereiche für die beiden geänderten Funktionen sein. Für die Definitionsbereiche gilt das nach Vorraussetzung. Der Bildbereich von U' kann ebenfalls $A^* = \mathcal{N}^{kr}$ nicht überschreiten. (Hierfür wurde die Einschränkung an $\mathcal{N}^{kr}$ gemacht). Außerdem sind nach Definition eines digitalen Signatursystems auch $U^{kr'}$ und $Test^{kr'}$ effizient berechenbar.

Die Gleichung „Test (t(s), N, U(s,N))=wahr” wird von kr' erfüllt, weil U' und Test' zusammen mit t^{kr} ein digitales Signatursystem bilden. In allen übrigen Gleichungen aus *Spez* treten weder U noch Test auf, also sind diese durch kr' genauso wie durch kr erfüllt. ◊

4.1.3 Entsprechung zwischen Termen und kryptographischen Bauarten

Bisher hat man einem Term bei jeder Schlüsselbelegung eine andere kryptographische Nachricht zugeordnet. Indem man alle diese Nachrichten zu einer Bauart zusammenfaßt, gewinnt man eine belegungsunabhängige Entsprechung, die man wieder durch ^{kr} bezeichnen kann. Dies ist nötig, um z.B. einer algebraischen Aussage, daß ein Angreifer die durch einen Term T repräsentierte Nachricht nicht kennt, eine kryptographische zuzuordnen, daß er keine Nachricht der Bauart T^{kr} kennt.

Da dies formal kompliziert aussieht, obwohl es ganz natürlich ist, sei es zuerst am üblichen Beispiel des Terms T = U(s, N) dargestellt: Für eine Belegung B ist

$$T^B = U^{kr}(s^B, N^{kr})$$

(vgl. 3.4), was aus der ganzen Belegung nur vom Wert des einen Schlüssels s abhängig ist. Diese Nachrichten bildeten im kryptographischen Modell zusammen gerade eine Bauart $\mathcal{B}$ (vgl. 4.1.1), die man jetzt zur Unterscheidung der beiden Modelle als

$$\mathcal{B}: (\mathcal{S}^{kr})^1 \to \mathsf{P}(\mathcal{N}^{kr}) \quad \text{mit} \quad \mathcal{B}(s') = \{U^{kr}(s', N^{kr})\} \text{ für alle } s' \in \mathcal{S}^{kr}$$

bezeichnen würde. Es ist also $T^B = \mathcal{B}(s^B)$, wobei auch die s^B sämtliche Werte aus $\mathcal{S}^{kr}$ annehmen. Natürlich soll T^{kr} dieses $\mathcal{B}$ sein.

Zur Vereinfachung, und weil er für die Gegenbeispiele genügt, wird auch in der Verallgemeinerung des Beispiels nur der Fall betrachtet, daß der Angreifer keinen der Schlüssel kennt, die im betrachteten Term vorkommen. Daß dies die Diskussion vereinfacht, sieht man am Vergleich der Beispiele 1) bis 3) mit Beispiel 4) aus 3.4. Da die Nachrichten anfangs durch den Term Auth(s_Z, s_B) dargestellt sind, kann man alle als Beispiele für die hier gewünschte Zuordnung auffassen, wobei 4) den Fall behandelt, daß der Angreifer beide Schlüssel nicht kennt, während 1) bis 3) Versionen davon sind, daß er nur s_Z nicht kennt.

Damit könnte man erwarten, daß man einem Term, in dem x Schlüssel vorkommen, eine Bauart mit Urbildbereich $(\mathcal{S}^{kr})^x$ zuordnen möchte. Allerdings muß man beachten, daß bei Termen mit mehreren Schlüsseln diese bei Bauarten durch ihre Reihenfolge, bei Belegungen hingegen durch ihre Namen aus *Spez* unterschieden werden, so daß sich ein einziger Term mit k Schlüsseln durch k! Bauarten darstellen läßt (beispielsweise kann man U(s_Z, s_B) durch $\mathcal{B}(s_1, s_2) = \{U^{kr}(s_1, s_2)\}$ oder $\mathcal{B}(s_1, s_2) = \{U^{kr}(s_2, s_1)\}$ darstellen). Man muß sich hier also irgendwie festlegen; Lemma 2 aus 3.4 garantiert jedoch, daß alle möglichen Festlegungen die gleichen Folgen haben.

Entsprechend der Diskussion vor Definition 5 in 3.4 ist klar, daß bei der Betrachtung von Ungleichungen oder Nachrichtenunkenntnis zwischen mehreren Nachrichten aus einem Protokoll die Festlegung, welcher Schlüssel aus *Spez* durch welche Position in der Bauart dargestellt wird, für alle diese Nachrichten gleich zu treffen ist. Dies läßt sich am einfachsten erreichen, wenn man nicht zuerst Bauarten mit minimaler Schlüsselanzahl für jeden Term definiert und diese im kryptographischen Modell erweitert, sondern sofort jedem Term eine Bauart mit sämtlichen überhaupt in der Betrachtung vorkommenden Schlüsseln zuordnet, wie bei den Beispielen $\mathcal{B}_5$ und $\mathcal{B}_6$. Da Bauarten sinnvollerweise aber nur mit einer endlichen Schlüsselanzahl definiert werden konnten, können dies i.allg. nicht alle Schlüssel aus $\mathcal{S}$ sein. Damit erhält man folgende Definition:

Definition 7: Sei eine Spezifikation *Spez* gegeben, $\mathcal{S}_{end} \subset \mathcal{S}$ eine endliche Menge von Schlüsseln, $k = |\mathcal{S}_{end}|$, und sei eine Numerierung dieser Schlüssel gegeben, d.h. eine bijektive Abbildung

$$nr: \quad \mathcal{S}_{end} \to \{1,\ldots,k\}.$$

Zudem gebe es eine durch „kr" bezeichnete korrekte kryptographische Interpretation von *Spez*. Für jeden Term T über *Spez*, dessen sämtliche Schlüssel in $\mathcal{S}_{end}$ liegen, sei dann die zu T gehörige (einelementige) Bauart definiert als

$$T^{kr}: \quad (\mathcal{S}^{kr})^k \to \mathsf{P}(\mathcal{N}^{kr})$$
$$(s'_1,\ldots, s'_k) \to \{T^B\},$$

wobei B eine Belegung mit

$$B(s) = s'_{nr(s)} \text{ für alle } s \in \mathcal{S}_{end}$$

ist. Falls die verwendeten $\mathcal{S}_{end}$ und nr nicht klar sind, muß präziser $T^{kr,nr}$ geschrieben werden. ◊

Anmerkung 2: T^{kr} aus Definition 7 ist (bei festem nr) wohldefiniert, obwohl die verwendete Belegung B nicht eindeutig bestimmt ist, da T^B nur von der Belegung der in T vorkommenden Schlüssel abhängt. Diese sind alle in $\mathcal{S}_{end}$, also bei allen möglichen B gleich belegt. ◊

Nun sei noch die obige Aussage, daß die Wahl von $\mathcal{S}_{end}$ und der Numerierung nr keine Rolle spielt, solange alle beteiligten Schlüssel enthalten sind, formal klargestellt.

Anmerkung 3: Ist $\mathcal{S}(T)$ die genaue Menge der in T vorkommenden Schlüssel, $k^* = |\mathcal{S}(T)|$ deren

Anzahl und nr* eine feste Numerierung von $\mathcal{S}(T)$, so geht jedes mögliche $T^{kr,nr}$ aus $T^{kr,nr*}$ durch Erweiterung und Schlüsselpermutation hervor (vgl. Definition 5 in 3.4).

Damit folgt aus Lemma 2 in 3.4, daß Aussagen über Nachrichtenunkenntnis oder Ungleichungen zwischen Bauarten, die als kryptographische Entsprechungen bestimmter algebraischer Terme definiert wurden, nicht von der Wahl von $\mathcal{S}_{end}$ und nr abhängen. ◊

Beweis: Als Zwischenstufe wird ein $T^{kr,nr\wedge}$ definiert, das durch Erweiterung aus $T^{kr,nr*}$ hervorgeht und aus dem dann durch Schlüsselpermutation $T^{kr,nr}$ wird. Dazu sei

$$\begin{aligned} nr\wedge: \quad & \mathcal{S}_{end} \to \{1,\ldots k\} \\ & s \to nr\wedge(s) = nr*(s) \text{ für alle } s \in \mathcal{S}(T),\ nr\wedge(s) \text{ beliebig für } s \in \mathcal{S}_{end}\backslash\mathcal{S}(T) \end{aligned}$$

Um zu zeigen, daß $T^{kr,nr\wedge}$ eine Erweiterung von $T^{kr,nr*}$ ist, muß nachgewiesen werden, daß

$$T^{kr,nr\wedge}(s'_1,\ldots,s'_k) = T^{kr,nr*}(s'_1, \ldots,s'_{k*}) \text{ für alle } s'_1,\ldots,s'_k \in \mathcal{S}^{kr},$$

d.h. jeweils

$$T^{B\wedge} = T^{B*},$$

wobei

$$B\wedge(s) = s'_{nr\wedge(s)} \text{ für alle } s \in \mathcal{S}_{end} \text{ und } B*(s) = s'_{nr*(s)} \text{ für alle } s \in \mathcal{S}(T).$$

Da der Wert von T bei einer Belegung nur von den Werten der Schlüssel aus $\mathcal{S}(T)$ abhängt und B^ und B* auf diesen wegen nr^(s) = nr*(s) übereinstimmen, gilt die letzte Gleichung.

Nun ist also noch zu zeigen, daß $T^{kr,nr}$ durch eine Schlüsselpermutation π aus $T^{kr,nr\wedge}$ hervorgeht. Das bedeutet, daß

$$T^{kr,nr}(s'_1,\ldots,s'_k) = T^{kr,nr\wedge}(s'_{\pi(1)}, \ldots,s'_{\pi(k)}) \text{ für alle } s'_1,\ldots,s'_k \in \mathcal{S}^{kr}$$

gelten soll, d.h. jeweils

$$T^{B} = T^{B\wedge},$$

wobei

$$B(s) = s'_{nr(s)} \text{ und } B\wedge(s) = s'_{\pi(nr\wedge(s))} \text{ für alle } s \in \mathcal{S}_{end}.$$

Die Formel für B^(s) sieht man dabei am einfachsten ein, wenn man zunächst die $s'_{\pi(i)}$ umschreibt als s''_i: Dann soll nach Definition 5 $B\wedge(s) = s''_{nr\wedge(s)}$ sein, was gerade $s'_{\pi(nr\wedge(s))}$ war.

Da der Wert von T bei einer Belegung nur von den Werten der Schlüssel aus $\mathcal{S}(T) \subset \mathcal{S}_{end}$ abhängt, erreicht man die Gültigkeit der letzten Gleichung, wenn man

$$\pi(nr\wedge(s)) = nr(s)$$

verlangt. Da nr^ bijektiv ist, kann man dies auch ausdrücken als

$$\pi = nr \circ (nr\wedge)^{-1}. \quad ◊$$

4.1.4 Entsprechungen zwischen algebraischem und kryptographischem Modell

Bisher wurde im engeren Sinne keine Entsprechung zwischen den beiden Modellen definiert, sondern Entsprechungen für Terme einerseits im algebraischen Modell (die Klassen), andererseits im kryptographischen (die Bauarten). Als Abschluß von 4.1 wird nun gezeigt, wie sich eine direkte

Entsprechung zwischen einem algebraischen Modell und einer korrekten kryptographischen Interpretation seiner Spezifikation, d.h. vor allem zwischen Termklassen und Bauarten, bilden läßt. Im folgenden werden hiervon allerdings nur die ersten beiden Absätze benötigt.

Zunächst sei eine feste Belegung B der Schlüssel und Variablen angenommen. Daraus soll eine Abbildung von den algebraischen Termklassen auf kryptographische Nachrichten gewonnen werden. Für die kyptographische Entsprechung der Klasse K eines Terms T liegt dabei die Definition

$$K^B := T^B$$

nahe. Zu zeigen ist nur die Wohldefiniertheit, d.h. $T_1{}^B = T_2{}^B$ für $T_1, T_2 \in K$.

Diese folgt direkt aus der Definition des algebraischen Modells als initiale Semantik von *Spez* (vgl. 1. am Ende von 2.2, genauer [EhMa_85, Definition 2.4]), und zwar trotz der anderen Behandlung der Schlüssel, da eine korrekte kryptographische Interpretation zusammen mit einer Belegung der Schlüssel gerade ein Modell im Sinne der abstrakten Datentypen darstellt.

Für Funktionen ist die Entsprechung leicht, da die Funktionen des algebraischen Modells genau den Funktionssymbolen aus *Spez* entsprechen und deren Interpretationen (die von der Belegung natürlich unabhängig sind) übernehmen können.

Wollte man Nachrichtenunkenntnis und Ungleichungen zwischen beiden Modellen direkt, d.h. ohne den Umweg über Terme, vergleichen, so müßte man natürlich auch den Klassen Bauarten zuordnen. Dazu liegt für die Bauart K^{kr}, die der Klasse K eines Terms T entspricht, eine Definition $K^{kr} := T^{kr}$ nahe. Dies muß allerdings genauer als

$$K^{kr,nr} := T^{kr,nr}$$

geschrieben werden, wobei nr eine Numerierung einer endlichen Schlüsselmenge $\mathcal{S}_{end} \supset \mathcal{S}(T)$ war. Gilt $T_1, T_2 \in K$ und $\mathcal{S}_{end} \supset \mathcal{S}(T_1) \cup \mathcal{S}(T2)$, wobei der Angreifer alle Schlüssel aus $\mathcal{S}_{end}$ nicht kennt, so ist die Definition von der Auswahl von T_1 oder T_2 unabhängig, d.h. $T_1{}^{kr,nr} = T_2{}^{kr,nr}$, denn

$$T_i{}^{kr,nr}(s'_1,\ldots, s'_k) = \{T^{B_i}\}$$

wobei B_i eine Belegung mit

$$B_i(s) = s'_{nr(s)} \text{ für alle } s \in \mathcal{S}_{end}$$

ist. Auf $\mathcal{S}_{end}$ stimmen also B_1 und B_2 überein, und da T_2 nur Schlüssel aus $\mathcal{S}_{end}$ enthält, ist $T_2{}^{B2} = T_2{}^{B1}$. Daß letzteres gleich $T_1{}^{B1}$ ist, wurde bereits oben im Absatz über feste Belegungen gezeigt.

Hiermit ist in gewisser Weise Wohldefiniertheit gezeigt. Da jedoch i.allg. die Terme einer Klasse zusammen unendlich viele Schlüssel enthalten können (indem man z.B. Unterterme der Form s - s hinzufügt), und dabei auch welche, die der Angreifer kennt, scheint es nicht sinnvoll, eine feste Numerierung auszuwählen und wieder nur K^{kr} zu schreiben. Auch die Schreibweise $K^{kr,nr}$ ist jedoch häßlich, denn wenn man die Klasse durch einen Term darstellt, der mehr Schlüssel enthält als nr numeriert, kann man diesen Ausdruck auf Anhieb gar nicht auswerten. Insofern würde es sich, wenn man diese Entsprechung weiterverfolgen müßte, anbieten, einer Termklasse auch eine Klasse von Bauarten zuzuordnen, nämlich alle, die man auf obige Weise als $K^{kr,nr}$ enthalten kann, und die dann wieder alle durch Erweiterungen und Schlüsselpermutationen auseinander hervorgingen.

4.2 Einfache Vergleichsergebnisse

Der Vergleich soll anhand von Aussagen über Nachrichtenunkenntnis vorgenommen werden, da diese als Gegenbeispiele überzeugender wirken als die einfacheren über Ungleichungen. Unter diesen

genügen aber besonders einfache, bei denen es im algebraischen Modell um die Kenntnis einer einzigen Nachricht geht.

Die zwei unterschiedlichen Typen von Gegenbeispielen werden in 4.2.3 bzw. 4.2.4 dargestellt. Um sie gemeinsam zu vereinfachen, wird in 4.2.1 zunächst eine Aussage gezeigt, die etwa besagt, daß das algebraische Modell Nachrichtenunkenntnis höchstens dann korrekt modellieren kann, wenn es Ungleichungen zwischen einzelnen Nachrichten korrekt modelliert. (Wem falsch modellierte Ungleichungen als Gegenbeispiele genügen, benötigt von 4.2.1 nur Definition 9). In 4.2.2 finden sich Bemerkungen zu Alternativen beim Beweis der Aussage aus 4.2.1.

Die Gegenbeispiele werden zunächst so gewählt, daß die Darstellung möglichst kurz wird. In 4.2.5 folgt im Rahmen der Diskussion der Ergebnisse eine Einbettung der Gegenbeispiele in mittelmäßig realistische Protokollstücke. Zudem wird diskutiert, ob man sie durch Änderungen an den zulässigen Protokollen oder Signatursystemen ausschließen könnte, was aber nicht der Fall zu sein scheint.

4.2.1 Zusammenhang von Nachrichtenunkenntnis und Ungleichungen

In den Gegenbeispielen soll gezeigt werden, daß man in manchen Fällen im algebraischen Modell nicht merkt, daß ein Angreifer in der Realität nach Erhalt gewisser Nachrichten $N_1,\ldots,N_m$ eine realistische Chance hat, sich auch eine Nachricht N zu konstruieren, die er eigentlich nicht besitzen sollte. Alle im folgenden angeführten Beispiele verwenden ein N der Gestalt $U(s, N_{neu})$ und im wesentlichen nur ein N_1, und zwar von der Form $U(s, N_{alt})$, also eine Unterschrift mit demselben Schlüssel, aber unter eine andere, uninteressantere (in der Praxis oft aus einem vorigen Protokolldurchlauf stammende) Nachricht.

Desweiteren sind die Beispiele so, daß der Angreifer gar nicht erst etwas konstruieren muß, sondern mit einer gewissen, nicht zu tolerierenden Chance einfach $U(s, N_{alt})$ anstelle von $U(s, N_{neu})$ verwenden kann, weil im kryptographischen Modell schon $N_{alt} = N_{neu}$ ist (im algebraischen natürlich nicht).

Um zu zeigen, daß zwei irgendwie gewählte N_{alt} und N_{neu} auf diese Art tatsächlich zu einem Gegenbeispiel führen, muß man jeweils zeigen, daß

a) im kryptographischen Modell $U(s, N_{alt})$ und $U(s, N_{neu})$ wirklich hinreichend oft gleich sind und
b) der Angreifer im algebraischen Modell $U(s, N_{neu})$ wirklich nicht aus $U(s, N_{alt})$ konstruieren kann.

Ersteres wird, dank der expliziten Konstruktion von $U(s, N_{alt})$ und $U(s, N_{neu})$, meist leicht sein, letzteres ist infolge der etwas komplizierten Definition algebraischer Modelle schwieriger (obwohl es anschaulich immer einleuchtet, insbesondere wenn aus 2.2 die Definition des algebraischen Modells mittels des Gleichungskalküls betrachtet wird, in dem das einzige, was man direkt mit Termen der Form U(…) machen kann, ist, sie durch Anwendung von Test in wahr zu verwandeln).

Es war beabsichtigt, in allen Einzelfällen zunächst zu zeigen, daß

a') N_{alt} und N_{neu} im kryptographischen Modell hinreichend oft gleich und
b') N_{alt} und N_{neu} im algebraischen Modell wirklich ungleich sind,

und in diesem Abschnitt vorweg zu beweisen, daß daraus die vorigen beiden Aussagen folgen, um so die Beispiele von technischen Details zu befreien. Allerdings erwies es sich als geschickter, statt b') eine etwas stärkere Aussage der Art

b") „N_{alt} und N_{neu} sind mindestens einmal in einem kryptographischen Modell nicht gleich"

zu verwenden.

Als gemeinsame Darstellung einer Nachricht für beide Modelle wird hier die Darstellung als Term über der Spezifikation benutzt (d.h. die N_{alt} und N_{neu} aus der obigen informellen Betrachtung werden als Terme T_{alt} und T_{neu} formalisiert). Kryptographischer und algebraischer Teil werden zunächst getrennt bewiesen und dann zum Satz 1 über einen Zusammenhang der Korrektheit des algebraischen Modells bzgl. der Modellierung von Nachrichtenunkenntnis bzw. Ungleichungen vereint.

Die Idee des Beweises für den kryptographischen Teil ist trivial: natürlich haben Nachrichten, die manchmal gleich sind, ebensooft auch gleiche Unterschriften; es ist also im Grunde nur zu zeigen, daß die gewählten Definitionen zusammenpassen.

Lemma 4: Seien eine Spezifikation $\mathcal{S}pez$ mit einer durch „^{kr}" bezeichneten korrekten kryptographischen Interpretation und ein Schlüssel $s^* \in \mathcal{S}$ gegeben. Für zwei Terme T_{alt} und T_{neu} über $\mathcal{S}pez$ gelte, daß die Bauarten T_{alt}^{kr} und T_{neu}^{kr} nicht kryptographisch verschieden sind.

Dann kann ein Angreifer, der alle in T_{alt} und T_{neu} vorkommenden Schlüssel und s nicht kennt, trotzdem aus einer Nachricht der Bauart $\mathcal{A}:=(U(s^*, T_{alt}))^{kr}$ eine der Bauart $\mathcal{B}:=(U(s^*, T_{neu}))^{kr}$ erzeugen. ◊

Beweis: Genauer ist gemäß Definition 7 (in 4.1.3) das Lemma so zu verstehen, daß $\mathcal{S}(T_{alt})$ und $\mathcal{S}(T_{neu})$ die Mengen der in T_{alt} bzw. T_{neu} vorkommenden Schlüssel sind, eine endliche Menge von Schlüsseln $\mathcal{S}_{end} \supset \mathcal{S}(T_{alt}) \cup \mathcal{S}(T_{neu}) \cup \{s^*\}$ gewählt wurde, $k = |\mathcal{S}_{end}|$ ist und eine Numerierung $nr: \mathcal{S}_{end} \to \{1,\ldots,k\}$ vorliegt.

Dies konnte in der Formulierung des Lemma eingespart werden, da gemäß Anmerkung 3 aus 4.1.3 die Aussagen nicht von der genauen Wahl von $\mathcal{S}_{end}$ und nr abhängen, auch nicht davon, ob die Verschiedenheit von T_{alt} und T_{neu} noch mit anderen $\mathcal{S}_{end}'$ und nr' bestimmt wird, die s^* nicht enthalten.

Es muß nun gemäß Definition 3 (in 3.4) eine höchstens polynomial wachsende Schaltnetzfolge C_n konstruiert werden, die für unendlich viele Sicherheitsparameter bei Eingabe von kryptographischen Testschlüsseln $t(s'_1),\ldots,t(s'_k)$ und einer alten Unterschrift $N_1 \in \mathcal{A}(s'_1,\ldots,s'_k)$ hinreichend oft eine neue Unterschrift $N \in \mathcal{B}(s'_1,\ldots,s'_k)$ erzeugt.

Dabei ist nach Definition von $\mathcal{A}$ immer

$$N_1 = (U(s^*, T_{alt}))^B = U^{kr}(s^{*B}, T_{alt}^B))$$

für eine Belegung B mit

$$B(s) = s'_{nr(s)} \text{ für alle } s \in \mathcal{S}_{end}$$

und analog

$$N = U^{kr}(s^{*B}, T_{neu}^B)).$$

Als C_n wird einfach das Schaltnetz verwendet, das die Eingabe N_1 auf die Ausgabe N durchschaltet, also ohne Beachtung der Testschlüssel und ohne jegliches Gatter (und das somit auf keinen Fall zu groß ist). Es ist mindestens dann korrekt, wenn $T_{alt}^B = T_{neu}^B$.

Nach Voraussetzung und Definition 4 (in 3.4) gilt, daß ein Polynom P existiert, so daß für unendlich viele Sicherheitsparameter der Anteil an Tupeln $(s'_1,\ldots,s'_k)$ von Schlüsseln der Länge n mit $T_{alt}^{kr}(s'_1,\ldots,s'_k) = T_{neu}^{kr}(s'_1,\ldots,s'_k)$ mindestens 1/P(n) beträgt. Dabei ist

$$T_{alt}^{kr}(s'_1,\ldots,s'_k) = \{T_{alt}^B\} \text{ und } T_{neu}^{kr}(s'_1,\ldots,s'_k) = \{T_{neu}^B\}$$

mit demselben B wie oben, also ist, wie gewünscht, $T_{alt}^B = T_{neu}^B$ für genau diese Schlüsseltupel.

Deren Anteil genügt auch gemäß Definition 3. ◊

Nun soll der sogenannte algebraische Teil bewiesen werden. Dabei wird im folgenden die kanonische Interpretation (im Sinne der abstrakten Datentypen) einer Spezifikation *Spez*, die jeder Grundnachricht, und als Folge davon jedem Term, ihre bzw. seine Klasse zuordnet, stets mit „^{alg}" (für „algebraisch") bezeichnet. Dies ist analog zur Bezeichnung „^{kr}" für eine kryptographische Interpretation, da hier jedem Term sein algebraisches Modell zugeordnet wird.

Gemäß den diesen Abschnitt einleitenden Bemerkungen ist folgendes zu zeigen:

Lemma 5: Gegeben sei eine Spezifikation *Spez* mit einer durch „^{kr}" bezeichneten korrekten kryptographischen Interpretation. Für zwei Terme T_{alt} und T_{neu} gebe es zu unendlich vielen Sicherheitsparametern n eine Belegung B_n der Schlüssel mit

$$T_{alt}{}^{B_n} \neq T_{neu}{}^{B_n}.$$

Seien $\mathcal{S}(T_{alt})$ und $\mathcal{S}(T_{neu})$ die Mengen der in T_{alt} bzw. T_{neu} vorkommenden Schlüssel. Außerdem sei ein zusätzlicher Schlüssel $s^* \in \mathcal{S} \setminus \mathcal{S}(T_{alt}) \cup \mathcal{S}(T_{neu})$ gewählt.

Dann kann ein Angreifer, der alle in T_{alt} und T_{neu} vorkommenden Schlüssel und s^* nicht kennt, im algebraischen Modell $U(s^*, T_{neu})^{alg}$ aus $U(s^*, T_{alt})^{alg}$ nicht erzeugen. Genauer darf man dabei auch annehmen, daß er $T_{alt}{}^{alg}$ und $T_{neu}{}^{alg}$ und alle Testschlüssel kennt (was die Aussage nur stärker macht), d.h.

$$U(s^*, T_{neu})^{alg} \notin [\ \{ U(s^*, T_{alt})^{alg}, T_{alt}{}^{alg}, T_{neu}{}^{alg} \} \cup \{ t(s)^{alg} \mid s \in \mathcal{S}(T_{alt}) \cup \mathcal{S}(T_{neu}) \cup \{s^*\} \}\ ] \ \Diamond$$

Beweis: Die Grundidee ist, zu zeigen, wie man unter der gegenteiligen Annahme das digitale Signatursystem brechen könnte.

Annahme: $U(s^*, T_{neu})^{alg}$ läge doch in jener Hülle.

Dann gibt es einen Term „Baum" über *Spez*, in dem s^* nur als $t(s^*)$ oder $U(s^*, T_{alt})$ vorkommt, mit

$$U(s^*, T_{neu})^{alg} = Baum^{alg}.$$

(Beweis: Ursprünglich bedeutet $U(s^*, T_{neu})^{alg} \in [\mathcal{E}]$, daß man N durch eine endliche Anzahl von Funktionsanwendungen auf Nachrichten aus $\mathcal{E}$ und die gemeinsamen Grundnachrichten im algebraischen Modell gewinnen kann, d.h. $U(s^*, T_{neu})^{alg} = Baum'$, wobei Baum' ein Term im algebraischen Modell ist. Die Nachrichten aus $\mathcal{E}$ sind in diesem Fall alle schon als N^{alg} dargestellt, und die Funktionen des algebraischen Modells gerade die mit „^{alg}" versehenen aus *Spez*, die durch $f^{alg}(N^{alg}) := f(N)^{alg}$ definiert wurden. Zieht man mittels dieser Gleichungen alle „^{alg}" aus Baum' nach außen, so erhält man $Baum^{alg}$.)

Dieser Baum entspricht der Konstruktionsvorschrift, wie der Angreifer im algebraischen Modell die Unterschrift unter T_{neu} durch Berechnung von Baum fälschen kann, und soll nun in die entsprechende kryptographische verwandelt werden.

Zunächst folgt gemäß 4.1.4 (3. Absatz) aus der algebraischen Gleichung die entsprechende kryptographische:

$$U(s^*, T_{neu})^B = Baum^B$$

für jede Belegung B der Schlüssel, d.h. $U^{kr}(s^{*B}, T_{neu}{}^B) = Baum^B$.

Es genügt gemäß Definition 1 und 2 (in 3.1.2 bzw. 3.2), eine höchstens polynomial wachsende Folge von Schaltnetzen C_n zu konstruieren, die für unendlich viele Sicherheitsparameter n für jedes s' $\in \mathcal{S}^{kr}$ als Eingabe $t^{kr}(s')$ erhalten, eine Nachricht $N \in \mathcal{N}^{kr}$ und $U^{kr}(s', N)$ ausgeben und zwischendurch Unterschriften unter andere Nachrichten als N erfragen. (Speziell hier sollen also Unterschriften nicht nur mit einer gewissen Wahrscheinlichkeit, sondern immer gefälscht werden.) Die folgende Konstruktion ist in Bild 6 an einem Beispiel dargestellt.

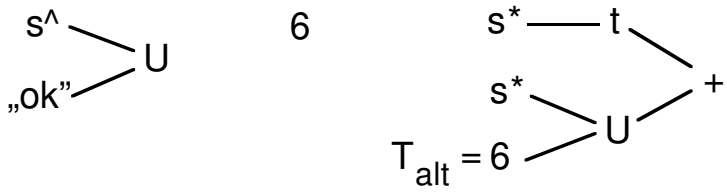
Natürlich soll $N := T_{neu}{}^{B_n}$ gewählt werden. Ist B* eine Belegung, für die einerseits $s^{B*} := s^{B_n}$ für alle $s \in \mathcal{S}(T_{alt}) \cup \mathcal{S}(T_{neu})$ ist, andererseits $s^{*B*} := s'$ (B* existiert wegen $s^* \notin \mathcal{S}(T_{alt}) \cup \mathcal{S}(T_{neu})$), so gilt

$$U^{kr}(s', N) = U^{kr}(s^{*B*}, T_{neu}{}^{B*}) = Baum^{B*}$$

und $T_{alt}{}^{B*} = T_{alt}{}^{B_n} \neq T_{neu}{}^{B_n} = T_{neu}{}^{B*}$, d.h. die Unterschrift unter $T_{alt}{}^{B*}$ darf erfragt werden.

Der Teil von C_n, der N erzeugt, soll nun genau der Baumstruktur von T_{neu} entsprechen, der für $U^{kr}(s', N)$ der Baumstruktur von Baum. Wo in den Termen eine Grundnachricht $N' \notin \mathcal{S}$ vorkommt, wird einfach konstant N'^{kr} angeschlossen, entsprechend s^{B*} für $s \neq s^*$. Kommt s* in Baum vor, so nur als $t(s^*)$ oder $U(s^*, T_{alt})$. Für $t(s^*)$ wird die Eingabe $t^{kr}(s')$ angeschlossen; die Unterschrift $U(s^*, T_{alt})$ wird ggf. erfragt. Für die vorkommenden Funktionssymbole (außer bei $t(s^*)$ oder $U(s^*, T_{alt})$ natürlich) braucht man nur die gemäß Definition 6 (in 4.1.1) existierenden Schaltnetze zusammenzuschalten; da die Anzahl der Funktionen in den Termen unabhängig von n ist, wächst auch die Größe der resultierenden Schaltnetze höchstens polynomial in n. ◊

Beispiel mit : T_{neu} T_{alt} Baum



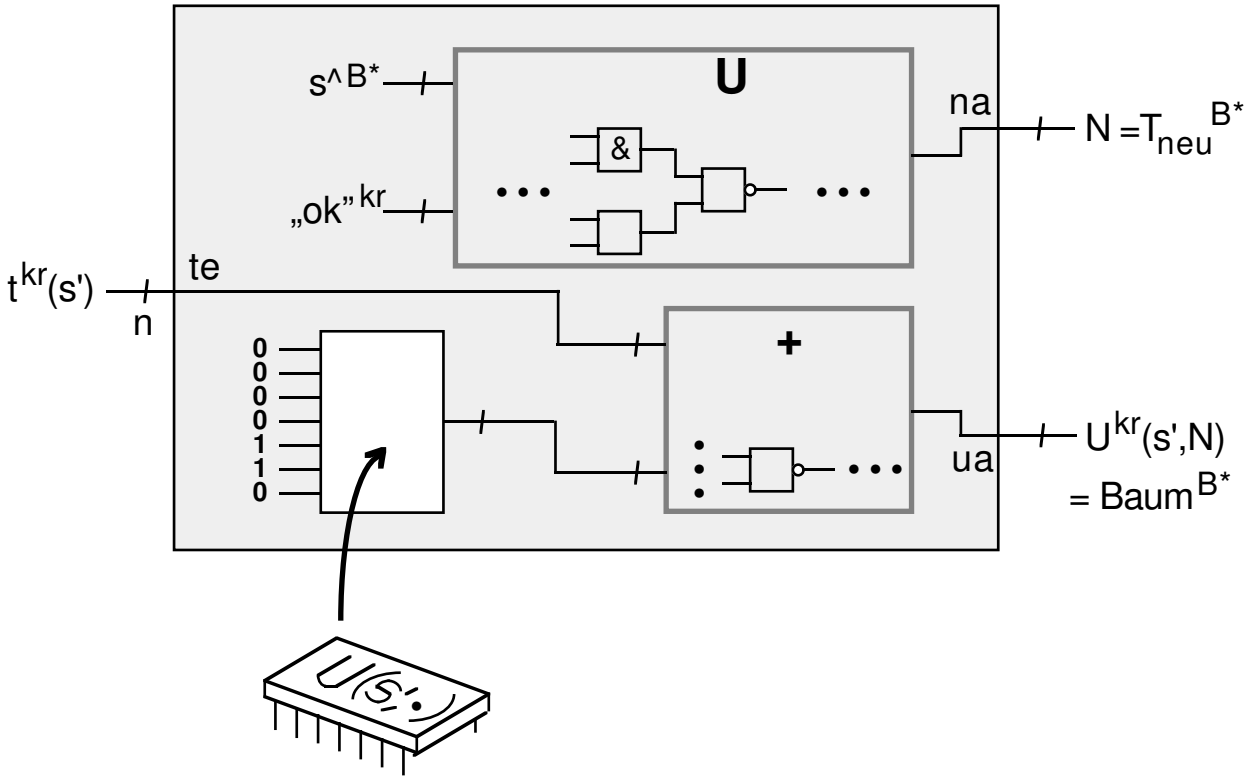


Bild 6 Beispiel zu Lemma 5

Bevor nun die beiden Lemmata zu einem Satz über den Zusammenhang zwischen Nachrichtenunkenntnis und Ungleichungen zusammengefaßt werden können, muß die Korrektheit eines algebraischen Modells genügend weit formalisiert werden:

Definition 8: Folgendes ist eine notwendige Bedingung dafür, daß das algebraische Modell zu einer Spezifikation *Spez*, die eine durch „kr" bezeichnete korrekte kryptographische Interpretation besitzt, Nachrichtenunkenntnis korrekt modelliert:

Sind T_{neu} und T_{alt} zwei Terme ohne Variablen über *Spez*, in denen höchstens Schlüssel aus der endlichen Menge $\mathcal{S}_{end} \subset \mathcal{S}$ vorkommen, $k = |\mathcal{S}_{end}|$ und nr eine Numerierung von $\mathcal{S}_{end}$, so gilt:

$$T_{neu}^{alg} \notin [\{T_{alt}^{alg}\} \cup \{ t(s)^{alg} \mid s \in \mathcal{S}_{end}\}]$$

$$\Rightarrow$$

Ein Angreifer, der die Schlüssel $s_1,\ldots,s_k$ nicht, aber eine Nachricht der Bauart T_{alt}^{kr} kennt, kann keine Nachricht der Bauart T_{neu}^{kr} erzeugen. ◊

Die formulierte Bedingung ist nicht notwendigerweise auch hinreichend, weil sie z.B. nur von Nachrichten handelt, in denen keine Schlüssel des Angreifers vorkommen, oder auch, weil der Angreifer außer den Testschlüsseln nur eine einzige Nachricht kennen soll, in der diese Schlüssel vorkommen.

Die Auswahl des Teils b) der folgenden Definition ist natürlich gemäß Lemma 5 getroffen.

Definition 9: a) Folgendes ist eine notwendige Bedingung dafür, daß das algebraische Modell zu einer Spezifikation *Spez*, die eine durch „kr" bezeichnete korrekte kryptographische Interpretation besitzt, Ungleichungen korrekt modelliert:

Sind T_{neu} und T_{alt} zwei Terme ohne Variablen über *Spez*, so gilt:

$$T_{neu}^{alg} \neq T_{alt}^{alg}$$

$$\Rightarrow$$

Die Bauarten T_{alt}^{kr} und T_{neu}^{kr} sind kryptographisch verschieden. ◊

b) Unter denselben Bedingungen bedeute, daß das algebraische Modell im schwachen Sinne Ungleichungen korrekt modelliert, folgendes:

Für irgendeine korrekte kryptographische Interpretation kr' von *Spez* gibt es zu unendlich vielen Sicherheitsparametern n eine Belegung B_n mit $T_{neu}^{B_n} \neq T_{alt}^{B_n}$

$$\Rightarrow$$

Die Bauarten T_{alt}^{kr} und T_{neu}^{kr} sind kryptographisch verschieden. ◊

Anmerkung: Werden gemäß Definition 9 Ungleichungen korrekt modelliert, so werden, wie die Bezeichnungen schon suggerieren, auch im schwachen Sinne Ungleichungen korrekt modelliert. Dies folgt aus 4.1.4, wo gezeigt wurde, daß $T_{neu}^{alg} = T_{alt}^{alg}$ für jede Belegung $T_{neu}^{B} = T_{alt}^{B}$ impliziert. Umgekehrt folgt also aus $T_{neu}^{B_n} \neq T_{alt}^{B_n}$, daß $T_{neu}^{alg} \neq T_{alt}^{alg}$, und daraus, wenn Ungleichungen korrekt modelliert werden, daß T_{alt}^{kr} und T_{neu}^{kr} kryptographisch verschieden sind. ◊

Satz 1: Gegeben sei eine Spezifikation *Spez*, die eine durch „kr" bezeichnete korrekte

kryptographische Interpretation besitzt. Wenn Ungleichungen nicht einmal im schwachen Sinne korrekt modelliert werden, dann wird auch Nachrichtenunkenntnis nicht korrekt modelliert. ◊

Beweis: Hier sind im wesentlichen die Lemmata 4 und 5 zusammenzufassen. Sei dazu $T_{alt} \neq T_{neu}$ eine Ungleichung, die nicht im schwachen Sinne korrekt modelliert wird.

Einerseits folgt aus der Voraussetzung, daß $T_{alt}{}^{B_n} \neq T_{neu}{}^{B_n}$ für unendlich viele n, wenn s* ein zusätzlicher Schlüssel ist, nach Lemma 5 insbesondere

$$U(s^*, T_{neu})^{alg} \notin [\{ U(s^*, T_{alt})^{alg}\} \cup \{ t(s)^{alg} \mid s \in \mathcal{S}(T_{alt}) \cup \mathcal{S}(T_{neu}) \cup \{s^*\}\}].$$

Andererseits sind nach Voraussetzung die Bauarten $T_{alt}{}^{kr}$ und $T_{neu}{}^{kr}$ nicht kryptographisch verschieden. Lemma 4 besagt somit, daß ein Angreifer aus einer Nachricht der Bauart $(U(s^*, T_{alt}))^{kr}$ eine der Bauart $(U(s^*, T_{neu}))^{kr}$ erzeugen kann, auch wenn er alle in T_{alt} und T_{neu} vorkommenden Schlüssel und s* nicht kennt.

Diese beiden Aussagen über die Terme $U(s^*, T_{neu})$ und $U(s^*, T_{alt})$ bedeuten gerade, daß Nachrichtenunkenntnis nicht korrekt modelliert wird. ◊

Im folgenden brauchen deshalb als Gegenbeispiele zum algebraischen Modell nur noch Ungleichungen gesucht zu werden, die nicht im schwachen Sinne korrekt modelliert werden.

4.2.2 Anmerkungen zu einem rein algebraischen Lemma

Es hätte als hübscher gelten können, wenn der algebraische Teil von 4.2.1 als rein algebraisches Lemma anstelle des dortigen Lemma 5 formuliert und nur im algebraischen Modell bewiesen worden wäre; gemäß den dortigen einleitenden Bemerkungen hätte man etwa folgendes Lemma erwarten können:

Pseudolemma: Wenn im algebraischen Modell zu einer Spezifikation *Spez* für zwei Terme $T_{alt}{}^{alg} \neq T_{neu}{}^{alg}$ gilt, so kann ein Angreifer, der s nicht kennt, $U(s, T_{neu})^{alg}$ aus $U(s, T_{alt})^{alg}$ nicht erzeugen, d.h. $U(s, T_{neu})^{alg} \notin [U(s, T_{alt})^{alg}, t(s)^{alg}]$. ◊

In diesem Abschnitt seien kurz die Schwierigkeiten erläutert, auf die dieser Ansatz stieß. Dies wird zwar für den Rest der Arbeit nicht eigentlich benötigt, jedoch wird so wenigstens ein Beweis innerhalb des algebraischen Modells vorgeführt, und die Schwierigkeiten sind ein Hinweis darauf, daß das algebraische Modell nicht nur nicht korrekt ist, sondern nicht einmal unbedingt einfacher zu benutzen als das kryptographische.

Anmerkung 4: Das Pseudolemma gilt nicht. Eine notwendige Zusatzforderungen ist z.B., daß

$$s^{*alg} \notin [U(s^*, T_{alt})^{alg}, t(s^*)^{alg}],$$

d.h. der Schlüssel s* nicht schon im algebraischen Modell aus einer einzigen Unterschrift zu bestimmen ist. ◊

Beweis: Zuerst wird die Notwendigkeit gezeigt: Wenn $s^{*alg} \in [U(s^*, T_{alt})^{alg}, t(s)^{alg}]$, der Angreifer also s* erzeugen kann, so kann er natürlich auch jede Unterschrift mit s* unter eine ihm bekannte Nachricht erzeugen, d.h. $U(s^*, T_{neu})^{alg} \in [U(s^*, T_{alt})^{alg}, t(s^*)^{alg}]$, falls $T_{neu} \in [U(s^*, T_{alt})^{alg}, t(s^*)^{alg}]$, insbesondere also, wenn T_{neu} keine Schlüssel enthält.

Um zu zeigen, daß die Zusatzforderung nicht aus den Voraussetzungen des Pseudolemmas folgt, genügt das triviale Beispiel, daß es in *Spez* eine Gleichung $\forall s \in \mathcal{S}$: s = 1 geben könnte. ◊

Hätte man in *Spez* eine eigene Sorte $\mathcal{U}$ für Unterschriften vorgesehen, so hätte man mindestens noch die Zusatzforderung benötigt, daß $\mathcal{U}$ im algebraischen Modell mehr als ein Element hat. Wenn $\mathcal{U}$ einelementig wäre, so lägen nämlich sämtliche Unterschriften in derselben Klasse, d.h. der Angreifer kann für $U(s^*, T_{neu})$ jede beliebige andere nehmen, z.B. $U(s^*, T_{alt})$. Die Notwendigkeit jener Zusatzforderung ergäbe sich aus dem, wieder trivialen, Gegenbeispiel, daß es eine Gleichung $\forall u, u' \in \mathcal{U}: u = u'$ geben könnte.

Auch bei der gewählten Modellierung, in der die Unterschriften schon wieder als Nachrichten aufgefaßt werden, dürfte $\mathcal{N}$ nicht einelementig sein; dies ist aber schon durch die Voraussetzung $T_{alt}^{alg} \neq T_{neu}^{alg}$ ausgeschlossen.

Gezeigt werden konnte nur die folgende, in Voraussetzung 1. noch stärker abgeschwächte Version des Pseudolemmas. Dafür wird die Menge $\mathcal{E}$ der dem Angreifer explizit bekannten Nachrichten wieder auf die Version aus Lemma 5 ergänzt; der Beweis gilt jedoch genauso mit $\mathcal{E} = \{U(s^*, T_{alt})^{alg}, t(s^*)^{alg}\}$.

Lemma 6: Es seien eine Spezifikation *Spez* und ein Schlüssel s*, d.h. eine Grundnachricht der Sorte $\mathcal{S}$, gegeben. Im algebraischen Modell zu *Spez* gelte für zwei Terme $T1^{alg} \neq T2^{alg}$. Zusätzlich habe im algebraischen Modell zu *Spez*

1. die Ergebnissorte von U unendlich viele Elemente und
2. $s^{*alg} \notin [\mathcal{E}]$ gelte, d.h. der Schlüssel s* sei nicht schon im algebraischen Modell aus einer einzigen Unterschrift zu bestimmen.

Dann kann man T1 und T2 so als T_{alt} und T_{neu} sortieren, daß für

$$\mathcal{E} := \{U(s^*, T_{alt})^{alg}, T_{alt}^{alg}, T_{neu}^{alg}\} \cup \{t(s)^{alg} \mid s \in \mathcal{S}(T_{alt}) \cup \mathcal{S}(T_{neu}) \cup \{s^*\}\}$$

gilt:

$$U(s^*, T_{neu})^{alg} \notin [\mathcal{E}]. \quad \Diamond$$

Beweis: Die Idee des Beweises findet sich, mit bis dahin eingeführten Bezeichnungen, die in a) und b) eingeführt werden, in c) bei „genügt es".

a) Zunächst wird gezeigt, daß es nichts schadet, T_{alt} und T_{neu} etwas abzuändern:

Sei $\mathcal{S}_{end} := \mathcal{S}(T_{alt}) \cup \mathcal{S}(T_{neu}) \cup \{s^*\}$. Sind T_a und T_n zwei Terme aus denselben Klassen wie T_{alt} bzw. T_{neu}, d.h.

$$T_a^{alg} = T_{alt}^{alg} \text{ und } T_n^{alg} = T_{neu}^{alg}$$

und $\mathcal{E}' := \{U(s^*, T_a)^{alg}, T_a^{alg}, T_n^{alg}\} \cup \{t(s)^{alg} \mid s \in \mathcal{S}_{end}\}$, so ist $U(s^*, T_n)^{alg} = U(s^*, T_{neu})^{alg}$ und $\mathcal{E}' = \mathcal{E}$, also genügt es, zu beweisen, daß

$$U(s^*, T_n)^{alg} \notin [\mathcal{E}'].$$

b) Im folgenden (bei (*)) wird benötigt, daß weder T_n noch T_a einen Unterterm aus der Klasse T_n^{alg} haben. Dies soll nun durch geeignete Wahl von T_a und T_n erreicht werden. Dazu bezeichne man einen Term als <u>minimal</u>, wenn er keinen Unterterm hat, der zur selben Klasse gehört.

Allgemein gilt, daß jeder Term T entweder selbst minimal ist oder einen Unterterm aus derselben Klasse hat, der minimal ist, denn da Terme endlichen Bäumen entsprechen und Unterterme den Unterbäumen davon, kann man von T aus solange zu immer kleineren Unterbäumen übergehen, bis man einen minimalen erhält.

Die Wahl von T_a und T_n beginnt man mit einem minimalen $T \in T1^{alg}$. Wenn es keinen Unterterm aus $T2^{alg}$ hat, so setzt man $T_a := T$ und wählt ein beliebiges in $T2^{alg}$ minimales T_n dazu. Andernfalls

wähle man einen Unterterm $T' \in T2^{alg}$ von T. Wegen der Minimalität von T hat T' keinen Unterterm aus $T1^{alg}$. Deswegen kann man $T_a := T'$ setzen und als T_n diesmal einen minimalen Term aus $T1^{alg}$ wählen.

c) Annahme: Es gilt $U(s^*, T_n)^{alg} \in [\mathcal{E}]$.

Dann gibt es einen Term „Baum", in dem s* nur als t(s*) oder $U(s^*, T_a)$ oder innerhalb von T_a oder T_n vorkommt, mit

$$U(s^*,T_n)^{alg} = Baum^{alg}.$$

(Beweis wie in Lemma 5; dieser Baum entspricht wieder der Konstruktionsvorschrift, wie der Angreifer die Unterschrift fälschen kann).

Um diese Gleichung zu widerlegen, genügt es nach Definition der initialen Semantik (vgl. 2.2 bzw. [EhMa_85, Definition 2.4]), für *Spez* eine korrekte Interpretation I (im Sinne der abstrakten Datentypen) mit $U(s^*,T_n)^I \neq Baum^I$ zu finden.

I soll aus alg gewonnen werden und auf allen Objekten aus *Spez* außer auf den Funktionssymbolen U und Test mit alg übereinstimmen.

Da U und Test in *Spez* nur in der einen Gleichung „Test (t(s), N, U(s,N))=wahr" vorkommen sollten, sind die übrigen Gleichungen auch nach der Änderung von alg zu I noch gültig. I ist also eine korrekte Interpretation, sofern man noch die Gültigkeit der letzten Gleichung garantiert.

Um $U(s^*,T_n)^I \neq Baum^I$ zu erreichen, d.h. $U^I(s^{*alg},T_n^I) \neq Baum^I$ (denn $s^{*I}=s^{*alg}$), liegt es nahe,

1. $T_n^I = T_n^{alg}$ und $Baum^I = Baum^{alg}$ zu verlangen,
2. U^I genau an dem einen Wertepaar (s^{*alg}, T_n^{alg}) gegenüber U^{alg} zu ändern (wie, siehe (**)), und
3. $Test^I$ nur ggf. am Wertetripel $(t(s^*)^{alg}, T_n^{alg}, U^I(s^{*alg}, T_n^{alg}))$ gegenüber $Test^{alg}$ zu ändern, indem es dort den Wert „wahr" erhält.

2. und 3. zusammen garantieren dann, daß I eine korrekte Interpretation ist.

Enthalten T_{neu} und Baum die Symbole U und Test nicht, so ist 1. klar. Genauer kann es, wenn 2. und 3. erfüllt sind, bei 1. nur dann ein Problem geben, wenn bei der Auswertung (s. [EhMa_85, Definition 1.10]) von T_{neu}^I oder $Baum^I$ zwischendurch

$$U^I(s^{*alg}, T_n^{alg})$$

oder

$$Test^I (t(s^*)^{alg}, T_n^{alg}, U^I(s^{*alg}, T_n^{alg}))$$

vorkommt, d.h. irgendwo erstmalig die geänderten Funktionswerte vorkommen.

- Bei der Auswertung von T_n^I und innerhalb der von $Baum^I$ bei Unterbäumen T_a^I und T_n^I ist dies nicht möglich: Es hieße nämlich, daß T_n bzw. T_a einen Unterterm T_n' mit $T_n'^{alg} = T_n^{alg}$ hätte (im ersten Fall den rechten am entsprechenden Knoten „U", im zweiten den mittleren am entsprechenden Knoten „Test") und stünde im Widerspruch zur Wahl von T_n in b) (*).
- Nun wird betrachtet, ob bei der Auswertung von $Baum^I$, aber außerhalb sämtlicher Unterterme T_a und T_n, der Ausdruck $U^I(s^{*alg}, T_n^{alg})$ vorkommen kann. Dies kann

höchstens bei der Auswertung eines Knotens U geschehen.

Der eine Fall wäre, daß dieser die Wurzel eines Unterterms $U(s^*, T_a)$ ist. Dann müßte aber $T_a^{alg} = T_n^{alg}$ sein, im Widerspruch zur Hauptvoraussetzung $T1^{alg} \neq T2^{alg}$ des Lemmas.

Daraus folgt, daß der linke Unterterm s' des entsprechenden U nicht s* selbst ist, denn s* kommt in Baum nur als $t(s^*)$ oder $U(s^*, T_{alt})$ vor (oder innerhalb von T_a und T_n, was aber in diesem Punkt nicht betrachtet wird).

Somit ist s' ein Unterterm von Baum, dessen Wurzel nicht innerhalb eines $t(s)$, $U(s^*, T_{alt})$, T_a oder T_n liegt. Folglich gilt auch $s'^{alg} \in [\mathcal{E}] = [\mathcal{E}]$. Andererseits ist $s'^{alg} = s^{*alg}$, und $s^{*alg} \in [\mathcal{E}]$ steht im Widerspruch zur Voraussetzung 2.

- Es bleibt der Fall, daß bei der Auswertung von $Baum^I$, außerhalb der T_a und T_n, der Ausdruck $Test^I (t(s^*)^{alg}, T_n^{alg}, U^I(s^{*alg}, T_n^{alg}))$ vorkommt. Dabei ist $U^I(s^{*alg}, T_n^{alg})$ die in 2. neu gewählte Unterschrift, Minimalitätsüberlegungen für Baum helfen also nicht, und auch das Vorkommen der ersten beiden Argumente für Test ist völlig realistisch, da damit Test prüft, ob die gewünschte Unterschrift schon erzeugt wurde.

  Deswegen muß dieser Fall schon durch die Wahl von $U^I(s^{*alg}, T_n^{alg})$ in 2. ausgeschlossen werden, d.h. $U^I(s^{*alg}, T_n^{alg})$ ist verschieden von den Klassen aller Unterterme von Baum zu wählen (oder zumindest von den rechten Unterbäumen der Knoten Test, deren andere Unterbäume den Klassen $t(s^*)^{alg}$ und T_n^{alg} angehören). (**) Dies ist möglich, weil nach Voraussetzung 2. dieses Lemmas unendlich viele Klassen zur Auswahl stehen.

Nun ist gezeigt, daß jedes I, das man aus alg gemäß 2. und 3. incl. (**) bildet, die eingangs gewünschte Eigenschaft hat. Damit hat man auch einen Widerspruch zur Annahme erreicht. ◊

Die Gültigkeit der Zusatzvoraussetzungen zu Lemma 6 wären für jede konkret verwendete Spezifikation extra zu prüfen. Dies könnte vielleicht geschehen, indem man die zulässigen Gleichungen geeignet einschränkt. Da allerdings kaum jemand zweifeln wird, daß beide Voraussetzungen erfüllt sein müssen, wenn *Spez* in irgendeiner sinnvollen Weise ein digitales Signatursystem modelliert, liegt es hier wiederum nahe, zu beweisen, daß es aus der Existenz einer korrekten kryptographischen Interpretation folgt. Da damit jedoch gegenüber 4.2.1 nichts gewonnen wird, wird dies hier nicht weiterverfolgt.

Selbstverständlich soll nicht behauptet werden, daß der Beweis zu Lemma 6 der kürzest mögliche zu einer sinnvollen Konkretisierung des Pseudolemma sein müsse. Insbesondere könnte der Weg über den Gleichungskalkül (vgl. 2. in 2.2) beschritten werden. Dazu müßte man wohl Invarianten unter den Einsetzungsschritten mit Gleichungen aus *Spez* suchen. Zumindest die Zusatzvoraussetzung an *Spez* aus Anmerkung 4 und die Notwendigkeit, sie für jede Spezifikation einzeln zu beweisen oder in eine besser überprüfbare Form zu bringen, blieben aber auch dabei erhalten.

4.2.3 Nichtkorrektheit bei Funktionen mit kleinen Bildbereichen

Diese Tatsache wird an einem einfachen Beispiel dargestellt, da dieses allein bereits zeigt, daß die algebraische Modellierung ohne jede Einschränkung nicht korrekt sein kann.

Dazu werden die beiden Ungleichungen

$$\text{Parität}\,(U(s, 6)) \neq 0 \quad \text{und} \quad \text{Parität}\,(U(s, 6)) \neq 1$$

betrachtet, wobei „Parität" die übliche Bedeutung haben soll.

Es ist anschaulich einleuchtend, daß im algebraischen Modell beide Ungleichungen gelten müßten,

da z.B. eine Gleichung Parität (U(s, 6)) = 0 zur Folge hätte, daß bei sämtlichen Implementierungen des Signatursystems und sämtlichen Schlüsseln die Unterschrift unter die Zahl 6 gerade Parität hätte, was nicht zu erwarten ist. Ebenso verständlich ist es, daß man im kryptographischen Sinne auf keine dieser Ungleichungen vertrauen kann, da sie in etwa der Hälfte der Fälle nicht gelten.

Formal soll gemäß dem Schluß von 4.2.1 nun gezeigt werden, daß durch jedes passende algebraische Modell mindestens eine dieser beiden Ungleichungen nicht im schwachen Sinne korrekt modelliert wird.

Satz 2: Sei *Spez* eine Spezifikation, in der neben dem Signatursystem Grundnachrichten $0,1,6 \in \mathcal{N}$ und $s \in \mathcal{S}$ und ein Funktionssymbol Parität: $\mathcal{N} \rightarrow \mathcal{N}$ vorkommen. *Spez* besitze eine korrekte kryptographische Interpretation, für die Parität^{kr} die übliche Quersummenbildung über den als binär codiert angenommenen Nachrichten ist und 0^{kr} und 1^{kr} die (ebenfalls binär codierten) Zahlen 0 und 1 sind.

Dann modelliert das algebraische Modell zu *Spez* mindestens eine der Ungleichungen

$$\text{Parität}\,(U(s, 6)) \neq i, \quad i \in \{0,1\}$$

nicht im schwachen Sinne korrekt. ◊

Beweis: Zu zeigen ist, daß einerseits die Bauarten $(\text{Parität}\,(U(s, 6)))^{kr}$ und i^{kr} nicht kryptographisch verschieden sind, andererseits eine korrekte kryptographische Interpretation kr' von *Spez* existiert, bei der es zu jedem Sicherheitsparameter n eine Belegung B' mit $(U(s,6))^{B'} \neq i^{B'}$ gibt.

Ersteres bedeutet gemäß Definition 4 (in 3.4), daß für ein Polynom P und unendlich viele Sicherheitsparameter n der Anteil an Schlüsseln s_1' der Länge n mit

$$(\text{Parität}\,(U(s, 6)))^{kr}\,(s_1') = i^{kr}\,(s_1')$$

mindestens 1/P(n) beträgt, was nach Definition 7 (in 4.1.3)

$$\text{Parität}^{kr}\,(U^{kr}(s_1', 6^{kr})) = i^{kr}$$

bedeutet. Da die Parität nur 0 oder 1 sein kann, gilt für jedes n eine der beiden Gleichungen für mindestens die Hälfte aller Schlüssel, für mindestens ein i also für unendlich viele n. Man braucht also nur P(n)=2 zu wählen. Auch sei jetzt solch ein i fest gewählt.

Für dieses i wird nun andererseits kr' gesucht. Statt zu untersuchen, ob bei einem schon existierenden kryptographisch sicheren digitalen Signatursystem zu jedem n ein Schlüssel existiert, für den die Unterschrift unter 6 die Parität 1-i hat (was durchaus plausibel ist), ist es einfacher, aus einem beliebigen kryptographisch sicheren digitalen Signatursystem ein zweites zu konstruieren, in dem alle Unterschriften diese Parität haben, indem die bisherigen Unterschriften durch ein weiteres Bit auf die passende Parität ergänzt werden. Da eine sehr ähnliche Konstruktion in 4.2.4 wieder benutzt wird, werden beide zusammen in Lemma 7 durchgeführt.

Nun wendet man Lemma 7 speziell auf U^{kr} und Test^{kr} an (genauer siehe Anmerkung 5). Da sich dabei der Schlüsselraum und das Nachrichtenalphabet nicht ändern, kann man kr' aus kr bilden, indem man dieses neue digitale Signatursystem anstelle des bisherigen verwendet, und Lemma 3 besagt, daß kr' wieder eine korrekte kryptographische Interpretation von *Spez* ist. Daher folgt dann die Behauptung. ◊

Lemma 7: U, Test und t mögen ein kryptographisch sicheres digitales Signatursystem bilden. f sei eine Funktion, die durch eine höchstens polynomial wachsende Folge von Schaltnetzen berechnet werden kann.

Man erhält ein neues kryptographisch sicheres digitales Signatursystem, wenn man an jede

Unterschrift ihr Bild unter f anhängt, d.h. Nachrichten- und Schlüsselräume und t beibehält und U_f und $Test_f$ definiert durch

$$U_f(s, N) := U(s, N) \,\&\, f(U(s, N))$$

und

$$Test_f\,(\,t(s), N, N'\,) = \text{wahr} \quad :\Leftrightarrow \quad Test\,(\,t(s), N, \pi_1(N')\,) = \text{wahr} \;\wedge\; \pi_2(N') = f(\pi_1(N')),$$

wobei „&" Konkatenation und π_1 und π_2 die Projektionen auf die erste und zweite Komponente bedeuten (wozu die Konkatenation in einer Form erfolgen muß, die das Wiederauseinandernehmen gestattet). ◊

Beweis: Es wird gezeigt, daß aus jeder Schaltnetzfolge, die das neue System bricht, eine Folge von nicht wesentlich größeren Schaltnetzen konstruiert werden kann, die das ursprüngliche System bricht.

Formal bedeutet Unsicherheit des neuen Systems gemäß Definition 1, daß es Polynome P und Q gibt, so daß für unendlich viele Sicherheitsparameter n ein C_n mit $Gr(C_n) < Q(n)$ das System für einen Anteil $\geq 1/P(n)$ der Schlüssel der Länge n bricht.

Aus einem solchen C_n soll nun ein D_n konstruiert werden, das das ursprüngliche System für dieselben Schlüssel bricht. Die Konstruktion ist in Bild 7 dargestellt. Das zu konstruierende D_n ist genau wie in Bild 4 als Platine abgebildet, die einen Testschlüssel t(s) als Eingabe erhält, an na eine Nachricht N und an ua ihre Unterschrift U(s, N) ausgeben soll, in die aber dazu noch Chips eingesetzt werden müssen, die U realisieren.

Dabei kann man vom Schaltnetz C_n ausgehen (das in Bild 7 innerhalb der Gesamtplatine dunkler eingezeichnet ist). Dieses benötigt zunächst dieselbe Eingabe t(s), die man daher einfach mit der von D_n verbindet.

Außerdem erwartet es das Einsetzen von U_f-Chips (im Bild an der Stelle, wo der Untergrund von D_n durchscheint). Solche dürfen aber in D_n nicht verwendet werden, man muß sich daher mit den U-Chips und einfachen Gattern behelfen. Man verwendet jeweils ein U-Chip, das U(s, N) erzeugt. Dies wird verzweigt, auf die zweite Version f angewendet und dies mit der ersten konkateniert.

Zuletzt erzeugt C_n eine U_f-Unterschrift, von der als Ausgabe von D_n nur die erste Komponente benötigt wird.

Da sowohl die angefragten Nachrichten als auch die ausgegebene bei C_n und D_n gleich sind, ist die Unterschrift auch bei D_n neu. Da D_n nur um eine konstante Anzahl von Schaltnetzen für f und Konkatentionen größer als C_n ist, wächst es auch höchstens polynomial. ◊

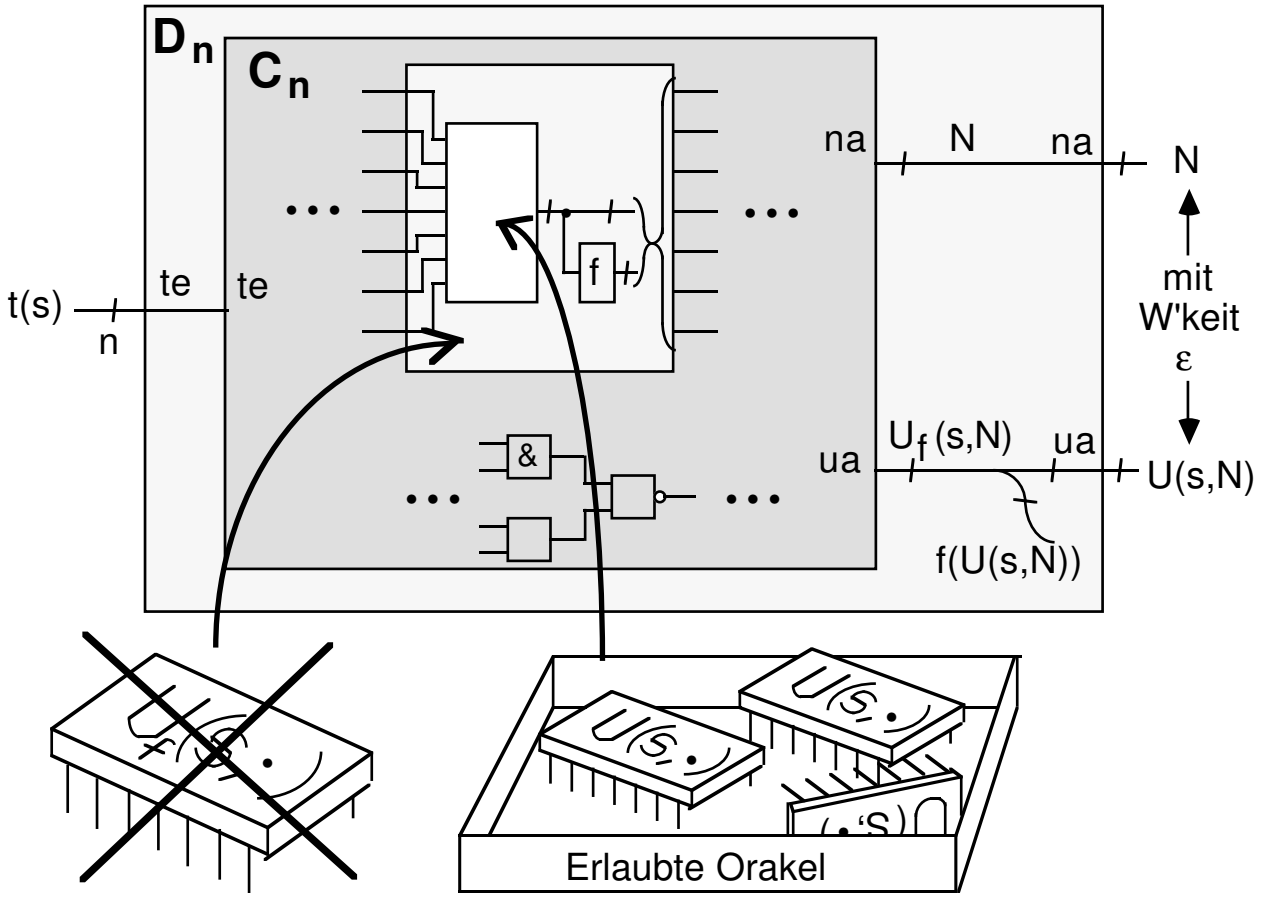


Bild 7 Sicherheitsbeweis für Signatursysteme mit Anhängsel

Anmerkung 5: Bei der Anwendung von Lemma 7 im Beweis von Satz 2 ist die benötigte Funktion f natürlich die, die das auf die Parität 1-i ergänzende Bit berechnet, d.h. f(N) := (1-i) ⊕ Parität(N). Die Konkatenation kann z.B. durch einfaches Hintereinanderschreiben erfolgen, da die zweite Komponente immer das letzte Bit ist. ◊

Man sieht nun sofort, daß man in diesem Beispiel statt „Parität" auch jede andere Funktion f mit (in der Realität bzw. dem kryptographischen Modell) endlichem Bildbereich hätte nehmen können; im Beweis muß nur P bei m-elementigem Bildbereich auf 1/m verkleinert werden. (Und daß es sich um die Nachricht „6" handelte, spielte sowieso keine Rolle.)

4.2.4 Nichterfassung möglicher innerer Struktur der Unterschriften

Funktionen mit kleinen Bildbereichen sind aber nicht die einzige Quelle für Inkorrektheiten. Eine zweite besteht darin, daß Unterschriften auf der kryptographischen Seite eine innere Struktur haben können, die auf der algebraischen nicht modelliert wurde.

Als Beispiel kann man eine Funktion „umdreh" betrachten, die das gewöhnliche Schreiben der Nachrichten von hinten nach vorn modellieren soll. Die Gleichung

$$\text{umdreh}\,(U(s, 17)) = (U(s, 17))$$

bedeutet dann, daß U(s, 17) ein Palindrom ist. Wieder leuchtet anschaulich ein, daß die Gleichung algebraisch nicht gelten kann, da sonst für jedes kryptographische digitale Signatursystem und jeden

Schlüssel die Unterschrift unter 17 ein Palindrom wäre, was unwahrscheinlich ist. Um zu zeigen, daß man sich kryptographisch jedoch nicht auf die entsprechende Ungleichung verlassen kann, wird ein kryptographisch sicheres digitales Signatursystem konstruiert, in dem alle Unterschriften Palindrome sind. Formal ist gemäß dem Schluß von 4.2.1 folgendes zu zeigen:

Satz 3: Sei *Spez* eine Spezifikation, in der neben dem Signatursystem Grundnachrichten $17 \in \mathcal{N}$ und $s \in \mathcal{S}$ und ein Funktionssymbol umdreh: $\mathcal{N} \rightarrow \mathcal{N}$ vorkommen. *Spez* besitze eine korrekte kryptographische Interpretation kr", für die $\text{umdreh}^{kr"}$ das übliche Rückwärtsschreiben über den als binär codiert angenommenen Nachrichten ist.

Dann modelliert das algebraische Modell zu *Spez* die Ungleichung

$$\text{umdreh}\,(U(s, 17)) \neq U(s, 17)$$

nicht im schwachen Sinne korrekt. ◊

Beweis: Es genügt zu zeigen, daß es korrekte kryptographische Interpretationen kr und kr' gibt, so daß die Bauarten $(\text{umdreh}\,(U(s,17)))^{kr}$ und $(U(s,17))^{kr}$ nicht kryptographisch verschieden sind, aber bei kr' für alle Belegungen $(\text{umdreh}\,(U(s,17)))^{B} \neq (U(s,17))^{B}$ gilt.

Gemäß Lemma 7 kann man jedem kryptographisch sicheren digitalen Signatursystem mit U, Test und t ein neues mit gleichem Schlüsselraum und Nachrichtenalphabet und den Funktionen U_{umdreh} und $Test_{umdreh}$ machen. In diesem sind alle Unterschriften Palindrome. (Die Konkatenation kann man, da beide Komponenten gleichlang sind, wieder ohne Trennzeichen durchführen.)

Ebenso kann man ein Signatursystem mit U_{inv} und $Test_{inv}$ bilden, in dem keine einzige Unterschrift ein Palindrom ist, indem man als „inv" die Funktion wählt, die jeder Nachricht das Inverse des ersten Bits zuordnet.

Wendet man diese Konstruktionen auf kr" an, d.h. ersetzt $U^{kr"}$ und $Test^{kr"}$ durch $U^{kr"}{}_{umdreh}$ und $Test^{kr"}{}_{umdreh}$ bzw. $U^{kr"}{}_{inv}$ und $Test^{kr"}{}_{inv}$, so erhält man nach Lemma 3 zwei neue korrekte kryptographische Interpretationen von *Spez*. Die erste davon kann man als kr, die zweite als kr' verwenden. ◊

4.2.5 Diskussion des Ergebnisses

In 4.2.3 und 4.2.4 wurden zwei Gegenbeispiele angegeben, von denen jedes zusammen mit 4.2.1 besagt, daß das algebraische Modell gemäß den verwendeten Definitionen Nachrichtenunkenntnis nicht korrekt modelliert.

Hier soll noch informell die Relevanz dieser Aussagen diskutiert und weitere Aspekte des Vergleichs der beiden Modelle erwähnt werden.

Zunächst wird die Relevanz des Paritätsbeispiels aus 4.2.3 betrachtet. Hier ging es (zusammen mit 4.2.1) um die unerlaubte Kenntnis einer Nachricht der Bauart U(s*, Parität (U(s, 6))).

Ein erster möglicher Kritikpunkt an der Relevanz dieses Beispiels ist, daß hier die Bauart explizit definiert wird, wogegen in 3.4 diskutiert wurde, daß in Aussagen über Nachrichtenunkenntnis an dieser Stelle eine implizite Bauartsdefinition sinnvoller ist, d.h. die Angabe eines Tests, den der Angreifer mit dieser Nachricht unerlaubterweise besteht. Ein trivialer solcher Test ist durch „Ist N = U(s*, Parität (U(s, 6))) ?" gegeben, doch muß man zu dessen Durchführung beide Schlüssel s und s* kennen, d.h., man testet, ob man selbst eine Nachricht unterschrieben hat, was kaum sinnvoll klingt.

Sinnvoller ist der Test „Gilt Test (t(s*), Parität (U(s, 6)), N) = wahr ?", den der Besitzer X des Schlüssels s, aber nicht von s*, durchführen kann. Ersetzt man nun Parität durch eine Hash-

Funktion, die Nachrichten auf 32 bit kürzt, was nach dem Schluß von 4.2.3 immer noch ein Gegenbeispiel ist, so könnte man sich vorstellen, daß dies ein Protokollteil ist, in dem X prüft, daß der Besitzer Y von s* die Nachricht U(s, 6) von X empfangen hat und in diesem Zusammenhang für irgendetwas garantiert. Falls Y manchmal in anderem Zusammenhang 32-bit-Nachrichten unterschreibt, könnte der Angreifer diese Garantie evtl. vortäuschen und dadurch X zu einer Handlung veranlassen, die dieser andernfalls unterließe. Dies würde im algebraischen Modell nicht bemerkt.

Selbstverständlich wäre dieses Beispiel ein schlecht entworfenes Protokoll, aber ein Beweissystem muß natürlich nicht nur kluge, sondern auch dumme Fehler finden.

Ein Ausschließen der Beispiele aus 4.2.3 über zusätzliche Regeln für den Protokollentwurf wäre möglich, z.B. indem man die Verwendung von Funktionen mit endlichen Bildbereichen verböte (d.h. hier etwa die Länge der Ergebnisse der Hash-Funktion mit dem Sicherheitsparameter wachsen ließe). An anderen Protokollstellen sind solche Funktionen jedoch sicherlich erwünscht, z.B. die Funktion „Test" des Signatursystems. Muß man aber beim Protokollentwurf kompliziertere Regeln verwenden, so lohnt sich das Modell wohl kaum noch.

Zur Diskussion der Relevanz des Palindrombeispiels kann man das Beispiel zunächst ähnlich wie das vorige in ein Protokoll einbetten: Diesmal geht es um die unerlaubte Kenntnis einer Nachricht der Bauart U(s*, umdreh(U(s, 6))) bei Kenntnis von U(s*, U(s, 6)).

Wieder könnten die Nachrichten bedeuten, daß Y, der Besitzer von s*, dem Besitzer X von s in Bezug auf dessen Anfrage U(s, 6) eine Garantie gibt. Dabei gebe es zwei Typen von Garantien (z.B. über Kreditwürdigkeit bis zu verschiedenen Beträgen), von denen die weitergehende durch U(s*, umdreh(U(s, 6))), die eingeschränktere durch U(s*, U(s, 6)) ausgedrückt wird. X testet sie mittels „Ist Test (t(s*), N, umdreh (U(s, 6))) = wahr ?" bzw. „Ist Test (t(s*), N, U(s, 6)) = wahr ?". In diesem Fall könnte derjenige Z, um dessen Kreditwürdigkeit es geht (wozu man am besten „Z" statt „6" verwendet), die geringere Bescheinigung für die stärkere ausgeben, was im algebraischen Modell nicht bemerkt würde.

Ob dieser Fehler in der Praxis entstehen könnte, ist zu bezweifeln, da man die Eigenschaft eines digitalen Signatursystems, daß alle Unterschriften Palindrome sind, sicher bald feststellen würde. Das Beispiel kann aber wiederum auf andere innere Strukturen der Unterschriften übertragen werden, solange diese sich durch Gleichungen ausdrücken lassen.

Wollte man alle diese inneren Strukturen gemeinsam ausschließen, müßte man wohl eine Art Zufallseigenschaft der Unterschriften zeigen. Dies würde spätestens, wenn die Unterschriften irgendwie formatiert dargestellt werden müssen, einfach nicht gelten, und auch, falls es einmal gelten sollte, zusätzliche kryptographische Definitionen und Beweise verlangen.

Die Alternative, das verwendete digitale Signatursystem nur auf die Abwesenheit der im verwendeten Protokoll störenden Eigenschaft zu prüfen (hier also darauf, daß nicht alle Unterschriften Palindrome sind), ist noch weniger gangbar, da das algebraische Modell diese Strukturen eben nicht bemerkt.

Selbst wenn die Gegenbeispiele nicht wären, spräche noch manches gegen das algebraische Modell:

Verlangt man, daß auch Ungleichungen korrekt modelliert sein müssen (nicht nur schwach, vgl. Definition 9a in 4.2.1), weil auch dies bei Beweisen mancher Protokolleigenschaften eine Rolle spielen könnte, so müßte insbesondere für alle Terme T1, T2 ohne Schlüssel und Variable, d.h. für die $T1^{kr}$ bzw. $T2^{kr}$ jeweils eine feste Nachricht ist, aus $T1^{kr} = T2^{kr}$ auch $T1^{alg} = T2^{alg}$ folgen. Dies besagt etwa, daß die Gleichungen in $\mathcal{Spez}$ die nicht-kryptographischen Funktionen vollständig beschreiben müssen. Man dürfte also nicht, wie bei abstrakten Datentypen sonst, in die Spezifikation nur die Gleichungen aufnehmen, die man innerhalb der Anwendung (hier des Protokollablaufs im

korrekten Fall) tatsächlich benötigt, und Gültigkeit oder Ungültigkeit der übrigen der Implementierung anheimstellen. Dies allein dürfte für manche Fälle problematisch sein.

Auch zeigen die Bemühungen in 4.2.2, daß der Umgang mit dem algebraischen Modell nicht unbedingt leichter sein muß als mit dem kryptographischen, obwohl seine Notationen, wie die Darstellung der Nachrichten als Terme, viel näher an der üblichen Art, mit Protokollen umzugehen, sind.

5 Ausblick

Nach den vorangegangenen Kapiteln, insbesondere der Diskussion in 4.2.5, erscheint es für Sicherheitsbeweise nicht sinnvoll, nun, da es auch die kryptographische Modellierung und hinreichend gute digitale Signatursysteme gibt, mit der algebraischen Modellierung weiterzuarbeiten, genauer mit der initialen Semantik, d.h. den Termklassen, als Modell. (Hingegen kann das Modell für schnelle, z.T. automatisierte Unsicherheitsbeweise weiterhin sinnvoll sein.)

Das ändert nichts daran, daß eine Darstellung von Nachrichten in Protokollen durch Terme aus Grundnachrichten und darauf angewandten Operatoren nett und relativ natürlich ist (was man z.B. daran sehen kann, daß das Beispiel im nur vom kryptographischen Modell handelnden Abschnitt 3.4 mit einer Termdarstellung eingeleitet wurde). Sie sollte also nicht unbedingt mitabgeschafft werden.

Man könnte also Protokolle weiterhin in der Termdarstellung formulieren, alle Sicherheitsbeweise aber im kryptographischen Modell vornehmen. Dazu würde man weiterhin die Definitionen aus 3.4 benötigen (evtl. an einigen, in 3.4 teilweise schon gekennzeichneten Stellen erweitert) und die Zuordnungen zwischen Termen und dem kryptographischen Modell aus 4.1.1 bis 4.1.3.

Damit könnte man auch versuchen, die Gültigkeit der Sicherheitsaussagen aus Arbeiten, die das algebraische Modell für Protokolle mit digitalen Signatursystemen verwenden, also [Waid_85, Bürk_87], zu untermauern, insbesondere, da dort sowieso schon mehr mit Termen als mit Klassen gearbeitet wird. Die dort zugrundegelegten Aussagen über Nachrichtenunkenntnis, die teilweise auch schon im algebraischen Modell nicht mit voller Strenge bewiesen wurden, müßten dazu im kryptographischen Modell bewiesen werden. Die darauf aufbauenden Protokollbeweise, auf denen das Schwergewicht jener Arbeiten liegt, könnten, falls obiges gelingt, wahrscheinlich ziemlich unverändert bleiben.

Statt jede einzelne Aussage über Nachrichtenunkenntnis (die über eine simple Unterschrift hinausgeht) einzeln zu beweisen, scheint es möglich zu sein, Lemmata über Nachrichtenunkenntnis von Termen gewisser Strukturen vorweg zu beweisen (z.B. daß ein Angreifer aus Unterschriften unter Nachrichten, die in einem bestimmten Format einen Zeitstempel tragen, keine unter Nachrichten mit einem anderen Zeitstempel erzeugen kann). So könnte man evtl. viele der Dinge, die man im algebraischen Modell sofort zu erkennen meinte, auch künftig verwenden.

Ein vollständig anderer (neuerer) Ansatz für sichere Protokolle besteht in der Verwendung universeller Übersetzungstechniken, die aus einer Dienstspezifikation direkt eine sichere und zudem datenschützende Implementierung erzeugen (z.B. [GoMW_87, BrCC_88]). Dies ist in vieler Hinsicht einfacher als der bisherige Ansatz. Speziell für komplexe Dinge wie digitale Zahlungssysteme und Informationskauf ist der auch bei diesem Ansatz anfallende Teil der Arbeit, nämlich die Erzeugung der vollkommen formalisierten Dienstspezifikation, bereits nicht trivial.

Aus zwei Gründen könnte aber eine Erzeugung der Protokolle von Hand mittels einer Aneinanderreihung von Unterschriften (wie etwa in Bild 1) mit anschließendem Beweis noch eine Weile neben den allgemeinen Übersetzungstechniken bestehen:

Zum einen sind die automatisch entstehenden Protokolle zur Zeit noch sehr ineffizient, da der Ansatz der Übersetzungstechniken darin besteht, die in der Dienstspezifikation vorkommenden Funktionen als Schaltnetze (also wieder keine Schaltwerke!) darzustellen und diese Darstellung gatterweise zu verschlüsseln. Bei digitalen Zahlungssystemen müßten als Funktionen wohl Unterschriften vorkommen, da z.B. das Geld oder eine Quittung, die einer der Beteiligten erhält, durch unterschriebene Nachrichten dargestellt werden. Nach 3.3 hätte beim Sicherheitsparameter n die Darstellung $O(n^3)$ Gatter, die alle verschlüsselt und als Nachrichten versandt werden müßten.

Zum anderen sind die entstehenden Protokolle unanschaulich, während von Hand mit einzelnen Unterschriften solche entworfen werden können, die (wie das in Bild 1) weitgehend den bisherigen, nicht digitalen Abläufen entsprechen. Die gesellschaftliche oder juristische Akzeptanz solcher Protokolle könnte also höher sein.

6 Literatur

BeMi_88 Mihir Bellare, Silvio Micali: How to sign given any trapdoor function; 20th Symposium on Theory of Computing (STOC) 1988, ACM, New York 1988, Seite 32 bis 42.

BlFM_88 Manuel Blum, Paul Feldman, Silvio Micali: Non-interactive zero-knowledge and its applications (extended abstract); 20th Symposium on Theory of Computing (STOC) 1988, ACM, New York 1988, Seite 103 bis 112.

Bras_88 Gilles Brassard: Modern Cryptology - A Tutorial; LNCS 325, Springer-Verlag, Berlin 1988.

BrCC_88 Gilles Brassard, David Chaum, Claude Crépeau: Minimum Disclosure Proofs of Knowledge; Journal of Computer and System Sciences 37 (1988) 156-189.

Bric_85 Ernest F. Brickell: Breaking Iterated Knapsacks; Crypto '84, LNCS 196, Springer-Verlag, Heidelberg 1985, Seite 342 bis 358.

Bürk_87 Holger Bürk: Modellierung und Verifikation der Sicherheit und Lebendigkeit von Kryptoprotokollen; Diplomarbeit am Institut für Rechnerentwurf und Fehlertoleranz, Universität Karlsruhe, März 1987.

ChEv_87 David Chaum, Jan-Hendrik Evertse: A secure and privacy-protecting protocol for transmitting personal information between organizations; Crypto '86, LNCS 263, Springer-Verlag, Berlin 1987, Seite 118 bis 167.

DaHo_88 James A. Davis, Diane B. Holdridge: Factorization of Large Integers on a Massively Parallel Computer; Eurocrypt '88, LNCS 330, Springer-Verlag, Berlin 1988

DiHe_76 Whitfield Diffie, Martin E. Hellman: New Directions in Cryptography; IEEE Transactions on Information Theory, Band IT-22, Nr. 6, November 1976, Seite 644 bis 654.

DoEK_82 D. Dolev, S. Even, R. M. Karp: On the Security of Ping-Pong Protocols; Information and Control Band 55, 1982, Seite 57 bis 68.

DoYa_81 D. Dolev, A. C. Yao: On the Security of Public Key Protocols; 22nd Symposium on Foundations of Computer Science (FOCS) 1981, IEEE Computer Society, 1981, Seite 350 bis 357.

DoYa_83 Danny Dolev, Andrew C. Yao: On the Security of Public Key Protocols; IEEE Transactions on Information Theory, Band IT-29, Nr. 2, March 1983, Seite 198 bis 208.

EhMa_85 H. Ehrig, B. Mahr: Fundamentals of Algebraic Specifications 1; EATCS Monographs on Theoretical Computer Science Band 6, W. Brauer, G. Rosenberg, A. Salomaa (eds.), Springer-Verlag Berlin, Heidelberg, Tokyo 1985.

EvGS_85 Shimon Even, Oded Goldreich, Adi Shamir: On the security of ping-pong protocols when implemented using RSA (extended abstract); Crypto 85, Abstract Band, Tagungsband erscheint 1986 im Springer-Verlag in der Reihe LNCS.

GaJo_79 Michael R. Garey, David S. Johnson: Computers and Intractability - A Guide to the Theory of NP-Completeness; W.H. Freeman and Company, New York 1979.

Gol1_87 Oded Goldreich: Two Remarks Concerning the Goldwasser-Micali-Rivest Signature Scheme; Crypto '86, LNCS 263, Springer-Verlag, Berlin 1987, Seite 104 bis 110.

GoMi_82 Shafi Goldwasser, Silvio Micali: Probabilistic Encryption & How to Play Mental Poker Keeping Secret All Partial Information; 14th Symposium on Theory of Computing (STOC) 1982, ACM, New York 1982, Seite 365 bis 377.

GoMi_84 Shafi Goldwasser, Silvio Micali: Probabilistic Encryption; Journal of Computer and System Sciences Band 28, 1984, Seite 270 bis 299.

GoMR_84 Shafi Goldwasser, Silvio Micali, Ronald L. Rivest: A "Paradoxical" Solution to the Signature Problem; 25th Symposium on Foundations of Computer Science (FOCS) 1984, IEEE Computer Society, 1984, Seite 441 bis 448.

GoMR_88 Shafi Goldwasser, Silvio Micali, Ronald L. Rivest: A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks; SIAM J. Comput. 17/2 (1988) Seite 281 bis 308.

GoMT_82 Shafi Goldwasser, Silvio Micali, Po Tong: Why and How to establish a Private Code On a Public Network; 23rd Symposium on Foundations of Computer Science (FOCS) 1982, IEEE Computer Society, 1982, Seite 134 bis 144.

GoMW_87 Oded Goldreich, Silvio Micali, Avi Wigderson: How to solve any Protocol Problem (extended abstract); 19th STOC 1987, Seite 218 - 229.

KuRo_86 Herbert Kubicek, Arno Rolf: Mikropolis; Mit Computernetzen in die "Informations-gesellschaft"; 2. Auflage, VSA-Verlag, Hamburg 1986.

Lens_86 H. W. Lenstra, Jr.: Factoring Integers with Elliptic Curves; Version 19860512, erhalten von W. Trinks 1986.

Merr_83 Michael John Merritt: Cryptographic Protocols; Ph. D. Dissertation, School of Information and Computer Science, Georgia Institute of Technology, February 1983.

MiRS_88 Silvio Micali, Charles Rackoff, Bob Sloan: The Notion of Security for Probabilistic Cryptosystems; SIAM J. Comput. 17/2 (1988) Seite 412 bis 426.

Pome_84 Carl Pomerance: Analysis and comparison of some integer factoring algorithms; Computational Methods in Number Theory Band 1; H. W. Lenstra jr., R. Tijdeman (eds.); Mathematical Centre Tracts 154 CWI Amsterdam (ehemals Mathematisch Centrum), Amsterdam 1984, Seite 89 bis 139.

PoST_88 Carl Pomerance, J. W. Smith, Randy Tuler: A pipeline architecture for factoring large integers with the Quadratic Sieve algorithm; SIAM J. Comput. 17/2 (1988) Seite 387 bis 403.

PWP_87 Birgit Pfitzmann, Michael Waidner, Andreas Pfitzmann: Rechtssicherheit trotz Anonymität in offenen digitalen Systemen; Computer und Recht 3/10,11,12 (1987) Seiten 712 bis 717, 796 bis 803, 898 bis 904.

Sch4_85 Uwe Schöning: Complexity and Structure; LNCS 211, Springer-Verlag, Berlin 1985.

Sham_84 Adi Shamir: A Polynomial-Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem; IEEE Transactions on Information Theory Band IT-30, Nu. 5, September 1984, Seite 699 bis 704.

Shan_49 Claude E. Shannon: Communication in the Presence of Noise; Proceedings of the Institute of Radio Engineers, Band 37, Nummer 1, Januar 1949, Seite 10 bis 21; reprinted in Proceedings of the IEEE Band 72, Nr. 9, September 1984, Seite 1192 bis 1201.

Sper_84 Volker Sperschneider: Logik; Manuscript zur Vorlesung WS 84/85, Institut für Informatik I, Universität Karlsruhe.

SteW_85 Wilhelm Steinmüller: Soziale Beherrschbarkeit offener Netze; Proceedings der 1. GI Fachtagung Datenschutz und Datensicherung im Wandel der Informationstechnologien, München, Oktober 1985, herausgegeben von P.P.Spies, Informatik-Fachberichte Band 113, Springer-Verlag Berlin Heidelberg New-York Tokyo 1985, Seite 237 bis 249.

Waid_85 Michael Waidner: Datenschutz und Betrugssicherheit garantierende Kommunikationsnetze. Systematisierung der Datenschutzmaßnahmen und Ansätze zur Verifikation der Betrugssicherheit; Diplomarbeit am Institut für Informatik IV, Universität Karlsruhe, August 1985, Interner Bericht 19/85 der Fakultät für Informatik.

WaPf_85 Michael Waidner, Andreas Pfitzmann: Betrugssicherheit trotz Anonymität. Abrechnung und Geldtransfer in Netzen; 1. GI Fachtagung Datenschutz und Datensicherung im Wandel der Informationstechnologien, IFB 113, Springer-Verlag, Berlin 1985, Seite 128 bis 141; Überarbeitung in: Datenschutz und Datensicherheit DuD /1 (1986), Seite 16 bis 22.