Betrugssicherheit trotz Anonymität
Abrechnung und Geldtransfer in Netzen

Michael Waidner, Andreas Pfitzmann

Kurzfassung

Ein formales Modell zur Beschreibung von Wertetransferprotokollen in anonymen Kommunikationssystemen und zum Beweis der Betrugssicherheit dieser Protokolle wird informell vorgestellt. Das Modell wird auf ein Beispielprotokoll angewandt.

1 Einleitung

Diensteintegrierende digitale Rechnernetze, kurz ISDN genannt, sollen in Zukunft einen großen Teil der Aufgaben des täglichen Lebens wie Briefverkehr, Warenbestellung und -bezahlung, Geldüberweisung und Erfragen von Daten aller Art übernehmen. Dies geschieht durch Verwendung verschiedener *Dienste*, z. B. zur Übermittlung von Nachrichten, zur Transferierung von (digitalem) Geld, zur Abfrage von Daten (Adreßauskunft, Börsenkurse) usw.

Die Inanspruchnahme solcher Dienste durch einen Benutzer des ISDN's verursacht diesem auf zweierlei Arten Kosten:

- Der Netzbetreiber, d. h. die Post, erhebt für die Übermittlung der notwendigen Nachrichten Gebühren (*Abrechnung*).
- Der Anbieter des Dienstes erwartet vom Benutzer eine angemessene Bezahlung. Dies erfordert Verfahren (Protokolle) zum *Geldtransfer.*

In der Literatur finden sich für diese Abrechnungs- und Geldtransferaufgaben verschiedene Lösungen: In Pfit_83 wird ein Abrechnungsprotokoll mit lokalen Gebührenzählern sowie eines über Vermittlungszentralen vorgeschlagen. Protokolle zum Geldtransfer finden sich in Pfit_83 (anonyme Nummernkonten, unmanipulierbare Zähler), Riha_83, Riha_85 (Zentralbanken), Chau_83, Chau_85 (Bargeldersatz) und EvGY_84 (elektronische Brieftasche). In allen Fällen stellen sich zwei Fragen:

- Sind die Verfahren *betrugssicher* oder kann der Abrechnungsmechanismus umgangen, ein Dienst ohne angemessene Bezahlung erschwindelt oder die Bezahlung eines Dienstes herbeigeführt werden, ohne diesen zu erbringen?
- Garantieren die Verfahren die *Anonymität* des Dienstbenutzers vor dem Dienstanbieter und dem Netzbetreiber bzw. die des Dienstanbieters vor dem Dienstbenutzer und dem Netzbetreiber?

Die Notwendigkeit der ersten Frage liegt auf der Hand. Daß ein Protokoll auch die zweite Frage mit einem „Ja" zu beantworten gestatten sollte, bedarf einiger Überlegung. Angenommen, die Anonymität gegenüber dem Netz wäre nicht gesichert, so könnte der Betreiber (oder durch „Trojanische Pferde" der Hersteller der Netz-Software und -Hardware) leicht das wesentliche Kommunikationsverhalten eines Benutzers protokollieren. Da auch Warenbestellungen, Zeitschriftenlektüre und Fernsehprogramme über das Netz vermittelt werden, rückt die Möglichkeit der Erstellung von Persönlichkeitsbildern, das Ende der Privatsphäre, bedrohlich in den Bereich des Realisierbaren. Entsprechendes gilt für die Anbieter der Dienste selbst (vgl. Chau_85, Pfit_85).

In dieser Arbeit sollen die Möglichkeiten eines formalen *Beweises* der Betrugssicherheit von Geldtransferprotokollen untersucht werden unter der Annahme eines anonymen Kommunikationssystems. Möglichkeiten zur Realisierung eines anonymen Kommunikationssystems sind z. B. beschrieben in Pfit_85, Pfi1_85 und Waid_85.

2 Modellierung

In diesem Kapitel wird informell ein Modell zur Beschreibung von Wertetransferprotokollen beschrieben. Eine genauere und formale Darstellung dieses Modells findet sich in Waid_85.

2.1 Protokolle

Unter einem *Protokoll* versteht man eine Menge von *Vorschriften*, nach denen in einem Kommunikationssystem (KomSy) verschiedene *Instanzen* miteinander *Nachrichten* austauschen, um ein gemeinsames Ziel zu erreichen. Dieses Ziel kann der einfache Transport von Information sein oder, darauf aufbauend, der Transfer von Werten.

Im folgenden wird dem KomSy, auf dem die Protokolle zum Wertetransfer aufsetzen, stets unterstellt, daß es

- integren, d. h. fehlerfreien und gegen Veränderungen geschützten Transport von Nachrichten leistet,
- alle Nachrichten an alle Instanzen verteilt,
- fair arbeitet, d. h. jeder sendewilligen Instanz in begrenzter Zeit die Möglichkeit gibt, wirklich zu senden und

Dies ist eine Überarbeitung unseres Beitrags zur 1. GI-Fachtagung „Datenschutz und Datensicherung im Wandel der Informationstechnologien"

– anonym arbeitet, d. h. den Sender und Empfänger einer Nachricht vor einem Angreifer auf der Ebene des KomSy geheimhält.

Integrität und Fairness werden von realen KomSy's gemeinhin erfüllt. Die Forderung nach Anonymität kann durch geeignete Maßnahmen befriedigt werden (vgl. Pfit_85), ist aber für den Nachweis der Betrugssicherheit nicht notwendig, sondern eher erschwerend. Die Anonymität impliziert dabei bereits, daß das KomSy Nachrichten verteilt (vgl. Pfit_85), wenn auch nicht notwendig an alle (broadcasting), sondern nur an hinreichend viele (multicasting). Das Modell unterstellt vollständige Verteilung, doch kann diese Forderung ohne große Schwierigkeiten aufgegeben werden (vgl. DoWi_83).

Auf diesem KomSy sollen *Werte sicher* und *anonym* transferiert werden. Die hierzu geeigneten Protokolle benötigen einige Hilfsmittel aus der Kryptographie zur Geheimhaltung (Konzelation) und Authentikation von Nachrichten. Diese Hilfsmittel sind z. B. in AkI_83, Denn_82, DiHe_79 und Hors_85 ausführlich beschrieben. Im folgenden benötigen wir hiervon:

– Ein *Kryptosystem mit öffentlichen Schlüsseln*, kurz KröS genannt. Die Schlüsselpaare dieses Systems werden mit $(\ddot{o}_x, p_x)$ bezeichnet, wobei $\ddot{o}_x$ den öffentlichen, p_x den privaten Schlüssel meint. Statt durch eigene (mit Schlüsseln parametrisierte) Ver- und Entschlüsselungsoperationen, wird die Verschlüsselung der Nachricht n mit dem Schlüssel $\ddot{o}_x$ mit $\ddot{o}_x(n)$ bezeichnet, die Entschlüsselung mit p_x mit $p_x(n)$. Es wird angenommen, daß für jedes Schlüsselpaar $\ddot{o}_x p_x = p_x \ddot{o}_x = id$ gilt (id = Identität). Das RSA-System (RSA_78) erfüllt diese Annahme.
– Ein *Signatursystem* zur Authentikation. Der Einfachheit halber wird im folgenden stets das KröS zur Signaturbildung verwendet. Ist der öffentliche Schlüssel $\ddot{o}_x$ einer Instanz I_x bekannt, so kann diese eine Nachricht n unterschreiben, indem sie $p_x(n)$ bildet. Diese Unterschrift ist für jede andere Instanz, die $\ddot{o}_x$ kennt, überprüfbar.
– Eine sichere *Einwegfunktion* ew, d. h. eine Funktion, die zwar leicht zu bilden, deren Umkehrung aber für jeden Wert sehr schwer ist. Im folgenden kann statt des Wertes ew(n) der Einwegfunktion stets auch n selbst verwendet werden, d. h. ew wird nicht zur Authentikation verwendet. Gründe für die Verwendung von ew sind hier, daß ew zum einen überflüssige Information verbirgt, zum anderen, daß ew in der Realität als komprimierende Funktion angenommen werden kann, so daß es effizienter sein kann, ew(n) statt n zu übertragen.

Das verwendete Modell stellt Kryptosysteme und ähnliches durch Operationen auf einer Nachrichtenmenge dar, bildet also eine abstrakte Algebra (Herm_67). Indem die Nachrichtenmenge nicht jeder Instanz voll bekannt gemacht wird, entfällt im Modell der Angriff auf das Kryptosystem durch vollständige Suche (Merr_83), der in der Realität aus Zeitgründen ausscheidet.

Seinem Wesen nach kann ein KomSy nur Nachrichten, also Informationen transportieren, keine materiellen Güter. Aus diesem Grunde stellt das Modell Werte durch Nachrichten dar. Diese Betrachtungsweise ist auch in der Realität üblich, wo beispielsweise eine Besitzurkunde oder eine Banknote einen bestimmten Wert darstellen, der von ihrem Papierwert verschieden ist.

Der Besitz einer Instanz wird durch die Nachrichten dargestellt, die sie kennt. So kann ein Kontoguthaben durch die Menge der (von einer Bank unterschriebenen) Gut- und Lastschriften dieses Kontos dargestellt werden.

Diese Betrachtungsweise findet sich im Modell wieder durch die *Zähler*. Ein Zähler ist eine Funktion, die für eine bestimmte Instanz jeder Nachricht einen Wert zuordnet. Definiert man zum Beispiel einen Zähler K, der Kontowerte zählt, so ordnet K einer Gutschrift über d Einheiten für Instanz I_x eben den Wert d für Instanz I_x zu. Da einer Gutschrift für I_x gemeinhin eine Lastschrift für eine Instanz I_y entspricht, könnte K derselben Nachricht für I_y den Wert -d zuordnen. Der Kontostand von I_x ergibt sich dann als der *Zählerstand* des Zählers K: er ist gleich der Summe der Werte, die der Zähler K für die Instanz I_x den von I_x gesendeten und empfangenen Nachrichten zuordnet.

Neben dem Kontozähler müssen je nach Anwendung und Protokoll weitere Zähler eingeführt werden, so etwa ein *Wertezähler* W, der Betriebsmittelwerte einer Instanz zählt. Er könnte z. B. den Antworten einer Datenbank die Kosten zuordnen, die zur Erzeugung dieser Antworten in Form von Rechenleistung aufgebracht werden mußten.

Die eigentlichen Protokolle werden in der Literatur im allgemeinen durch Angabe eines korrekten, also fehlerfreien Ablaufs des Protokolls angegeben. Häufig geschieht dies aus der Sicht einer der beteiligten Instanzen, z. B. bei der Befragung einer Datenbank aus der Sicht des Fragestellers. Ein sehr primitives Protokoll hierfür könnte lauten:

Pseudoprotokoll „Datenbankanfrage"

1. Schritt Erzeuge eine Datenbankanfrage f und einen Scheck c in Höhe der für die Beantwortung der Frage anfallenden Kosten. Sende f und c an die Datenbank.
2. Schritt Erhalte von der Datenbank die Antwort a auf die Frage f.

Solch eine sehr vage informelle Beschreibung muß im Modell in eine formale Beschreibung umgesetzt werden. Hierzu muß das Protokoll zumindest auch den Betrugsfall behandeln. Um einen Betrug behandeln zu können, muß dieser *entdeckbar* sein, d. h. im Protokoll müssen entsprechende *Tests* enthalten sein. Im obigen Beispiel könnte der Fragesteller im zweiten Schritt prüfen, ob die Antwort a befriedigend war und andernfalls c zurückziehen.

Die einzelnen *Protokollabläufe* dieses Protokolls werden nicht streng sequentiell ausgeführt, sondern eine Datenbank wird mehrere Anfragen gleichzeitig beantworten wollen. Zudem wird, beachtet man das Ziel der Betrugssicherheit, die Sicht jeder Instanz interessieren, nicht nur die der zufällig zur Beschreibung des Protokolls ausgewählten. Aus diesen Gründen spaltet man Protokolle für n Instanzen in n *Teilprotokolle* auf, die jeweils aus einzelnen *Regeln* bestehen. Eine Protokollausführung besteht dann darin, daß jede beteiligte Instanz aus den bereits erhaltenen Nachrichten ermittelt, welche Regel des Protokolls gerade anwendbar ist und diese dann anwendet. Eine Regelanwendung hat im allgemeinen das Senden einer Nachricht zur Folge.

Die Herstellung des Kontextes einer Regel, d. h. die Feststellung, welche Nachrichten zu einem Protokollablauf ge-

hören, geschieht im Modell durch die Folge der bereits empfangenen Nachrichten zusammen mit dem Anfangswissen der einzelnen Instanzen (z.B. deren private Schlüssel).

Gegenüber der Realität ist dies keine Einschränkung, da ein Protokollablauf, an dem mehrere Instanzen beteiligt sind, notwendig durch die ausgetauschten Nachrichten festgelegt ist. Weitere, durch die Instanzen während eines Protokollablaufs erzeugte und nur lokal vorhandene Information wäre überflüssig, da sie aus den ausgetauschten Nachrichten jederzeit neu erzeugt werden kann. Umgekehrt stellt die Annahme, daß alle jemals ausgetauschten Nachrichten gespeichert werden, ebenfalls keine starke Einschränkung dar. Für praktische Zwecke kann die Menge der gespeicherten Nachrichten begrenzt werden, indem von Zeit zu Zeit die in diesen Nachrichten gespeicherte relevante Information extrahiert und von den betroffenen Instanzen anerkannt wird. So kann ein Kontoguthaben zwar durch die Menge aller Gut- und Lastschriften beschrieben werden, die Bank und der Kontobesitzer können sich aber auch von Zeit zu Zeit auf den gerade gültigen Kontostand einigen und die vorherigen Gut- und Lastschriften vergessen, d. h. die entsprechenden Nachrichten löschen.

Um nicht alles als graue Theorie erscheinen zu lassen, betrachte man folgendes Beispiel, das Pfit_83 entlehnt ist und die Erfragung kostenpflichtiger Dienste über Anonyme Nummernkonten erlaubt: das

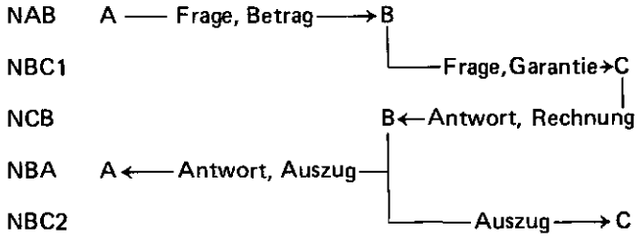
Protokoll „AnoNuKo"

Das Szenario des Protokolls besteht aus beliebig vielen Instanzen, unter denen drei für einen Protokollablauf ausgezeichnet werden: der Dienstkunde Andi, die Bank B und der Dienstanbieter Claus. Außerhalb des Modells haben Andi und Claus bei der Bank B (anonyme) *Nummernkonten* eingerichtet.

Ein Nummernkonto eines Kunden X der Bank B ist gekennzeichnet durch eine eindeutige Kontonummer k_x, ein Schlüsselpaar $(\ddot{o}_{xB}, p_{xB})$ des Kunden X und ein Schlüsselpaar $(\ddot{o}_B, p_B)$ der Bank B. Dem Kunden X sind k_x, $\ddot{o}_{xB}$, p_{xB}, $\ddot{o}_B$ bekannt, der Bank k_x, $\ddot{o}_{xB}$, $\ddot{o}_B$, p_B. Die Kontonummer dient beiden zur Identifikation, die Schlüsselpaare sowohl zur Geheimhaltung (und impliziten Adressierung, vgl. Pfit_85) als auch zur Authentikation. Da zur Eröffnung eines Kontos zumindest die Bank bekannt sein muß, also nicht anonym sein kann, wird das Paar $(\ddot{o}_B, p_B)$ als für alle Kunden X gleich betrachtet. Die Kunden hingegen werden als anonym angenommen, ihre Schlüssel können exklusiv für den Bankverkehr verwendet werden.

Der Dienstkunde Andi möchte nun auf eine Frage f vom Diensterbringer Claus eine Antwort a erhalten. Diese Antwort kostet r Einheiten, die von Andis Konto auf das Konto von Claus transferiert werden müssen. Der Dienstkunde Andi sendet seine Frage f zusammen mit einem Betrag b, den er maximal für die Beantwortung von f zu zahlen bereit ist, an die Bank B. Diese prüft, ob sie für diesen Betrag noch garantieren kann, und leitet gegebenenfalls die Frage f an den Diensterbringer Claus weiter. Dieser bestimmt die zur Beantwortung aufzubringenden Kosten r, prüft, ob der Betrag b reicht, und erzeugt gegebenenfalls die Antwort a. Die Antwort a und die Kosten r sendet er an die Bank B, die die Kosten r von Andis Konto auf das Konto des Erbringers Claus transferiert. Die Bank sendet schließlich an Andi die Antwort von Claus und an beide Kontoauszüge.

Die Grobstruktur des Protokolls ist



Am linken Rand sind Kennungen für die einzelnen Nachrichten angegeben, wobei künftig stets eine Nachricht mit Kennung NX durch nX bezeichnet wird. Mit diesen Kennungen lauten die Regeln des Protokolls informell:

Regel RA1: Andi sendet eine Frage an die Bank

Der Dienstkunde Andi erzeugt eine Frage f und ein noch nicht verwendetes Schlüsselpaar $(\ddot{o}_f, p_f)$. Dank $\ddot{o}_f$ kann er die Antwort erhalten, ohne daß sie die Bank b lesen kann. Er verschlüsselt die Frage f und den „Antwortschlüssel" $\ddot{o}_f$ mit dem öffentlichen Schlüssel $\ddot{o}_C$ des Diensterbringers Claus. Weiter legt er den maximalen Betrag b fest, den er zu zahlen bereit ist. Sodann schickt er alles unterschrieben und mit seiner Kontonummer k_A versehen an die Bank B.

$$nAB\ A \text{ —— } \ddot{o}_B(\ddot{o}_{AB}, p_{AB}(\ddot{o}_C(f, \ddot{o}_f), \ddot{o}_C, k_A, b, NAB)) \rightarrow B$$

Regel RB1: Die Bank sendet Frage und Garantie an Claus

Die Bank erhält eine Nachricht nAB mit Kennung NAB, sieht die Frage aber nur verschlüsselt als $F = \ddot{o}_C(f, \ddot{o}_f)$. Sie prüft, ob sie für den Kunden mit der Kontonummer k_A noch für b Einheiten garantieren kann, und leitet gegebenenfalls die Frage unterschrieben an den durch $\ddot{o}_C$ angegebenen Diensterbringer weiter.

$$nBC1\ B \text{ —— } \ddot{o}_C(\ddot{o}_B, p_B(F, b, NBC1)) \rightarrow C$$

Regel RC1: Claus sendet die Antwort an die Bank

Der Diensterbringer Claus erhält eine Nachricht nBC1 mit Kennung NBC1, also eine Frage der Bank. Er entschlüsselt die Frage und prüft, ob der Betrag b ausreicht, um die Frage zu beantworten. Falls ja, so sendet er die Antwort a verschlüsselt mit $\ddot{o}_f$, damit die Bank sie nicht lesen kann, sowie die Kosten r und seine Kontonummer an die Bank B. Durch ew(nBC1) kann die Bank später die Antwort der entsprechenden Frage zuordnen. C unterschreibt alles sowohl mit p_C, damit seine Identität als intendierter Beantworter der von A gestellten Frage nachgewiesen wird, als auch mit seinem Bankschlüssel p_{CB}, damit die Bank seinen Rechnungsbetrag von A's Konto auf seines umbuchen kann.

$$nCB\ C \text{ —— } \ddot{o}_B(\ddot{o}_C, p_C(\ddot{o}_{CB}, p_{CB}(ew(nBC1), \ddot{o}_f(a), k_C, r, NCB))) \rightarrow B$$

Regel RB2/3: Die Bank sendet Andi die Antwort und Andi und Claus je einen Auszug

Die Bank erhält eine Nachricht nCB mit Kennung NCB, entnimmt ihr eine Antwort, verschlüsselt als $An = \ddot{o}_f(a)$, und testet, ob es eine dazu passende Frage gab, prüft, ob der Rechnungsbetrag klein genug ist, und bucht gegebenenfalls den Rechnungsbetrag r von Andis Konto auf das Konto von Claus. Sodann sendet sie die Antwort (oder das, was die

Bank dafür hält) zusammen mit einem Kontoauszug über den Kontostand kSA an Andi und schickt als Bestätigung für Claus auch diesem einen Auszug über den Kontostand kSC.

nBA B——$\ddot{o}_{AB}(\ddot{o}_B,p_B(ew(nAB),An,r,k_A,kSA,NBA))$↛A

nBC2 B——$\ddot{o}_{CB}(\ddot{o}_B,p_B(ew(nCB),r,k_C,kSC,NBC2))$ → C

Der Beweis dieses Protokolls benötigt insgesamt drei Zähler: die bereits beschriebenen Zähler K und W und einen „Garantiezähler" G, der wie K gebildet wird. Er zählt den Betrag, für den die Bank noch garantieren kann. Dies ist notwendig, da die Bank bei überlappenden Anfragen nicht stets für den vollen Kontostand garantieren kann.

Die Werte werden wie folgt an die Nachrichten gebunden: (k_x = Kontonummer von x, x = A,C)

	Zählerwerte					
Nachrichten-kennungen	K für k_A	G für k_A	W für k_A	K für k_C	G für k_C	W für k_C
NAB						
NBC1		−b				
NCB						−r
NBA	−r	b−r	r			
NBC2				r	r	

Werte der Nachrichten für die einzelnen Zähler

Das Protokoll AnoNuKo garantiert unter der Annahme eines anonymen KomSy's den Kunden der Bank B, daß sie bei B anonym ein Konto unterhalten können. Die Bank ist aber in der Lage, zu einem bestimmten Konto anzugeben, von welchen anderen Konten Geld auf dieses Konto und an welche anderen Konten Geld von diesem Konto überwiesen wurde. Damit kann die Bank bezüglich des Geldverkehrs *anonyme Persönlichkeitsbilder* ihrer Kunden erstellen.

Gegenüber dem KomSy wie auch untereinander sind die Kunden der Bank anonym.

Das Protokoll AnoNuKo ist in Waid_85 ausführlich behandelt und in einer dort eingeführten Pseudosprache formal beschrieben.

2.2 Betrug und Betrugssicherheit

Üblicherweise wird der Betrugsfall in Wertetransferprotokollen wie dem in 2.1 angegebenen nur informell behandelt. Das hier übliche Muster ist:

- *Prüfe*, ob der bis jetzt erreichte Stand des Protokollablaufs zufriedenstellend ist. Falls nicht, dann *protestiere*.
- Ansonsten führe die nächste anzuwendende Regel aus.

Unter dem Begriff *Betrug* versteht man hier also einen Oberbegriff für alle Ursachen, die einer Instanz einen Schaden zufügen können. Dieser Schaden wird aufgrund eines Tests erkannt und eines Protests behoben.

Die Verwendung von „Betrug" unterscheidet sich somit erheblich von der Verwendung im juristischen Bereich, wo hierfür die Schädigungsabsicht notwendig ist. Der „informatische" Betrug kann aber auch ein Fehler, z.B. des KomSy's, oder ein Versehen sein. Ein Fehler des KomSy's ist im Modell allerdings per Annahme ausgeschlossen. Als Betrug kommt im Modell jedes Fehlverhalten in Frage, das eine Instanz I nachweislich schädigt und nicht durch die Instanz I selbst verursacht wurde. Als Betrüger kommen dann alle Instanzen außer I in Frage. Die betrügenden Instanzen dürfen alles tun, was sie aufgrund ihres *Wissens* tun können, müssen sich also insbesondere nicht an die Regeln des zu beweisenden Protokolls halten. Das Wissen einer Instanz und damit die Einschränkung der Fähigkeiten eines Angreifers oder Betrügers wird explizit in Form von Teilalgebren (s. 2.1) angegeben. Betrügt eine Instanz, so verfügt sie zumindest zu Beginn nicht über mehr Wissen, als wenn sie nicht betrügen würde. Betrügen mehrere Instanzen, so gilt dasselbe für die Vereinigung deren Wissens.

Das *Erkennen* eines Betruges in diesem Sinne erfolgt durch einen Schadenstest. Dies scheint die einzige durchführbare Möglichkeit zu sein. Insbesondere ist die Umkehrung hiervon, d. h. die vollständige Auflistung aller Betrugsmöglichkeiten, undurchführbar. Wer sollte eine solche Liste als vollständig nachweisen? Dieser Schadenstest wird in der Literatur üblicherweise explizit in das Protokoll aufgenommen und unterscheidet sich für jede Regel des Protokolls. Das verwendete Modell faßt alle diese Tests zusammen in einen Test, beschrieben durch die sogenannten *Konsensbedingungen*. Diese Bedingungen müssen im korrekten Fall für jede Regel des Protokolls invariant sein und müssen im Schadensfall diesen Schaden anzeigen. Die Erstellung dieser Konsensbedingungen entspricht einer Sicherheitsspezifikation des Protokolls: was muß alles gelten, damit kein Betrug vorliegt?

Diese Spezifikation stellt den Spezifizierer vor dasselbe Problem wie jede andere Spezifikation: er muß feststellen, ob sie vollständig und korrekt ist. Beides ist anwendungs- und zum Teil sogar protokollabhängig. Eine einfache Invariante, wie z. B. die Bedingung „Summe über alle Zählerstände konstant", scheidet dabei meist aus. Diese Invariante berücksichtigt zum Beispiel nicht, daß zum einen die geforderte Konstanz selbst im korrekten Fall nicht erreicht wird. Es wird häufig der Fall auftreten, daß eine Instanz eine Nachricht eines bestimmten Wertes gesendet und damit einen Wert verloren, die diesen Verlust ausgleichende Nachricht aber noch nicht erhalten hat. Zum anderen würde diese Bedingung alleine es erlauben, einer Instanz gegen deren Willen z. B. Geldwerte in Sachwerte, z. B. Tausende von Zeitungen, umzutauschen, ohne daß die Invariante verletzt wäre.

Für das oben angeführte Beispiel des Protokolls AnoNuKo lauten die Konsensbedingungen informell:

Dienstkunde Andi

KA_1 Erhält Andi eine Antwort a, so war diese von ihm bestellt und kostet nicht mehr, als er zu zahlen bereit war.

KA_2 Erhält Andi eine Antwort a, so ist diese neu.

KA_3 Erhält Andi eine Antwort a, für die der Antworter eine Rechnung über r Einheiten stellt, so war a auch r Einheiten wert.

KA_4 Erhält Andi von B einen Wert übermittelt, der den Kontostand von Andi darstellt, so stimmt dieser Wert.

KA_5 Ändert sich der Konto- oder Wertestand von Andi aufgrund einer Nachricht n um −r Einheiten, so gibt es eine Nachricht n', aufgrund der sich der Werte- oder Kontostand von Andi entsprechend um +r Einheiten ändert (z. B. n = Anfrage, n' = Antwort oder n' = Rücküberweisung).

Bank B

KB_1 Alle Konten führen nicht-negative Werte. (Wie sollten Schulden von einem anonymen Kunden eingetrieben werden?)

KB_2 Die Kontoänderungen der Bankkunden lassen sich so paaren, daß sie sich gegenseitig aufheben. (Es gibt in diesem Protokoll keine Aus- oder Einzahlungen!)

Dienstanbieter Claus

KC_1 Erhält Claus von B einen Wert übermittelt, der den Kontostand von Claus darstellt, so stimmt dieser Wert.

KC_2 Ändert sich der Konto- oder Wertestand von Claus aufgrund einer Nachricht n um $-r$ Einheiten, so gibt es eine Nachricht n', aufgrund der sich der Werte- oder Kontostand von Claus entsprechend um $+r$ Einheiten ändert (z. B. n = Antwort, n' = Überweisung).

Die Bedingungen KA_5 und KC_2 fordern, daß sich Wertestand und Kontostand ausgleichen. Jeder Lastschrift muß genau eine Erhöhung des Wertestandes (oder eine Rücküberweisung) entsprechen, jeder Gutschrift genau eine Senkung (oder eine Lastschrift) und umgekehrt.

Ist eine dieser Konsensbedingungen während eines Protokollablaufs verletzt, so *protestiert* die betroffene Instanz. In der Realität ginge dieser Protest an den vermuteten Verursacher des angezeigten Schadens, und wenn das nichts nützt, letztendlich an ein Gericht. Diesem Gericht entspricht in der Welt der Wertetransferprotokolle die *Schiedsstelle*. Diese hat die Aufgabe, nach einem Protest die Berechtigung dieses Protests zu prüfen, die Ursache des angezeigten Schadens zu *lokalisieren*, d. h. den *Betrüger* festzustellen, und den Schaden zu *beheben*. Existiert für ein gegebenes Protokoll eine korrekte und realisierbare Schiedsstelle, so wird dieses Protokoll als *betrugssicher* bezeichnet. Eine Instanz ist dann vor jedem Betrug geschützt, der durch Regelverletzungen beliebiger anderer Instanzen verursacht wurde. Betrachtet man die Schiedsstelle als Instanz, so muß diese allerdings als nicht betrügend angenommen werden.

2.3 Beweis der Betrugssicherheit

Um die Betrugssicherheit zu beweisen, gibt es prinzipiell zwei Möglichkeiten. Zum einen kann man versuchen, die Schiedsstelle zu konstruieren, indem man ihr Protokoll angibt. Zum anderen kann man einfach die Existenz einer solchen Schiedsstelle nachweisen. Da im ersten Fall zusätzlich die Korrektheit der Schiedsstelle nachgewiesen werden müßte, geht das hier vorgestellte Modell den zweiten Weg. Die Existenz einer Schiedsstelle wird bewiesen, indem zu jeder Konsensverletzung gezeigt wird, daß sie auf eine Abweichung einer Instanz, des *Betrügers*, von den Regeln des Protokolls konstruktiv zurückzuführen ist. Eine solche Abweichung wird *Regelverletzung* genannt, deren Nachweis durch die Schiedsstelle ein *Regelverletzungsbeweis*, kurz RVBeweis. Gelingt der Schiedsstelle ein RVBeweis gegen die Instanz I, so ist ein solcher Beweis offensichtlich nur dann sinnvoll, wenn I *belastbar* ist, d.h. wenn der Schaden, den I möglicherweise durch die Regelverletzung verursacht hat, behoben werden kann. Dies ist dann der Fall, wenn I *identifizierbar* ist oder wenn I durch eine andere belastbare Instanz, z. B. die Bank, etwas „weggenommen" werden kann. Ist I belastbar, so wird der RVBeweis als *effektiver RVBeweis* bezeichnet. In einem nicht anonymen KomSy ist jede Instanz belastbar, also jeder RVBeweis effektiv. Im wesentlichen kann der Existenzbeweis der Schiedsstelle als deren Konstruktion betrachtet werden.

Die Behandlung eines Protestes durch eine Schiedsstelle erfolgt nach folgendem Schema.

Die Schiedsstelle erhält von einer Instanz I_i die Folge der bis dahin gesendeten Nachrichten mit dem Hilferuf: *„Ich bin betrogen!"*

Die Verletzung der Konsensbedingung, die diesen Hilferuf verursachte, kann im wesentlichen zwei Ursachen haben:

- Eine Nachricht mit einer bestimmten Kennung, die aufgrund des bisherigen Protokollablaufs erwartet wurde, blieb aus.
- Eine empfangene Nachricht n_t war „falsch", d. h. durch eine Regelverletzung zustande gekommen.

Im ersten Fall muß nachgewiesen werden, daß eine Regel nicht angewandt wurde, obwohl sie hätte angewandt werden müssen (Zeitschranken!), im zweiten Fall, daß eine Regel falsch angewandt wurde.

Beide Nachweise basieren auf zwei Voraussetzungen:

- Das Senden oder Nichtsenden einer bestimmten Nachricht kann durch die Schiedsstelle nachgeprüft werden. Im Modell wird dies durch die Annahme eines fehlerfreien, verteilenden KomSy's erreicht: erhält jeder alle Nachrichten, so kann jeder Senden oder Nichtsenden nachprüfen (round table environment, vgl. DoWi_83).
- Der Sender einer korrekten Nachricht ist authentizierbar, d. h. alle Nachrichten sind unterschrieben.

Sind alle Nachrichten unterschrieben, so kann zu einer Nachricht n_t die Instanz I_u, die n_t ursprünglich erzeugte, bestimmt, wenn auch nicht notwendig identifiziert werden (*Ursprungsbeweis*, kurz UBeweis).

Durch Verwendung von *Nachrichtenkennungen* kann zu n_t auch die genaue Regel von I_u angegeben werden, die zur korrekten Erzeugung führte. Dient n_t als *Indiz* eines effektiven RVBeweises, so hat die Schiedsstelle verschiedene Möglichkeiten.

Kann bereits aus dem bisherigen Wissen der Schiedsstelle und der Struktur der Nachricht n_t gefolgert werden, daß I_u eine Regel verletzt haben mußte, so liegt ein *direkter RVBeweis* vor.

Ist dies nicht der Fall, so kann die Schiedsstelle, da die Instanz I_u für einen effektiven RVBeweis belastbar sein muß, I_u zu einer *Rechtfertigung* der Sendung von n_t auffordern (*indirekter RVBeweis*).

Verweigert I_u diese, so wird I_u als betrügend betrachtet. Zur Rechtfertigung gibt es für I_u zwei Möglichkeiten.

I_u gibt zu, n_t gesendet zu haben, und rechtfertigt dies.

Die Rechtfertigung besteht darin, daß I_u die Struktur von Nachrichten, die sie zuvor erhalten hat, und die es ihr erlaubten, n_t zu senden, offenlegt, z. B. verschlüsselte Nachrichten entschlüsselt. Die Schiedsstelle kann die offengelegten Nachrichten wie vorher n_t auf Regelverletzungen hin untersuchen. Führt dies zu keinem Erfolg und liegen keine weiteren Indizien vor, so ist das Protokoll *unsicher*.

I_u behauptet, eine andere Instanz habe n_t gesendet.

Hierzu muß I_u zeigen, daß es n_t zuvor schon als Splitter einer anderen Nachricht n_s gesendet hat und folglich auch der Empfänger von n_s als Sender von n_t in Frage kommt.

Die einzige Möglichkeit hier ist, wie n_t jetzt die Nachricht n_s auf Regelverletzungen hin zu untersuchen. Führt dies zu keinem Erfolg und liegen keine weiteren Indizien vor, so ist das Protokoll *unsicher*.

Die als existent nachzuweisende Schiedsstelle geht also nach dem Rechtfertigungsprinzip das Protokoll rekursiv bis zur Lokalisierung des Betrügers zurück. Die Existenz einer Schiedsstelle ist dann nachgewiesen, wenn zu jeder Verletzung einer Konsensbedingung die Existenz eines effektiven RVBeweises gezeigt werden kann. In Waid_85 ist dies für das Protokoll AnoNuKo getan. Die wesentlichen Methoden sind in Kap. 3 skizziert.

3 Beweisskizze des Protokolls AnoNuKo

Der Beweis des Protokolls AnoNuKo erfolgt, indem gezeigt wird, daß aus jeder negierten Konsensbedingung aus 2.2 die Existenz eines effektiven Regelverletzungsbeweises folgt. Für das Protokoll AnoNuKo ist dabei jeder RVBeweis effektiv, da er entweder gegen die Bank B geführt wird, die nicht anonym ist, oder gegen einen Kunden der Bank B, der durch B folglich belastbar ist. Hierbei wird unterstellt, daß Dienstanbieter erst nach einer gewissen Zeit über gutgeschriebenes Geld verfügen können. Dies ist im Modell, da es keine Zeit modelliert, nicht ausdrückbar (siehe auch denkbare Erweiterungen des Modells in Kap. 4). Dürften Dienstanbieter sofort über gutgeschriebenes Geld verfügen, z. B. ihr Konto sofort vollständig leeren, wären sie natürlich danach nicht mehr durch die Bank belastbar.

Da ohne ein formaleres Gerüst diese Beweise nicht durchzuführen sind, sollen hier lediglich die hierzu notwendigen Ideen dargestellt werden. In Waid_85 ist ein ausführlicher Beweis enthalten.

Die Konsensbedingungen KA_i, $i = 1, \ldots, 4$, und KC_1 werden durch einfaches Rückverfolgen behandelt, d. h. man zeigt, daß im Falle der Verletzung eine andere Instanz eine Regel verletzt hat.

Als Beispiel betrachte man KA_3. Ist KA_3 verletzt, so muß A eine Nachricht n_t besitzen mit der Kennung NBA (d. h. eine Antwortnachricht), deren Rechnungswert r aber für die darin enthaltene Antwort a falsch ist. Die Nachricht n_t ist von B unterschrieben, also ein UBeweis gegen B. Wird B zur Rechtfertigung aufgefordert, so muß B Nachrichten nCB und nAB mit den Kennungen NCB und NAB vorlegen. Ansonsten hat B betrogen. Der Nachricht nCB sind die von C erzeugte Antwort $An' = \ddot{o}_f(a')$ und der Rechnungsbetrag r' zu entnehmen. Gilt $r' \neq r$ oder $a' \neq a$, so hat B betrogen, ansonsten hat C betrogen, da nach KA_3 r nicht angemessen war. Wann eine Rechnung angemessen ist oder nicht, ist im Modell durch eine Funktion, die jeder Antwort ihren Wert zuordnet, definiert.

Die Bedingungen KA_5, KB_1, KB_2 und KC_2, d. h. die Bedingungen, die eine Art Konstanz fordern, werden durch „Paarung" bewiesen. Damit ist gemeint, daß die sich gegenseitig aufhebenden Nachrichten in einer eineindeutigen Relation zusammengefaßt, gepaart werden. Dabei dürfen relativ neue Nachrichten ungepaart bleiben, da deren Partnernachricht noch nicht eingetroffen sein muß. Für KA_5 werden die Nachrichten mit der Kennung NBA mit sich selbst gepaart, für KC_2 die Nachrichten mit den Kennungen NCB und NBC2. Für KB_1 ist die Paarrelation mehrstellig. Sie paart alle vorkommenden Nachrichten.

4 Ausblick

Das vorgestellte Modell entstand im Rahmen einer Diplomarbeit (Waid_85) und stellt somit erst den Anfang, nicht das Ende eines sicher sinnvollen und lohnenden Versuchs dar, die Eigenschaft der Betrugssicherheit zu formalisieren und deren Beweisbarkeit zu untersuchen.

Bezüglich des Modells sind natürlich noch zahlreiche Fragen offen, insbesondere die nach seiner Validierung und Akzeptanz durch Banken, Bürger und Gerichte, sowie die der Überprüfung von Beweisen der bei realistischen Protokollen zu erwartenden Länge. Ein realistisches Protokoll sollte zumindest in einer zweistufigen Bankenhierarchie (Banken mit privaten Kunden und Zentralbanken zum Clearing zwischen Banken) arbeiten. Um diese Probleme zu lösen, sollte die gewählte Modellierung nochmals unter diesen Gesichtspunkten überdacht werden.

Erweiterungen und Verbesserungen des Modells sind in mehreren Richtungen denkbar:

Ohne das Modell ändern zu müssen, könnte man versuchen, statt der im Beispiel gewählten syntaktischen Definition der Konsensbedingungen diese syntaxunabhängig nur in der die Welt modellierenden Algebra zu definieren. Hierdurch erhielte man protokollunabhängige Konsensbedingungen.

Eine weitere wesentliche Verbesserung wäre erreicht, wenn die Anforderungen des Modells an das verwendete KomSy modifiziert werden könnten. Denkbar wären hier die Behandlung von groben Übertragungsfehlern (Ausbleiben einer erwarteten Nachricht) und der Verzicht auf die Forderung der Verteilung. Da in einem fehlerfreien Verteilnetz der Schiedsstelle alle gesendeten Nachrichten in ihrer zeitlichen Abfolge vorliegen, verfügt sie über eine globale Sicht des Geschehens im KomSy. Betreibt das KomSy keine Verteilung, so verliert die Schiedsstelle diese globale Sicht. Hierdurch wird die Schiedsstelle sicher realistischer, der Beweis der Betrugssicherheit dafür etwas schwieriger (DoWi_83).

Ebenfalls wünschenswert wäre eine stärkere Berücksichtigung zeitlicher Abfolgen innerhalb des Modells, indem für die Regeln eines Protokolls explizite Zeitschranken angegeben werden, innerhalb derer sie zur Anwendung kommen müssen. Soll das Modell auch Übertragungsfehler des KomSy's oder Lebendigkeitseigenschaften eines Protokolls berücksichtigen, so sind Erweiterungen in dieser Richtung unumgänglich.

Explizite Zeitschranken können auch für den (genaueren) Beweis von Betrugssicherheit notwendig sein, z. B. muß Geld gegebenenfalls eine gewisse Zeit t auf einem anonymen Nummernkonto bleiben, damit der Inhaber im Falle eines dann ebenfalls nur innerhalb einer gewissen Zeit $t' < t$ immer erfolgreichen Protestes belastbar ist.

Neben diesen modellinternen Erweiterungen stellt sich die Frage, wie sich die Sicherheitseigenschaften der verwendeten Kryptosysteme auf die Sicherheit von Wertetransferprotokollen auswirken. Modelle der verwendeten Art (vgl.

auch Merr_83) können die Struktur dieser Kryptosysteme nur sehr grob wiedergeben. Im Beispiel wird dem Kryptosystem als einzige Eigenschaft die Beziehung $\ddot{o}_x p_x = p_x \ddot{o}_x = id$ unterstellt. Diese Beziehung wird durch das RSA-System (RSA_78) erfüllt. Das Modell berücksichtigt aber nicht die weiteren algebraischen Eigenschaften des Systems. So bildet RSA einen Homomorphismus bezüglich der Multiplikation, was einerseits zu Angriffsmöglichkeiten führt, die im Modell nicht vorkommen (Denn_84), andererseits auch zu neuen Anwendungsmöglichkeiten (Chau_85). Allgemein ausgedrückt wird dem verwendeten Kryptosystem unterstellt, daß es absolut sicher sei, was in der Realität praktisch nicht erreicht wird.

Die Garantie der Betrugssicherheit eines Protokolls aufgrund des hier vorgestellten Modells ist somit immer nur relativ zur Sicherheit der zur Implementierung des Protokolls verwendeten Kryptosysteme zu verstehen.

Wir danken den Mitgliedern unserer Datenschutzarbeitsgruppe, insbesondere Birgit Pfitzmann, für ihre Unterstützung und ebenso zahlreiche wie konstruktive Kritik.

5 Literatur

Akl_83 Selim G. Akl: Digital Signatures: A Tutorial Survey; Computer, IEEE, Vol. 16, Nr. 2, Februar 1983, Seite 15 bis 24

Chau_83 David Chaum: Blind Signatures for untraceable payments; Advances in Cryptology, Proceedings of Crypto 82, A Workshop on the Theory and Application of Cryptographic Techniques, 23.–25. August 1982, University of California, Santa Barbara, herausgegeben von David Chaum, Ronald L. Rivest, and Alan T. Sherman, Plenum Press, New York, 1983, Seite 199 bis 203

Chau_85 David Chaum: Security without Identification: Transaction Systems to make Big Brother Obsolete; Communications of the ACM Vol. 28, Nu. 10, October 1985, Seite 1030 bis 1044

Denn_82 Dorothy E. Denning: Cryptography and Data Security; Addison-Wesley Publishing Company, Reading, Mass.; 1982

Denn_84 Dorothy E. Denning: Digital Signatures with RSA and ther Public-Key Cryptosystems; Communications of the ACM, Vol. 27, Nr. 4, April 1984, Seite 388 bis 392

DiHe_79 Whitfield Diffie, Martin E. Hellman: Privacy and Authentication: An Introduction to Cryptography; Proceedings of the IEEE, Vol. 67, Nr. 3, März 1979, Seite 397 bis 427

DoWi_83 Danny Dolev, Avi Wigderson: On the security of multiparty protocols in distributed systems; Advances in Cryptology, Proceedings of Crypto 82, A Workshop on the Theory and Application of Cryptographic Techniques, 23.–25. August 1982, University of California, Santa Barbara, herausgegeben von David Chaum, Ronald L. Rivest, and Alan T. Sherman, Plenum Press, New York, 1983, Seite 167 bis 175

EvGY_84 S. Even, O. Goldreich, Y. Yacobi: Electronic Wallet; 1984 International Zurich Seminar on Digital Communications, Applications of Source Coding, Channel Coding and Secrecy Coding, 6.–8. März 1984, Zürich, Schweiz, Swiss Federal Institute of Technology, Proceedings IEEE Catalog Nr. 84CH1998–4, Seite 199 bis 201

Herm_67 Hans Hermes: Einführung in die Verbandstheorie; Springer-Verlag Heidelberg, New York 1967 Grundlehren der mathematischen Wissenschaften, Band 73

Hors_85 Patrick Horster: Kryptologie; Reihe Informatik/47, Herausgegeben von K. H. Böhling, U. Kulisch, H. Maurer, Bibliographisches Institut, Mannheim, 1985

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

Pfit_83 Andreas Pfitzmann: Ein dienstintegriertes digitales Vermittlungs-/Verteilnetz zur Erhöhung des Datenschutzes; Fakultät für Informatik, Universität Karlsruhe, Interner Bericht 18/83, Dezember 1983

Pfit_85 Andreas Pfitzmann: Technischer Datenschutz in diensteintegrierenden Digitalnetzen — Problemanalyse, Lösungsansätze und eine angepaßte Systemstruktur; 1. GI-Fachtagung „Datenschutz und Datensicherung", Informatik-Fachberichte 113, Springer-Verlag, Heidelberg 1985, Seite 96 bis 112

Pfi1_85 Andreas Pfitzmann: How to implement ISDNs without user observability — Some remarks; Fakultät für Informatik, Universität Karlsruhe, Interner Bericht 14/85, 1985

Riha_83 Karl Rihaczek: OSIS — Open Shops for Information Services; DuD Datenschutz und Datensicherung, Informationsrecht, Kommunikationssysteme, Friedr. Vieweg & Sohn, Braunschweig, Heft 2, April 1983, Seite 116 bis 125

Riha_85 Karl Rihaczek: Der Stand von OSIS; DuD, Datenschutz und Datensicherung, Informationsrecht, Kommunikationssysteme, Friedr. Vieweg & Sohn, Braunschweig, Heft 4, August 1985, Seite 213 bis 217

RSA_78 R. L. Rivest, A. Shamir, L. Adleman: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems; Communications of the ACM Vol. 21, Nr. 2, Februar 1978, Seite 120 bis 126, nachgedruckt in Vol. 26, Nr. 1, Januar 1983, Seite 96 bis 99

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