Diplomarbeit:

Spezifikation von Geheimhaltungseigenschaften für reaktive kryptologische Systeme

Universität Hildesheim
Institut für Informatik
Samelsonplatz 1
D–31141 Hildesheim

Vorgelegt von: Matthias Schunter
Teichstr. 1
D-26122 Oldenburg

Zeitraum: 09.08.1993 – 26.01.1994

Gutachter: Prof. Dr. Joachim Biskup
Zweitgutachter: Prof. Dr. Eike Best
Betreuung: Birgit Pfitzmann

Zusammenfassung

In dieser Diplomarbeit wird der Begriff der Geheimhaltung für kryptologische Systeme formalisiert und an Beispielen erläutert.

Nach einer Einleitung mit bestehenden Ansätzen (Kap. 1) wird ein stochastisches Systemmodell entwickelt (Kap. 2) und auf kryptologische Systeme angewendet (Kap. 3). Anschließend wird eine modular aufgebaute Geheimhaltungsdefinition für kryptologische Systeme vorgestellt (Kap. 4), einige Beispiele konstruierter Definitionen ausführlich behandelt (Kap. 5) und der in dieser Arbeit entwickelte Geheimhaltungsbegriff auf Konzelationssysteme angewendet (Kap. 6).

Abschließend folgt ein Ausblick mit noch offenen Fragen (Kap. 7).

Die Arbeit wird durch ein Glossar mit vorausgesetzten Begriffen der Kryptografie, ein Literaturverzeichnis und einen Index ergänzt.

Erklärung

Hiermit versichere ich, daß ich diese Diplomarbeit selbständig erstellt und keine anderen als die angegeben Quellen verwendet habe.

Sibbesse, den 23. Januar 1994

Kurze Inhaltsübersicht

Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 Ziel dieser Arbeit

Ziel dieser Diplomarbeit ist es, eine einheitliche und formale Geheimhaltungsdefinition für beliebige reaktive*[1] kryptologische Systeme zu entwickeln.

Diese formalen Geheimhaltungsdefinitionen dienen dazu, die Geheimhaltung bestehender und zukünftiger Systeme zu klassifizieren und hierdurch eine Einordnung der Systeme zu ermöglichen.

Unter „einheitlich“ wird verstanden, daß nicht für jede Systemart eine Definition oder ein Spezialfall angegeben werden soll, sondern daß eine Definition auf beliebige Systeme angewendet werden kann.

1.2 Aufbau der Arbeit

Nach einer Einleitung mit bestehenden Ansätzen (Kap. 1) wird ein stochastisches Systemmodell entwickelt (Kap. 2) und auf kryptologische Systeme angewendet (Kap. 3). Anschließend wird eine modular aufgebaute Geheimhaltungsdefinition für kryptologische Systeme vorgestellt (Kap. 4), einige Beispiele konstruierter Definitionen ausführlich behandelt (Kap. 5) und der in dieser Arbeit entwickelte Geheimhaltungsbegriff auf Konzelationssysteme angewendet (Kap. 6). Zur Vervollständigung dieses Beispiels wird auch der Service von Konzelationssystemen ansatzweise definiert.

[1] Im Glossar erklärte Begriffe werden durch einen Stern gekennzeichnet.

Abschließend folgt ein Ausblick mit noch offenen Fragen (Kap. 7).

Die Arbeit wird durch ein Glossar mit vorausgesetzten Begriffen der Kryptografie, ein Literaturverzeichnis und einen Index ergänzt.

Häufig verwendete Symbole im Text und in den Abbildungen werden in der Legende auf Seite 127 erläutert, der Definitionsort aller anderen Symbole ist im Symbolverzeichnis am Anfang des Index nachzuschlagen.

1.3 Anmerkungen

Trotz intensiver Literaturrecherche in der Kryptologie und ihren verwandten Gebieten[2] habe ich kaum verwertbare Ergebnisse gefunden. Keiner der bestehenden Ansätze behandelt Geheimhaltung in der hier geforderten Allgemeinheit. Bestehende Definitionen sind nur für sehr spezielle Systemtypen angegeben, behandeln nur unrealistische Geheimhaltungsstufen oder geben Logiken an, ohne Eigenschaften der zugrundeliegenden kryptologischen Verfahren zu berücksichtigen [BAN90a, Syv92, BC91a].

Einzig Ansätze aus der Informationsflußsicherheit und der Kryptologie waren für meine Arbeit bedingt verwendbar. Darum habe ich auf die Darstellung der anderen — gänzlich unverwertbaren — Ansätze verzichtet.

Die eigentlichen Geheimhaltungsdefinitionen wurden in dieser Arbeit vollständig neu entwickelt[3] und lehnen sich nur in einzelnen Spezialfällen an bestehende Definitionen an.

[2] Protokollogiken, Datenbank– und Informationsflußsicherheit, Rechnersicherheit, u.a.

[3] Mit sehr hilfreicher Betreuung durch Birgit Pfitzmann.

1.4 Bestehende Ansätze

1.4.1 Kryptologische Definitionen

Bisher wurde in der Kryptografie noch keine einheitlich formalisierte Geheimhaltungsdefinition vorgeschlagen. Selbst Geheimhaltung und Korrektheit einzelner Systemarten wurden nur zum Teil formal spezifiziert. Keine der bereits vorhandenen Definitionen ist auf beliebige Systeme übertragbar.

Analog wurden Systemeigenschaften wie Anonymität, Geheimhaltung mangels allgemeingültiger Definition für jedes System neu definiert. Die Geheimhaltung vorgeschlagener Systeme wurde oft in Bezug auf informelle Geheimhaltungsdefinitionen „bewiesen“ [ZS93].

In den folgenden Abschnitten werde ich existierende Ansätze für Geheimhaltung und Korrektheit nach Systemarten geordnet skizzieren.

1.4.1.1 Konzelationssysteme

Als Beispiel für bestehende Definitionen wird hier die Definition für komplexitätstheoretische Geheimhaltung bei passivem Angreifer angegeben. Dies ist der einzige komplexitätstheoretische Fall, für den eine formale und allgemein akzeptierte Geheimhaltungsdefinition existiert.

Für die Geheimhaltung von Konzelationssystemen ohne aktiven Angriff wurden bereits Definitionen von Goldwasser und Micali [GM84b] („semantic security“ und „indistinguishable encryptions“) und von Yao [Yao82] vorgeschlagen.

Micali, Rackoff und Sloan haben beweisen [MRS88], daß alle drei Definitionen äquivalent sind.

Goldreich zeigt die Äquivalenz der beiden Definitionen unter schwächeren Voraussetzungen [Gol93].

Da die Äquivalenz dieser drei Definitionen ein Indiz für eine gelungene Formalisierung ist, kann im Bereich der Konzelationssysteme davon ausgegangen werden, daß eine ausreichend exakte Sicherheitsdefinition für passive Angreifer existiert. Diese ist in dieser Arbeit auf das zu erstellende Systemmodell zu übertragen, oder als Spezialfall einer allgemeineren Definition zu beweisen.

Da eine allgemeinere Sicherheitsdefinition diese Definitionen beinhalten sollte, wiederhole ich sie kurz:

Def. 1.1 (Semantic Security (Aus [Gol93]))
An encryption scheme (G, E, D) [Schlüsselgenerator, Verschlüsselung, Entschlüsselung] is semantically secure, if for every probabilistic polynomial-time algorithm A, there exists a probabilistic polynomial-time algorithm A', such that for every polynomial-time random variable $\{X_n\}_{n \in I\!N}$, every polynomial-time computable function $h : \{0,1\}^* \to \{0,1\}^*$, every function $f : \{0,1\}^* \to \{0,1\}^*$, every constant $c > 0$, and all sufficiently large n,

$$P(A(E_{G_1(1^n)}(X_n), h(X^n), 1^n) = f(X_n)) < P(A'(|X_n|, h(X_n), 1^n) = f(X_n)) + \frac{1}{n^c}$$

The probability space in the above terms is taken over the probability space underlying X_n and the internal coin tosses of G, E, A, and A'.

Die Idee dieser Definition ist, daß der Algorithmus A bei gleicher partieller Information $h()$ trotz Schlüsseltext* eine Eigenschaft des Klartextes* nicht besser rät als Algorithmus A', dem statt dessen nur die Länge der Nachricht zur Verfügung stand.

Def. 1.2 (Indistinguishable Encryption (Aus [Gol93]))
An encryption scheme (G, E, D) has indistinguishable encryption if, for every polynomial-time random variable $\{T_n = X_n Y_n Z_n\}_{n \in I\!N}$ with $|X_n| = |Y_n|$, every probabilistic polynomial-time algorithm A, every constant $c > 0$, and all sufficiently large n,

$$|P(A(Z_n, E_{G_1(1^n)}(X_n)) = 1) - P(A(Z_n, E_{G_1(1^n)}(Y_n)) = 1)| < \frac{1}{n^c}.$$

The probability in the above terms is taken over the probability space underlying T_n and the internal coin tosses of G, E, and A.

Idee ist, daß kein Algorithmus existiert, der effektiv bestimmen kann, welcher der zwei Schlüsseltexten die Verschlüsselung eines gegebenen Klartextes ist.

Die Definitionen beziehen sich auf Private-Key Kryptosysteme, lassen sich jedoch einfach auf Public–Key Systeme übertragen, indem den Algorithmen A zusätzlich der Verschlüsselungsalgorithmus $E_{G_1(1^n)}$ eingegeben wird.

Für aktive Angriffe existieren Ansätze von Naor und Yung [NY90], Rackoff und Simon [RS92] und Galil et. al. [GHY91], die jedoch jeweils nur einzelne aktive Angriffsarten* abdecken. Eine allgemeine Definition existiert nicht und die bestehenden sind zum Teil nicht realistisch:

[NY90] modelliert nur nichtadaptiven gewählten-Schlüsseltext-Klartext Angriff bei nichtinteraktiven Systemen,

[RS92] behandelt nur nichtinteraktive Systeme.

[GHY91] ist nicht realistisch und behandelt nur Geheimhaltung bei nichtadaptiv gewählter-Schlüsseltext-Klartext Angriff (vgl. Bsp. 3.2 auf S. 53),

[GS88] betrachtet das totale Brechen eines Konzelationssystems aus Sicht der Komplexitätstheorie.

1.4.1.2 Signatursysteme

Signatursysteme sind hier als Beispiel für eine Systemart mit formal definierten Serviceanforderungen angeführt. Die hier angegebenen Literaturstellen haben als Grundlage der Servicespezifikationen für Konzelationssysteme in Abschnitt 6 gedient.

Formale Serviceanforderungen für Signatursysteme wurden von Pfitzmann in [Pfi93b, Pfi93a] angegeben.

Eine allgemeine Sicherheitsdefinition für verschiedene Angriffsarten existiert nicht.

1.4.2 Sicherer Informationsfluß

Diese Gruppe bestehender Definitionen stammt aus dem militärischen Bereich. Die vom U.S. Department of Defense praktizierte Mehrschichtensicherheitsstrategie wurde erstmals 1973 von Bell und La Padula in [BL83] formalisiert. Diese Arbeiten dienten als Grundlage zur Entwicklung von Zugriffskontrollmechanismen wie „Mandatory Access Control[4] (MAC)“. Die Ergebnisse wurden auch im „Government of Defense Trusted Computer System Evaluation Criteria“ (TCSEC[5], [DoD85]) verwendet, welches sieben Sicherheitsstufen für sichere Systeme definiert (A1, B1, B2, B3, C1, C2, D).

[4] Im Gegensatz zu „Discretionary Access Control“ (DAC), bei dem jeder Benutzer bestimmen kann, wer auf seine Daten zugreifen kann, kontrolliert bei MAC das System die Zugriffsrechte.

[5] Teil der „Rainbow Series“ (wg. der Einbandfarben). Aufgrund seines orangenen Einbands auch als „Orange Book“ bezeichnet.

Die Informationsquellen– und Senken werden bei diesem Definitionstyp in zwei oder mehr Ebenen unterteilt. Ein Informationsfluß von einer höheren (geheimeren) in eine tiefere Ebene darf nicht stattfinden. Verringerung der Geheimhaltungsstufe von Informationen ist in der Regel nicht vorgesehen.

Die zugrundeliegenden Artikel kann man nach folgenden Kriterien unterteilen:

- Anzahl der Ebenen
- stochastisch/deterministisch

Zwei Sicherheitsdefinitionen wurden von Goguen und Mesequer in [GM82] und von Mc Lean in [Lea90] vorgestellt. Beide Definitionen teilen die Informationsquellen und -senken in die zwei Sicherheitsstufen High und Low.

Goguen und Mesequers „Probabilistic Noninterference“ (PNI) definiert ein System als sicher, wenn die Wahrscheinlichkeit eines Low–Ereignisses für je zwei High–Eingabeverhalten identisch ist, d.h. daß die Wahrscheinlichkeiten aller Low-Ereignisse unabhängig von High-Eingaben sind. Gray formalisiert diese Eigenschaft wie folgt:

$$P_{\mathcal{H},\mathcal{L}}(e) = P_{\mathcal{H}',\mathcal{L}}(e).$$

Dies bedeutet, daß die Wahrscheinlichkeit eines Low–Ereignisses bei vorgegebenem Low–Eingabeverhalten und zwei verschiedenem High–Eingabeverhalten gleich ist.

McLeans „Flow–Model“ (FM) definiert ein System als sicher, wenn gilt:

$$P(L|H \cap L) = P(L_t|L),$$

also wenn die Wahrscheinlichkeit einer L–Ausgabe unabhängig von der H–Ein-/Ausgabegeschichte ist.

Gray formalisiert in [Gra92] beide Sicherheitseigenschaften mit einem einheitlichen Modell und beweist, daß Mc Leans Definition stärker als die Definition von Goguen und Messequer ist. Die „probabilistic noninterference“ schützt nur vor verdeckten Kanälen von High nach Low, während Mc Leans „flow model“ jegliche Information, die an einer H–Quelle oder –Senke übertragen wurde, schützt. Wesentlicher Unterschied ist, daß die „probabilistic noninterference“ nur Unabhängigkeit vom High–Eingabeverhalten fordert, wohingegen McLeans „Flow–Model“ auch die von jeglichen Eingaben unabhängigen Ausgaben des Systems einbezieht. Das System in Abbildung 1.1 erfüllt

Goguen und Mesequers „Probabilistic Noninterference“, jedoch nicht Mc Leans „flow model“, da die geheimzuhaltende Information von keiner Eingabe abhängt, sondern im System erzeugt wird.

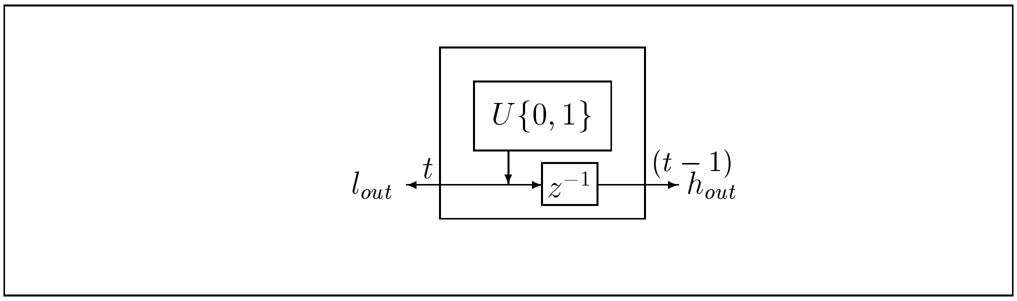

Abbildung 1.1: System, welches PNI aber nicht FM erfüllt [Gra92, S. 279] (U ist Zufallszahlengenerator; z^{-1} Verzögerer)

Gray und Syverson erweitern Gray's Modell in [GS92] auf mehrere Ebenen.

Der Begriff des sicheren Informationsflusses ist in dieser Form zur Formalisierung des kryptologischen Geheimhaltungsbegriffs nicht geeignet, da sich hiermit kein kontrollierter Informationsfluß erzielen läßt und komplexitätstheoretische Geheimhaltung* nicht modelliert ist. Einige der hier vorgeschlagenen Definitionen, sollten daher in der Lage sein, kontrollierten Informationsfluß zu ermöglichen und trotzdem unerwünschten Informationsfluß zu verhindern.

Kapitel 2

Stochastisches Maschinenmodell

Um Aussagen über das Verhalten kryptologischer Systeme zu ermöglichen, wird in diesem Abschnitt ein formales Systemmodell vorgestellt. Es handelt sich um ein wahrscheinlichkeitstheoretisches Rechnermodell, das den deterministischen Fall als Spezialfall enthält. Es ist betont allgemein gehalten, die Anwendung auf Kryptosysteme erfolgt in Kapitel 3. Das Modell besteht aus einer Schnittstellendefinition und zwei Beschreibungsansätzen für das interne Systemverhalten. Alle Aussagen über ein System beziehen sich auf diese Schnittstelle. Ob ein System eine gegebene Aussage erfüllt, hängt dann vom internen Verhalten ab. Das Modell basiert auf dem Rechnermodell aus [Gra92] und Ansätzen aus [Ros93]. Das Modell von Gray wurde um Instanzen und um zusammengesetzte Systeme erweitert.

2.1 Schnittstelle und Schnittstellenablauf

Ein– und Ausgabe erfolgen über Ein– bzw. Ausgabekanäle. In jeden Eingabekanal wird o.B.d.A. zu jedem Zeitpunkt genau eine Operation eingegeben. Analog gibt jeder Ausgabekanal zu jedem Zeitpunkt eine Operation aus. Keine Ein– bzw. Ausgabe wird durch die Operation `nop` modelliert.

Beispiel 2.1 (Addierer) *Abbildung 2.1 zeigt einen Addierer als schematisches Beispiel. Zur Außenwelt führen drei Ein-/Ausgabekanäle, wobei der Kanal* `c` *Ausgabekanal ist und die übrigen Kanäle Eingabekanäle sind.*

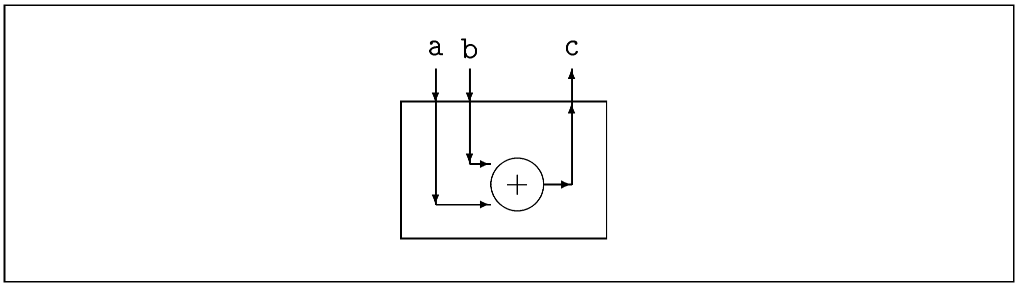

Abbildung 2.1: Systemskizze

Es wird angenommen, daß durch die angegebenen Kanäle der gesamte Informationsfluß in das und aus dem System erfaßt wird, d.h. daß auch verdeckte Kanäle benannt und beschrieben werden.

Die Systemdefinition gliedert sich in eine Definition der Schnittstelle und eine Definition des Verhaltens an dieser Schnittstelle.

Def. 2.1 (Schnittstelle) *Eine Schnittstelle S ist ein Tupel*

$$S = < In, Out, \Gamma, \mathbb{T} >,$$

wobei In die endliche Menge der Eingabekanäle, Out die endliche Menge der Ausgabekanäle, $\Gamma \neq \emptyset$ ein endliches Alphabet und $\mathbb{T} = I\!N^+$ oder $\mathbb{T} = \{1, 2, \ldots, N\}$ die Menge der diskreten Zeitpunkte ist.

$$\mathbb{C} := In \cup Out$$

bezeichne die Menge aller Ein-/Ausgabekanäle.

$$OP := \Gamma^+ \cup \{\texttt{nop}\}$$

ist die Menge der Ein– bzw. Ausgaben (`nop` ist die leere Operation).

Im folgenden werden Ein– und Ausgabe einer Schnittstelle näher beschrieben. Zu jedem Zeitpunkt kann durch jeden Eingabekanal eine Operation eingegeben werden. Jeder Ausgabekanal kann zu jedem Zeitpunkt eine Operation senden. Hierbei ist zu beachten, daß nicht nach Ein– und Ausgabeoperationen unterschieden wird. Die Operation `nop` sagt aus, daß über den Kanal keine Aus– bzw. Eingabe erfolgt. Es wird somit davon ausgegangen, daß jeder Kanal zu jedem Zeitpunkt eine Operation sendet und empfängt.

Da OP abzählbar ist, können mehrargumentige Ein– oder Ausgaben durch einargumentige ersetzt werden.

Beispiel 2.2 (Schnittstelle des Addierers) *Die Schnittstelle des Addierers aus Beispiel 2.1 sei wie folgt definiert:*

$$S =< \{a, b\}, \{c\}, I\!N, I\!N^+ > .$$

Die Schnittstelle ist allgemein und könnte auch zu beliebigen anderen Systemen mit zwei Eingabekanälen und einem Ausgabekanal gehören.

Wie in Beispiel 2.7 gezeigt wird, legt die Wahl der Schnittstelle das Verhalten des Systems noch nicht fest: Das System mit der Schnittstelle des Addierers kann intern beliebig agieren.

Def. 2.2 (Instanz) *Eine Instanz zu einer Schnittstelle S ist eine Menge $A \subseteq \mathbb{C} \times \mathbb{T}$.*

Idee dieses Instanzbegriffes ist, den „Einflußteilbereich“ einer reellen Person oder eines an die Schnittstelle angeschlossenen Systems anzugeben. Als Instanzen können in kryptologischen Systemen z.B. Sender, Empfänger, Signierer und Angreifer (bzw. entsprechende Systeme) auftreten. Eine Instanz gibt an, welche Kanäle zu welchen Zeitpunkten benutzt werden können.

Abb. 2.2 zeigt ein Instanzschema zu einer einfachen Schnittstelle. Aus der Abbildung soll deutlich werden, daß der Einfluß einzelner Instanzen im zeitlichen Verlauf zu– bzw. abnehmen kann. Ein praktisches Beispiel hierfür wäre ein Angreifer, der die „friedlichen“ Benutzer nach und nach von einzelnen Knoten eines Netzwerks verdrängt.

Ist $(c, t) \in A$ mit $c \in In$ so bedeutet dies, daß die Instanz A zur Zeit t die Eingabe des Kanals c bestimmt. Ist $(t, c) \in A$ mit $c \in Out$ so bedeutet dies, daß die Instanz A Kanal c zum Zeitpunkt t beobachten kann.

Beispiel 2.3 (Instanzen zur Schnittstelle des Addierers)
Instanzen zur Schnittstelle aus Beispiel 2.2 wären z.B. Zahleneingabe

$$Z := \{a, b\} \times I\!N^+$$

und Ergebnisausgabe

$$E := \{c\} \times I\!N^+.$$

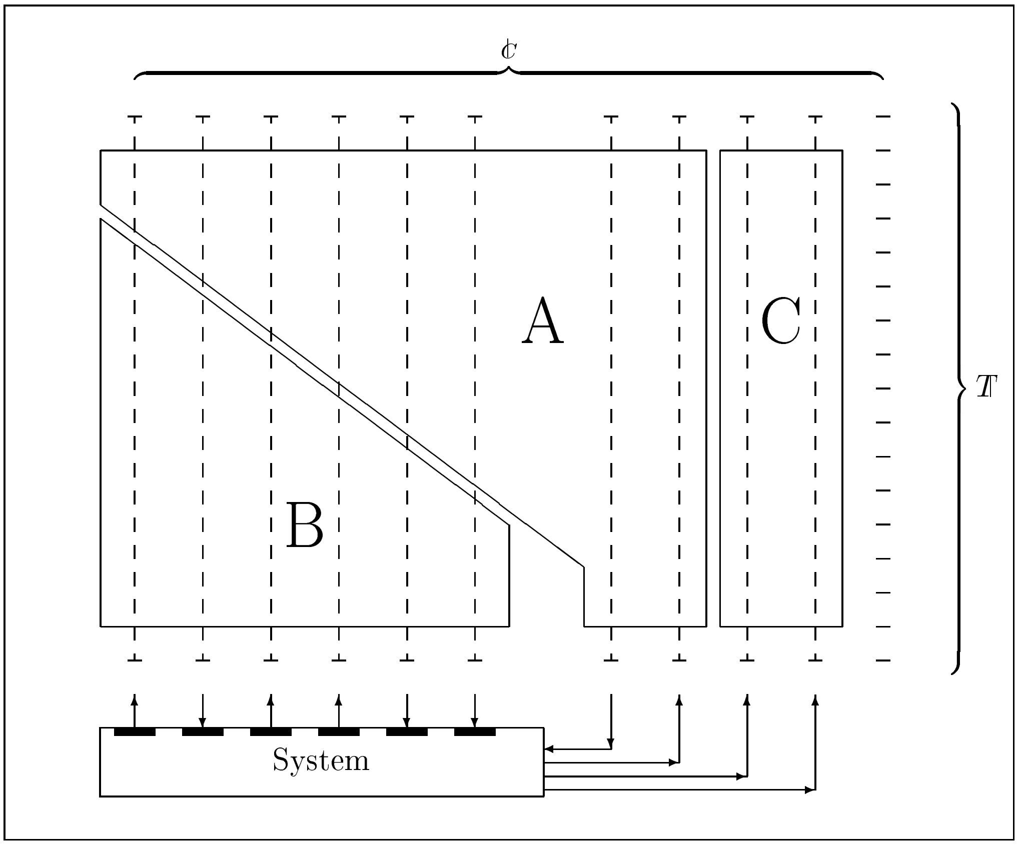

Abbildung 2.2: Instanzschema zu einem Beispielsystem

Die Instanz zu einer festen Kanalmenge K und zum Zeitraum $t = a, \ldots, b$ wird kurz als $K[a, b]$ bezeichnet. Handelt es sich um eine Menge Z von Zeitpunkten, so wird dies mit $K[Z]$ bezeichnet. Häufig werden in diesem Zusammenhang die Instanzen $In[\mathbb{T}] = In \times \mathbb{T}$ und analog $Out[\mathbb{T}]$ und $\mathbb{C}[\mathbb{T}]$ auftreten. Sei M Menge von Zeitpunkten, A eine Instanz, so wird $A \cap \mathbb{C} \times M$ als $A[M]$ bezeichnet.

Die Mächtigkeit von Instanzen läßt sich direkt an der Anzahl kontrollierter Kanäle und der Zeitdauer der Kontrolle ablesen. Eine allmächtige Angreiferinstanz ist durch die Instanz $\mathbb{C}[\mathbb{T}]$ gegeben.

Def. 2.3 (Eingabe) *Eine Eingabe $i_A(S)$ einer Instanz A zu einer Schnittstelle $S =< In, Out, \Gamma, \mathbb{T} >$ ist eine Funktion, die allen $(c, t) \in A$; $c \in In$ eine Operation $op \in OP$ zuordnet.*

Die Menge aller möglichen Eingaben zu einer Instanz A bis zum Zeitpunkt t zu einer Schnittstelle S wird als $I_{A[1,t]}(S)$ bezeichnet:

$$I_{A[1,t]}(S) := \{i_{A[1,t]} | i_{A[1,t]} : A \cap In[1, t] \rightarrow \ OP\}.$$

Die Menge aller möglichen endlichen Eingaben einer Instanz A zu einer Schnittstelle S wird mit $I_A(S)$ bezeichnet. Es gilt:

$$I_A(S) := \bigcup_{t \in \mathbb{T}} I_{A[1,t]}(S).$$

Falls klar ist, welche Schnittstelle zugrundeliegt, werden Eingaben ohne Angabe der Schnittstelle S geschrieben.

Anmerkung 2.1 *$I_{A[1,t]}$ enthält ausschließlich Eingaben der Länge t. I_A enthält Eingaben jeder endlichen Länge (bei unendlichem $\mathbb{T}$).*

Es gilt also für $\mathbb{T} = I\!N^+$ und unendliche A:

$$i_A \notin I_A,$$

ja sogar

$$I_{A[\mathbb{T}]} \cap I_A = \emptyset.$$

Die Eingabe einer Instanz zu einer vorgegebenen Kanalmenge K zu einem Zeitpunkt t kann zur Vereinfachung als Vektor $i_t = i_{K[t,t]}(S)$ und analog $i_{A[1,t]}$ als Matrix $< i_1, \ldots, i_t >= i_{K[1,\ldots,t]}(S)$ geschrieben werden. Die Eingabe über vorgegebene Kanäle und einen Zeitraum $[a, b]$ wird mit $< i_a, \ldots, i_b >$ bezeichnet. Ist $c \in In$ Eingabekanal und $t \in \mathbb{T}$, so bezeichnet $i_{c,t} = i_{\{c\}[t]}$ die Eingabe über diesen Kanal zum Zeitpunkt t.

Beispiel 2.4 (Beispieleingabe für Addierer) *Ein Beispiel für eine Eingabe der Zahleneingabeinstanz Z ist:*

$$i_{Z[1,2]} =< (1,2), (25,8) > \quad \textit{jeweils in der Reihenfolge} \quad (a,b).$$

Def. 2.4 (Ausgabe) *Eine Ausgabe $o_A(S)$ zu einer Schnittstelle*

$$S =< In, Out, \Gamma, \mathbb{T} >$$

an eine Instanz A ist eine Funktion, die allen $(c,t) \in A$; $c \in Out$ eine Operation $op \in OP$ zuordnet.

Die Menge aller möglichen Ausgaben einer Instanz A bis zum Zeitpunkt t zu einer Schnittstelle S wird als $O_{A[1,t]}(S)$ bezeichnet:

$$O_{A[1,t]}(S) := \{o_{A[1,t]} | o_{A[1,t]} : A \cap Out[1,t] \rightarrow \ OP\}.$$

Die Menge aller endlichen Ausgaben einer Instanz A zu einer Schnittstelle S wird mit $O_A(S)$ bezeichnet. Es gilt:

$$O_A(S) := \bigcup_{t \in \mathbb{T}} O_{A[1,t]}(S).$$

Für Ausgaben können die gleichen Schreibweisen und Vereinfachungen wie für Eingaben verwendet werden.

Def. 2.5 (Ablaufprotokoll) *Die Ein– und Ausgabe einer Instanz an einer Schnittstelle wird als Ablaufprotokoll $s_A(S)$ zusammengefaßt. $s_A(S)$ ist also eine Funktion, die allen $(c,t) \in A$ eine Operation zuordnet. $s_A(S)$ bezeichnet ein Ablaufprotokoll ohne Zeitbeschränkung.*

Die Menge aller möglichen Ablaufprotokolle zu einer Instanz A bis zum Zeitpunkt t über eine Schnittstelle S wird als $S_{A[1,t]}(S)$ bezeichnet:

$$S_{A[1,t]}(S) := \{s_{A[1,t]} | s_{A[1,t]} : A \cap \mathbb{C}[1,t] \rightarrow \ OP\}.$$

Die Menge aller endlichen Ablaufprotokolle einer Instanz A und einer Schnittstelle S wird mit $S_A(S)$ bezeichnet. Es gilt

$$S_A(S) := \bigcup_{t \in \mathbb{T}} S_{A[1,t]}(S).$$

Anmerkung 2.2 *Da ein Ablaufprotokoll gleichlange Ein– und Ausgaben besitzt, gibt es den Zustand nach erfolgter Ausgabe wieder.*

Anmerkung 2.3 *$S_A(S)$ enthält Ablaufprotokolle jeder endlichen Länge.*

$S_{\mathbb{C}[\mathbb{T}]}$ enthält nur Protokolle der Länge $|\mathbb{T}|$ (i.d.R. unendlich).

Um ein gegebenes Ablaufprotokoll $s_{\mathbb{C}[\mathbb{T}]}(S)$ des Gesamtsystems auf einzelne Instanzen einzuschränken, wird ein Sichtoperator definiert. Dieser Operator schränkt den Definitionsbereich des Ablaufprotokolls auf die gegebene Instanz ein. Er ist auf der Menge der Ablaufprotokolle zu allen Instanzen definiert:

Def. 2.6 (Sichtoperator) *Sei $S =< In, Out, \Gamma, \mathbb{T} >$ Schnittstelle,*

$$\mathbb{C} := In \cup Out,$$

so ist der Sichtoperator $\lceil$ wie folgt definiert:

$$\lceil : \left(\bigcup_{A \subseteq C[\mathbb{T}]} S_A(S) \right) \times \mathcal{P}(\mathbb{C}[\mathbb{T}]) \to \bigcup_{A \subseteq C[\mathbb{T}]} S_A(S),$$

$$s_A \lceil A' := s_A \cap (A' \times OP).$$

$s_A \lceil A'$ ist also nur für Tupel der Instanz $A \cap A'$ definiert und ansonsten undefiniert.

2.2 Systemmodell

2.2.1 Modellierung des Systemverhaltens

Der im vorigen Abschnitt eingeführte Schnittstellenbegriff wird in den Abschnitten 2.2.2 und 2.4.2 durch Angabe von Funktionen, die das System– und Umgebungsverhalten beschreiben, zu einem Systemmodell erweitert.

Um das Verhalten des Systems zu modellieren und ein Wahrscheinlichkeitsmaß herzuleiten bieten sich zwei Systemmodelle an. Das erste Modell (Abschnitt 2.2.2) betrachtet nur Ein– und Ausgaben; modelliert das Verhalten an der Schnittstelle also als „Black Box“. Eine in Abschnitt 2.4.2 vorgestellte Verfeinerung hiervon modelliert das System mit internen Systemzuständen.

Idee der Definitionen ist, zu gegebenem Ablaufprotokoll und jeder möglichen Ausgabe eine Wahrscheinlichkeit anzugeben.

2.2.2 Systemmodell mit stochastischer Ausgabefunktion

Das hier vorgestellte Systemmodell interagiert mit seiner Umwelt wie folgt:

1. Der bisherige Ablauf bis zum Zeitpunkt t sei auf einem gemeinsamen Band gespeichert.

2. System und Umwelt entnehmen dem Band den bisherigen Ablauf.

3. System und Umwelt berechnen unabhängig voneinander, die Ausgaben für Ihre Kanäle zum Zeitpunkt $t+1$ (Umweltausgabe = Eingabe des Systems).

4. Das Ablaufprotokoll des Zeitpunktes $t+1$ wird an das Band angehängt.

5. Weiter mit Schritt 2.

Es muß bei der Realisierung der Systeme darauf geachtet werden, daß keiner zum Zeitpunkt $t+1$ noch schnell Ausgaben eines anderen Systems zum Zeitpunkt in die Berechnung seiner Ausgabe zum Zeitpunkt $t+1$ einbeziehen kann (vgl. Beavers „rushing adversaries“ [Bea91]). Dies kann z.B. durch „simultaneous broadcasting“ sichergestellt werden.

Die Reaktion eines Systems auf bestimmte Eingaben, wird durch die Ausgabefunktion $\mathcal{O}$ festgelegt. Sie gibt für alle Zeitpunkte $t \in \mathbb{T}$ zu vorgegebenem Ein– und Ausgabeablaufprotokoll bis zum Zeitpunkt t die Ausgabewahrscheinlichkeit jeder möglichen Ausgabe zum Zeitpunkt $t+1$ an.

Def. 2.7 (Ausgabefunktion) *Eine Funktion*

$$\mathcal{O} : \bigcup_{t \in \mathbb{T}} O_{Out[t+1]} \times I_{In[1,t]} \times O_{Out[1,t]} \to [0,1]$$

heißt Ausgabefunktion zu einer Schnittstelle S, falls für alle $t \in \mathbb{T}$ für alle

$$\{< i_1, \ldots, i_t >, < o_1, \ldots, o_t >\} \in I_{In[1,t]} \times O_{Out[1,t]}$$

gilt

$$\sum_{o_{t+1} \in O_{Out[t+1]}} \mathcal{O}(o_{t+1} | < i_1, \ldots, i_t >, < o_1, \ldots, o_t >) = 1.$$

Beispiel 2.5 (Ausgabefunktion des Addierers)
Folgende Ausgabefunktion dient dazu, aus der Schnittstelle aus Bsp. 2.2 einen Addierer zu konstruieren:

$$\mathcal{O}(o_{c,t+1} | < i_1, \ldots, \begin{pmatrix} i_{a,t} \\ i_{b,t} \end{pmatrix} >, < o_1, \ldots, o_t >) :=$$

$$\begin{cases} 1 & \textit{falls} \quad \underset{n_1, n_2 \in \mathbb{N}}{\exists} \quad \begin{array}{l} i_{a,t} = n_1 \wedge i_{b,t} = n_2 \wedge \\ o_{c,t+1} = n_1 + n_2 \end{array} \\ 1 & \textit{falls} \quad \begin{array}{r} o_{c,t+1} = \mathtt{nop} \wedge \\ \underset{n_1, n_2 \in \mathbb{N}}{\neg\exists} \; i_{a,t} = n_1 \wedge i_{b,t} = n_2 \end{array} \\ 0 & \textit{sonst} \end{cases} .$$

Def. 2.8 (System) *Ein System ist ein Paar $\Sigma = (S, \mathcal{O})$, wobei S Schnittstelle gemäß Definition 2.1 und $\mathcal{O}$ eine Ausgabefunktion gemäß Definition 2.7 ist.*

Anmerkung 2.4 *Bezeichnungen wie $i_A(\Sigma)$, $o_A(\Sigma)$ beziehen sich auf die im System enthaltene Schnittstelle.*

Def. 2.9 (Abgeschlossenes System) *Ein System*

$$\Sigma = (< \emptyset, Out, \Gamma, \mathbb{T} >, \mathcal{O})$$

heißt abgeschlossen.

Ein abgeschlossenes System ist also ein System ohne Eingabekanäle. Es ist unabhängig von seiner Umgebung.

Beispiel 2.6 (Addierer) *Die komplette Beschreibung des Addiersystems ist:*

$$\Sigma_{Add} := (S, \mathcal{O}),$$

wobei S die Schnittstelle des Addierers (Bsp. 2.2) und $\mathcal{O}$ die Ausgabefunktion des Addierers (Bsp. 2.5) ist.

2.2.3 Systemfamilien

Def. 2.10 (Systemfamilie)
Sei I abzählbare Menge. Eine Folge $\Sigma = (\Sigma_n)_{n\in I}$, $\Sigma_n = (S, \mathcal{O}_n)$ von Systemen mit einheitlicher Schnittstelle S heißt Systemfamilie. $n \in I$ heißt Sicherheitsparameter. Sei $\mathcal{O}$ eine universelle Ausgabefunktion für die gilt:

$$\mathcal{O}_n = \mathcal{O}(n),$$

so wird eine derartige Systemfamilie kurz mit $\Sigma = (S, \mathcal{O})$ bezeichnet.

Anmerkung 2.5 *Indexmenge I ist in der Regel die Menge der positiven natürlichen Zahlen IN^+. Es können jedoch auch Systeme mit zusammengesetztem Sicherheitsparameter definiert werden.*

Eine polynomielle Systemfamilie ist eine Menge von Systemen, die sich nur durch unterschiedliche Sicherheitsparameter unterscheiden und deren Reaktionszeiten polynomial im Sicherheitsparameter und in der Länge der Eingaben sind. Die Definition läßt nahezu beliebige Systemfolgen zu. Systemfolgen, die die Forderung nach steigender Sicherheit nicht erfüllen, werden jedoch durch die später geforderte steigende Geheimhaltung ausgeschlossen. Es können auch Systemfamilien mit mehreren Sicherheitsparametern angegeben werden. Solche Systeme müssen dann in allen Sicherheitsparametern polynomiell sein.

Def. 2.11 (Polynomielle Systemfamilie) *Eine polynomielle Systemfamilie $(\Sigma_n)_{n\in I}$ ist eine Systemfamilie $(S, \mathcal{O})$ mit einer im Sicherheitsparameter und in den Eingaben polynomiell zeitbeschränkten „universellen“ Ausgabefunktion $\mathcal{O}$.*

Anmerkung 2.6 (Mögliche Variante) *Wie* B. PFITZMANN *andeutet [Pfi93a, §4.4.3], kann auch gefordert werden, daß ein System nur im Sicherheitsparameter polynomiell sein darf.*

2.3 Modellierung der Umgebung

Das Umgebungsverhalten wird analog zum Systemverhalten als Funktion definiert, die bei gegebenem Ablaufprotokoll jeder Eingabe eine Wahrscheinlichkeit zuweist.

Die Eingabefunktion $\mathcal{I}$ gibt für alle Zeitpunkte $t \in I\!N$ zu vorgegebenem Ein– und Ausgabeablaufprotokoll bis zum Zeitpunkt t die Eingabewahrscheinlichkeit jeder möglichen Eingabe zum Zeitpunkt $t+1$ an. Sie ist nicht Teil des Systems.

Def. 2.12 (Eingabefunktion) *Eine Funktion*

$$\mathcal{I} : \bigcup_{t \in I\!N} I_{In[t+1]} \times I_{In[1,t]} \times O_{Out[1,t]} \to [0,1]$$

heißt Eingabefunktion eines Systems S, falls für alle $t \in \mathbb{T}$ für alle

$$\{< i_1, \ldots, i_t >, < o_1, \ldots, o_t >\} \in S_{\mathbb{C}[1,t]}$$

gilt:

$$\sum_{i_{t+1} \in I_{In[t+1]}} \mathcal{I}(i_{t+1} | < i_1, \ldots, i_t >, < o_1, \ldots, o_t >) = 1.$$

Anmerkung 2.7 *Die Umgebung kann ein anderes System sein, da das System Teil eines größeren Systems sein kann (Man beachte die Ähnlichkeit der Eingabefunktion mit einer Ausgabefunktion!).*

Anmerkung 2.8 *Bei abgeschlossenen Systemen (Def. 2.9) ist eine Eingabefunktion nicht nötig.*

In dieser Arbeit werden nur abgeschlossene Systeme betrachtet, da hier alle Aussagen über das Systemverhalten auch ohne eine vorgegebene Eingabefunktion möglich sind.

2.4 Zusammengesetzte Systeme

In diesem Abschnitt wird erläutert, wie aus mehreren Teilsystemen und einer Verbindungsfunktion ein Gesamtsystem erstellt werden kann. Dies ist die

Grundlage, um Teilsysteme zu einem Gesamtsystem zu vereinigen. Ein Zero-Knowledge-Proof* System könnte z.B. ein Teilsystem eines Signatursystems* sein.

Die zugrundeliegende Idee ist, die Systeme so zu „verschalten“, daß Ausgaben eines Kanals zum nächsten Zeitpunkt an dem mit ihm verbundenen Kanal, der in der Regel zu einem anderen System gehört, wieder eingegeben werden. Ist ein Eingabekanal eines Systems unverbunden, so handelt es sich um einen Eingabekanal des Gesamtsystems. Je nach „Verschaltung“ verbleiben einzelne Eingabekanäle als Eingabekanäle des verschalteten Systems. Ein Verschaltungsbeispiel zeigt Abb 2.3.

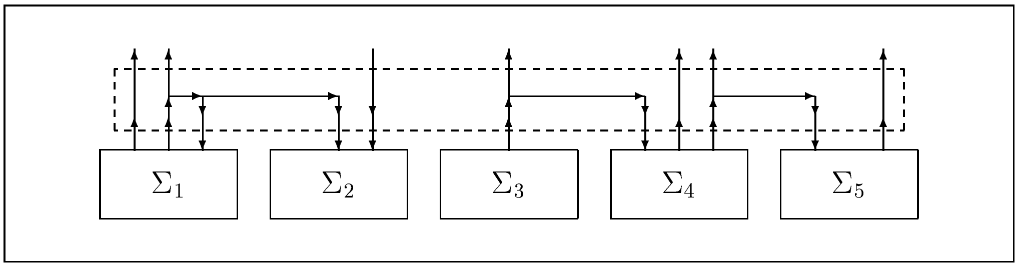

Abbildung 2.3: Konstruktion eines Gesamtsystems aus Einzelsystemen

2.4.1 Teilsysteme und Teilsystemfamilien

Ein Teilsystem zu einer Instanz A ist ein System zu einer Schnittstelle, welches nur auf Eingaben dieser Instanz reagiert und dessen Ausgaben nur im Rahmen dieser Instanz stattfinden. Ausgaben für $(c,t) \notin A$ sind nicht definiert. Ebenso ist laut der folgenden Definition nicht gestattet, mehr als ein endliches Ablaufprotokoll zu dieser Instanz einzugeben.

Def. 2.13 (Teilsystem zur Instanz A) *Ein Tupel $\Sigma_A = (S, A, \mathcal{O}_A)$ mit $S = (In, Out, \Gamma, \mathbb{T})$ heißt Teilsystem zur Instanz $A \subseteq (In \cup Out) \times \mathbb{T}$ gdw. $\mathcal{O}_A$ Ausgabefunktion ist und gilt:*

$$\mathcal{O}_A : \bigcup_t O_{A[t+1]} \times I_{A[1,t]} \times O_{A[1,t]} \to [0,1]$$

und für alle $t \in \mathbb{T}$ gilt:

$$\forall_{\substack{i_{A[1,t]} \in I_{A[1,t]} \\ o_{A[1,t]} \in O_{A[1,t]}}} \sum_{o_{A[t+1]} \in O_{A[t+1]}} \mathcal{O}_A(o_{A[t+1]} | i_{A[1,t]}, o_{A[1,t]}) = 1.$$

Lemma 2.1 (Teilsystem zu einem System) *Ist*

$$\Sigma = ((In, Out, \Gamma, \mathbb{T}), \mathcal{O})$$

System nach Def. 2.8, so ist $\Sigma' = ((In, Out, \Gamma, \mathbb{T}), (In \cup Out) \times \mathbb{T}, \mathcal{O})$ Teilsystem nach Definition 2.13.

Beweis: Folgt direkt aus den Definitionen. ■

Das folgende System agiert „auf der Instanz" wie ein Teilsystem und gibt sonst nichts aus:

Lemma 2.2 (System zu einem Teilsystem) *Ist*

$$\Sigma_A = ((In, Out, \Gamma, \mathbb{T}), A, \mathcal{O}_A)$$

Teilsystem zur Instanz A, so ist $(S, \mathcal{O})$ mit

$$\mathcal{O}(o_{t+1} | < i_1, \ldots, i_t >, < o_1, \ldots, o_t >) =$$
$$= \begin{cases} 0 \qquad \textit{für} \qquad \exists_{(c,t+1) \in o_{t+1} \lceil (\mathbb{C}[\mathbb{T}] - A)} \quad o_{c,t+1} \neq \texttt{nop} \\ \mathcal{O}_A(o_{t+1} | < i_1, \ldots, i_t > \lceil A, < o_1, \ldots, o_t > \lceil A) \\ \qquad \textit{für} \qquad \neg\exists_{(c,t+1) \in o_t + 1 \lceil (\mathbb{C}[\mathbb{T}] - A)} \quad o_{c,t+1} \neq \texttt{nop} \end{cases}$$

System zur Schnittstelle S.

Beweis: $\mathcal{O}$ ist Ausgabefunktion, da sowohl für $o_{t+1} \lceil A$ (wegen Def. 2.13) als auch für $o_{t+1} \lceil (\mathbb{C} \times \mathbb{T} - A)$ (Laut Definition) die Summe der Ausgabewahrscheinlichkeiten 1 ist. ■

Lemma 2.3 *Ist*

$$\Sigma_A = ((In, Out, \Gamma, \mathbb{T}), A, \mathcal{O}_A)$$

Teilsystem zur Instanz A, so ist auch

$$\Sigma_A = ((In', Out', \Gamma, \mathbb{T}), A, \mathcal{O}_A)$$

mit $In' \supseteq In$ und $Out' \supseteq Out$ Teilsystem zur Instanz A.

Beweis: Direkt aus Def. 2.13.

Def. 2.14 (Teilsystemfamilie) *Sei I abzählbare Menge. Eine Folge $\Sigma_A = (\Sigma_n)_{n \in I}$, $\Sigma_n = (S, A, \mathcal{O}_{A,n})$ von Teilsystemen mit einheitlicher Schnittstelle S und einheitlicher Instanz A heißt Teilsystemfamilie. $n \in I$ heißt Sicherheitsparameter. Sei $\mathcal{O}$ universelle Ausgabefunktion mit*

$$\mathcal{O}_{A,n} = \mathcal{O}(n),$$

so wird eine derartige Teilsystemfamilie kurz mit $\Sigma = (S, A, \mathcal{O}_A)$ bezeichnet.

Anmerkung 2.9 *Bezeichnungen für Systemfamilien, wie „polynomiell", werden analog auf Teilsysteme übertragen.*

Anmerkung 2.10 *Falls Verwechslungen ausgeschlossen sind, werden Teilsysteme kurz als Systeme bezeichnet.*

2.4.1.1 Zusammensetzen von Teilsystemen

Idee der folgenden Definition ist, die unabhängigen Teilsysteme zu den Teilinstanzen über eine partielle Eingabefunktion zu einer Gesamtausgabefunktion zu verschalten. Dies wird Bezeichnungen wie „das aus Σ_A und Σ_B zusammengesetzte System Σ" ermöglichen. Voraussetzung ist ein einheitliches Alphabet und einheitliche Zeitpunkte.

Bestehende Modelle (z.B. [BD87, Man91, Hoa85]) zur Verbindung von Systemen konnten nicht verwendet werden, da sie

- nichtdeterministische und nicht, wie hier benötigt, probabilistische Systeme verknüpfen,
- nicht auf Instanzen übertragbar sind[1] oder
- nicht auf das Systemmodell nach Gray angewendet werden können.

Das gewählte Modell besitzt

- globale Synchronität und

[1] Hierzu wird eine zeitabhängig wechselnde Verschaltung benötigt

- nichtspeichernde Leitungen.

Die folgenden Definitionen verbinden die Systeme, indem

- eine Verbindungsfunktion v definiert wird, die zu jedem Eingabekanal und jedem Zeitpunkt festlegt, ob er mit einem Ausgabekanal verbunden werden soll, ob er weiterhin Eingabekanal des Gesamtsystem sein soll oder ob er durch ständige Eingabe von `nop` gelöscht werden soll,
- eine Gesamtschnittstelle definiert wird, die alle Ausgabekanäle und alle herausgeführten Eingabekanäle (je nach $v()$) des Gesamtsystems enthält und
- eine Ausgabefunktion $\mathcal{O}$ des Gesamtsystems aus den Einzelausgabefunktionen konstruiert wird.

  Die Konstruktion dieser Ausgabefunktion folgt dem Schema, welches bei der Eingabefunktion (Def. 2.12) skizziert wurde: Alle Systeme schreiben ihre Ausgaben auf ein Band; einige Ausgaben werden auf die Eingaben umkopiert; alle Systeme lesen das (z.T. abgeänderte) Ablaufprotokoll zu ihrer Instanz.

Def. 2.15 (Verbindungsfunktion) *Seien* $(\Sigma_1, \Sigma_2, \ldots, \Sigma_n)$ *Teilsysteme mit*

$$\Sigma_i = ((In_i, Out_i, \Gamma, \mathbb{T}), A_i, \mathcal{O}_i),$$

und

$$\biguplus_i A_i \cap (Out_i \times \mathbb{T}) \wedge \biguplus_i In_i \tag{2.1}$$

so heißt eine Funktion

$$v : \bigcup_i In_i \times \mathbb{T} \rightarrow \{\texttt{nop}, \uparrow\} \cup \bigcup_i Out_i$$

Verbindungsfunktion der Systeme.

Ist $v(i,t) = o$, so wird der Eingabekanal i zum Zeitpunkt t die Ausgabe des Kanals o zum Zeitpunkt t als Eingabe. Ist $v(i,t) = \texttt{nop}$ so soll der Eingabekanal durch Eingabe von `nop` „gelöscht" werden. Ist $v(i,t) = \uparrow$ so bleibt der Kanal Eingabekanal des Gesamtsystems. Abbildung 2.4 veranschaulicht diese drei Fälle.

Die Schnittstellen der Teilsysteme werden in Abhängigkeit von $v()$ kombiniert:

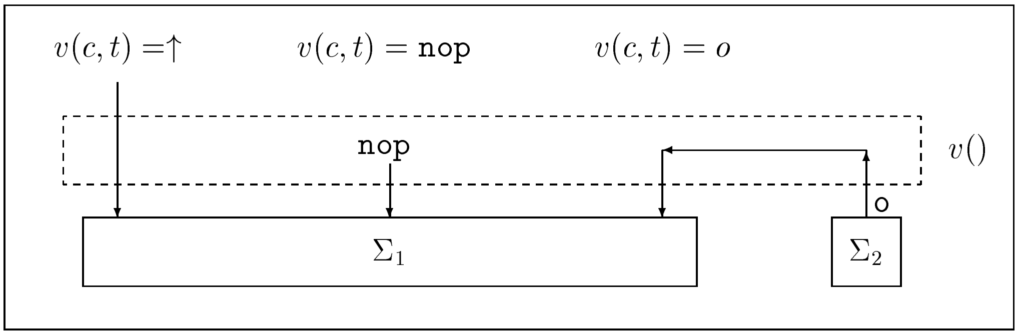

Abbildung 2.4: Die drei Möglichkeiten, einen Eingabekanal zu verbinden

Def. 2.16 (Gesamtschnittstelle) *Seien* $(\Sigma_1, \Sigma_2, \ldots, \Sigma_m)$ *Teilsysteme und* $v()$ *Verbindungsfunktion nach Def. 2.15 oder 2.18, mit*

$$\Sigma_i = ((In_i, Out_i, \Gamma, \mathbb{T}), A_i, \mathcal{O}_i),$$

so heißt

$$S_v = (In, Out, \Gamma, \mathbb{T})$$

mit

$$In := \bigcup_{i,t} \{c | c \in In_i \wedge v(c,t) = \uparrow\}$$

und

$$Out := \bigcup_i Out_i$$

Gesamtschnittstelle der Systeme (zu v*).*

Ist ein Eingabekanal zu einem beliebigen Zeitpunkt Eingabekanal des Gesamtsystems, so muß er in die Gesamtkanalmenge aufgenommen werden. Dies kann dazu führen, daß ein Kanal, der nur zu einem Zeitpunkt eine Eingabe verlangt Eingabekanal des Gesamtsystems ist. Dies ist aber nicht unerwartet, da analog hierzu ein Teilsystem zur Instanz A alle Kanäle, die in der Instanz vorkommen, enthalten muß.

Def. 2.17 (Zusammengesetztes System) *Seien* $(\Sigma_1, \Sigma_2, \ldots, \Sigma_n)$ *Teilsysteme mit*

$$\Sigma_i = ((In_i, Out_i, \Gamma, \mathbb{T}), A_i, \mathcal{O}_i).$$

Sei $v()$ Verbindungsfunktion und S_v die Gesamtschnittstelle hierzu. Für $v()$ gelte für alle (c,t) mit $v(c,t) \notin \{\texttt{nop}\ ,\uparrow\}$:

$$\exists_j (v(c,t),t) \in A_j.$$

Das aus den Teilsystemen $(\Sigma_1, \Sigma_2, \ldots, \Sigma_n)$ zusammengesetzte Teilsystem ist wie folgt definiert:

$$\Sigma = (S_v, A, \mathcal{O}) =: \Sigma_1 \bowtie_v \Sigma_2 \bowtie_v \ldots \bowtie_v \Sigma_n$$

mit

$$A = \{(c,t) | t \in \mathbb{T}, c \in In, v(c,t) = \uparrow\} \cup \bigcup_i A_i \cap Out \times \mathbb{T}.$$

Für $\mathcal{O}$ gilt für alle $t \in \mathbb{T}$ und für alle

$$s_{A[1,t]} = \{i_{A[1,t]}, o_{A[1,t]}\} \in I_{A[1,t]} \times O_{A[1,t]} :$$

$$\begin{aligned} &\mathcal{O}(o_{t+1} | s_{A[1,t]}) := \\ &= \prod_i \mathcal{O}_i(o_{t+1} \lceil A_i | v^*(s_{A[1,t]}) \lceil A_i) \end{aligned}$$

mit

$$v^*(s_{c,t}) := \begin{cases} s_{o,t} & \textit{für} \quad c \in In \wedge v(c,t) = o & \textit{(Wiedereingabe)} \\ \texttt{nop} & \textit{für} \quad c \in In \wedge v(c,t) = \texttt{nop} & \textit{(Löschung)} \\ s_{c,t} & \textit{sonst} & \textit{(Weiter– oder Ausgabe)} \end{cases} .$$

Die Idee der Definition von $v^*()$ ist, die Sicht der entsprechenden Teilinstanz aufzubereiten. $\mathcal{O}_i$ sieht also nur das „korrigierte" Ablaufprotokoll. Da alle Teilsysteme einheitliches Alphabet besitzen, können keine Verständigungsprobleme auftreten. Die Wahrscheinlichkeit einer Ausgabe errechnet sich als Produkt der Einzelwahrscheinlichkeiten, da die Einzelsysteme voneinander unabhängig sind. Falls die Ausgabeteilinstanzen die Ausgabekanalmenge nicht zu allen Zeitpunkten abdecken, ist die Gesamtausgabefunktion für einzelne (c,t) nicht definiert. Sie kann jedoch nach Lemma 2.2 vervollständigt werden.

Die Bedingung an $v()$ stellt sicher, daß diese undefinierten Ausgaben nicht als Eingabe eines anderen Teilsystems benötigt werden.

Um die Verbindung realer Systeme zu vereinfachen, wird im folgenden eine Funktion definiert, die Kanäle mit gleichen Namen verbindet. Eine „geschickte" Namensgebung der Kanäle soll mit der folgenden Definition die Verbindung der Systeme implizieren. Hierzu wird erlaubt, daß Kanalnamen mehrfach auftreten. Idee ist, daß wenn die Tupel (c,t) mit (c Eingabekanal) und (c,t) mit (c Ausgabekanal) in den Instanzen sind, so werden sie verbunden ($v(c,t) = c$). Besitzt ein Tupel (c,t) (c Eingabekanal) kein „passendes" Ausgabetupel, so wird nichts eingegeben ($v(c,t) = \texttt{nop}$). Da nach diesem Vorgehen kein freier Eingabekanal existiert, ist das so konstruierte System abgeschlossen.

Def. 2.18 (Kanonische Verbindungsfunktion)
Seien $(\Sigma_1, \Sigma_2, \ldots, \Sigma_n)$ *Teilsysteme mit*

$$\Sigma_i = ((In_i, Out_i, \Gamma, \mathbb{T}), A_i, \mathcal{O}_i)$$

und

$$\biguplus_i Out_i \cap (A_i \times \mathbb{T}),$$

so ist die kanonische Verbindungsfunktion wie folgt definiert:

$$v^+(c,t) := \begin{cases} c & \textit{falls} \quad \exists_i (c,t) \in A_i \cap Out \times \mathbb{T} \\ \texttt{nop} & \textit{sonst} \end{cases}$$

Anmerkung 2.11 (Erinnerung) *Hier werden keine disjunkten Eingabekanalmengen und somit auch keine disjunkten Eingabeinstanzen gefordert. Einzige Bedingung ist, daß keine zwei Instanzen zu einem Zeitpunkt in ein und denselben Kanal eingeben.*

Möglich und sogar erwünscht ist, daß für einen Eingabekanal c, der bei zwei Systemen vorkommt, gilt $(c,t) \in A_1$ *und* $(c,t) \in A_2$. *In diesem Fall wird die entsprechende Ausgabe* ($o_{v(c,t),t}$) *in beide Kanäle eingegeben.*

Def. 2.19 (Kanonisch zusammengesetztes System)
Seien $(\Sigma_1, \Sigma_2, \ldots, \Sigma_n)$ *Teilsysteme. Das aus diesen Systemen kanonisch zusammengesetzte System*

$$\Sigma = (S, \mathcal{O}) := \Sigma_1 \bowtie_+ \Sigma_2 \bowtie_+ \ldots \bowtie_+ \Sigma_n$$

ist die Vervollständigung gemäß Lemma 2.2 des folgenden Teilsystems:

$$\Sigma_1 \bowtie_{v^+} \Sigma_2 \bowtie_{v^+} \ldots \bowtie_{v^+} \Sigma_n.$$

Anmerkung 2.12 *Ein kanonisch zusammengesetztes System ist abgeschlossen.*

Beispiel 2.7 (Zähler aus Addierer) *Wir wollen nun ein spezielles Teilsystem Σ_Z (Zahlengenerator) und eine Verschaltungsfunktion v angeben, so daß das folgende System ein Zähler[2] ergibt:*

$$\Sigma_{Count} := \Sigma_Z \bowtie_v \Sigma_{Add}$$

Das folgende System gibt immer „1“ aus (Abb. 2.5):

$$\Sigma_Z = ((\emptyset, \{a\}, I\!N, I\!N), \{a\} \times I\!N, \mathcal{O}_Z)$$

mit

$$\mathcal{O}_Z(o_{t+1} | <>, < o_1, \ldots, o_t >) := \begin{cases} 1 & \textit{Für } o_{t+1} = 1 \\ 0 & \textit{sonst} \end{cases}$$

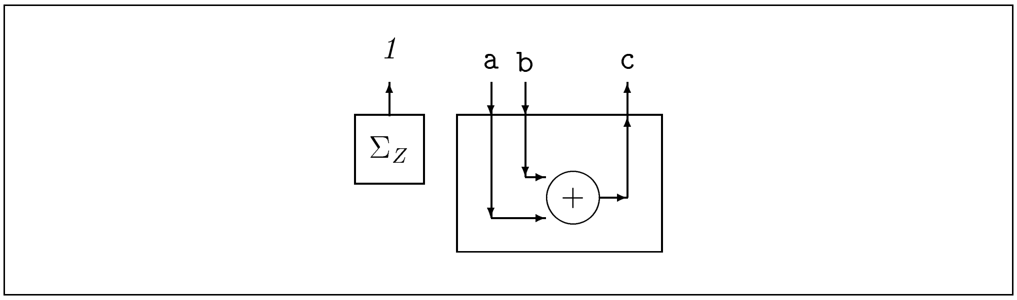

Abbildung 2.5: „1“ Generator und Addierer

Die folgende Verbindungsfunktion verbindet zum Zeitpunkt $t = 1$ den Ausgabekanal des Zahlengenerators mit den Eingabekanälen des Addierers. Später wird die Addiererausgabe in einen Eingabekanal des Addierers wieder eingegeben und der andere Eingabekanal weiter mit dem Zahlengenerator verbunden:

$$v_1(c', t) = \begin{cases} \mathtt{a} & \textit{für } t = 2 \\ \mathtt{a} & \textit{für } t \neq 2 \textit{ und } c' = \mathtt{a} \\ \mathtt{c} & \textit{für } t \neq 2 \textit{ und } c' = \mathtt{b} \end{cases}$$

[2]Da die Eingabefunktion des Addierers zwei Zahlen fordert, fängt er bei 2 an zu zählen...

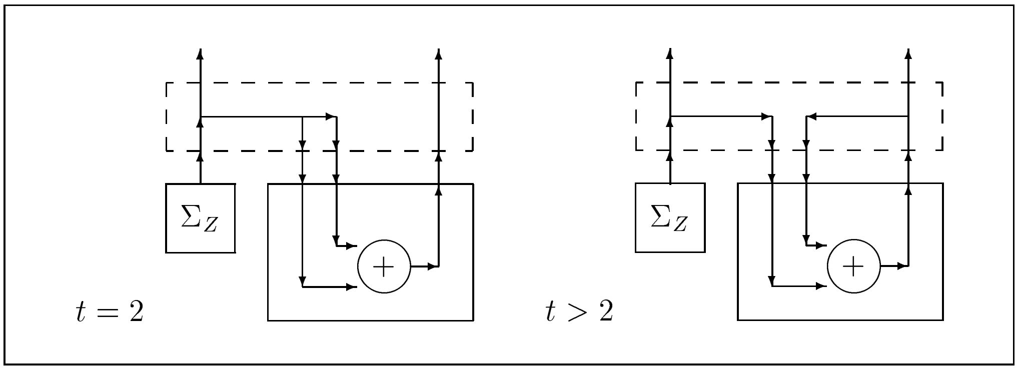

Abbildung 2.6: „1“ Generator und Addierer als Zähler (Verbindung $v_1()$)

Beispiel 2.8 *Durch geringfügige Änderung der Verschaltungsfunktion erhält man ein System, welches zum Zeitpunkt ($t \geq 2$) über den Kanal* `c` 2^{t-2} *ausgibt (Abb. 2.7):*

$$v_2(c', t) = \begin{cases} \mathtt{a} & \textit{für } t = 2 \\ \mathtt{c} & \textit{für } t \neq 2 \end{cases} .$$

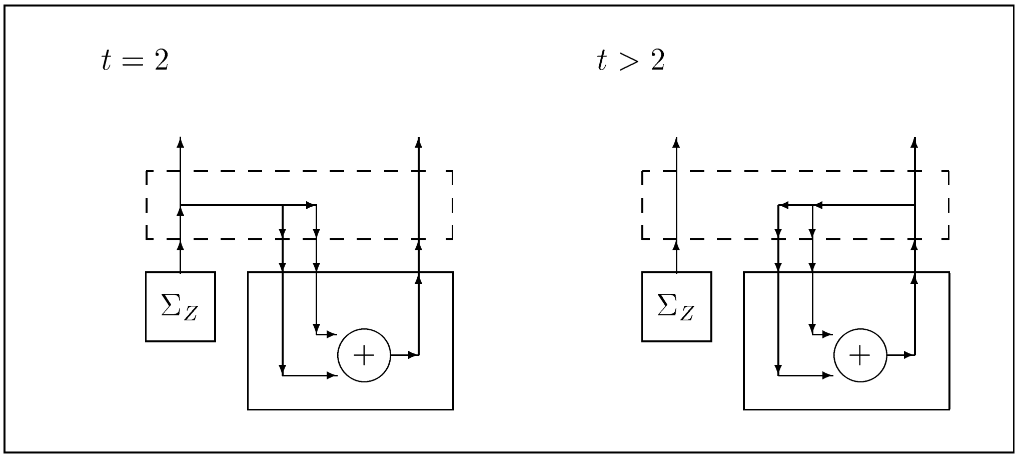

Abbildung 2.7: „1“ Generator und Addierer als Zweierpotenzgenerator (Verbindung $v_2()$)

Def. 2.20 (Zusammengesetzte Systemfamilie)
Seien $(\Sigma_1, \Sigma_2, \ldots, \Sigma_n)$ Teilsystemfamilien mit einheitlicher Indexmenge I, so

erhält man die zusammengesetzte Systemfamilie durch Zusammensetzen der Einzelsysteme für jedes $i \in I$ analog zu Def. 2.17.

Lemma 2.4 *Seien $(\Sigma_1, \Sigma_2, \ldots, \Sigma_n)$ polynomielle Teilsystemfamilien, v Verbindungsfunktion, so ist Σ mit*

$$\Sigma = ((In, Out, \Gamma, \mathbb{T}), \mathcal{O}) =: \Sigma_1 \bowtie_v \Sigma_2 \bowtie_v \ldots \bowtie_v \Sigma_n$$

polynomielle Teilsystemfamilie.

Beweisidee: Folgt mit dem „größten“ der Polynome p_1 bis p_n direktblackbox

2.4.2 Herleitung der Ausgabefunktion $\mathcal{O}$ aus Zuständen und Übergangsfunktion

Als Beispiel für eine andere Darstellung der Ausgabefunktion, wird sie in diesem Abschnitt aus einer (globalen) Zustandsmenge und einer Übergangsfunktion hergeleitet. Dies soll nur als Beispiel für die Verfeinerung der globalen Ausgabefunktion dienen. Andere Verfeinerungen (insbesondere unter Berücksichtigung von Netzeigenschaften) sind denkbar.

Dieser Abschnitt soll exemplarisch darstellen, wie diese „Black–Box“ innerhalb des Systems gefüllt werden kann. Das hier vorgestellte Beispiel wird in den späteren Abschnitten nicht vorausgesetzt.

Bei diesem Modell wird das Verhalten des Systems durch eine Menge von Zuständen und eine Übergangsfunktion modelliert.

Def. 2.21 (Reaktionsfunktion) *Sei $Q \neq \emptyset$ abzählbare Menge von Zuständen des Systems, $q_0 \in Q$ der Anfangszustand und sei eine Schnittstelle S gegeben. Eine Reaktionsfunktion ist eine Funktion $\mathcal{T}$:*

$$\mathcal{T} : I_{In[t]} \times Q \times O_{Out[t+1]} \times Q \to [0, 1]$$

mit:

$$\underset{(i_t, q_t) \in I_{In[t]} \times Q}{\forall} : \left[\sum_{(o_{t+1}, q_{t+1}) \in O_{Out[t]} \times Q} \mathcal{T}(i_t, q_t, o_{t+1}, q_{t+1}) = 1 \right]$$

Die Reaktionsfunktion gibt zu einem Zustand, einer Eingabe, einer Ausgabe und einem Folgezustand die Wahrscheinlichkeit an, mit der diese Reaktion stattfindet.

Beispiel 2.9 (Reaktionsfunktion des Addierers)
Die Reaktionsfunktion eines Addierers (Bsp. 2.1) könnte wie folgt definiert sein ($Q = \{q_0\}$; Schnittstelle wie in Bsp. 2.2):

$$\mathcal{T}(i_t, q_0, o_{t+1}, q_0) := \begin{cases} 1 & \textit{falls} \quad \underset{n_1,n_2 \in I\!N}{\exists} i_t = \binom{n_1}{n_2} \wedge o_{t+1} = n_1 + n_2 \\ 1 & \textit{falls} \quad o_{t+1} = \texttt{nop} \ \wedge \underset{n_1,n_2 \in I\!N}{\neg\exists} i_t = \binom{n_1}{n_2} \\ 0 & \textit{sonst} \end{cases}$$

Die folgende Funktion gibt zu einer Eingabe, einer Ausgabe und einem Zustand die Wahrscheinlichkeit an, mit der das System sich nach dieser Ein-/Ausgabe in diesem Zustand befindet:

Def. 2.22 (Zustandsfunktion) *Gegeben sei eine Schnittstelle S. Die Zustandsfunktion ist eine Funktion q mit:*

$$q : \bigcup_{t \in T\!I} O_{Out[1,t]} \times I_{In[1,t]} \times Q \to [0,1]$$

und

$$q(<>, <>, q_0) = 1$$
$$q(<>, <>, q_i) = 0 \textit{ für } q_i \neq q_0$$

$$\begin{aligned} & q(< i_1, \ldots, i_n >, < o_1, \ldots, o_n >, q_{n+1}) = \\ = & \sum_{q_n, o_{n+1}} q(< i_1, \ldots, i_{n-1} >, < o_1, \ldots, o_{n-1} >, q_n) \cdot \mathcal{T}(i_n,\, q_n,\, o_{n+1},\, q_{n+1}). \end{aligned}$$

Für den späteren Korrektheitsbeweis wird noch ein Lemma benötigt:

Lemma 2.5 *Für alle $< i_1, \ldots, i_n >$ und $< o_1, \ldots, o_n >$ gilt:*

$$\sum_{q_{n+1}} q(< i_1, \ldots, i_n >, < o_1, \ldots, o_n >, q_{n+1}) = 1.$$

Für $n = 0$ gilt die Behauptung. Sei also:

$$\sum_{q_n} q(< i_1, \ldots, i_{n-1} >, < o_1, \ldots, o_{n-1} >, q_n) = 1.$$

Es gilt:

$$
\begin{aligned}
&\sum_{q_{n+1}} q(<i_1,\ldots,i_n>,<o_1,\ldots,o_n>,q_{n+1}) = \\
&= \sum_{q_{n+1}} \sum_{q_n,o_{n+1}} q(<i_1,\ldots,i_{n-1}>,<o_1,\ldots,o_{n-1}>,q_n) \cdot \mathcal{T}(i_n,\, q_n,\, o_{n+1},\, q_{n+1}) \\
&= \sum_{q_n} q(<i_1,\ldots,i_{n-1}>,<o_1,\ldots,o_{n-1}>,q_n) \cdot \\
&\qquad \sum_{o_{n+1},q_{n+1}} \mathcal{T}(i_n,\, q_n,\, o_{n+1},\, q_{n+1}) \\
&= \sum_{q_n} q(<i_1,\ldots,i_{n-1}>,<o_1,\ldots,o_{n-1}>,q_n) \cdot 1 \\
&= 1. \blacksquare
\end{aligned}
$$

Def. 2.23 (Ausgabefunktion mit Zuständen)
Eine Ausgabefunktion mit Zuständen ist eine Funktion $\mathcal{O}$ mit:

$$\mathcal{O} : \bigcup_{t \in T|} O_{Out[t+1]} \times O_{Out[1,t]} \times I_{In[1,t]} \to [0,1]$$

$$
\begin{aligned}
&\mathcal{O}'(o_{n+1}| <i_1,\ldots,i_n>, <o_1,\ldots,o_n>) := \\
&\quad \sum_{q_n,q_{n+1}} q(<i_1,\ldots,i_n>,<o_1,\ldots,o_n>,q_n) \cdot \mathcal{T}(i_n,q_n,o_{n+1},q_{n+1}).
\end{aligned}
$$

Lemma 2.6 (Korrektheit) *Die Ausgabefunktion mit Zuständen gemäß Definition 2.23 ist eine Ausgabefunktion gemäß Definition 2.7.*

Beweis:

Es gilt:

$$
\begin{aligned}
&\sum_{o_{t+1} \in O_{Out[t+1]}} \mathcal{O}'(o_{t+1}| <i_1,\ldots,i_t>,<o_1,\ldots,o_t>) = \\
&= \sum_{q_n,q_{n+1}} q(<i_1,\ldots,i_t>,<o_1,\ldots,o_t>,q_n) \cdot \mathcal{T}(i_t,q_n,o_{t+1},q_{n+1}) \\
&= \sum_{q_n} q(<i_1,\ldots,i_t>,<o_1,\ldots,o_t>,q_n) \cdot \sum_{q_n,\,q_{n+1}} \mathcal{T}(i_t,q_n,o_{t+1},q_{n+1}) \\
&= 1 \cdot 1. \blacksquare
\end{aligned}
$$

Lemma 2.7 (Vollständigkeit) *Zu jeder Ausgabefunktion (Def. 2.7) existiert eine äquivalente Ausgabefunktion mit Zuständen (Def. 2.23).*

Beweisidee: Da $\bigcup_t I_{\mathbb{C}[1,t]} \times O_{\mathbb{C}[1,t]}$ abzählbar ist, kann es als Zustandsmenge aufgefaßt werden. Aus $\mathcal{O}$ läßt sich hierdurch direkt eine Funktion $\mathcal{O}'$ konstruieren.

2.5 Ereignisse und Wahrscheinlichkeiten

In diesem Abschnitt wollen wir unter anderem Begriffe wie „Ereignis" und „Wahrscheinlichkeit von Ereignissen" für das vorgeschlagene Systemmodell definieren. Grundlage ist eine σ–Algebra über $S_{\mathbb{C}[\mathcal{T}|]}$.

2.5.1 Ereignisse

Im folgenden werden $s_A(S)$, $i_A(S)$ u.ä. mit s_A und i_A abgekürzt, da es sich in der Regel um ein festes System handelt.

Def. 2.24 (Ereignis mit Zeitbegrenzung) *Das zu einem Ablaufprotokoll beschränkter Länge einer Instanz gehörige Ereignis mit Zeitbegrenzung $[\![s_{A[1,t]}]\!]$ ist wie folgt definiert:*

$$[\![s_{A[1,t]}]\!] := \{s | s \in S_{\mathbb{C}[1,t]}, s\lceil A = s_{A[1,t]}\}.$$

Diese Ereignisräume werden nun zusammengefasst:

Def. 2.25 (Ereignis zu Ablaufprotokoll) *Das zu einem Ablaufprotokoll einer Instanz A gehörige Ereignis $[\![s_A]\!]$ ist wie folgt definiert:*

$$[\![s_A]\!] := \{s | s \in S_{\mathbb{C}[\mathcal{T}|]}, s\lceil A = s_A\}$$

Ein Ereignis zu einem Ablaufprotokoll s_A einer Instanz sind alle (unendlichen) Ablaufprotokolle des Gesamtsystems, die aus Sicht der Instanz nicht unterscheidbar sind, d.h. bei denen die Ein– und Ausgaben der Instanz mit s_A übereinstimmen. Es gilt wie bisher, daß $[\![s_A]\!]$ der Schnitt der $[\![s_{A[1,t]}]\!]$ ist.

Es gilt außerdem: $\emptyset \subseteq E \subseteq E\lceil A \subseteq S_{\mathbb{C}[\mathcal{T}|]}$, da eine eingeschränkte Sicht mehr Ablaufmöglichkeiten offenläßt.

Beispiel 2.10 *Ein Ereignis zum Addiererbeispiel (Def. 2.6):*

$$[\![s\lceil\{(c,1)\} = 5]\!] = <(0,5,5),(1,4,5),(2,3,5),(3,2,5),(4,1,5),(5,0,5)>.$$

Def. 2.26 (Ereignisraum) *Der Ereignisraum $I\!E(S)$ zu einem gegebenen System ist die kleinste Teilmenge der Potenzmenge von $S_{\mathbb{C}[\mathcal{T}]}$, die das zugehörige Ereignis zu jedem endlichen Ablaufprotokoll enthält und abgeschlossen unter Komplement und abzählbarer Vereinigung ist.*

Def. 2.27 (Sichtoperator für Ereignisse) *Sei E ein Ereignis und A eine Instanz, so gilt:*

$$E\lceil A := \bigcup_{s \in E} [\![s\lceil A]\!]$$

Nun wird kurz gezeigt, daß $I\!E$ tatsächlich eine σ–Algebra ist:

Lemma 2.8 *$I\!E(S)$ ist eine σ–Algebra über $S_{\mathbb{C}[\mathcal{T}]}(S)$.*

Zu zeigen [Hin85, Bau78, S.16]:

1. Abgeschlossenheit unter Komplement
2. Abgeschlossenheit unter abzählbarer Vereinigung
3. $\emptyset \in I\!E(S)$

Da die ersten zwei Punkte bereits nach Definition gelten, ist nur die dritte zu zeigen.

Es gilt:

$$[\![<>,<>]\!] = S_{\mathbb{C}[\mathcal{T}]} \quad \text{und} \quad [\![<>,<>]\!] \in I\!E(S)$$

Also folgt mit Punkt 1:

$$\emptyset = S^{C}_{\mathbb{C}[\mathcal{T}]} \in I\!E(S). \blacksquare$$

Der zur σ–Algebra gehörige Meßraum ist $(S_{\mathbb{C}[\mathcal{T}]}, I\!E(S))$. Zu gegebenem $\mathcal{I}$ wird nun ein Wahrscheinlichkeitsmaß $P_{\mathcal{I}}$ auf $I\!E(S)$ definiert. Man erhält somit den Wahrscheinlichkeitraum $(S_{\mathbb{C}[\mathcal{T}]}, I\!E(S), P_{\mathcal{I}})$.

Anmerkung 2.13 *Die Systeme einer Systemfamilie nach Definition 2.10 besitzen identische Ereignisräume und unterscheiden sich durch unterschiedliche Wahrscheinlichkeitsräume.*

2.5.2 Ein Wahrscheinlichkeitsmaß auf $\mathbb{E}(S)$

Die Wahrscheinlichkeit endlicher Systemabläufe läßt sich aus einer Ein– und Ausgabefunktion wie folgt herleiten:

Def. 2.28 (Wahrscheinlichkeit eines endlichen Protokolls) *Sei Σ ein System mit der Ausgabefunktion $\mathcal{O}$ und $\mathcal{I}$ die Eingabefunktion der Umgebung, so ist die Wahrscheinlichkeit eines endlichen Ablaufs induktiv über seine Länge definiert:*

$$P_{\mathcal{I}} : \mathbb{E} \to [0,1]$$

$$P_{\mathcal{I}}([\![<>,<>]\!]) = 1$$

$$\begin{aligned} P_{\mathcal{I}}([\![< i_1, i_2, \ldots, i_{n+1} >, < o_1, o_2, \ldots, o_{n+1} >]\!]) :=& \\ P_{\mathcal{I}}([\![< i_1, i_2, \ldots, i_n >, < o_1, o_2, \ldots, o_n >]\!]) \cdot& \\ \mathcal{I}(i_{n+1} | < i_1, i_2, \ldots, i_n >, < o_1, o_2, \ldots, o_n >) \cdot& \\ \mathcal{O}(o_{n+1} | < i_1, i_2, \ldots, i_n >, < o_1, o_2, \ldots, o_n >).& \end{aligned}$$

Die Wahrscheinlichkeiten für negierte und zusammengesetzte Ereignisse werden wie folgt definiert:

$$P_{\mathcal{I}}(\neg A) := 1 - P_{\mathcal{I}}(A),$$

Sei $(A_i)_{i \in I}$ eine Folge von disjunkten Ereignissen:

$$P_{\mathcal{I}}(\bigcup_{i \in I} A_i) := \sum_{i \in I} P_{\mathcal{I}}(A_i).$$

Anmerkung 2.14 *Wie weiter oben bereits angemerkt, erfolgt erst eine Eingabe und dann eine Ausgabe.*

Anmerkung 2.15 (Abgeschlossene Systeme) *Ist ein System abgeschlossen (Def. 2.9), so ist der Wahrscheinlichkeitsraum bereits durch $\mathcal{O}$ bestimmt.*

Lemma 2.9 *$P_{\mathcal{I}}$ ist ein Wahrscheinlichkeitsmaß auf $\mathbb{E}(S)$ und somit*

$$(S_{\mathbb{C}}[\mathbb{T}](S), \mathbb{E}(S), P_{\mathcal{I}})$$

ein Wahrscheinlichkeitsraum.

Hierzu muß nur noch gezeigt werden, daß $P_{\mathcal{I}}$ ein Wahrscheinlichkeitsmaß definiert.

Dies folgt mit dem Übergangswahrscheinlichkeitsmaß $\mathcal{U}$ direkt aus dem Satz von Ionescu/Tulcea (Satz 20.12 [Hin85, S. 134]):

$$\begin{gathered}\mathcal{U}(i_{n+1}, o_{n+1} | < i_1, i_2, \ldots, i_n >, < o_1, o_2, \ldots, o_n >) := \\ \mathcal{I}(i_{n+1} | < i_1, i_2, \ldots, i_n >, < o_1, o_2, \ldots, o_n >) \cdot \\ \mathcal{O}(o_{n+1} | < i_1, i_2, \ldots, i_n >, < o_1, o_2, \ldots, o_n >).\end{gathered}$$

Dies ist ein Übergangswahrscheinlichkeitsmaß [Hin85, S. 128]. Dies folgt direkt aus der Definition der $\mathcal{I}$ und $\mathcal{O}$ und aus der Definition des Ereignisbegriffs. (Das Urbild einer Ein– bzw. Ausgabe zum Zeitpunkit t ist ein Ereignis.)

2.5.3 Weitere Ereignisse und Wahrscheinlichkeiten

Als Hilfsmittel für die Definitionen des nächsten Abschnitts benötigen wir noch weitere Ereignisbegriffe und Wahrscheinlichkeiten.

Der Ereignisraum und eine Schnittstelle $S =< In, Out, \Gamma, \mathbb{T} >$ incl. den oben angegebenen Bezeichnungen sei gegeben.

2.5.3.1 Bedingte Wahrscheinlichkeit

Eine bedingte Wahrscheinlichkeit ist wie folgt definiert:

Def. 2.29 (Bedingte Wahrscheinlichkeit) *Seien $A, B \subseteq \mathbb{E}$ Ereignisse, so gilt:*

$$P(A|B) := \frac{P(A \cap B)}{P(B)} \quad mit\ \frac{0}{0} := 0$$

2.5.3.2 Prädikate und Ereignisse hierzu

Um Eigenschaften von Ereignissen überprüfen zu können, werden Prädikate auf Ereignisse übertragen:

Def. 2.30 (Anwendung eines Prädikats auf ein Ereignis)
Sei $Pred : S_{\mathbb{C}[T]} \rightarrow \{True, False\}$ *ein Prädikat auf Abläufen mit abzählbarem Gültigkeitsbereich, so gilt:*

$$[\![Pred]\!] := \{s | s \in S_{\mathbb{C}[T]} \wedge Pred(s)\}.$$

$[\![Pred]\!]$ beschreibt die Menge aller Ablaufprotokolle eines Schnittstelle, für die das Prädikat gültig ist.

Lemma 2.10 *Für zwei Prädikate* A *und* B *gilt:*

$$A \rightarrow B \Rightarrow [\![A]\!] \subseteq [\![B]\!]$$

Beweis: Ist $s \in [\![A]\!]$, so gilt A und somit B; also gilt auch $s \in [\![B]\!]$. ■

2.5.3.3 Temporale Ereignisse

Da für die Beschreibung von Schnittstellenabläufen auch Aussagen über zeitliche Abfolgen benötigt werden, werden in diesem Abschnitt zeitliche Eigenschaften von Ereignissen definiert.

Im folgenden werden zeitlicher Abstand von Ereignissen definiert, um das zeitlich polynomielle Auseinanderliegen von zu formalisieren (vgl. [MP91, Kap. 3]).

Idee der Definition ist, daß es einen Zeitraum von t bis t' gibt, so daß beide Ereignisse vollständig in diesem Zeitraum liegen.

Def. 2.31 (Zeitlicher Abstand von Ereignissen) *Seien* $A, B \in \mathbb{E}(S)$ *Ereignisse und* $n \in \mathbb{N}$ *eine Zahl, so gilt*

$$\begin{aligned} A \sim_n B :=& \\ & \underset{t,t' \in T}{\exists} 0 \leq t \leq t' \leq t + n : \quad [\![A \lceil \mathbb{C} \times \{t, \ldots, t'\}]\!] = A \wedge \\ & \qquad\qquad\qquad\qquad\qquad [\![B \lceil \mathbb{C} \times \{t, \ldots, t'\}]\!] = B \end{aligned}$$

Def. 2.32 (Zeitliche Implikation von Ereignissen) *Seien $A, B \in I\!E(S)$ Ereignisse und $n \in I\!N$ eine Zahl, so gilt*

$$A \rightarrow_n B := A \sim_n B \wedge A \subseteq B$$

$$A \leftarrow_n B := A \sim_n B \wedge B \subseteq A$$

Die folgende Funktion gibt einen Zeitraum an, in dem ein Ereignis geschieht:

Def. 2.33 (Zeitraum) *Sei A Ereignis, so ist der Zeitraum von A wie folgt definiert:*

$$\begin{array}{cl} time(A) := & \{t_i, t_i + 1, \ldots, t_j - 1, t_j\} \\ \Longleftrightarrow & [\![A \lceil \mathbb{C} \times \{t_i, \ldots, t_j\}]\!] = A \\ \wedge & [\![A \lceil \mathbb{C} \times \{t_i + 1, \ldots, t_j\}]\!] \neq A \\ \wedge & [\![A \lceil \mathbb{C} \times \{t_i, \ldots, t_j - 1\}]\!] \neq A \end{array}$$

Analog wird auch ein Zeitbegriff für Instanzen eingeführt:

$$time(A) := \{t | (c, t) \in A, t \in \mathbb{T}\}$$

Kapitel 3

Kryptosystemmodell

In diesem Abschnitt soll der stochastische Systembegriff aus Kapitel 2 auf Kryptosysteme angewendet werden. Es werden zuerst das Kryptosystem und die Angreifer– und Benutzerteilsysteme erläutert und anschließend zu einem Gesamtsystem integriert.

Benutzer und Angreifer sind Systeme mit den Schnittstellenkanälen als Eingabe, zwei zusätzlichen Kanälen zur Kommunikation zwischen Angreifer und Benutzer und einem Kanal zur Ergebnisausgabe durch den Angreifer.

Die Struktur des Gesamtsystems zeigt Abbildung 3.1. Das so konstruierte System besitzt keinen Eingabekanal. Es ist also abgeschlossen (Def 2.9).

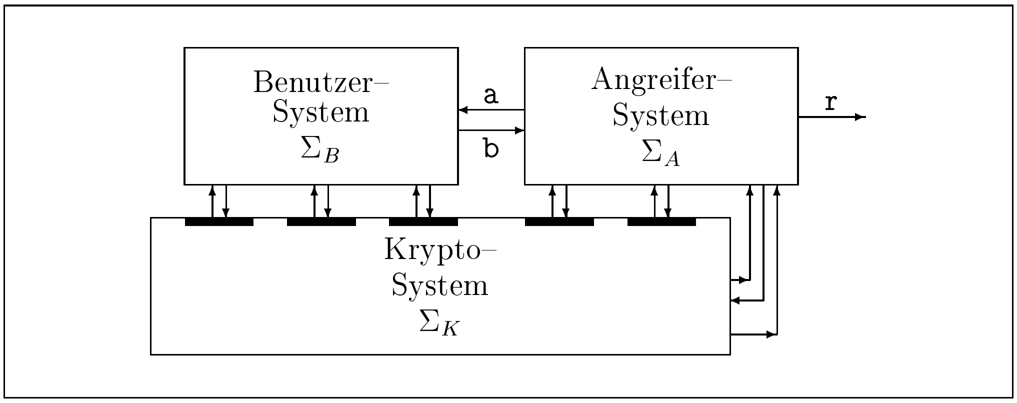

Abbildung 3.1: Systemstruktur eines aus Kryptosystem, Angreifer und Benutzer zusammengesetzten Systems

3.1 Kryptosystem

Das hier beschriebene Kryptosystemmodell besteht aus einer Schnittstelle, die die Verbindung zur Umgebung modelliert und einer Ausgabefunktion, die die Reaktion des Systems auf die Umgebung zusammenfaßt. Ein Kryptosystem wird hier als „Black Box“ betrachtet, d.h. der interne Aufbau wird nicht genauer modelliert.

Eine Teilmenge der Kanalmenge des Systems sind die Kanäle U der regulären Benutzerschnitstelle. Über diese Kanäle soll die „Bedienung“ des Systems erfolgen. Beispiel hierfür wären der Ein– und Ausgabekanal eines Konzelationssystems. Der Angreifer kann natürlich auch Teile dieser Benutzerschnittstelle kontrollieren (z.B. bei einem Netzwerk, bei dem einzelne Knoten vom Angreifer kontrolliert sein können).

Bei Kanälen des Systems, die nicht Teil der Benutzerschnittstelle sind, handelt es sich in der Regel um „illegale“ Kanäle, wie abgehörte bzw. angezapfte Leitungen.

Def. 3.1 (Kryptosystem) *Ein Kryptosystem ist ein Paar* (Σ_K, U) *wobei* $\Sigma_K = (S, \mathcal{O})$ *Systemfamilie,* $S = (In, Out, \Gamma, \mathbb{T})$ *Schnittstelle und* $U \subseteq In \cup Out$ *ist.* U *wird als Benutzerschnittstelle bezeichnet. O.B.d.A. sei* $(In \cup Out) \cap \{a, b, r\} = \emptyset$.

3.2 Benutzer

Der Benutzer eines Kryptosystems* wird hier als Teilsystem zur Benutzerinstanz modelliert. Im Gegensatz zum Begriff des Angreifers, wird dieser Begriff in der Literatur in dieser allgemeinen Form bisher nicht verwendet. Im folgenden wird Angreifer– und Benutzerteilsystem auch kurz als „Benutzer“ bzw. „Angreifer“ bezeichnet.

3.2.1 Benutzerinstanz

Die Benutzerinstanz $B \subseteq U \times \mathbb{T}$ zu einem Kryptosystem (Σ_K, U) gibt an, welche Kanäle der Benutzerschnittstelle vom Benutzer besetzt[1] sind. In der

[1] D.h. der Benutzer wählt die Eingaben und erhält die Ausgaben des Kanals

Regel wird die Instanz B für ein $C_B \subseteq U$ die Form $C_B \times \mathbb{T}$ haben d.h. daß der Benutzer kontrolliert die ihm zugeordneten Kanäle zu allen Zeitpunkten.

Abb. 3.2 veranschaulicht die Benutzerinstanz an einem Beispiel.

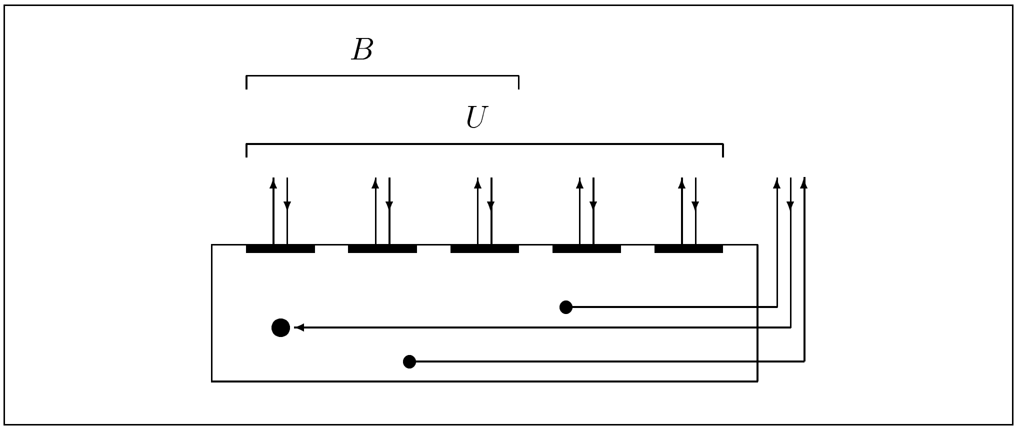

Abbildung 3.2: Benutzerinstanz eines Kryptosystems

3.2.2 Benutzerklassen

3.2.2.1 Vorbemerkungen

In diesem Abschnitt werden Benutzer nach Ihrer Angreifbarkeit klassifiziert werden. Es werden einige grundlegende Klassen formal definiert, die sich durch ihre Angreifbarkeit, d.h. dem Grad der Kooperation mit dem Angreifer über die Kanäle `a`[2] und `b`[3], unterscheiden. „Angriffsarten" wurden ursprünglich für Konzelationssysteme definiert und werden hier auf beliebige Systeme verallgemeinert. (Spezielle Klassen für Konzelationssysteme sind in Kapitel 6.4 ab Seite 106 angegeben.) Eine derart allgemeine Auffassung von „Angriffsarten" ist neu. Bisher wurden „Angriffsarten" nur für einzelne, nichtreaktive Systeme angegeben. Die hier angegebene Definition für beliebige reaktive Systeme ist die erste ihrer Art.

Eine Modellierung der Angriffsarten als Anforderungen an den Angreifer erscheint mir nicht sinnvoll. Es ist unrealistisch von einem sich zurückhaltenden Angreifer, der nicht alle seine Möglichkeiten voll ausschöpft, auszugehen.

[2] Dies ist der Kanal vom Angreifer zum Benutzer

[3] Dies ist der Kanal vom Benutzer zum Angreifer

Die Einteilung nach Benutzerklassen ist orthogonal zur Entscheidung, ob Benutzer informationstheoretische oder komplexitätstheoretische Systeme sein sollen (Abschnitt 3.2.2.2). Sowohl die Menge der informationstheoretischen Benutzer, als auch die Menge der komplexitätstheoretischen Benutzer enthält die definierten Benutzerklassen als Unterklassen. Eine komplexitätstheoretische Benutzerklasse ist Unterklasse der informationstheoretischen.

Im folgenden wird zu den meisten Klassen je ein positives und ein negatives Prädikat angegeben. Das positive Prädikat garantiert die Möglichkeit für diese Angriffsart, das negative verhindert die jeweilige Angriffsart.

Bezeichnungen der letzten Abschnitte, wie z.B. der Begriff der Schnittstelle mit ihren Bezeichnungen, werden im folgenden vorausgesetzt.

Def. 3.2 (Benutzerklasse zu einer Formel) *Sei F eine der im folgenden angegebenen Formel, $\mathcal{C}$ eine der zwei angegebenen Grundklassen $\mathcal{P}$ oder $\mathcal{F}$. So ist ein $\mathcal{O}_B$ in $\mathcal{C}_F$ gdw.*

$$\underset{\mathcal{I}}{\forall} F$$

wobei die enthaltenen Wahrscheinlichkeiten zum System $(S_B, \mathcal{O}'_B)$ mit der Eingabefunktion $\mathcal{I}$ gebildet werden und $\mathcal{O}'_B$ die Vervollständigung der Benutzerausgabefunktion $\mathcal{O}_B$ nach Lemma 2.2 ist. Die Schnittstellenbezeichnungen Out, In und $\mathbb{T}$ in den Formeln beziehen sich auf die Schnittstelle S_B des Benutzerteilsystems.

Idee dieser Definition ist, daß die Formel trotz beliebigem Umgebungsverhalten (d.h Angreifer) gilt.

3.2.2.2 Komplexitäts– oder informationstheoretisch

Die Menge der Benutzer wird in Teilsysteme mit unbegrenzter Rechenleistung und Teilsysteme mit polynomiell zeitbeschränkter Rechenleistung unterteilt.

Eine komplexitätstheoretische Aussage ist für ein einzelnes festes System nicht sinnvoll. Komplexitätstheoretische Aussagen werden daher immer bezüglich eines Sicherheitsparameters n gemacht. Wir betrachten daher Klassen von Systemen. Die Systeme Σ_n einer Systemklasse unterscheiden sich nur durch ihren Sicherheitsparameter n (vgl. Def. 2.10).

Informationstheoretische Systeme wurden hier analog mit einem Sicherheitsparameter modelliert, um den allgemeinsten Fall zu erfassen.

Informationstheoretische Benutzer unterscheiden sich von komplexitätstheoretischen Benutzern durch erweiterte Handlungsfähigkeit. Dies wirkt sich vor allem dann aus, wenn das Benutzerverhalten nicht eingeschränkt ist. In diesem Fall darf beliebige Benutzeraktion die Geheimhaltung nicht gefährden.

$\mathcal{P}$ bezeichne die Menge der polynomiell beschränkten universellen Ausgabefunktionen und $\mathcal{F}$ die Menge aller Ausgabefunktionen.

3.2.2.3 Benutzer führt alle Anweisungen aus

Die Benutzer der folgenden Klasse müssen beliebige Anweisungen der Angreifer ausführen.

Idee dieser Definition ist, daß der Angreifer über den Kanal `a` ein Ausgabeablaufprotokoll (Anweisung) an den Benutzer sendet. Dieser führt das Ablaufprotokoll dann aus. Anweisungen für die Vergangenheit müssen (und können) nicht ausgeführt werden.

Die Definition schließt adaptiven Angriff mit ein: der Angreifer kann immer wenn der Benutzer keine Anweisung abarbeitet, eine Anweisung senden.

Def. 3.3 (Gewählte Eingabe) *Sei $cs \in O_B$ ein endliches Ausgabeablaufprotokoll.*

$$\begin{aligned}
&\texttt{chosen-input} \iff \\
\underset{t}{\forall} \underset{cs \in O_B}{\forall} \quad &(min(time([\![cs]\!])) > t \wedge \\
&i_{a,t} = \texttt{do}\ cs \wedge \\
&\underset{t \in time([\![cs]\!])}{\forall} i_{a,t} = \texttt{nop}\ ) \rightarrow \\
&P([\![cs]\!]) = 1.
\end{aligned}$$

Anmerkung 3.1 *Dadurch, daß der Angreifer während dem Ablauf des angeforderten Eingabeablaufprotokolls keine weiteren Anweisungen geben darf, wird verhindert, daß der Angreifer dem Benutzer widersprüchliche Anweisungen gibt oder daß der Angreifer den Benutzer durch andauerndes Senden von langen Anweisungen überfordert.*

Die Einschränkung, daß time(cs) nach t beginnt, verhindert, daß der Angreifer Anweisungen rückwirkend gibt (die Formel wäre ohne diese Zusatzbedingung unerfüllbar).

Idee der folgenden Definition ist, daß der Benutzer seine Eingaben stochastisch unabhängig von den Anweisungen des Angreifers wählt.

Durch dieses Prädikat wird nicht ausgeschlossen, daß der Angreifer durch interne Kanäle Befehle an den Benutzer sendet.

Def. 3.4 (Keine gewählte Eingabe)

$$
\begin{aligned}
& \texttt{no-chosen-input} \iff \\
& \qquad \underset{s \in S_{\llbracket C[T] \rrbracket}}{\forall} \; \underset{t}{\forall} P(\llbracket s \lceil (B \cap Out[t+1]) \rrbracket | \llbracket s \lceil \{a\}[1,t] \rrbracket) \\
= & \quad P(\llbracket s \lceil (B \cap Out[t+1]) \rrbracket).
\end{aligned}
$$

Anmerkung 3.2 *Das negative Prädikat ist nicht die Negation des positiven! Es gibt also Benutzer, die etwas tun können, ohne es zu müssen...*

Anmerkung 3.3 *In der Praxis sind diese Benutzer Systeme, die die Eingabe aus* `a` *nicht verwerten. Das Prädikat* $\underset{t}{\forall} a_t = \texttt{nop}$ *garantiert somit* `no-chosen-input`.

3.2.2.4 Benutzer teilt nichtgeheime Ein-/Ausgaben mit

Das Prädikat, welches hier angegeben wird, soll den Benutzer *zwingen*, alle vom Angreifer gewünschten Beobachtungen mitzuteilen. Wird diese Forderung nicht eingeschränkt, so ist keine Geheimhaltung mehr möglich. Daher werden hier nur Ergebnisse zurückgegeben, die nicht zum geheimen Ereignis gehören.

Dies ist die Nachbildung eines gewählten–Schlüsseltext Angriffs. Da in unserem Systemmodell keine „Schlüsseltexte“ existieren, kann der Angreifer beliebige Aktionen innerhalb des Systems durchführen und sich dann (die daraus resultierenden) Beobachtungen vom Benutzer mitteilen lassen.

Die Operationsmenge einer Ereignismenge[4] ist wie folgt definiert:

$$Op(\mathcal{E}) : \mathcal{P}(\mathbb{E}) \to OP$$

$$Op(\mathcal{E}) := \bigcup_{t \in \mathbb{T}^{\dagger}; E \in \mathcal{E}; [\![E \lceil \mathbb{C}^{\dagger}[t]]\!] \neq \mathbb{E}} \{\texttt{op} \mid c \in \mathbb{C}; (c, t, \texttt{op}) \in E\}.$$

Die folgende Definition ist somit nicht unabhängig von der Ereignismenge $\mathcal{E}$. Die Idee dieser Definition ist, daß der Angreifer dem Benutzer mitteilt, welche Ausgaben er wissen will (Kanäle und Zeitpunkte). Der Benutzer wartet, bis der letzte Zeitpunkt vorbei ist und sendet dann seine Beobachtungen an den Angreifer. War eine geheimzuhaltende Operation dabei, so teilt er das Ergebnis nicht mit.

Def. 3.5 (Gewählte Ausgabe)

$$\texttt{chosen-output}(\mathcal{E}) \iff \forall_{s \in S_{\mathbb{C}^{\dagger}[\mathbb{T}^{\dagger}]}} \forall_{t \in \mathbb{T}^{\dagger}} \forall_{B' \subseteq B} (i_{a,t} = \texttt{get } B'$$
$$\wedge \forall_{t \in time(B')} i_{a,t} = \texttt{nop})$$
$$\to b_{max(time(B'))+1} = \texttt{receive } s\lceil(B' - \{a, b\} \times \mathbb{T}) - \mathbb{C} \times \mathbb{T} \times Op(\mathcal{E}).$$

Anmerkung 3.4 *Hier sind auch andere Foramlisierungen denkbar. Der Benutzer könnte z.B. alle gewünschten Ausgaben, außer den Ausgaben zu Zeitpunkten aus time($\mathcal{E}$), mitteilen. Ebenso „$(c, t) \in \mathcal{E}$". Dies muß eventuell von der Art des Ereignisses abhängig gemacht werden: Soll der Zeitpunkt der Einzelereignisse geheimgehalten werden, so ist die Zeitvariante sinnvoller als die Operationsvariante. Bei geheimen Nachrichteninhalten ist die Operationsvariante vorzuziehen.*

Anmerkung 3.5 *Gilt $Op(\mathcal{E}) = \Gamma^+$ so ist diese Definition nicht mehr anwendbar, da dann alle Operationen geheim sind.*

Dies wäre zum Beispiel bei $\Gamma = \{0, 1, \ldots, 9\}$ und

$$\{[\![i_{a,1} \; gerade]\!], [\![i_{a,1} \; ungerade]\!]\}$$

der Fall.

[4] Einige Ereignisse, die ununterscheidbar, d.h. geheim, sein sollen.

Die folgende Definition sagt aus, daß die Mitteilungen des Benutzers an den Angreifer unabhängig von den Beobachtungen des Benutzers sein müssen:

Def. 3.6 (Keine gewählte Ausgabe)

$$\texttt{no-chosen-output} \iff$$
$$\forall_{s \in S_{\mathbb{C}[\mathbb{T}]}} \quad P([\![s\lceil(\{b\} \times \mathbb{T})]\!] | [\![s\lceil(B \cap In[\mathbb{T}] - \{a,b\} \times \mathbb{T})]\!]) = P([\![s\lceil\{b\} \times \mathbb{T}]\!]).$$

Anmerkung 3.6 *Ein Benutzer, der dem Angreifer nichts mitteilt, erfüllt diese Forderung.*

Das Prädikat $\forall_t \, o_{b,t} = \texttt{nop}$ *garantiert also* `no-chosen-output`.

Anmerkung 3.7 `no-chosen-output` *kann analog auch getrennt für die Beobachtbarkeit der Ein– und Ausgabekanäle definiert werden.*

3.2.2.5 Keine Möglichkeit zu adaptivem Angriff

Bei den bisherigen Definitionen wurde die Anzahl der Anweisungen vom Angreifer an den Benutzer nicht eingeschränkt. Sie definieren daher aktive Angriffe. Zur Einschränkung der bisherigen Definitionen dient die folgende Definition, die dem Angreifer nur eine Anweisung bzw. Anfrage erlaubt. Da dies aber eine Forderung an den Angreifer wäre, wird hier gefordert, daß das Benutzerverhalten unabhängig von jeder weiteren Anweisung des Angreifers ist.

Def. 3.7 (Nicht-adaptiv)

$$\begin{aligned} \texttt{non-adaptive} \iff & \forall_{s \in S_{\mathbb{C}[\mathbb{T}]}} \\ & (\exists_{t_1 < t_2} i_{a,t_1} \neq \texttt{nop} \ \wedge i_{a,t_2} \neq \texttt{nop} \ ) \\ \rightarrow & \forall_{t > t_2} P([\![s\lceil B[t]]\!] | [\![s\lceil\{a\}[t_2]]\!]) = P([\![s\lceil B[t]]\!]). \end{aligned}$$

Diese Definition sagt aus, daß der Benutzer nur auf die erste Nachricht über `a` reagiert d.h. daß seine Handlungen stochastisch unabhängig von allen anderen Nachrichten sind.

3.2.2.6 Drei extreme Benutzerklassen

Das folgende Prädikat ist das Prädikat zur Menge aller Benutzer:

Def. 3.8 (Menge aller Benutzer)

$$\mathtt{all} \iff True.$$

Anmerkung 3.8 (Das Geheimnis darf verraten werden!) *In der Menge aller Benutzer sind auch die Benutzer enthalten, die voll mit dem Angreifer kooperieren. Das heißt, daß sie auch das geheime Ereignis mitteilen dürfen. Dies ist nur für einen später vorgestellten Spezialfall sinnvoll, der keine Verbindung zwischen Angreifer und Benutzer vorsieht („absoluter Angriff“).*

Der folgende Benutzer befolgt uneingeschränkt alle Aufträge des Angreifers.

Def. 3.9 (Total kooperativer Benutzer)

$$\mathtt{universal}(\mathcal{E}) \iff \mathtt{chosen\text{-}input} \wedge \mathtt{chosen\text{-}output}(\mathcal{E}).$$

Der folgende Benutzer kooperiert nicht mit dem Angreifer (die Bezeichnung „passive“ stammt vom passiven Angreifer* eines Konzelationssystems):

Def. 3.10 (Passiver Benutzer)

$$\mathtt{passive} \iff \mathtt{no\text{-}chosen\text{-}input} \wedge \mathtt{no\text{-}chosen\text{-}output}$$

3.2.2.7 Zeitschranken

Mit diesem Prädikat soll definiert werden, daß der Benutzer nur an bestimmten Zeitpunkten (in der Regel ein Zeitraum) auf Eingaben des Angreifers reagiert und Informationen an den Angreifer weiterleitet.

Def. 3.11 (Zeitbeschränkter Benutzer)

$$\texttt{time-constrained}(\mathbb{T}') \iff$$
$$P([\![s\lceil(B \cap Out[\mathbb{T}])]\!] \mid [\![s\lceil[\{a,b\} \times (\mathbb{T}/\mathbb{T}')]]\!]) = P([\![s\lceil(B \cap Out[\mathbb{T}])]\!]).$$

Anmerkung 3.9 *Die hier angegebenen Formeln sollen nur eine Anregung geben und sind nicht vollständig. Es könnten z.B. weitere Klassen definiert werden:*

- *Benutzer, die nur vor dem geheimen Ereignis Anfragen zulassen;*
- *Benutzer, die gleichverteilt ein Ereignis der geheimen Ereignismenge auswählen.*

3.2.3 Benutzerdefinition

Ein Benutzer wird durch eine Benutzerklasse und seine Instanz charakterisiert:

Def. 3.12 (Benutzer) *Sei ein Kryptosystem* (Σ_K, U)*, eine Benutzerklasse* $\mathcal{C}_F$ *und eine Instanz*

$$B \subseteq U \times \mathbb{T}$$

gegeben, so heißt

$$(\mathcal{C}_F, B)$$

Benutzer der Klasse $\mathcal{C}_F$ *zum Kryptosystem* (Σ_K, U) *und zur Instanz* B.

3.2.4 Benutzersysteme

Es wird nun die Klasse der Benutzersysteme zu einem Benutzer angegeben. Die vorgegebene Benutzerinstanz eines Kryptosystems wird um die Instanz $\{a, b\} \times \mathbb{T}$ erweitert, um später direkte Kommunikation mit dem Angreifer zu ermöglichen (vgl. Abb. 3.1).

Def. 3.13 (Benutzersysteme) *Sei $(\mathcal{C}_F, B)$ Benutzer der Klasse $\mathcal{C}_F$ zum Kryptosystem (Σ_K, U) und zur Instanz B, so ist die Menge der Benutzersysteme hierzu wie folgt definiert:*

$$\begin{aligned}&\mathcal{B}((\Sigma_K, U), (\mathcal{C}_F, B)) := \\ &\qquad \{(S, B', \mathcal{O}_{B'}) | \mathcal{O}_{B'} \in \mathcal{C}_B; (S_{B'}, B', \mathcal{O}_{B'}) \quad \textit{ist Teilsystemfamilie}\}\end{aligned}$$

mit

$$B' := (\{a, b\} \times \mathbb{T}) \cup B$$

und

$$S_{B'} = (Out \cup \{a\}, In \cup \{b\}, \Gamma, \mathbb{T}\},$$

wobei Out und In die Kanalmengen des Kryptosystems sind.

Anmerkung 3.10 *Eingabekanäle des Systems sind Ausgabekanäle des Benutzersystems und umgekehrt.*

Anmerkung 3.11 (Namen der Kanäle) *Die gleiche Benennung dient der einfacheren Verschaltung mit der kanonischen Verschaltungsfunktion.*

3.3 Angreifer

3.3.1 Angreiferinstanz

Der hier vorgeschlagene Angreifer ist nur durch seine Rechenkapazität und die ihm zur Verfügung stehenden Angriffskanäle eingeschränkt. Rechenleistung und Verbreitungsgrad überschreitende Anforderungen an den Angreifer sind unrealistisch.

Ein Angreifer zu einem Benutzer hat die Kontrolle über alles, was vom Benutzer nicht kontrolliert wird. Als weiteres Merkmal kann ein Teil A_U des Angreifers als „legal" bzw. geduldet definiert werden. Sinn dieser Aufteilung der Angreiferinstanz in zwei Teilinstanzen ist, daß die Information, die der „legale" Angreiferanteil erhält, von der Geheimhaltungsforderung quasi ausgenommen wird. Hierdurch kann Geheimhaltung für beliebiges Benutzerverhalten auch in Systemen mit Informationsfluß vom Benutzer zum „legalen" Angreifer definiert werden. Die zugrundeliegende Idee ist, daß Informationen, vom Benutzer zum „legalen“ Angreifer, vom Benutzer „freiwillig“ erteilt werden. Eine Folge ist, daß Informationen, die der Angreifer über diese Instanz erfährt, die Geheimhaltung nicht verringern können. Die Angreiferinstanzen werden in Abb. 3.3 verdeutlicht, wobei A_U „legale“ Teilinstanz des Angreifers ist. Sie ist die Restinstanz zur Benutzerinstanz zusammen mit den Kanälen $\{a, b, r\}$, die der Kommunikation mit dem Benutzer und der Ausgabe von Ergebnissen dienen. Ist ein Kanal zu einem Zeitpunkt nicht vom Benutzer besetzt, so wird er vom Angreifer besetzt. Es gilt also:

$$A := \mathbb{C} \times \mathbb{T} - B$$

Der „legale“ Teil des Angreifers, wird mit A_U bezeichnet. Es gilt:

$$A_U \subseteq A.$$

Abb. 3.3 veranschaulicht die Instanzen eines Kryptosystems.

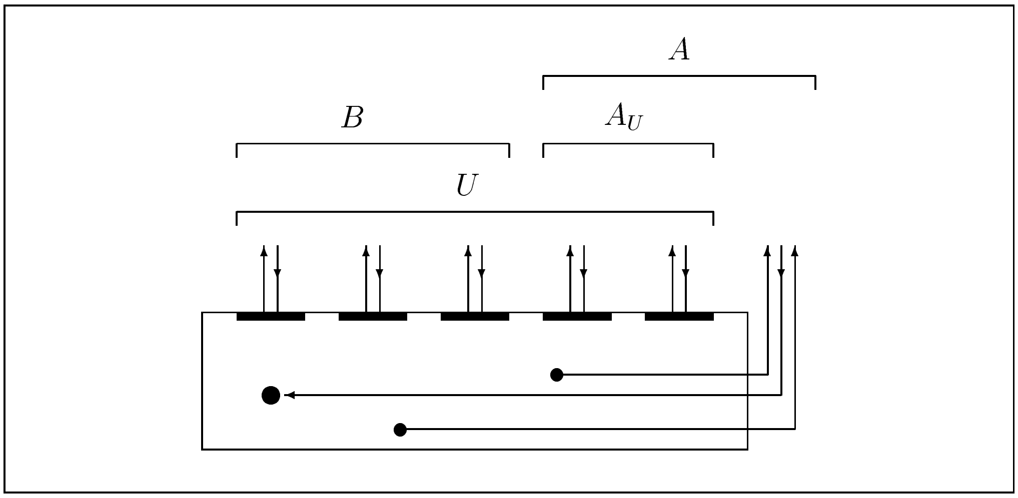

Abbildung 3.3: Instanzen eines Kryptosystems

Das folgende Beispiel soll dies verdeutlichen:

Beispiel 3.1 (Unterschied zwischen A und A_U) *Angreifer A will Informationen über die Firma F sammeln. Er sammelt in Radio und Zeitung. Unter der Voraussetzung, daß die Firma keine Geheimnisse veröffentlicht, ist* *dies die Instanz A_U. Zusammen mit den Informanten innerhalb der Firma erhält man den Angreifer A. Erhält A nicht mehr Information als A_U, so kann man seinen Angriff als gescheitert bezeichnen.*

Wäre in diesem Fall Radio und Zeitung nicht Teil von A_U, so wäre der Angriff bereits geglückt, wenn dort Informationen (wie z.B. Werbung) über die Firma verbreitet würden. □

3.3.2 Komplexitäts– oder informationstheoretisch

Analog zu den Benutzern können Angreifer informations– oder komplexitätstheoretisch sein. Der informationstheoretische Angreifer ist stärker als der komplexitätstheoretische, da er in seiner Rechenleistung nicht eingeschränkt ist.

3.3.2.1 Angreiferdefinition

Größe des legalen Teils und die Rechenkapazität werden zusammen als Angreifer bezeichnet:

Def. 3.14 (Angreifer) *Sei ein Kryptosystem (Σ_K, U) und eine Benutzerinstanz B gegeben. Für*

$$A_U \subseteq (U \times \mathbb{T} - B)$$

heißt $(\mathcal{F}, A_U)$ informationstheoretischer Angreifer,
$(\mathcal{F}_f, A_U)$ informationstheoretischer Angreifer mit Fehler $(f : I\!N \to [0,1])$ und
$(\mathcal{P}, A_U)$ komplexitätstheoretischer Angreifer zu A_U.

Def. 3.15 (Absolute Angreifer)
$(\mathcal{F}, \emptyset)$ heißt informationstheoretisch absoluter Angreifer.
$(\mathcal{P}, \emptyset)$ heißt komplexitätstheoretisch absoluter Angreifer.

Der Angreifer ohne „legale" Teilinstanz, ist ein wichtiger Spezialfall, der in Abschnitt 4.3.3 ausführlich behandelt wird.

3.3.3 Angreifersysteme

Analog zum Benutzersystem kann das Angreifersystem um zwei Kanäle zur direkten Kommunikation zwischen Angreifer und Benutzer erweitert werden. Zusätzlich besitzt das Angreifersystem einen Ausgabekanal r.

Def. 3.16 (Angreifersysteme) *Die Menge der Angreifersysteme $\mathcal{A}$ zu einem Angreifer $(\mathcal{C}, A_U)$, einer Benutzerinstanz B und einem Kryptosystem (Σ_K, U) ist:*

$$\begin{aligned}&\mathcal{A}((\Sigma_K, U), B, (\mathcal{C}, A_U)) := \\ &\quad \{(S_A, A, \mathcal{O}_A) | \mathcal{O}_A \in \mathcal{C}; (S, A, \mathcal{O}_A) \quad \textit{ist Teilsystemfamilie}\}\end{aligned}$$

mit

$$A := (\mathbb{C} \times \mathbb{T} - B) \cup \begin{cases} \{a, b, r\} \times \mathbb{T} & \textit{für } A_U \neq \emptyset \\ \{\quad r\} \times \mathbb{T} & \textit{sonst} \end{cases}$$

und

$$S_A = (Out \cup \{b\}, In \cup \{a, r\}, \Gamma, \mathbb{T}),$$

wobei Out und In die Kanalmengen des Kryptosystems sind.

Def. 3.17 („legale" Angreifersysteme) *Die Menge der „legalen" Angreifersysteme $\mathcal{A}_{A'_U}((\Sigma_K, U), (\mathcal{C}_F, B), (\mathcal{C}, A_U))$ zu einem Angreifer $(\mathcal{C}, A_U)$, einer Benutzerinstanz B und einem Kryptosystem (Σ_K, U) ist:*

$$\begin{aligned}&\mathcal{A}_{A'_U}((\Sigma_K, U), B, (\mathcal{C}, A_U)) := \\ &\quad \{(S, A'_U, \mathcal{O}_{A'_U}) | \mathcal{O}_{A'_U} \in \mathcal{C}; (S_A, A'_U, \mathcal{O}_{A'_U}) \quad \textit{ist Teilsystemfamilie} A'_U\}\end{aligned}$$

mit

$$A'_U = A_U \cup \begin{cases} \{a, b, r\} \times \mathbb{T} & \textit{für } A_U \neq \emptyset \\ \{\quad r\} \times \mathbb{T} & \textit{sonst} \end{cases}$$

und

$$S_A = (Out \cup \{b\}, In \cup \{a, r\}, \Gamma, \mathbb{T}),$$

wobei Out und In die Kanalmengen des Kryptosystems sind.

Durch die hier angegebenen Angreifer wird eine vorgegebene Benutzerfamilie (Def. 3.13) angegriffen.

3.3.4 Angreiferaktionen während dem geheimen Ereignis

Für einen aktiven Angriff*, muß der Angreifer die Möglichkeit haben das System zu „testen“. Wird ein interaktiver Eingriff des Angreifers beim Ablauf des geheimen Ereignisses nicht zugelassen, so führt dies zu einer zu schwachen Definition. Als Beispiel für eine derartige Definition soll die Definition der interaktiven Geheimhaltung bei chosen–ciphertext* bei Konzelationssystemen von Galil, Haber und Yung [GHY91] vorgestellt werden:

Galil et. al. haben vorgeschlagen, dem Angreifer ein System zum Ausprobieren zu geben. Nachdem der Angreifer beliebig probiert hat, wird ein Ereignis ausgewählt und dann dieses Ereignis ungestört ausgeführt (der Angreifer beobachtet). Anhand dieser Beobachtung und anhand der Erfahrungen, die der Angreifer durch das vorherige Testen gemacht hat, soll er schließlich bestimmen, welches der Ereignisse er beobachtet hat. Folgendes Beispiel soll zeigen, daß diese Definition (zumindest für interaktive Systeme) zu schwach ist:

Beispiel 3.2 (Sicheres System nach [GHY91])
Sei G das *perfekte interaktive Konzelationsprotokoll (falls existent). Wir konstruieren ein neues Protokoll, das exakt wie G arbeitet, jedoch bei einer kleinen Übertragungsverzögerung eine Kopie der geheimzuhaltenden Nachricht an den Angreifer sendet.*

Anmerkung 3.12 *Das so konstruierte System ist intuitiv nicht sicher, da eine kleine Übertragungsverzögerung die Geheimhaltung nicht völlig zerstören sollte (es handelt sich primär um einen Angriff auf die Verfügbarkeit* und nicht gegen die Vertraulichkeit*). Verzögert der Angreifer eine Nachricht, so wird ihm umgehend das Geheimnis mitgeteilt.*

Lemma 3.1 *Das System aus Beispiel 3.2 erfüllt die Definition der interaktiven Geheimhaltung aus [GHY91], obwohl es intuitiv nicht sicher ist.*

Beweisskizze: Der Angreifer erfährt durch Verschlüsselung beliebiger Nachrichten vor dem geheimen Ereignis entweder garnichts, da das ursprüngliche System sicher war, oder bei kleineren

Verzögerungen die Testnachricht, die er ohnehin schon wußte (er hat sie gewählt).

Beim ungestörten Ablauf mit einer geheimen und zufällig gewählten Nachricht tritt keine Verzögerung und somit kein Geheimnisverrat auf, da der Angreifer nur beobachtet. Über die geheime Nachricht erhält er somit keinerlei Information.

Wie uns dieses Beispiel gezeigt hat, muß für eine realistische Definition der Angreifer auch während des Ablaufs des geheimzuhaltenden Ereignisses eingreifen können. Dies ist auch realistischer, da kein Angreifer seine Beobachtungen und Korruptionsversuche gerade während dem für ihn interessanten Ereignis unterbricht.

3.4 Zusammensetzen der drei Teilsysteme

Das stochastische Systemmodell aus Kapitel 2 setzt *eine* Ausgabefunktion des Systems voraus. Es wird nun kurz angedeutet, wie ein abgeschlossenes System aus Krypto–, Angreifer– und Benutzersystem zusammengesetzt wird.

Gegeben sind:

- Ein Kryptosystemfamilie (Σ_K, U)
- Eine Benutzersystemfamilie Σ_B
- Eine Angreifersystemfamilie Σ_A

Diese Teilsysteme sollen nun gemäß Abbildung 3.1 auf Seite 39 zu einem Gesamtsystem Σ verbunden werden. Da jedem Eingabekanal eines Teilsystems ein Kanal eines anderen Systems entspricht, kann dies durch das kanonische Zusammenfügen der Teilsysteme erreicht werden.

Es gilt also:

$$\Sigma_n := \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A,n}.$$

Hierdurch werden Kopien aller Verbindungskanäle nach außen geführt. Das so entstandene System besitzt keine Eingabekanäle und ist somit abgeschlossen (Def. 2.9 vgl. auch 2.15).

Falls beim legalen Angreifer einige Kanäle unbelegt sind, wird über diese (illegale) Teilinstanz $(A - A_U)$ ständig `nop` eingegeben (vgl. Def. 2.19).

3.5 Beispiel für internen Systemaufbau

Um die Allgemeinheit dieses Modells zu verdeutlichen, wird in diesem Abschnitt kurz angedeutet, daß auch ein Netzwerk durch eine Ausgabefunktion darstellbar ist.

Ausgangsbasis der Geheimhaltungsdefinitionen ist ausschließlich die Ausgabefunktion $\mathcal{O}$ des Kryptosystems; wie diese Funktion intern zustandekommt, ist für die Geheimhaltungs– und Servicedefinitionen nicht relevant.

3.5.1 Ausgabefunktion

3.5.1.1 Innerer Aufbau

Ein Kryptosystem kann zum Beispiel ein Netzwerk von interaktiven Turingmaschinen (vgl. z.B. [Bea91]) sein. Idee ist, daß ein Netz den Service eines bestimmten Kryptosystemtyps bietet, wenn man an der Benutzerschnittstelle die gewünschten Operationen durchführen kann. Die Geheimhaltung eines Kryptosystems wird dann mit den in Abschnitt 4 eingeführten Begriffen definiert. Abb. 3.4 zeigt die Struktur des Beispielsystems mit eingezeichneten Angriffspunkten. Wie aus der Abbildung ersichtlich ist, müssen nicht alle Leitungen vom Angreifer beobachtbar oder veränderbar sein.

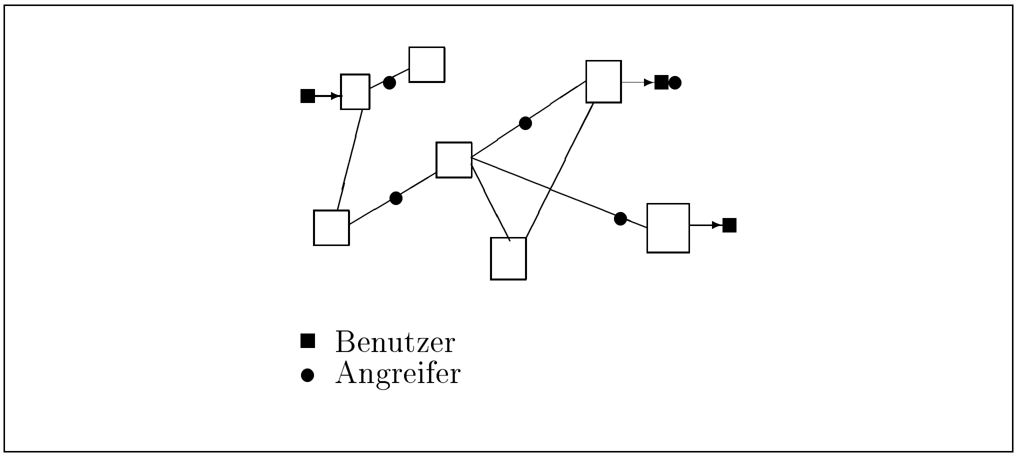

Abbildung 3.4: Innere Struktur eines Kryptosystems mit eingezeichnetem Angreifer

Zur Spezifikation der Serviceeigenschaften genügt es, die Benutzerschnittstelle des Systems anzugeben. Hierbei wird die Schnittstellendefinition aus Kapitel 2 verwendet.

3.5.1.2 Knoten

Die Knoten sind hier interaktive Turingmaschinen. Falls der Angreifer Interna der Knoten beeinflussen kann, so muß dies über die regulären Ein– und Ausgabekanäle des Knotens möglich sein oder durch extra Leitungen zum Angreifer explizit modelliert werden.

3.5.1.3 Leitungen

Das Leitungsmodell ist frei wählbar. Als Beispiel kann jede Leitung als Übertragungsfunktion modelliert sein (Abb. 3.5).

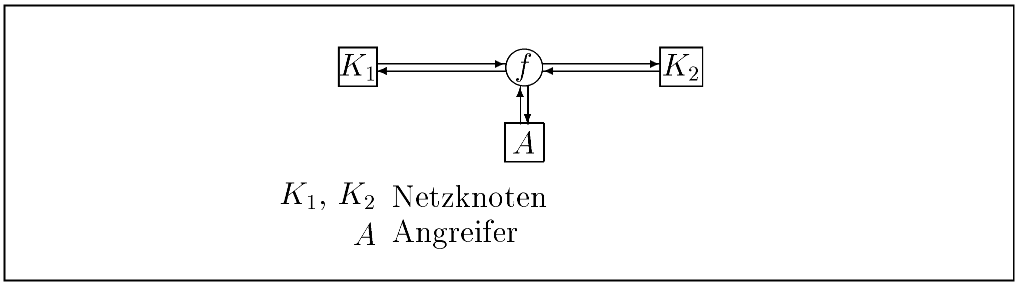

Abbildung 3.5: Beispiel für ein Leitungsmodell

Kapitel 4

Geheimhaltung

In diesem Abschnitt werden Definitionen für die Geheimhaltung von Ereignissen angegeben. Die angegebene modulare Definition wird im nächsten Kapitel durch Beispiele veranschaulicht. Grundlage ist das aus Kryptosystem, Benutzer und Angreifer zusammengesetzte System aus Kapitel 3.

4.1 Anforderungen an eine Definition

Ziele einer Definition für Geheimhaltung sind:

- unerwünschten Informationsfluß zu verhindern,
- kontrollierten Informationsfluß zu gewährleisten und
- den intuitiven Begriff der Geheimhaltung möglichst genau zu erfassen.

Hierbei muß nach Stärke und Verbeitungsgrad des Angreifers und nach Kooperationsbereitschaft und Stärke der Benutzer unterschieden werden. Ich will nun die beiden letzten Ziele ausführlicher erläutern:

4.1.1 Geheimhaltung ist kein Widerspruch zu Informationsfluß

Bei direkter Betrachtung eines minimalen Konzelationssystems* mit einem Sende– und Empfangskanal sollte keine Information über die versendete Nachricht an den Angreifer* weitergegeben werden.

Da Systeme in der Praxis jedoch oft Informationsfluß zwischen allen Teilnehmern ermöglichen soll (z.B. bei Netzwerken), muß nur verhindert werden, daß ein Angreifer gegen den Willen des Benutzers Geheimnisse erfährt. Hierbei muß natürlich gewährleistet sein, daß willentlicher Geheimnisverrat die Geheimhaltung des Systems nicht beeinflußt. Das folgende kleine Beispiel verdeutlicht dies:

Beispiel 4.1 (Informationsfluß soll möglich sein!)
Alice schickt Bob eine Nachricht. Eve versucht, etwas über den Inhalt der Nachricht zu erfahren. Obwohl Alice auch die Möglichkeit haben muß, an Eve Nachrichten zu senden, soll die an Bob gesendete Nachricht geheim bleiben.

Falls Alice eine Kopie der geheimen Nachricht an Eve sendet[1]*, so erfährt Eve die Nachricht, ohne daß hierdurch die Geheimhaltung des Systems beeinträchtigt sein muß, da in diesem Fall Alice und nicht das System schuld ist.*

Anmerkung 4.1 *Den Geheimnisverrat durch den Benutzer kann man in einer allgemeinen Definition nicht ausschließen, wäre er ausgeschlossen, so wäre keine Definition für beliebige Systeme mehr möglich.*

4.1.1.1 Eine Geheimhaltungsdefinition muß der Intuition entsprechen

Die hier anzugebende Definition soll den Sicherheitsbegriff möglichst genau erfassen. Manche bereits bestehende Definitionen erfüllt diese Forderung nicht.

Die Vorstellung, daß eine Stufe der Geheimhaltung genügt, ist unrealistisch. Es werden hier mehrere Stufen vorgestellt, die unterschiedliche Geheimhaltungsstärken formalisieren. Kriterien sind, ob Benutzer alle Freiheiten haben oder genaue „Bedienungsanleitungen“ befolgen müssen, sowie die Mächtigkeit und Verbreitung von Benutzern und Angreifern. Es ist hierbei im Zweifelsfall besser, eine zu starke Definition zu wählen, was zur Folge hat, daß ein System, welches diese Definition erfüllt, auf alle Fälle sicher ist.

Manche in der Literatur vorgeschlagenen Definitionen erfüllen dieses Kriterium nicht. So sind Systeme, die der von Galil et. al. in [GHY91] vorgeschlagenen Geheimhaltungsdefinition genügen in der Realität nicht unbedingt sicher (vgl. Bsp. 3.2 auf S. 53).

[1] Alice hat von einem „guten Freund“ erfahren, daß dies die Geheimhaltung erhöht.

4.1.2 Wieso erfüllen bestehende Geheimhaltungsdefinitionen diese Forderungen nicht?

Es existiert bis heute noch keine einheitliche Definition der Geheimhaltung. Es wurden bereits einige Geheimhaltungsdefinitionen für Spezialfälle vorgestellt. Diese sind aber nicht übertragbar, da sie

- im nachhinein die Sicherheit *eines* vorgeschlagenen Kryptosystemtyps definieren und nicht auf andere Systemtypen übertragbar sind,
- Geheimhaltung bezüglich einer informellen Angriffsdefinition „beweisen“.
- nur einzelne Angriffsarten oder nur wenige spezielle Angreifer betrachten [GHY91, MRS88] oder
- nur nichtinteraktive Systeme betrachten die, wie von Deffie und Hellman in [DH76, S. 648, Formel (2) & (3)] vorgeschlagen, eine Trennung des Public–Key Kryptosystems in zwei Teilfunktionen voraussetzen [MRS88]. (Auf interaktive Protokolle, die nicht als zwei Funktionen darstellbar sind, sind diese Definitionen nicht anwendbar.)

4.2 Aufbau der Definition

Die Idee der hier angegebenen Definitionen ist, geheime Informationen vor dem Angreifer zu schützen und trotzdem Informationsfluß zwischen Angreifer und Benutzern zu ermöglichen.

Da der Angreifer für die aktiven Angriffe auf die „Mitarbeit“ der Benutzer angewiesen ist, wird die Sicherheit gegenüber diesen Angriffen definiert, indem die Mächtigkeit zweier Angreifer verglichen wird. Der „friedliche“ Angreifer erhält nur die Informationen direkt von den Benutzern. Der „illegale“ Angreifer erhält zusätzlich Informationen aus dem System. Ziel muß sein, daß der zweite Angreifer nicht mehr Information erhält als der erste (vgl. 4.1).

Um das Verständnis der im folgenden angegebenen Definitionen zu erleichtern, wird die Struktur kurz in einem Überblick dargestellt.

Die angegebene Definition ist modular aufgebaut. Sie setzt sich bei gegebenem Kryptosystem aus Teildefinitionen für Angreifer* und Benutzer* zusammen. Eine zu den Gegebenheiten passende Definition wird aus den angegebenen Teildefinitionen zusammengesetzt. Zu frei wählbaren Angreifer– und Benutzertypen kann somit eine Geheimhaltungsdefinition konstruiert werden.

Benötigte Fakten sind das Krypsosystem (Def. 3.1), die Benutzerinstanz und somit die Verbreitung* des Angreifers[2], „legale“ Teilinstanz des Angreifers, die Rechenleistung von Angreifer und Benutzer, eine geeignete Benutzerklasse (z.B. `passive`), die angibt, wie stark der Benutzer mit dem Angreifer kooperiert, und, ob Geheimhaltung für alle oder nur für einige Benutzer dieser Gruppe gefordert wird. Durch die angegebenen Teildefinitionen wird hieraus eine Geheimhaltungsdefinition erstellt.

Beispiel 4.2 (Geheime Addition) *Gegeben unser Addierer (Bsp. 2.1) mit einem Zusatzkanal* `d` *der einen Summanden ausgibt (Abb. 4.1).*

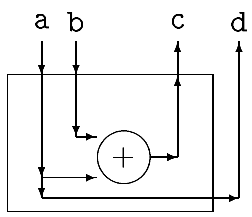

Abbildung 4.1: Modifizierter Addierer

Der informationstheoretische Benutzer wählt die Eingaben für die Kanäle `a` *und* `b`*, die Eingaben in Kanal* `b` *seien geheim. Es gilt also:*

$$B := \{a, b\} \times \mathbb{T}.$$

Die Summe der Zahlen sei nicht geheim. Es gilt also:

$$A_U := \{c\} \times \mathbb{T}.$$

Es ergibt sich aus der Schnittstelle des Kryptosystems und der Benutzerinstanz:

$$A = \{c, d\} \times \mathbb{T}.$$

[2]Die Verbreitung des Angreifers folgt direkt aus Kryptosystemschnittstelle und Benutzerinstanz.

Der Angreifer sei komplexitätstheoretisch. Benutzerklasse sei `passive` *Abbildung 4.2 zeigt den „legalen“ und „illegalen“ Angreifer. Bei den hier angegebenen Gegebenheiten ist der Angriff geglückt, da der „illegale“ Angreifer, im Gegensatz zum „legalen“, den geheimen Summanden* `b` *berechnen kann. Eine Geheimhaltungsdefinition würde bei diesen Gegebenheiten aus den Teil-*

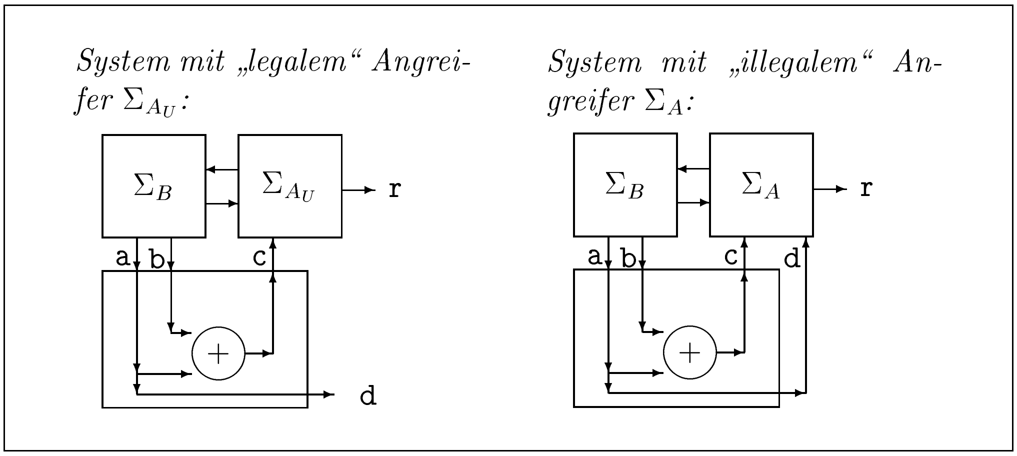

Abbildung 4.2: Addierer mit Benutzer und Angreifern

definitionen 3.10, 4.4 und 4.5 zusammengesetzt.

4.3 Angreiferteil der Definition

4.3.1 Grundidee: Geheimhaltung bei festem Benutzer und zwei verwandten Angreifern

Die hier zu formalisierende Vorstellung ist die, daß ein Benutzer gegeben ist und zwei Angreifer Σ_A und $\Sigma_{A'}$ miteinander konkurrieren. Σ_A ist der vollständige Angreifer (incl. Zugang zu Systeminternas), der alle seine Möglichkeiten ausschöpft, und Σ_{A_U} der „legale“, d.h. Σ_A bekommt die Informationen direkt vom Benutzer, von seiner regulären Systemschnittstelle und von Kanälen innerhalb des Systems. $\Sigma_{A'}$ ist wie Σ_A, hat jedoch keinen Zugang zu Systeminternas. Informationen die er erhält, erteilt der Benutzer sozusagen freiwillig.

Der Angreifer soll als „distinguisher“ zwischen zwei Ereignissen E_1 und E_2 dienen. Ziel ist, daß E_1 und E_2 aus Sicht des Angreifers ununterscheidbar sind. Falls E_1 eingetreten ist, soll er über den speziellen Kanal r (wie Resultat) eine „1“ ausgeben. Bei E_2 nicht. Die Abbildungen 4.3 und 4.4 veranschaulichen dieses Modell.

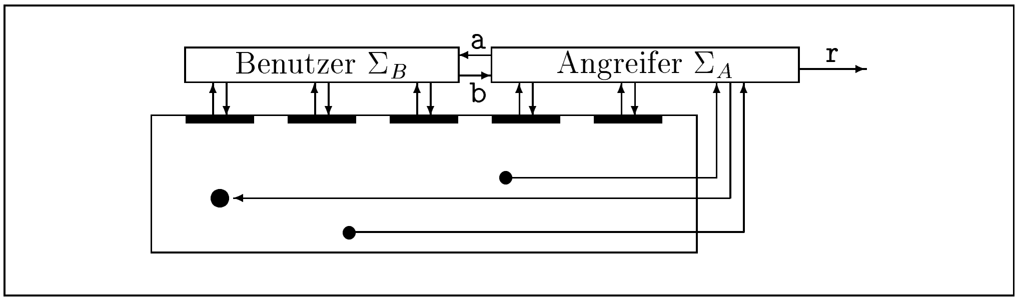

Abbildung 4.3: Systemskizze mit Angreifer Σ_A

Es werden drei Angreifertypen vorgestellt:

- Informationstheoretisch
- Informationstheoretisch mit Fehlerwahrscheinlichkeit f
- Komplexitätstheoretisch

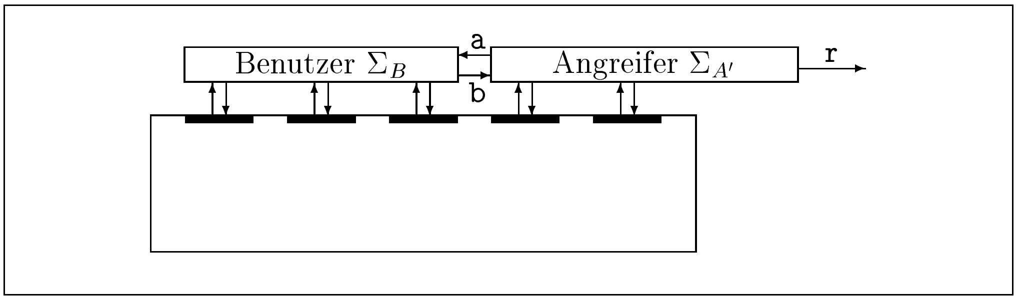

Abbildung 4.4: Systemskizze mit Angreifer $\Sigma_{A'}$

Anmerkung 4.2 *Wird im folgenden $P(E) > 0$ gefordert, so gilt dies, gdw. gilt:*

$$\underset{\Sigma_A}{\forall} P_{\Sigma_K \bowtie_+ \Sigma_B \bowtie_+ \Sigma_A}(E) > 0.$$

Es gilt also nur gdw. es bei beliebigem Angreiferverhalten gilt.

Anmerkung 4.3 *Für die im folgenden benutzten Ereignismengen $\mathcal{E}$ wird meistens für alle $E \in \mathcal{E}$ gelten:*

$$P_{\Sigma_n}(E) = P_{\Sigma'_n}(E)$$

mit:

$$\Sigma_n := \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A,n},$$
$$\Sigma'_n := \Sigma_{K,n} \bowtie_+ \Sigma_{B,n}.$$

Dies bedeutet, daß der Angreifer keinen Einfluß darauf hat, welches Ereignis der Benutzer auswählt.

Ist dies nicht der Fall und wählt der „legale“ Angreifer das Ereignis aus, so ist jedes System sicher.

4.3.2 Geheimhaltung bei festem Benutzer

Die folgenden Definitionen formalisieren Geheimhaltung bei einem festen Benutzersystem. $\mathcal{E}$ ist die Menge der geheimzuhaltenden Ereignisse. Der Angreifer versucht Informationen zu erhalten, welches Ereignis aus $\mathcal{E}$ stattgefunden hat.

Def. 4.1 (Informationstheoretischer Angriff) *Gegeben sei ein Kryptosystem (Σ_K, U), ein Angreifer $(\mathcal{F}, A_U)$, ein fester Benutzer $\Sigma_B = (S_B, B, \mathcal{O}_B)$ und eine Menge $\mathcal{E}$ von möglichen[3] Ereignissen, so ist die Formel F wie folgt definiert:*

$$F_{(\Sigma_K,U),\Sigma_B,(\mathcal{F},A_U),\mathcal{E}} :=$$

$$\underset{\Sigma_A \in \mathcal{A}}{\forall} \; \underset{\Sigma_{A'_U} \in \mathcal{A}_U}{\exists} \; \underset{n}{\forall} \; \underset{s \in \bigcup_{E \in \mathcal{E}} E}{\forall} \; \underset{\substack{E_1, E_2 \in \mathcal{E} \\ P_{\Sigma_n}(E_1) > 0 \\ P_{\Sigma_n}(E_2) > 0}}{\forall} \frac{P_{\Sigma_n}(E_1 | [\![s \lceil A]\!])}{P_{\Sigma_n}(E_2 | [\![s \lceil A]\!])} = \frac{P_{\Sigma'_n}(E_1 | [\![s \lceil A'_U]\!])}{P_{\Sigma'_n}(E_2 | [\![s \lceil A'_U]\!])}$$

mit:

$$\Sigma_n := \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A,n},$$

$$\Sigma'_n := \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A_U,n},$$

$$\mathcal{A} = \mathcal{A}((\Sigma_K, U), B, (\mathcal{F}, A_U)),$$

$$\mathcal{A}_U = \mathcal{A}_U((\Sigma_K, U), B, (\mathcal{F}, A_U))$$

und

$$\Sigma_{A_U} = (S_{A'_U}, A'_U, \mathcal{O}_{A'_U}).$$

Diese Definition ist verletzt, wenn ein Angreifer Σ_A existiert, der besser als alle „legalen“ Angreifer $\Sigma_{A'_U}$ ist, d.h. aus seiner Sicht E_1 und E_2 besser unterscheiden kann, als der legale Angreifer.

Anmerkung 4.4 *Würde die Gültigkeit der Bedingung auch für $P_{\Sigma_n}(E) = 0$ gefordert, so würde für eine Eingabefunktion $\mathcal{O}$ mit $P_{\Sigma_n}(E_1) = 0$ für ein E_1 und $P_{\Sigma_n}(E_2) > 0$ für ein E_2 die Ununterscheidbarkeitsbedingung immer verletzt werden da eine Seite der Gleichung positiv, und die andere per Definition null wäre ($\frac{0}{0} = 0$) wird. Dies ist aber nicht sinnvoll, da meist nicht der gesamte Ereignisraum ausgeschöpft wird (z.B. die Menge der Ereignisse zu beliebigen Texten).*

Anmerkung 4.5 *Auch beim informationstheoretischen Angreifer sind Kanäle für aktive Angriffe vorgegeben. Eine eventuelle Äquivalenz aktiver und passiver Angriffe müßte bewiesen werden.*

[3] D.h. $\underset{E \in \mathcal{E}}{\forall} \; \underset{\Sigma_B}{\exists} \; P(E) > 0$

Anmerkung 4.6 *Ist $P_{\Sigma_n}(E_2|[\![s\lceil A]\!])$ und $P_{\Sigma'_n}(E_2|[\![s\lceil A]\!])$ undefiniert, so ist die Bedingung erfüllt. Ist nur eine der beiden Seiten definiert so verletzt dies die Bedingung.*

Anmerkung 4.7 (Angreifer kennt Benutzerverhalten)
Diese Definition schließt den Fall ein, daß der Angreifer „passend" zum Benutzer gewählt wird. Dies würde in der Praxis bedeuten, daß der Angreifer die Benutzerausgabefunktion $\mathcal{O}_B$ kennt.

In der folgenden Definition wird der Sicherheitsparameter erstmals verwendet.

Def. 4.2 (Informationstheoretischer Angriff mit Fehler)
Gegeben sei ein Kryptosystem (Σ_K, U), ein fester Benutzer $\Sigma_B = (B, \mathcal{O}_B)$, ein Angreifer $(\mathcal{F}_f, A_U)$ und eine Menge $\mathcal{E}$ von möglichen Ereignissen. So ist die Formel F wie folgt definiert:

$$F_{(\Sigma_K,U),\Sigma_B,(\mathcal{F}_f,A_U),\mathcal{E}} :=$$

$$\underset{\Sigma_A\in\mathcal{A}}{\forall}\ \underset{\Sigma_{A'_U}\in\mathcal{A}_U}{\exists}\ \underset{n_0}{\exists}\ \underset{n>n_0}{\forall}\ \underset{s\in\bigcup_{E\in\mathcal{E}} E}{\forall}\ \underset{\substack{E_1,E_2\in\mathcal{E}\\ P_{\Sigma_n}(E_1)>0\\ P_{\Sigma_n}(E_2)>0}}{\forall}$$

$$\frac{P_{\Sigma_n}([\![r=1]\!]|E_1)}{P_{\Sigma_n}([\![r=1]\!]|E_2)} < \frac{P_{\Sigma'_n}([\![r=1]\!]|E_1)}{P_{\Sigma'_n}([\![r=1]\!]|E_2)} + f(n)$$

mit:

$$\Sigma_n := \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A,n},$$

$$\Sigma'_n := \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A_U,n},$$

$$\mathcal{A} = \mathcal{A}((\Sigma_K, U), B, (\mathcal{F}_f, A_U)),$$

$$\mathcal{A}_U = \mathcal{A}_U((\Sigma_K, U), B, (\mathcal{F}_f, A_U))$$

und

$$\Sigma_{A_U} = (S_{A'_U}, A'_U, \mathcal{O}_{A'_U}).$$

In der Regel werden Fehlerklassen angegeben. D.h. es wird nicht eine feste Funktion, sondern eine Funktionenklasse verwendet.

Def. 4.3 (Infoth. Angreifer mit polynomiellem Fehler) *Der Durchschnitt aller Angreifer mit Fehlerwahrscheinlichkeit f für $f = 1/p$ (p Polynom) heißt die Klasse der informationstheoretischen Angreifer mit polynomiell kleiner Fehlerwahrscheinlichkeit.*

Analog ist auch der informationstheoretische Angreifer mit exponentiell kleinem Fehler definiert.

Def. 4.4 (Komplexitätstheoretischer Angriff)
Gegeben sei ein Kryptosystem (Σ_K, U), ein fester Benutzer $\Sigma_B = (S_B, B, \mathcal{O}_B)$, ein Angreifer $(\mathcal{P}, A_U)$ und eine Menge $\mathcal{E}$ von möglichen Ereignissen. So ist die Formel F wie folgt definiert:

$$
\begin{aligned}
&F_{(\Sigma_K,U),\Sigma_B,(\mathcal{P},A_U),\mathcal{E}} := \\
&= \forall_{c>0} \forall_{\Sigma_A \in \mathcal{A}} \exists_{\Sigma_{A'_U} \in \mathcal{A}_U} \exists_{n_0} \forall_{n>n_0} \quad \forall_{\substack{E_1, E_2 \in \mathcal{E} \\ P_{\Sigma_n}(E_1) > 0 \\ P_{\Sigma_n}(E_2) > 0}} \\
&\qquad \frac{P_{\Sigma_n}([\![r=1]\!] | E_1)}{P_{\Sigma_n}([\![r=1]\!] | E_2)} < \frac{P_{\Sigma'_n}([\![r=1]\!] | E_1)}{P_{\Sigma'_n}([\![r=1]\!] | E_2)} + \frac{1}{n^c}
\end{aligned}
$$

mit:

$$
\begin{aligned}
\Sigma_n &:= \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A,n}, \\
\Sigma'_n &:= \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A_U,n}, \\
\mathcal{A} &= \mathcal{A}((\Sigma_K, U), B, (\mathcal{P}, A_U)), \\
\mathcal{A}_U &= \mathcal{A}_U((\Sigma_K, U), B, (\mathcal{P}, A_U))
\end{aligned}
$$

und

$$\Sigma_{A_U} = (S_{A'_U}, A'_U, \mathcal{O}_{A'_U}).$$

Anmerkung 4.8 (Unmögliche Ereignismengen) *Unmögliche Ereignisse sind immer ununterscheidbar, da $P(E)$ nie positiv ist.*

Lemma 4.1 (Möglichkeit von Ereignissen)
Die folgenden zwei Forderungen sind für endliche Ereignismengen $\mathcal{E}$ äquivalent:

$$\exists_{\Sigma_B} \forall_{E \in \mathcal{E}} P(E) > 0 \iff \forall_{E \in \mathcal{E}} \exists_{\Sigma'_B} P(E) > 0.$$

Beweis: „$\Rightarrow$“: Zu jedem E erfüllt Σ_B die Bedingung.
„$\Leftarrow$“: Sei $i_n \in E_n$ eine spezielle Eingabe des Systeme $\Sigma_{B'}$ für die $P(E_n) > 0$ ist. Für Σ_B welches gleichverteilt zwischen allen i_n auswählt, gilt nun

$$\underset{E \in \mathcal{E}}{\forall} P(E) > 0. \blacksquare$$

4.3.3 Spezialfall: absoluter Angreifer

Ein absoluter Angriff liegt vor, wenn die „legale“ Angreiferinstanz leer ist. Ist $A_U = \emptyset$, so ist das Angreifersystem Σ_{A_U} nicht mit dem Kryptosystem Σ_K verbunden! Beide arbeiten also stochastisch unabhängig.

Wie bereits in Abschnitt 3.3.1 erläutert, darf der Angreifer in diesem Fall keinerlei Information erhalten, welches der geheimen Ereignisse eingetreten ist. Informationsfluß ist nicht mehr möglich.

4.3.3.1 Informationstheoretischer Angreifer

Für den Spezialfall $A_U = \emptyset$ vereinfacht sich die Definition wie folgt:

Lemma 4.2 (Absoluter informationstheoretischer Angreifer)
Gilt $A_U = \emptyset$ und Anmerkung 4.3, so gilt für informationstheoretische Angreifer:

$$F_{(\Sigma_K,U),\Sigma_B,(\mathcal{F},\emptyset),\mathcal{E}} \Leftrightarrow \underset{\Sigma_A \in \mathcal{A}}{\forall} \ \underset{s \in \bigcup\limits_{E \in \mathcal{E}} E}{\forall} \ \underset{\substack{E_1, E_2 \in \mathcal{E} \\ P_{\Sigma_n}(E_1) > 0 \\ P_{\Sigma_n}(E_2) > 0}}{\forall} P_{\Sigma_n}([\![s\lceil A]\!] \mid E_1) = P_{\Sigma_n}([\![s\lceil A]\!] \mid E_2).$$

mit:

$$\Sigma_n := \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A,n},$$

$$\mathcal{A} = \mathcal{A}((\Sigma_K, U), B, (\mathcal{F}, \emptyset)).$$

Beweis:

1. Gelte Anmerkung 4.3. Da $A_U = \emptyset$ ist $[\![s\lceil A_U]\!] = S_{\mathbb{C}[\mathbb{T}]}$; also gilt:

   $$P_{\Sigma'_n}(E_1|[\![s\lceil A_U]\!]) = P_{\Sigma'_n}(E_1)$$

   und somit vereinfacht sich die Forderung zu

   $$\frac{P_{\Sigma_n}(E_1|[\![s\lceil A]\!])}{P_{\Sigma_n}(E_2|[\![s\lceil A]\!])} = \frac{P_{\Sigma'_n}(E_1)}{P_{\Sigma'_n}(E_2)}$$

   mit

   $$P_{\Sigma_n}(E_i) = P_{\Sigma'_n}(E_i)$$

   folgt die Behauptung.

2. (Skizze) Gelte Anmerkung 4.3 nicht, so gilt

   $$P_{\Sigma_n}(E) \neq P_{\Sigma'_n}(E)$$

   und somit gilt die angegebene Formel nicht. Da der Angreifer Σ_A Einfluß auf die Ereigniswahl des Benutzers hat und der „legale" Angreifer nicht, gilt auch $F_{(\Sigma_K,U),\Sigma_B,(\mathcal{F},\emptyset),\mathcal{E}}$ nicht. ■

Anmerkung 4.9 *Gilt bei informationstheoretischem Angreifer $A_U = \{r\}\times\mathbb{T}$ (wie in Def. 3.16 gefordert) so handelt es sich ebenfalls um einen absoluten Angreifer.*

Anmerkung 4.10 *Ein erster Ansatz ist, zu fordern, daß die Wahrscheinlichkeit beider Ereignisse nach der Beobachtung gleich wäre:*

$$\underset{\Sigma_A\in\mathcal{A}}{\forall}\ \underset{s\in\bigcup_{E\in\mathcal{E}} E}{\forall}\ \underset{\substack{E_1,E_2\in\mathcal{E}\\ P(E_1)>0\\ P(E_2)>0}}{\forall} P_{\Sigma_n}(E_1|[\![s\lceil A]\!]) = P_{\Sigma_n}(E_2|[\![s\lceil A]\!]),$$

dies ist aber nicht sinnvoll, da diese Definition, von einer a priori Gleichverteilung ausgeht.

4.3.3.2 Komplexitätstheoretischer Angreifer

Analog kann absolute Geheimhaltung auch für komplexitätstheoretische Angreifer definiert werden:

Lemma 4.3 (Absolute komplexitätstheoretische Angreifer)
Sei Σ_B Benutzer oder Benutzerfamilie. Sei A Angreiferinstanz und $\mathcal{E}$ eine Menge von möglichen Ereignissen. Gilt $A_U = \emptyset$, so gilt für ein festes Σ_B:

$$F_{(\Sigma_K,U),\Sigma_B,(\mathcal{P},\emptyset),\mathcal{E}} \Leftrightarrow$$

$$\underset{c>0}{\forall}\ \underset{\Sigma_A\in\mathcal{A}}{\forall}\ \underset{n_0}{\exists}\ \underset{n>n_0}{\forall}\ \underset{\substack{E_1,E_2\in\mathcal{E}\\ P_{\Sigma_n}(E_1)>0\\ P_{\Sigma_n}(E_2)>0}}{\forall} P_{\Sigma_n}(\llbracket r=1\rrbracket|E_1) < P_{\Sigma_n}(\llbracket r=1\rrbracket|E_2) + \frac{1}{n^c}$$

mit:

$$\Sigma_n := \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A,n},$$

$$\mathcal{A} = \mathcal{A}((\Sigma_K, U), B, (\mathcal{P}, \emptyset)).$$

Beweis: Da $A_U = \emptyset$ gilt:

$$\frac{P_{\Sigma'_n}(\llbracket r=1\rrbracket|E_1)}{P_{\Sigma'_n}(\llbracket r=1\rrbracket|E_2)} = 1. \blacksquare$$

4.4 Benutzerteil der Definition

In diesem Abschnitt werden die bisherigen Definitionen um geeignete Benutzer zu einer Geheimhaltungsdefinition erweitert. Es wird zwischen Geheimhaltung für bestimmte und Geheimhaltung für alle Benutzer unterschieden.

4.4.1 Geheimhaltung für alle oder nur für bestimmte Benutzer?

Wünschenswert wäre eine Definition, die den Benutzern alle Freiheiten läßt und dann fordert, daß der Angreifer durch seine „illegalen[4]“ Quellen keine zusätzliche Information erhalten darf, die er nicht ohnehin durch regelgerechte Systembenutzung erhält.

Diese Forderung ist aber für viele Anwendungen zu stark, da es nach dieser Definition nicht möglich sein darf, Information an den Angreifer im System zu übermitteln.

Für eine formelle Definition muß entschieden werden, wie über die Benutzer quantifiziert werden soll.

4.4.1.1 Wieso muß das Benutzerverhalten oft eingeschränkt werden?

Die stärkste Forderung an ein System ist, daß es für alle Benutzer einer Klasse (wie `chosen-input`) Geheimhaltung garantiert. Diese Forderung ist jedoch oft zu stark: läßt man beliebiges Benutzer– und Angreiferverhalten während dem geheimen Ereignis zu, so führt dies dazu, daß ein System nur dann sicher ist, wenn ein Benutzer keinerlei Information an den Angreifer im System übertragen kann, da dann für den Benutzer, der systematisch dem (illegalen) Angreifer verrät, welches Ereignis stattgefunden hat keine Geheimhaltung mehr besteht.

Um dies zu verdeutlichen, zeigt folgendes Beispiel daß das gebräuchliche One–Time Pad nach dieser Definition für uneingeschränkte Benutzer nicht sicher ist:

---

[4]Hier: $A - A_U$

Beispiel 4.3 *Sei* (E, D) *(für Verschlüsselungs– und Entschlüsselungsalgorithmus) das perfekte One-Time-Pad, welches zu einer Nachricht fester Länge einen Schlüsseltext fester Länge ausgibt.*

Das System agiert wie folgt:

- *Klartexteingabe* m *von Benutzer* B.
- *Schicke* $E(m)$ *über die vom Angreifer beobachtete Leitung.*
- *Entschlüssele und gebe* $D(E(m)) = m$ *aus.*

Die Tatsache ob eine Nachricht gesendet wurde soll hier nicht verborgen werden. Dies System sollte sicher sein, da der Angreifer keine Information über den Nachrichteninhalt erhält.

Das hier definierte System ist jedoch beliebige Benutzer nicht relativ (komplexitätstheoretisch!) sicher:

Begründung: Für den Benutzer, der durch getaktetes Abschicken von unsinnigen Nachrichten die geheime Nachricht an den Angreifer „morst", ist das System nicht sicher.

Um zu zeigen, daß es trotzdem Systeme gibt, die nach dieser Definition sicher sind, kann das obige System wie folgt modifiziert werden:

Beispiel 4.4 (Sicheres One–Time Pad)
Wird das One–Time Pad aus Bsp. 4.3 modifiziert, so daß das System zu jedem Zeitpunkt ein One–Time Pad oder einen One–Time Pad verschlüsselten Schlüsseltext fester Länge überträgt, so ist das so konstruierte System relativ informationstheoretisch sicher.

Begründung: Der Benutzer kann nun keinerlei Information an den Angreifer weitergeben. □

Analog hierzu kann der Benutzer auch dazu verpflichtet werden, zu jedem Zeitpunkt eine Nachricht zu versenden.

Die folgende Anmerkung gibt eine weitere Einschränkung an das Kryptosystem an:

Anmerkung 4.11 (Nichtprobabilistisches Konzelationssystem) *Sind beliebige Aktionen des Benutzers zugelassen, so darf der Benutzer bei nichtprobabilistischen Verfahren auf keinen Fall seinen geheimen Schlüssel des verwendeten Verschlüsselungsverfahrens wissen*[5]*. Er könnte sonst eventuell einen „entschlüsselten" Text verschlüsseln lassen und könnte somit dem Angreifer beliebige Nachrichten übermitteln. Unter anderem bei einem probabilistischen System muß dies nicht der Fall sein, da die Verschlüsselung eines „entschlüsselten" Klartextes in der Regel nicht wieder zum Klartext führt [GM84b].*

4.4.1.2 Geheimhaltung für einige Benutzerverhalten

In diesem Abschnitt soll kurz erläutert werden, wieso die Forderung, daß es mindestens einen nichttrivialen Benutzer gibt, für den das Ereignis geheim ist, sinnvoll ist.

Wie weiter oben gezeigt wurde, ist die Forderung nach Geheimhaltung für beliebiges Benutzerverhalten oft zu stark. Ein Ansatz, eine schwächere Definition zu konstruieren wäre, die Benutzerklassen so restriktiv zu wählen, daß der Benutzer ausschließlich das Ereignis durchführen kann und „sonst nichts". Eine derartige Definition müßte Geheimhaltung quasi für einen festen Benutzer definieren, welcher zu jeder Systemart erneut angegeben werden muß.

Dies ist jedoch schwer zu formalisieren und entspricht auch nicht der Zielsetzung dieser Arbeit. Es sollte genau spezifiziert werden, was ein Benutzer tun darf und was nicht. Die Allgemeinheit des Modells würde hierdurch stark eingeschränkt.

Falls die Benutzermenge auf Klassen reduziert wird, die gewisse „Verpflichtungen" erfüllen müssen, wird eine schwächere Geheimhaltungsdefinition sinnvoll, bei der nur für mindestens einen nichttrivialen Benutzer aus dieser Klasse Geheimhaltung gefordert wird. Die zugrundeliegende Überlegung ist, daß wenn es einen Benutzer gibt, für den das System sicher ist, muß es auch eine Klasse von Benutzern geben, für die Sicherheit gewährleistet ist. Das System kann somit nicht allzu unsicher sein.

Anmerkung 4.12 *Da der Allquantor durch einen Existenzquantor ersetzt wird, würde sich die Stärke der Forderungen bei Inklusionsbeziehung der Benutzerklassen umkehren.*

[5] Dies schließt aus, daß er ihn wählt, überträgt oder vom Kryptosystem mitgeteilt bekommt.

Zwischen aktiven und passiven Benutzerklassen besteht jedoch keine Inklusion.

Exemplarisch zeigt folgendes Lemma diesen Sachverhalt für `universal` und `passive`:

Lemma 4.4 *Es gibt keinen Benutzer, der sowohl* `passive` *als auch* `universal` *erfüllt.*

Beweisidee: Der Benutzer kann nicht Befehle befolgen und gleichzeitig nicht reagieren. ∎

Lemma 4.5 *In Kombination mit dem Angreiferteil der Definition stellt sich die erwartete Inklsusion ein: Ein gegen aktiven Angriff sicheres System ist auch gegen passiven Angriff sicher.*

Beweisidee: Der aktive Angreifer ist bei festem (aktiv angreifbarem) Benutzer stärker, da er auf seine „aktiven Mittel“ verzichten kann ($i_{a,t} = \texttt{nop}$) und somit einen passiven Angriff auf diesem Benutzer mit einschließt. ∎

4.5 Vollständige Geheimhaltungsdefinitionen

Def. 4.5 (Geheimhaltung)
Gegeben seien ein Kryptosystem (Σ_K, U), ein Benutzer $(\mathcal{C}_B, B)$ (vgl. Abschnitt 3.2.2 und Def. 3.2) und ein Angreifer $(\mathcal{C}_A, A_U)$. Sei

$$\mathcal{B} = \mathcal{B}((\Sigma_K, U), (\mathcal{C}_F, B)).$$

a) *Eine Menge $\mathcal{E}$ von möglichen Ereignissen heißt geheim bei beliebigem Benutzerverhalten gdw. gilt:*

$$\underset{\Sigma_B \in \mathcal{B}}{\forall} F_{(\Sigma_K, U), \Sigma_B, (\mathcal{C}_A, A_U), \mathcal{E}}.$$

b) *Eine Menge $\mathcal{E}$ von möglichen Ereignissen heißt geheim bei bestimmtem Benutzerverhalten gdw. gilt:*

$$\underset{\Sigma_B \in \mathcal{B}}{\exists} P_{\Sigma_B}(\mathcal{E}) > 0 \wedge F_{(\Sigma_K, U), \Sigma_B, (\mathcal{C}_A, A_U), \mathcal{E}}.$$

mit

$$P_{\Sigma_B}(\mathcal{E}) > 0 \iff \underset{n}{\forall} \underset{E \in \mathcal{E}}{\forall} P_{\Sigma''_n}(E) > 0$$

und

$$\Sigma''_n := \Sigma_{K,n} \bowtie_+ \Sigma_{B,n}.$$

Anmerkung 4.13 *Durch die Zusatzforderung $P(\mathcal{E}) > 0$ wird der „untätige“ Benutzer ausgeschlossen.*

4.5.1 Überblick über die modular konstruierbaren Definitionen

Tabelle 4.1 zeigt einen Überblick über alle aus den angegebenen Angreifer– und Benutzerdefinitionen konstruierbaren Geheimhaltungsdefinitionen und die im folgenden verwendeten Abkürzungen (Wie z.B. A:R,K,B:G,K,cc).

Einzelne wichtige Spezialfälle werden in Kapitel 5 genauer betrachtet.

Wer?		Abk.	**Art**	Anzahl
Angreifer:	A:	R,	Relativ	
		A,	Absolut (Def. 4.2)	2
		I	Informationsth. (Def. 4.1)	
		F	Informationsth. mit Fehler (Def. 4.2)	
		K	Komplexitätsth. (Def. 4.4)	3
Benutzer:	B:	A,	Alle (Def. 4.5)	
		E,	Einige (Def. 4.5)	2
		I,	Informationsth.	
		K,	Komplexitätsth.	2
Benutzerklassen:		a	`all` (Def. 3.8)	
		ci	`chosen-input` (Def. 3.3)	
		nci	`no-chosen-input` (Def. 3.4)	
		co	`chosen-output` (Def. 3.5)	
		nco	`no-chosen-output` (Def. 3.6)	≥ 7

Tabelle 4.1: Überblick über alle konstruierbaren Definitionen

4.6 Spezielle Ereignismengen $\mathcal{E}$

In diesem Abschnitt werden einige Beispiele für die Wahl geeigneter Ereignismengen vorgestellt.

4.6.1 Unbeobachtbarkeit

Def. 4.6 (Unbeobachtbarkeit eines Ereignisses) *Voraussetzungen wie in Def. 4.5. Ein Ereignis E eines Systems Σ heißt unbeobachtbar gdw. $\{E, \neg E\}$ ununterscheidbar ist.*

Diese Definition ist ein oft benötigter Spezialfall der Geheimhaltungsdefinitionen. Hiermit könnte zum Beispiel vor dem Angreifer verborgen werden, ob eine Nachricht gesendet wurde.

In Anmerkung 5.6 wird gezeigt werden, daß sehr starke Unbeobachtbarkeit äquivalent mit der stochastischen Unabhängigkeit der Angreiferbeobachtung vom Ereignis ist.

4.6.2 Geheimhaltung einer Nachricht

Als Beispiel werden in diesem Abschnitt einige Ereignismengen im Zusammenhang mit dem Versenden von Nachrichten angegeben.

Das senden einer Nachricht m eines Nachrichtenraums M wird hier als Eingabe $\texttt{send}\ m$ mit $m \in M$ zu einem bestimmten Kanal a zu einem bestimmten Zeitpunkt t modelliert. Sei $|\cdot| : M \to I\!N$ die Funktion, die einer Nachricht ihre Länge zuordnet.

4.6.2.1 Vollständige Geheimhaltung

Um Nachrichteninhalt, Nachrichtenlänge und die Tatsache, ob eine Nachricht gesendet wurde, geheimzuhalten, muß folgende Menge geheim sein:

$$\mathcal{E} := \{[\![i_{a,t} = \texttt{send}\ m]\!] | m \in M\} \cup \{[\![i_{a,t} = \texttt{nop}\ ]\!]\}$$

Die Idee ist, daß ein Angreifer keine zwei Nachrichten unterscheiden kann und auch nicht feststellen kann, ob eine Nachricht gesendet wurde oder nicht.

4.6.2.2 Geheimhaltung eines Nachrichteninhalts

Um nur den Nachrichteninhalt und nicht die Länge der Nachricht geheimzuhalten, müssen alle Mengen $\mathcal{E}_i$; $i \in I\!N$ geheim sein:

$$\mathcal{E}_i := \{[\![i_{a,t} = \texttt{send}\ m]\!] | m \in M \,\mathrm{mit}\, |m| = i\}$$

Die Idee ist, daß ein Angreifer die Nachrichtentexte einer Länge nicht unterscheiden kann. Er kann evtl. jedoch Texte verschiedener Länge unterscheiden.

4.6.2.3 Geheimhaltung der Nachrichtenlänge

Um die Länge der Nachricht geheimzuhalten, muß die folgende Menge $\mathcal{E}$ geheim sein:

$$\mathcal{E} := \{ \bigcup_{m \in M; |m| = i} [\![i_{a,t} = \texttt{send}\ m]\!] | i \in I\!N\}$$

Hierbei können je nach Nachrichtenraum M Teile des Nachrichtentextes an den Angreifer weitergegeben werden! Ist z.B. $|m| \geq 5$ für alle $m \in M$, so müssen 5 Zeichen nicht geheim sein.

Idee ist, daß ein Angreifer die Menge aller Nachrichten einer Länge nicht von der Menge der Nachrichten einer anderen Länge unterscheiden kann.

4.7 Sicherheit von Ereignissen

In diesem Abschnitt wird die Eigenschaft formalisiert, daß ein Ereignis eintritt. Diese Definition wird später auf die Erfüllung von Ereignissen zu Korrektheitsprädikaten angewendet.

Grundidee der Definitionen ist, daß trotz Angreifer die Wahrscheinlichkeit, daß das Ereignis eintritt (das Prädikat gilt) nahezi eins ist. Da geringe Störeinflüsse teilweise unvermeidbar sind (der Angreifer kann zum Beispiel eine Signatur raten), muß eventuell hier eine kleine Fehlerwahrscheinlichkeit zugelassen werden.

Diese Definition dient dazu, Aussagen, wie „ein Konzelationssystem überträgt eine Nachricht“ zu formalisieren.

Anmerkung 4.14 *Benutzer– und Angreiferklassen können wieder informations– oder komplexitätstheoretisch sein.*

4.7.1 Sicherheit eines Ereignisses

Die Sicherheit von Ereignisses wird hier nur grob behandelt, da sie für die Dienstdefinitionen für Konzelationssysteme nötig ist.

Beide Definitionen sind auf komplexitäts– bzw. informationstheoretische Benutzer und Angreifer anwendbar.

Def. 4.7 (Sicherheit von Ereignissen) *Sei Σ_K Systemfamilie. Ein Ereignis E heißt sicher bei Benutzern der Klasse $\mathcal{C}_B$ gegenüber einem Angreifer $(\mathcal{C}_A, A_U)$ gdw. gilt:*

$$\underset{c>0}{\forall}\ \underset{\Sigma_B \in \mathcal{B}}{\forall}\ \underset{\Sigma_A \in \mathcal{A}}{\forall}\ \underset{n_0}{\exists}\ \underset{n>n_0}{\forall} P_{\Sigma_n}(E) \geq 1 - \frac{1}{n^c}$$

mit

$$\begin{aligned}\Sigma_n &:= \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A,n},\\ \mathcal{A} &= \mathcal{A}((\Sigma_K, U), B, (\mathcal{C}_A, A_U)),\\ \mathcal{B} &= \mathcal{B}((\Sigma_K, U), (\mathcal{C}_F, B))\end{aligned}$$

und

$$\Sigma_{A_U} = (S_{A'_U}, A'_U, \mathcal{O}_{A'_U}).$$

4.7.2 Absolute Sicherheit eines Ereignisses

Obige Definition kann Fällen wie folgt verschärft werden:

Def. 4.8 (Absolute Sicherheit von Ereignissen) *Sei Σ_K Systemfamilie. Ein Ereignis E heißt absolut sicher bei Benutzern der Klasse $\mathcal{C}_B$ gegenüber einem Angreifer $(\mathcal{C}_A, A_U)$ gdw. gilt:*

$$\underset{\Sigma_B \in \mathcal{B}}{\forall}\ \underset{\Sigma_A \in \mathcal{A}}{\forall}\ \underset{n}{\forall} P_{\Sigma_n}(E) = 1$$

mit

$$\begin{aligned}\Sigma_n &:= \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A,n},\\ \mathcal{A} &= \mathcal{A}((\Sigma_K, U), B, (\mathcal{C}_A, A_U)),\\ \mathcal{B} &= \mathcal{B}((\Sigma_K, U), (\mathcal{C}_F, B))\end{aligned}$$

und

$$\Sigma_{A_U} = (S_{A'_U}, A'_U, \mathcal{O}_{A'_U}).$$

Als Anwendung dieser Definitionen werden in Kapitel 6 Dienstspezifikationen von Konzelationssystemen als Prädikate angegeben.

Kapitel 5

Beispiele konstruierbarer Geheimhaltungsdefinitionen

In diesem Abschnitt werden einige Beispiele für modular zusammengesetzte Geheimhaltungsdefinitionen vorgestellt und verglichen. Außerdem werden einige Eigenschaften dieser Geheimhaltungsdefinitionen angegeben und bewiesen.

5.1 Absolute Geheimhaltung

Der Begriff der absoluten Geheimhaltung soll sicherstellen, daß keinerlei Information vom Benutzer zum Angreifer fließen darf! Der Benutzer darf trotz beliebiger Aktion keine Möglichkeit erhalten, Information über die geheimen Ereignisse $\mathcal{E}$ an den Angreifer weiterzuleiten. Nahezu alle bisher bekannten Verfahren erfüllen die hier gestellten Forderungen im Dauerbetrieb nicht! Um zu zeigen, daß die Forderungen erfüllbar sind, wird als Beispiel unter anderem ein eingeschränktes One–Time–Pad vorgestellt, daß die vorgeschlagenen Definitionen erfüllt. Ein naiv eingesetztes One–Time–Pad erfüllt die Forderungen nicht, da der Benutzer durch die Nachrichtenlänge oder durch getaktetes Senden dem Angreifer „morsen“ kann, welches Ereignis stattgefunden hat (Bsp 4.3 & 4.4).

Die Angreiferinstanz kann auch hier Teile der Benutzerschnittstelle U des Kryptosystems besetzt halten. Da $A_U = \emptyset$ gilt (d.h. es existiert kein „legaler“ Teilangreifer) darf der Angreifer auch über diese Teile keine Information über das geheime Ereignis erhalten. Abb. 5.1 zeigt den Systemaufbau. Da alle Angreiferkanäle gleich behandelt werden (es spielt keine Rolle, ob es sich um Benutzer– oder interne Kanäle handelt), wurde in der Zeichnung nicht nach Kanalarten unterschieden.

Beispiel 5.1 (Multilevel–Security) *Ein Beispiel für einen Angreifer, der Teile der Benutzerschnittstelle besetzt, ist die Multilevel–Security.*

Hier kann der Angreifer ein regulärer Systembenutzer (mit einer niedrigen Sicherheitsstufe) sein. Es gilt jedoch zu verhindern, daß Benutzer höherer Sicherheitsstufen Informationen an den Angreifer übertragen können.

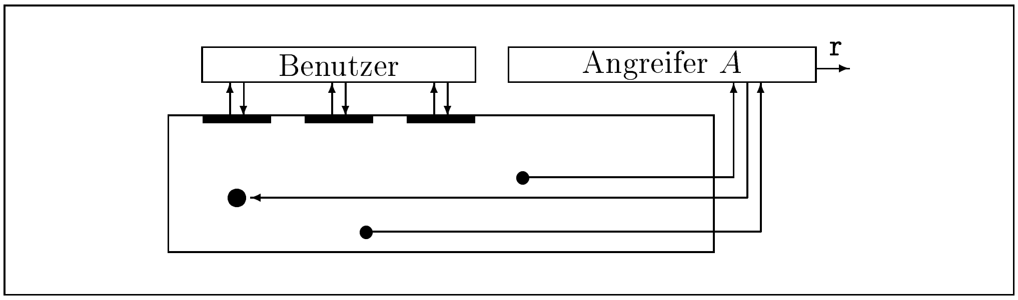

Abbildung 5.1: Systemskizze zu Def. 5.1

5.1.1 Absolute infoth. Geheimhaltung (A:AI, B:AIa)

Das Systemschema, welches dieser Definition zugrundeliegt, ist in Abb. 5.1 dargestellt.

Def. 5.1 (Absolute informationstheoretische Geheimhaltung) *Gegeben seien ein Kryptosystem (Σ_K, U) und eine Benutzerinstanz B. Sei*

$$\mathcal{B} = \mathcal{B}((\Sigma_K, U), (\mathcal{F}_{\texttt{all}}, B)).$$

Eine Menge von möglichen Ereignissen $\mathcal{E}$ eines Systems Σ heißt absolut informationstheoretisch geheim gegenüber einer Angreiferinstanz A gdw. gilt:

$$\underset{\Sigma_B \in \mathcal{B}}{\forall} \; \underset{\Sigma_A \in \mathcal{A}}{\forall} \; \underset{s \in \bigcup_{E \in \mathcal{E}} E}{\forall} \; \underset{\substack{E_1, E_2 \in \mathcal{E} \\ P_{\Sigma_n}(E_1) > 0 \\ P_{\Sigma_n}(E_2) > 0}}{\forall} \quad P_{\Sigma_n}(\llbracket s \lceil A \rrbracket \mid E_1) = P_{\Sigma_n}(\llbracket s \lceil A \rrbracket \mid E_2).$$

mit:

$$\Sigma_n := \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A,n}$$

und

$$\mathcal{A} = \mathcal{A}((\Sigma_K, U), B, (\mathcal{F}, \emptyset)).$$

Diese Definition sagt aus, daß je zwei stattgefundene Ereignisse aus Angreifersicht trotz beliebiger Benutzer– und Angreiferaktion nicht unterscheidbar sind.

Einzelne Ereignisse können bei bestimmten Eingabefunktionen nicht auftreten. Informationstheoretische Ununterscheidbarkeit ist jedoch trotzdem gewährleistet, wenn diese Ereignisse bei anderen Eingabefunktionen eine positive Wahrscheinlichkeit besitzen und hier die Bedingung erfüllen.

Anmerkung 5.1 (Beziehung zur Informationstheorie)
Diese Definition ist stärker als der bestehende Begriff der informationstheoretischen Geheimhaltung [Sha48, Sha49, Top74, Gra90]. Bei den bestehenden Ansätzen wird meist von eingeschränktem Benutzerverhalten ausgegangen (Benutzer wählt einmal ein Ereignis aus und tut sonst nichts). Dies ist hier nicht der Fall.

Anmerkung 5.2 (Kein Informationsfluß) *Ein System kann nur dann absolute informationstheoretische Geheimhaltung garantieren, wenn kein Kanal von den Benutzern zum Angreifer vorhanden ist. Falls ein Kanal vorhanden ist, können die Benutzer alle Ereignisse, die geheimgehalten werden sollen, dem Angreifer direkt mitteilen.*

Anmerkung 5.3 (Multilevel-Security) *Diese Definition ist sinnvoll für Fälle, die absolute Geheimhaltung fordern. Sie könnte z.B. Multilevel–Security garantieren.*

Der Benutzer ist hierbei H-Quelle, der Angreifer L-Senke. Beide versuchen gemeinsam *Information von H nach L zu übertragen [GM84a, Mil89, GS92].*

Anmerkung 5.4 (Angreifer kennt Benutzerverhalten) *Wie bereits in Anm. 4.7 erwähnt, kennt der Angreifer das Benutzerverhalten $\mathcal{O}_B$. Dies ist in der Definition daraus ersichtlich, daß zuerst über das Benutzerverhalten und dann über das Angreiferverhalten quantifiziert wird. Da Geheimhaltung für alle Angreifer gefordert wird, auch für den Angreifer „zum“ Benutzer.*

Es gibt jedoch keine expliziten Kanäle zwischen Angreifer und Benutzer.

Anmerkung 5.5 *In den Beispielen und Lemmas zur absoluten informationstheoretischen Geheimhaltung stehe P für P_{Σ_n}, da es sich nur um ein System handelt.*

Anmerkung 5.6 (Absolute informationsth. Unbeobachtbarkeit) *Informationstheoretische Unbeobachtbarkeit (Def. 4.6.1) von E bei absolutem Angreifer (Def. 5.1) ist äquivalent mit der Forderung nach stochastischer Unabhängigkeit des Ereignisses E mit der Angreiferbeobachtung $s\lceil A$.*

Begründung: (Äquivalenz mit Def. 5.1)

$$\begin{array}{rcl}
P(\llbracket s\lceil A\rrbracket | E) & = & P(\llbracket s\lceil A\rrbracket | \neg E) \\
\Leftrightarrow \quad \dfrac{P(\llbracket s\lceil A\rrbracket \cap E)}{P(E)} & = & \dfrac{P(\llbracket s\lceil A\rrbracket \cap \neg E)}{P(\neg E)} \\
\Leftrightarrow \quad \dfrac{P(\llbracket s\lceil A\rrbracket \cap E)}{P(E)} & = & \dfrac{P(\llbracket s\lceil A\rrbracket) - P(\llbracket s\lceil A\rrbracket \cap E)}{1 - P(E)} \\
\Leftrightarrow \quad (1 - P(E))P(\llbracket s\lceil A\rrbracket \cap E) & = & P(E)(P(\llbracket s\lceil A\rrbracket) - P(\llbracket s\lceil A\rrbracket \cap E)) \\
\Leftrightarrow \quad P(\llbracket s\lceil A\rrbracket \cap E) & = & P(E)P(\llbracket s\lceil A\rrbracket). \ \square
\end{array}$$

Lemma 5.1 *Definition 5.1 ist ein Spezialfall der Definitionen 4.1, 3.8 und 4.5.*

Beweis: Folgt direkt aus Lemma 4.2. ■

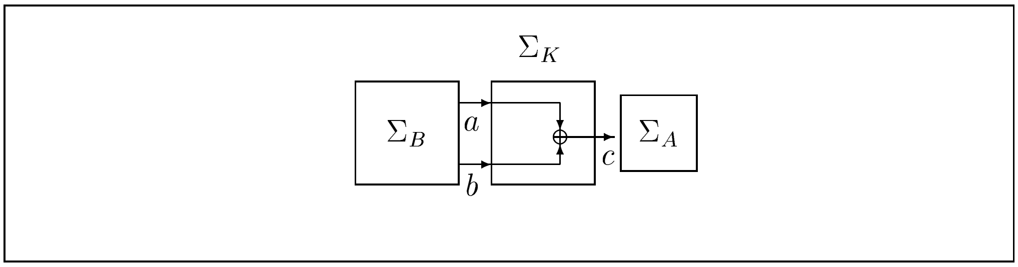

Abbildung 5.2: Beispielsystem zu Def. 5.1

Definition 5.1 soll dem System aus Abbildung 5.2 verdeutlicht werden. Angreifer ist $A = \{c\} \times \mathbb{T}$. Ausgabefunktion ist eine einfache xor–Verknüpfung. Da das System deterministisch ist, hängen die Wahrscheinlichkeiten der Abläufe nur von der Eingabefunktion ab. gegeben seien zwei Eingabefunktionen. Für die einzelnen Abläufe ergeben sich folgende Wahrscheinlichkeiten:

Beispiel 5.2 (xor-Gatter) *Sei* $S = (\{a, b\}, \{c\}, \{0, 1\}, \emptyset, \{1\})$ *und*

$$\mathcal{O}(c_t| < a_t, b_t >) := \begin{cases} 1 & \textit{falls } c_t = a_t \oplus b_t \textit{ (xor)} \\ 0 & \textit{sonst} \end{cases}.$$

so ist $(S, \mathcal{O})$ *eine Nachbildung eines xor-Gatters.*

	s		P(s)	
$< a,$	$b,$	$c >$	$\mathcal{O}_{B_1}$	$\mathcal{O}_{B_2}$
0	0	0	1/4	1/8
0	1	1	1/4	1/4
1	0	1	1/4	1/4
1	1	0	1/4	3/8
sonst			0	0

Beispiel 5.3 $E_3 := \{< 0, 0, 0 >\} \subseteq E_1$ *und* $E_4 := \{< 0, 1, 1 >\} \subseteq E_2$ *sind nicht informationstheoretisch geheim.*

Es gilt für $s\lceil A =< 0 >$:

$$\begin{aligned} P([\![s\lceil A]\!] \mid E_3) &= 1 \\ P([\![s\lceil A]\!] \mid E_4) &= 0 \neq 1. \ \square \end{aligned}$$

Beispiel 5.4 $E_5 = \{< 0, 0, 0 >\}$ *und* $E_6 = \{< 1, 1, 0 >\}$ *sind gegenüber* A *informationstheoretisch geheim.*

Sei $\mathcal{O}_B$ beliebig und $P(E_i) = p_i > 0$.

Es gilt für $s\lceil A =< 0 >$:

$$\begin{aligned} P([\![s\lceil A]\!] \mid E_1) &= P(\{< 0, 0, 0 >, < 1, 1, 0 >\} \mid \{< 0, 0, 0 >\}) = \\ &= \frac{P(\{< 0, 0, 0 >\})}{P(\{< 0, 0, 0 >\})} = 1 \\ P([\![s\lceil A]\!] \mid E_2) &= P(\{< 0, 0, 0 >, < 1, 1, 0 >\} \mid \{< 1, 1, 0 >\}) = \\ &= 1. \end{aligned}$$

Es gilt für $s\lceil A =< 1 >$:

$$\begin{aligned} P([\![s\lceil A]\!] \mid E_1) &= P(\{< 0, 1, 1 >, < 1, 0, 1 >\} \mid \{< 0, 0, 0 >\}) = 0 \\ P([\![s\lceil A]\!] \mid E_2) &= P(\{< 0, 1, 1 >, < 1, 0, 1 >\} \mid \{< 1, 1, 0 >\}) = 0. \ \square \end{aligned}$$

Beispiel 5.5 $E_1 := \{< 0,0,0 >, < 0,1,1 >, < 1,0,1 >\}$ *(höchstens eine Eingabe),* $E_2 := \{< 0,1,1 >, < 1,0,1 >, < 1,1,0 >\}$ *(mindestens eine Eingabe) sind im gegebenen System informationstheoretisch nicht geheim. Die Bedingung ist für* $\mathcal{O}_{B_1}$ *erfüllt, nicht jedoch für* $\mathcal{O}_{B_2}$*.*

Da der Angreifer keinen Eingabekanal kontrolliert, bezieht sich die Allquantifizierung der Eingaben nur auf die Eingabe über die Kanäle a und b. Zu zeigen ist für $s\lceil A = 0$ und $s\lceil A = 1$, daß gilt:

$$P(\llbracket s\lceil A \rrbracket \mid E_1) = P(\llbracket s\lceil A \rrbracket \mid E_2).$$

Es gilt für $\mathcal{O}_{B_1}$ und $s\lceil A =< 0 >$:

$$\begin{aligned}
&P(\llbracket s\lceil A \rrbracket \mid E_1) = \\
&= P(\{< 0,0,0 >, < 1,1,0 >\} \mid \{< 0,0,0 >, < 0,1,1 >, < 1,0,1 >\}) = \\
&= \frac{P(\{< 0,0,0 >\})}{P(\{< 0,0,0 >, < 0,1,1 >, < 1,0,1 >\})} = \frac{1/4}{3/4} = \frac{1}{3} \\
&P(\llbracket s\lceil A \rrbracket \mid E_2) = \\
&= P(\{< 0,0,0 >, < 1,1,0 >\} \mid \{< 0,1,1 >, < 1,0,1 >, < 1,1,0 >\}) = \\
&= \frac{P(\{< 1,1,0 >\})}{P(\{< 0,1,1 >, < 1,0,1 >, < 1,1,0 >\})} = \frac{1/4}{3/4} = \frac{1}{3}.
\end{aligned}$$

Es gilt für $\mathcal{O}_{B_1}$ und $s\lceil A =< 1 >$:

$$\begin{aligned}
&P(\llbracket s\lceil A \rrbracket \mid E_1) = \\
&= P(\{< 0,1,1 >, < 1,0,1 >\} \mid \{< 0,0,0 >, < 0,1,1 >, < 1,0,1 >\}) = \\
&= \frac{P(\{< 0,1,1 >, < 1,0,1 >\})}{P(\{< 0,0,0 >, < 0,1,1 >, < 1,0,1 >\})} = \frac{1/2}{3/4} = \frac{2}{3} \\
&P(\llbracket s\lceil A \rrbracket \mid E_2) = \\
&= P(\{< 0,1,1 >, < 1,0,1 >\} \mid \{< 0,1,1 >, < 1,0,1 >, < 1,1,0 >\}) = \\
&= \frac{P(\{< 0,1,1 >, < 1,0,1 >\})}{P(\{< 0,1,1 >, < 1,0,1 >, < 1,1,0 >\})} = \frac{1/2}{3/4} = \frac{2}{3}.
\end{aligned}$$

Also ist die Bedingung erfüllt. Die Gültigkeit der Bedingung wäre jedoch für alle Eingabefunktionen $\mathcal{O}_B$ zu zeigen. □

Für $\mathcal{O}_{B_2}$ und $s\lceil A =< 0 >$ gilt nun:

$$\begin{aligned}
&P(\llbracket s\lceil A\rrbracket \mid E_1) = \\
&\quad = \quad P(\{< 0,0,0 >, < 1,1,0 >\} \mid \{< 0,0,0 >, < 0,1,1 >, < 1,0,1 >\}) = \\
&\quad = \quad \frac{P(\{< 0,0,0 >\})}{P(\{< 0,0,0 >, < 0,1,1 >, < 1,0,1 >\})} = \frac{1/8}{5/8} = \frac{1}{5} \\
&P(\llbracket s\lceil A\rrbracket \mid E_2) = \\
&\quad = \quad P(\{< 0,0,0 >, < 1,1,0 >\} \mid \{< 0,1,1 >, < 1,0,1 >, < 1,1,0 >\}) = \\
&\quad = \quad \frac{P(\{< 1,1,0 >\})}{P(\{< 0,1,1 >, < 1,0,1 >, < 1,1,0 >\})} = \frac{3/8}{7/8} = \frac{3}{7} \neq \frac{1}{5}.
\end{aligned}$$

Die Bedingung ist für diese Eingabefunktion nicht erfüllt. Die Ereignisse sind somit nicht informationstheoretisch geheim gegenüber A. $\square$

5.1.2 Eigenschaften der absoluten informationstheoretischen Geheimhaltung

Die folgenden Lemmata zählen einige Eigenschaften der informationstheoretischen Geheimhaltung auf. Wie schon in Anm. 5.5 angemerkt, steht im folgenden P für P_{Σ_n}.

Lemma 5.2 *Für $A = \emptyset$ sind alle Mengen $\mathcal{E}$ von Ereignissen absolut informationstheoretisch geheim.*

Beweis: $A = \emptyset \Rightarrow \llbracket s\lceil A\rrbracket = I\!\!E \Rightarrow P(\llbracket s\lceil A\rrbracket \mid E) = \frac{P(I\!\!E \cap E)}{P(E)} = 1.$ $\blacksquare$

Lemma 5.3 *Falls $\mathcal{E}$ absolut informationstheoretisch geheim gegenüber A ist, so auch gegenüber $A' \subseteq A$.*

Beweis: Sei $\mathcal{E}$ informationstheoretisch geheim. Es gilt für alle s:

$$\llbracket s\lceil B\rrbracket = \biguplus_{s' \in \llbracket s\lceil A\rrbracket, s'\lceil A' = s\lceil A'} \llbracket s'\lceil A\rrbracket$$

Also gilt:

$$\begin{aligned} P(\llbracket s\lceil B\rrbracket|E_1) &= \biguplus_{s'\in\llbracket s\lceil A\rrbracket, s'\lceil A'=s\lceil A'} P(\llbracket s'\lceil A\rrbracket|E_1) = \\ \text{Mit Vorr.:} & \\ &= \biguplus_{s'\in\llbracket s\lceil A\rrbracket, s'\lceil A'=s\lceil A'} P(\llbracket s'\lceil A\rrbracket|E_2) = \\ &= P(\llbracket s\lceil B\rrbracket|E_2) \blacksquare \end{aligned}$$

Lemma 5.4 *Ist $|\mathcal{E}| = 1$, so ist $\mathcal{E}$ absolut informationstheoretisch geheim gegenüber allen Angreifern.*

Beweis: Folgt direkt aus der Definition. ■

Lemma 5.5 *Falls $\mathcal{E}$ absolut informationstheoretisch geheim gegenüber A ist, so auch $\mathcal{E}' \subseteq \mathcal{E}$.*

Beweis: Folgt mit Lemma 5.4 aus der Definition. ■

Lemma 5.6

$$\mathtt{iuu}_A \subseteq I\!E \times I\!E$$

mit E_1 $\mathtt{iuu}_A$ E_2 gdw. $\{E_1, E_2\}$ absolut informationstheoretisch geheim gegenüber Angreifer A ist, ist eine Äquivalenzrelation.

Beweis: Reflexivität und Symmetrie folgen direkt aus der Definition.

Definition 5.1 kann wie folgt verdeutlicht werden:

$$\underset{\Sigma_B\in\mathcal{B}}{\forall}\ \underset{\Sigma_A\in\mathcal{A}}{\forall}\ \underset{E_1,E_2\in\mathcal{E}}{\forall} P(E_1) = 0 \vee P(E_2) = 0 \vee P(\llbracket s\lceil A\rrbracket \mid E_1) = P(\llbracket s\lceil A\rrbracket \mid E_2)$$

Sei E_1 $\mathtt{iuu}_A$ E_2 und E_2 $\mathtt{iuu}_A$ E_3 aber nicht E_1 $\mathtt{iuu}_A$ E_3 so gibt es ein $\mathcal{O}_B$ und ein $\mathcal{O}_A$ (Hier zusammen als $\mathcal{O}_1$ bezeichnet) so daß gilt:

$$P_{\mathcal{O}_1}(E_1) \neq 0 \wedge P_{\mathcal{O}_1}(E_2) = 0 \wedge P_{\mathcal{O}_1}(E_3) \neq 0 \wedge P_{\mathcal{O}_1}(\llbracket s\lceil A\rrbracket \mid E_1) \neq P_{\mathcal{O}_1}(\llbracket s\lceil A\rrbracket \mid E_3)$$

Für ein $\mathcal{O}_1$ gilt nach Voraussetzung an $\mathcal{E}$ $P_{\mathcal{O}_2}(E_2) > 0$. Hier gilt auch nach Voraussetzung:

$$P_{\mathcal{O}_2}([\![s\lceil A]\!] \mid E_1) = P_{\mathcal{O}_2}([\![s\lceil A]\!] \mid E_2) = P_{\mathcal{O}_2}([\![s\lceil A]\!] \mid E_3)$$

Für eine Mischung (wie im Beweis zu Lemma 4.1) von $\mathcal{O}_1$ und $\mathcal{O}_2$ gilt zusätzlich noch:

$$P_{\mathcal{O}_1}([\![s\lceil A]\!] \mid E_1) \neq P_{\mathcal{O}_1}([\![s\lceil A]\!] \mid E_3) \Diamond \blacksquare$$

Lemma 5.7 *$\mathcal{E}$ ist absolut informationstheoretisch geheim bezüglich A gdw.*

$$\underset{\{E_1,E_2\}\subseteq\mathcal{E}}{\forall} E_1 \operatorname{iuu}_A E_2$$

Beweis: Folgt bis auf den Fall $|\mathcal{E}| = 1$ (Lemma 5.4) direkt aus der Definition.

Anmerkung 5.7 *Ist $\{E_1, E_2\}$ informationstheoretisch geheim gegenüber A so folgt hieraus nicht, daß $\{E_1', E_2'\}$ mit $E_1' \subseteq E_1$ und $E_2' \subseteq E_2$ informationstheoretisch geheim gegenüber A ist (Folgt mit $E_1 = E_2 = I\!\!E$ und Beispiel 5.5 sofort).*

Anmerkung 5.8 *Sind $\mathcal{E}_1$ und $\mathcal{E}_2$ informationstheoretisch geheim gegenüber A so folgt hieraus nicht daß $\mathcal{E}_1 \cup \mathcal{E}_2$ informationstheoretisch geheim sind (Lemma 5.4).*

Aus Lemma 5.6 folgt jedoch, daß falls $\mathcal{E}_1$ und $\mathcal{E}_2$ informationstheoretisch geheim gegenüber A sind und für beliebige $E_1 \in \mathcal{E}_1$ und $E_2 \in \mathcal{E}_2$ gilt $E_1 \operatorname{iuu}_A E_2$ so folgt hieraus, daß $\mathcal{E}_1 \cup \mathcal{E}_2$ informationstheoretisch geheim sind.

5.1.3 Alternative Definition der absoluten informationstheoretischen Geheimhaltung

Def. 5.2 (Absolute informationstheoretische Geheimhaltung) *Gegeben seien ein Kryptosystem (Σ_K, U) und eine Benutzerinstanz B. Sei*

$$\mathcal{B} = \mathcal{B}((\Sigma_K, U), (\mathcal{F}_{\mathsf{all}}, B)).$$

Eine Menge von möglichen Ereignissen $\mathcal{E}$ eines Systems Σ heißt absolut informationstheoretisch geheim gegenüber einer Angreiferinstanz A gdw. gilt:

$$\underset{\Sigma_B \in \mathcal{B}}{\forall} \; \underset{\Sigma_A}{\forall} \; \underset{s \in \bigcup_{E \in \mathcal{E}} E}{\forall} \; \underset{\substack{E_1, E_2 \in \mathcal{E} \\ P_{\Sigma_n}(E_1) > 0 \\ P_{\Sigma_n}(E_2) > 0}}{\forall} \; \frac{P_{\Sigma_n}(E_1 \,|\, [\![s\lceil A]\!])}{P_{\Sigma_n}(E_2 \,|\, [\![s\lceil A]\!])} = \frac{P_{\Sigma_n}(E_1)}{P_{\Sigma_n}(E_2)}.$$

mit:

$$\Sigma_n := \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A,n}$$

und

$$\mathcal{A} = \mathcal{A}((\Sigma_K, U), B, (\mathcal{F}, \emptyset)).$$

Lemma 5.8 *Definition 5.2 ist äquivalent zu Definition 5.1.*

Beweis: (Hier ohne Quantoren mit $P([\![s\lceil A]\!]) \neq 0$):

$$\begin{aligned}
\frac{P(E_1 \,|\, [\![s\lceil A]\!])}{P(E_2 \,|\, [\![s\lceil A]\!])} &= \frac{P(E_1)}{P(E_2)} \\
\Leftrightarrow \quad \frac{P(E_1 \cap [\![s\lceil A]\!]) P([\![s\lceil A]\!])}{P(E_2 \cap [\![s\lceil A]\!]) P([\![s\lceil A]\!])} &= \frac{P(E_1)}{P(E_2)} \\
\Leftrightarrow \quad \frac{P(E_1 \cap [\![s\lceil A]\!])}{P(E_1)} &= \frac{P(E_2 \cap [\![s\lceil A]\!])}{P(E_2)} \qquad (5.1) \\
\Leftrightarrow \quad P([\![s\lceil A]\!] \,|\, E_1) &= P([\![s\lceil A]\!] \,|\, E_2). \; \blacksquare
\end{aligned}$$

5.1.4 Absolute komplexitätsth. Geheimhaltung (A:AK, B:AKa)

Eine Menge von Ereignissen ist absolut komplexitätstheoretisch geheim, wenn bei jedem Benutzerverhalten ein Angreifer nur die Informationen über das eingetretene Ereignis erhält, die ihm direkt von den Benutzern mitgeteilt worden ist.

Diese Unterscheidung ist notwendig, da beim komplexitätsstheoretischen Modell für stärkere Angriffe ein Kanal zwischen Angreifer und Benutzern vorhanden sein muß.

Informationsaustausch darf hierbei nur über diese Steuerkanäle erfolgen. Ist Informationsfluß über andere Kanäle möglich, so ist das System nicht sicher.

Def. 5.3 (Absolute komplexitätstheoretische Geheimhaltung) *Gegeben seien ein Kryptosystem* (Σ_K, U) *und eine Benutzerinstanz* B *Sei*

$$\mathcal{B} = \mathcal{B}((\Sigma_K, U), (\mathcal{P}_{\text{all}}, B)).$$

Eine Menge $\mathcal{E}$ *von möglichen Ereignissen heißt absolut komplexitätstheoretisch geheim gdw. gilt:*

$$\underset{\Sigma_B \in \mathcal{B}}{\forall} \ \underset{c>0}{\forall} \ \underset{\Sigma_A \in \mathcal{A}}{\forall} \ \underset{\Sigma_{A'_U} \in \mathcal{A}_U}{\exists} \ \underset{n_0}{\exists} \ \underset{n>n_0}{\forall} \ \underset{\substack{E_1, E_2 \in \mathcal{E} \\ P_{\Sigma_n}(E_1) > 0 \\ P_{\Sigma_n}(E_2) > 0}}{\forall}$$

$$\frac{P_{\Sigma_n}([\![r=1]\!]|E_1)}{P_{\Sigma_n}([\![r=1]\!]|E_2)} < \frac{P_{\Sigma'_n}([\![r=1]\!]|E_1)}{P_{\Sigma'_n}([\![r=1]\!]|E_2)} + \frac{1}{n^c}$$

mit:

$$\begin{aligned}
\Sigma_n &:= \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A,n}, \\
\Sigma'_n &:= \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A_U,n}, \\
\mathcal{A} &= \mathcal{A}((\Sigma_K, U), B, (\mathcal{P}, \emptyset)), \\
\mathcal{A}_U &= \mathcal{A}_U((\Sigma_K, U), B, (\mathcal{P}, \emptyset))
\end{aligned}$$

und

$$\Sigma_{A_U} = (S_{A'_U}, A'_U, \mathcal{O}_{A'_U}).$$

Diese Definition sagt aus, daß kein polynomieller Angreifer existiert, der zwei Ereignisse besser unterscheiden kann als Angreifer, die nur die Informationen ausnutzen, die direkt vom Benutzer stammen (Σ_A und $\Sigma_{A'}$ sollen hier als „distinguisher" zwischen E_1 und E_2 dienen).

Lemma 5.9 (Komplexitätsth. Geheimhaltung für $P(\mathcal{E}) = 1$) *Sei*

$$\mathcal{B} = \mathcal{B}((\Sigma_K, U), (\mathcal{P}_{\text{all}}, B)).$$

Gilt $P(\mathcal{E}) = 1$ *so gilt Def. 5.3 gdw. gilt:*

$$\underset{\Sigma_B \in \mathcal{B}}{\forall} \ \underset{c>0}{\forall} \ \underset{\Sigma_A \in \mathcal{A}}{\forall} \ \underset{\Sigma_{A'_U} \in \mathcal{A}_U}{\exists} \ \underset{n_0}{\exists} \ \underset{n>n_0}{\forall} \ \underset{\substack{E_1, E_2 \in \mathcal{E} \\ P_{\Sigma_n}(E_1) > 0 \\ P_{\Sigma_n}(E_2) > 0}}{\forall}$$

$$P_{\Sigma_n}([\![r=1]\!]|E) < P_{\Sigma'_n}([\![r=1]\!]|E) + \frac{1}{n^c} \tag{5.2}$$

mit

$$\begin{aligned}\Sigma_n &:= \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A,n},\\ \Sigma'_n &:= \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A_U,n},\\ \mathcal{A} &= \mathcal{A}((\Sigma_K, U), B, (\mathcal{P}, \emptyset)),\\ \mathcal{A}_U &= \mathcal{A}_U((\Sigma_K, U), B, (\mathcal{P}, \emptyset))\end{aligned}$$

und

$$\Sigma_{A_U} = (S_{A'_U}, A'_U, \mathcal{O}_{A'_U}).$$

Beweis: Analog zu Beweis von Lemma 5.10.

Beispiel 5.6 (Def. 5.1 ist nicht schwächer als Def. 5.3) *Ein Konzelationssystem mit einem kryptografisch starken Pseudozufallszahlengenerator kann absolute komplexitätstheoretische Geheimhaltung bieten, nicht jedoch absolute informationstheoretische Geheimhaltung.*

Begründung: Komplexitätstheoretisch ist die Zufallszahlenfolge nicht vorhersagbar (Def. „kryptografisch stark"). Informationstheoretisch kann ein Benutzer festgelegte Nachrichten senden und der Angreifer kann sämtliche Startwerte des Pseudozufallszahlengenerators probieren und prüfen, ob eine Entschlüsselung den festgelegten Text ergibt (der Angreifer kennt diesen). □

5.2 Relative Geheimhaltung

Ziel einer Definition für relative Geheimhaltungseigenschaften ist, Informationsfluß vom Angreifer zu den Benutzern zuzulassen. Hierbei soll die Geheimhaltung eines Systems nicht verringert werden, wenn Benutzer dem Angreifer „freiwillig" Geheimnisse verraten. Die hier modellierten Angreifer haben Angriffspunkte im System und besetzen eventuell auch Teile der Benutzerschnittstelle. Über die Kanäle der Benutzerschnittstelle sollen die Benutzer die Möglichkeit besitzen, dem Angreifer beliebige Nachrichten zuzusenden (auch die geheime) ohne daß dann das System als unsicher bezeichnet werden muß. Die vorgestellte Definition ist vom Aufbau ähnlich wie Def. 5.3 auf Seite 91 (absolute komplexitätstheoretische Geheimhaltung).

Relative Geheimhaltung sichert Geheimhaltung bei beliebigem Benutzerverhalten. Da dies oft zu stark ist, fordert die schwache Geheimhaltung, die in Abschnitt 5.3 behandelt wird, nur Geheimhaltung bei einigen Benutzern.

5.2.1 Relative informationstheoretische Geheimhaltung (A:RI, B:AI)

Def. 5.4 (Relative informationstheoretische Geheimhaltung) *Gegeben sei ein Kryptosystem* (Σ_K, U)*, ein Benutzer* $(\mathcal{C}_F, B)$ *und ein Angreifer* $(\mathcal{F}, A_U)$*. Sei*

$$\mathcal{B} = \mathcal{B}((\Sigma_K, U), (\mathcal{C}_F, B)).$$

Eine Menge von möglichen Ereignissen $\mathcal{E}$ *heißt relativ informationstheoretisch geheim bei beliebigem Benutzerverhalten gdw. gilt:*

$$\underset{\Sigma_B \in \mathcal{B}}{\forall}\ \underset{\Sigma_A \in \mathcal{A}}{\forall}\ \underset{\Sigma_{A'_U} \in \mathcal{A}_U}{\exists}\ \underset{n}{\forall}\ \underset{s \in \bigcup\limits_{E \in \mathcal{E}} E}{\forall}\ \underset{\substack{E_1, E_2 \in \mathcal{E} \\ P_{\Sigma_n}(E_1) > 0 \\ P_{\Sigma_n}(E_2) > 0}}{\forall}$$

$$\frac{P_{\Sigma_n}(E_1 | [\![s \lceil A]\!])}{P_{\Sigma_n}(E_2 | [\![s \lceil A]\!])} = \frac{P_{\Sigma'_n}(E_1 | [\![s \lceil A'_U]\!])}{P_{\Sigma'_n}(E_2 | [\![s \lceil A'_U]\!])}$$

mit:

$$\Sigma_n := \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A,n},$$

$$\Sigma'_n := \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A_U,n},$$

$$\mathcal{A} = \mathcal{A}((\Sigma_K, U), B, (\mathcal{F}, A_U)),$$

$$\mathcal{A}_U = \mathcal{A}_U((\Sigma_K, U), B, (\mathcal{F}, A_U))$$

und

$$\Sigma_{A_U} = (S_{A'_U}, A'_U, \mathcal{O}_{A'_U}).$$

Es könnte hiermit ein Nachrichtentext in einem Netz geschützt werden. Sendet ein Benutzer dem Angreifer Nachrichten, verringert dies die Sicherheit nicht. Tauschen zwei Benutzer untereinander Nachrichten aus, so darf der Angreifer keine Information über den Nachrichteninhalt erhalten. Es ist jedoch trotzdem möglich, daß der Angreifer durch die Beobachtung der Leitungen erfährt, daß ein Ereignis aus $\mathcal{E}$ stattgefunden hat. War $P(\mathcal{E}) < 1$ müssen sich laut Definition alle bedingte Wahrscheinlichkeiten um den gleichen Faktor erhöhen, um die Definition zu erfüllen:

Lemma 5.10 (Relative infoth. Geheimhaltung bei $P(\mathcal{E}) = 1$)
Gelte Anm. 4.3. Gilt $P(\mathcal{E}) = 1$ so gilt Def. 5.4 gdw. gilt:

$$\forall_{\mathcal{O}_B \in \mathcal{B}} \forall_{\mathcal{O}_A \in \mathcal{A}} \exists_{\mathcal{O}_{A'_U} \in \mathcal{A}_U} \forall_{n} \forall_{s \in \bigcup_{E \in \mathcal{E}} E} \forall_{E \in \mathcal{E}} P_{\Sigma_n}(E | [\![s \lceil A]\!]) = P_{\Sigma'_n}(E | [\![s \lceil A_U]\!]) \quad (5.3)$$

mit:

$$\Sigma_n := \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A,n},$$

$$\Sigma'_n := \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A_U,n},$$

$$\mathcal{A} = \mathcal{A}((\Sigma_K, U), B, (\mathcal{F}, A_U))$$

und

$$\mathcal{A}_U = \mathcal{A}_U((\Sigma_K, U), B, (\mathcal{F}, A_U)),$$

wobei $\Sigma_A = (S_A, A, \mathcal{O}_A)$ und $\Sigma_{A_U} = (S_{A'_U}, A'_U, \mathcal{O}_{A'_U})$ die Angreifer zu (Σ_K, U), Σ_B und A_U sind.

Beweis: Def. 5.4 folgt direkt aus Formel 5.3. Zu zeigen ist also, daß für $|\mathcal{E}| > 1$ und $P(\mathcal{E}) = 1$ der Umkehrschluß gilt:

Gelte $P(\mathcal{E}) = 1$, Def. 5.4 aber nicht Formel 5.3, so gilt für ein $c \in I\!R; c > 0$ und alle $E \in \mathcal{E}$:

$$P_{\Sigma_n}(E | [\![s \lceil A]\!]) = c \cdot P_{\Sigma_n}(E | [\![s \lceil A'_U]\!])$$

Hieraus folgt mit $P_{\Sigma_n}(\mathcal{E} | [\![s \lceil A]\!]) = 1$ daß $P_{\Sigma'_n}(\mathcal{E} | [\![s \lceil A'_U]\!]) > 1.\diamondsuit$ ∎

Beispiel 5.7 (Def. 5.1 ist nicht schwächer als Def. 5.4)
Die Ereignismenge $\{E_1, E_2\}$ mit $E_1 = [\![i_{b,1} = 0]\!]$ und $E_2 = [\![i_{b,1} = 1]\!]$ ist im System aus Abbildung 5.3 nicht Informationstheoretisch geheim und trotzdem relativ informationstheoretisch geheim.

Die Ereignismenge ist im Beispielsystem nicht absolut informationstheoretisch geheim, da bei einem Benutzer, der in beide Eingabekanäle identische Befehle eingibt, beide Ereignisse für den Angreifer unterscheidbar werden.

Anhand folgender Systemdefinition soll nun Geheimhaltung nach Def. 5.4 gezeigt werden: $In = \{a, b\}$, $Out = \{c, d\}$, $\mathbb{T}$=$\{1\}$, $U = \{a, b, c\}$, $\mathcal{O}$ so daß $c_t = a_t$ und $d_t = b_t \oplus U\{0, 1\}$, $\Gamma = \{0, 1\}$. $A = \{(c, 1), (d, 1)\}$, $A_U = \{(c, 1)\}$ und $\mathcal{O}_A = \emptyset$, da der Angreifer keine Eingabekanäle kontrolliert.

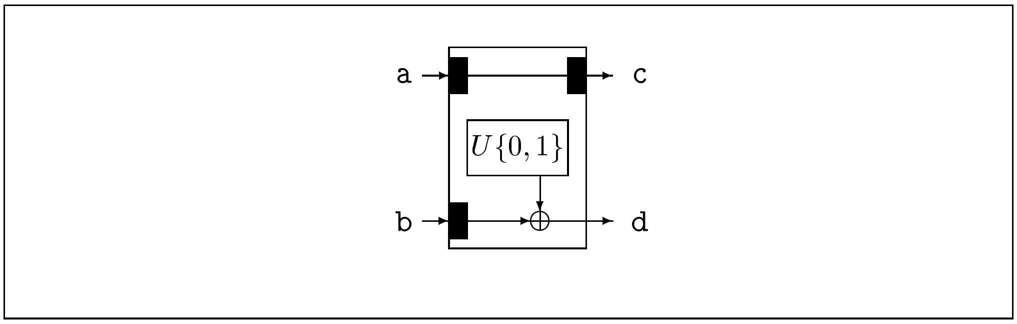

Abbildung 5.3: Beispielsystem zu Def. 5.4: Onetime-Pad und direkter Kanal zum Angreifer ($U\{0,1\}$ ist perfekter $\{0,1\}$–Zufallszahlengenerator, $\oplus$ ein xor-Gatter)

Für $A = \{c_1, d_1\}$ gilt mit beliebigem $\mathcal{O}_B$:

$$\frac{P_{\Sigma_n}(E_1|[\![< o_{c,1}, o_{d,1} >]\!])}{P_{\Sigma_n}(E_2|[\![< o_{c,1}, o_{d,1} >]\!])} =$$

$$= \frac{\frac{P_{\Sigma_n}([\![i_{b,1}=1]\!]\cap[\![<o_{c,1},o_{d,1}>]\!])}{P_{\Sigma_n}([\![<o_{c,1},o_{d,1}>]\!])}}{\frac{P_{\Sigma_n}([\![i_{b,1}=0]\!]\cap[\![<o_{c,1},o_{d,1}>]\!])}{P_{\Sigma_n}([\![<o_{c,1},o_{d,1}>]\!])}}$$

Mit stochastischer Unabh. von d und Anm. 4.3:

$$= \frac{P_{\Sigma'_n}([\![i_{b,1} = 1]\!] \cap [\![< o_{c,1} >]\!])P_{\Sigma'_n}([\![< o_{d,1} >]\!])}{P_{\Sigma'_n}([\![i_{b,1} = 0]\!] \cap [\![< o_{c,1} >]\!])P_{\Sigma'_n}([\![< o_{d,1} >]\!])}$$

$$= \frac{P_{\Sigma'_n}([\![o_{b,1} = 1]\!] \cap [\![< o_{c,1} >]\!])}{P_{\Sigma'_n}([\![i_{b,1} = 0]\!] \cap [\![< o_{c,1} >]\!])}$$

$$= \frac{P_{\Sigma'_n}(E_1|[\![< o_{c,1} >]\!])}{P_{\Sigma'_n}(E_2|[\![< o_{c,1} >]\!])}$$

Also sind die Ereignisse innerhalb dieses Systems relativ informationstheoretisch geheim. □

Wie wir an diesem Beispiel sehen konnten, ist Informationsfluß mit der Definition der relativen informationstheoretischen Geheimhaltung kontrollierbar! Obwohl Information vom Benutzer zum Angreifer übertragen werden konnte, war Geheimhaltung gewährleistet.

Um zu zeigen, daß die Definition der relativen informationstheoretischen Geheimhaltung trotzdem aussagekräftig ist, wird das Beispiel modifiziert:

Beispiel 5.8 *Wird der perfekte Zufallszahlengenerator in Beispiel 5.7 verunreinigt (z.B. mit $P(0) = 1/3$ und $P(1) = 2/3$), so ist das System nicht mehr relativ Informationstheoretisch sicher.*

Begründung: $o_{c,1}$ ist dann nicht mehr stochastisch unabhängig von $i_{b,1}$. Für einen Benutzer $\mathcal{O}_B$, der über $\mathtt{d}$ nichts verrät weiß Σ_A also mehr als $\Sigma_{A'_U}$. □

Lemma 5.11 *Ist $U \times \mathbb{T} \cap A = \emptyset$, so sind die Definitionen 5.1 und 5.4 äquivalent.*

Beweis: Folgt mit Lemma 5.8 direkt, da $P(E|[\![s\lceil\emptyset]\!]) = P(E)$ und $A' = \emptyset$ ■

5.2.2 Relative komplexitätstheoretische Geheimhaltung (A:RK, B:RK)

Eine Menge von Ereignissen ist relativ komplexitätstheoretisch geheim, wenn bei jedem Benutzerverhalten einer Benutzerklasse ein Angreifer nur die Informationen über das eingetretene Ereignis erhält, die ihm direkt von den Benutzern mitgeteilt worden ist.

Analog zur absoluten komplexitätstheoretischen Geheimhaltung sind zwei Kanäle zwischen Benutzer und Angreifer und ein Ergebniskanal vorhanden.

Diese Definition unterscheidet sich von der Definition der absoluten komplexitätstheoretischen Geheimhaltung dadurch, daß der Angreifer auch zusätzlich einen Teil der Benutzerschnittstellen besetzt hält. Hierdurch wird dem Benutzer ermöglicht, direkt mit dem Angreifer in Kontakt zu treten.

Ziel dieser Definition ist, daß ein Angreifer, der interne Informationen, die Information von mehreren Benutzerschnittstellen und die Information die direkt von den Benutzern erhält, nicht mächtiger ist, als ein Angreifer der keinen Zugriff auf die internen Informationen hat.

Def. 5.5 (Relative komplexitätstheoretische Geheimhaltung)
Gegeben sei ein Kryptosystem (Σ_K, U), ein Benutzer $(\mathcal{C}_F, B)$ und ein Angreifer $(\mathcal{P}, A_U)$. Sei

$$\mathcal{B} = \mathcal{B}((\Sigma_K, U), (\mathcal{C}_F, B)).$$

Eine Menge von möglichen Ereignissen $\mathcal{E}$ heißt relativ komplexitätstheoretisch geheim bei beliebigem Benutzerverhalten gdw. gilt:

$$\underset{\Sigma_B \in \mathcal{B}}{\forall}\ \underset{c>0}{\forall}\ \underset{\Sigma_A \in \mathcal{A}}{\forall}\ \underset{\Sigma_{A'_U} \in \mathcal{A}_U}{\exists}\ \underset{n_0}{\exists}\ \underset{n>n_0}{\forall}\ \underset{\substack{E_1, E_2 \in \mathcal{E} \\ P_{\Sigma_n}(E_1) > 0 \\ P_{\Sigma_n}(E_2) > 0}}{\forall}$$

$$\frac{P_{\Sigma_n}([\![r=1]\!]|E_1)}{P_{\Sigma_n}([\![r=1]\!]|E_2)} < \frac{P_{\Sigma'_n}([\![r=1]\!]|E_1)}{P_{\Sigma'_n}([\![r=1]\!]|E_2)} + \frac{1}{n^c}$$

mit:

$$\Sigma_n := \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A,n},$$

$$\Sigma'_n := \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A_U,n},$$

$$\mathcal{A} = \mathcal{A}((\Sigma_K, U), B, (\mathcal{P}, A_U)),$$

$$\mathcal{A}_U = \mathcal{A}_U((\Sigma_K, U), B, (\mathcal{P}, A_U))$$

und

$$\Sigma_{A_U} = (S_{A'_U}, A'_U, \mathcal{O}_{A'_U}).$$

Diese Definition sagt aus, daß kein polynomieller Angreifer existiert, der zwei Ereignisse besser unterscheiden kann als ein Angreifer, der nur die Informationen ausnutzt, die direkt vom Benutzer und den Benutzerschnittstellen des Angreifers stammt (Σ_A und $\Sigma_{A'}$ sollen hier als „distinguisher" zwischen E_1 und E_2 dienen).

Beispiel 5.9 (Def. 5.3 ist nicht schwächer als als Def. 5.5) *Dies folgt aus Beispiel 5.7.*

5.3 Schwache Geheimhaltung

Die schwache Geheimhaltung entspricht prinzipiell der relativen Geheimhaltung. Es wird jedoch nur gefordert, daß ein nichttrivialer[1] Benutzer existiert für den Geheimhaltung gewährleistet ist.

[1] D.h. ein Benutzer bei dem alle Ereignisse auch eintreten können

Dies ist eine schwächere Forderung als die relative Geheimhaltung, da Geheimhaltungt gegen beliebig bösartige Benutzer nun nicht mehr garantiert wird.

Diese Forderung basiert auf der Annahme, daß wenn Geheimhaltung für einen nichttrivialen Benutzer gewährleistet ist, auch Geheimhaltung für einige Benutzer gegeben ist und somit das System nicht ganz unsicher sein kann.

5.3.1 Schwache informationsth. Geheimhaltung (A:RI, B:EI)

Def. 5.6 (Schwache informationstheoretische Geheimhaltung)
Gegeben seien ein Kryptosystem (Σ_K, U), ein Benutzer $(\mathcal{C}_B, B)$ und ein Angreifer $(\mathcal{F}, A_U)$. Sei

$$\mathcal{B} = \mathcal{B}((\Sigma_K, U), (\mathcal{C}_F, B)).$$

Eine Menge von möglichen Ereignissen $\mathcal{E}$ heißt schwach informationstheoretisch geheim gdw. gilt:

$$\underset{\Sigma_B \in \mathcal{B}}{\exists} \left(P_{\Sigma_B}(\mathcal{E}) > 0 \wedge \underset{\Sigma_A \in \mathcal{A}}{\forall} \underset{\Sigma_{A'_U} \in \mathcal{A}_U}{\exists} \underset{n}{\forall} \underset{s \in \bigcup\limits_{E \in \mathcal{E}} E}{\forall} \underset{\substack{E_1, E_2 \in \mathcal{E} \\ P_{\Sigma_n}(E_1) > 0 \\ P_{\Sigma_n}(E_2) > 0}}{\forall} \frac{P_{\Sigma_n}(E_1 | [\![s \lceil A]\!])}{P_{\Sigma_n}(E_2 | [\![s \lceil A]\!])} = \frac{P_{\Sigma'_n}(E_1 | [\![s \lceil A'_U]\!])}{P_{\Sigma'_n}(E_2 | [\![s \lceil A'_U]\!])} \right)$$

mit:

$$\Sigma_n := \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A,n},$$

$$\Sigma'_n := \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A_U,n},$$

$$\mathcal{A} = \mathcal{A}((\Sigma_K, U), B, (\mathcal{F}, A_U)),$$

$$\mathcal{A}_U = \mathcal{A}_U((\Sigma_K, U), B, (\mathcal{F}, A_U)),$$

$$P_{\Sigma_B}(\mathcal{E}) > 0 \iff \underset{n}{\forall} \underset{E \in \mathcal{E}}{\forall} P_{\Sigma''_n}(E) > 0$$

und

$$\Sigma_{A_U} = (S_{A'_U}, A'_U, \mathcal{O}_{A'_U}).$$

Beispiel 5.10 (Def. 5.6 ist nicht stärker als Def. 5.4) *Das System aus Bsp. 5.5 leistet keine relative informationstheoretische Geheimhaltung. Für gleichverteilte Eingaben ($\mathcal{O}_B$) erfüllt es jedoch die Definition der schwachen informationstheoretischen Geheimhaltung.*

5.3.2 Schwache komplexitätsth. Geheimhaltung (A:AK, B:EK)

Def. 5.7 (Schwache komplexitätstheoretische Geheimhaltung) *Gegeben seien ein Kryptosystem (Σ_K, U), ein Benutzer $(\mathcal{C}_B, B)$ und ein Angreifer $(\mathcal{P}, A_U)$. Sei*

$$\mathcal{B} = \mathcal{B}((\Sigma_K, U), (\mathcal{C}_F, B)).$$

Eine Menge von möglichen Ereignissen $\mathcal{E}$ heißt schwach komplexitätstheoretisch geheim gdw. gilt:

$$\underset{\Sigma_B \in \mathcal{B}}{\exists} \left(P_{\Sigma_B}(\mathcal{E}) > 0 \wedge \underset{c>0}{\forall} \; \underset{\Sigma_A \in \mathcal{A}}{\forall} \; \underset{\Sigma_{A'_U} \in \mathcal{A}_U}{\exists} \; \underset{n_0}{\exists} \; \underset{n>n_0}{\forall} \; \underset{\substack{E_1, E_2 \in \mathcal{E} \\ P_{\Sigma_n}(E_1) > 0 \\ P_{\Sigma_n}(E_2) > 0}}{\forall} \right.$$
$$\left. \frac{P_{\Sigma_n}([\![r=1]\!] | E_1)}{P_{\Sigma_n}([\![r=1]\!] | E_2)} < \frac{P_{\Sigma'_n}([\![r=1]\!] | E_1)}{P_{\Sigma'_n}([\![r=1]\!] | E_2)} + \frac{1}{n^c} \right)$$

mit:

$$\Sigma_n := \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A,n},$$

$$\Sigma'_n := \Sigma_{K,n} \bowtie_+ \Sigma_{B,n} \bowtie_+ \Sigma_{A_U,n},$$

$$\mathcal{A} = \mathcal{A}((\Sigma_K, U), B, (\mathcal{P}, A_U)),$$

$$\mathcal{A}_U = \mathcal{A}_U((\Sigma_K, U), B, (\mathcal{P}, A_U)),$$

$$P_{\Sigma_B}(\mathcal{E}) > 0 \iff \underset{n}{\forall} \; \underset{E \in \mathcal{E}}{\forall} P_{\Sigma''_n}(E) > 0$$

und

$$\Sigma_{A_U} = (S_{A'_U}, A'_U, \mathcal{O}_{A'_U}).$$

Beispiel 5.11 (Sicheres System nach Def. 5.7) *Ein Beispiel hierfür ist ein System, welches vom Benutzer fordert zuerst das System zu initialisieren, bevor er eine Nachricht sendet.*

Beispiel 5.12 (Def. 5.7 ist nicht stärker als Def. 5.5) *Ein System wie in Beispiel 5.11 kann schwache Geheimhaltung garantieren. Es bietet jedoch keine relative Geheimhaltung, da es für einen Benutzer, der nicht initialisiert, keine Geheimhaltung bietet.*

Analog hierzu ist ein System, welches einen internen Kanal von Σ_B *zu* Σ_A *besitzt, über den beide* bei Kooperation *Informationen übertragen können, nur schwach und nicht relativ sicher.*

5.4 Vergleich der drei Geheimhaltungsstufen

Ein Überblick über die Geheimhaltungsstufen zeigt Abbildung 5.4. Alle Stufen werden zusätzlich nach Benutzerklassen gegliedert.

$$\begin{tikzcd}
\text{AI (Def. 5.1)} \arrow[r, "(5.12)"] \arrow[d, "(5.15)"'] & \text{AK (Def. 5.3)} \arrow[d, "(5.17)"] \\
\text{RI (Def. 5.6)} \arrow[r, "(5.13)"] \arrow[d, "(5.16)"'] & \text{RK (Def. 5.5)} \arrow[d, "(5.18)"] \\
\text{SI (Def. 5.6)} \arrow[r, "(5.14)"] & \text{SK (Def. 5.7)}
\end{tikzcd}$$

I	Informationstheoretisch
K	Komplexitätstheoretisch
A	Absolut
R	Relativ
S	Schwach

Abbildung 5.4: Geheimhaltungsstufen (Beweis)

Lemma 5.12 *Definition 5.3 (Absolute komplexitätstheoretische Geheimhaltung von Ereignissen) ist echt schwächer als Definition 5.1 (Absolute informationstheoretische Geheimhaltung von Ereignissen).*

Beweisskizze: Gibt es keinen informationstheoretischen Angreifer, der Information über das stattgefundene Ereignis erhalten kann, so auch keinen polynomiell beschränkten. Mit Beispiel 5.6 folgt die Behauptung. ■

Lemma 5.13 *Definition 5.5 (Relative komplexitätstheoretische Geheimhaltung von Ereignissen) ist echt schwächer als Definition 5.4 (Relative informationstheoretische Geheimhaltung von Ereignissen).*

Beweisskizze: Analog zu Beweis Lemma 5.12. ■

Lemma 5.14 *Definition 5.7 (Schwache komplexitätstheoretische Geheimhaltung von Ereignissen) ist echt schwächer als Definition 5.6 (Schwache informationstheoretische Geheimhaltung von Ereignissen).*

Beweisskizze: Wenn es keinen nichttrivialen Benutzer gibt, für den das Ereignis bei relativem, komplexitätstheoretischem Angreifer geheim ist, so gibt es auch keinen Benutzer, für den das Ereignis bei relativem, informationstheoretischem Angreifer geheim ist. ■

Lemma 5.15 *Definition 5.1 (Absolute informationstheoretische Geheimhaltung von Ereignissen) ist echt schwächer als Definition 5.4 (Relative informationstheoretische Geheimhaltung von Ereignissen).*

Beweis: Mit Lemma 5.5 und der Voraussetzung gilt

$$P_{\Sigma'_n}([\![s\lceil A'_U]\!] \mid E_1) = P_{\Sigma'_n}([\![s\lceil A'_U]\!] \mid E_2)$$

mit Beispiel 5.7 folgt die Behauptung. ■

Lemma 5.16 *Definition 5.6 (Schwache informationstheoretische Geheimhaltung von Ereignissen) ist echt schwächer als Definition 5.4 (Relative informationstheoretische Geheimhaltung von Ereignissen).*

Beweis: Laut Definition 5.4 gibt es ein Σ_B mit $P(E) > 0$ für $E \in \mathcal{E}$ ($\mathcal{E}$ ist möglich). Für dieses Σ_B gilt auch Definition 5.6. Mit Beispiel 5.10 folgt die Behauptung. ■

Lemma 5.17 *Definition 5.5 (Relative komplexitätstheoretische Geheimhaltung von Ereignissen) ist echt schwächer als Definition 5.3 (Absolute komplexitätstheoretische Geheimhaltung von Ereignissen).*

Beweis: Analog zu Beweis von Lemma 5.15. Mit Beispiel 5.9 folgt die Behauptung. ■

Lemma 5.18 *Definition 5.7 (Schwache komplexitätstheoretische Geheimhaltung von Ereignissen) ist echt schwächer als Definition 5.5 (Relative komplexitätstheoretische Geheimhaltung).*

Beweis: Analog zu Beweis von Lemma 5.16. Mit Beispiel 5.12 folgt die Behauptung. ■

Kapitel 6

Service und Sicherheit von Konzelationssystemen

In diesem Abschnitt wird Service und Geheimhaltung von reaktiven kryptologischen Systemen am Beispiel von Konzelationssystemen[1] behandelt.

Allgemein gliedert sich die Beschreibung eines Systems in zwei Teile. Als erstes wird die Benutzerschnittstelle mit ihren Kanälen und Operationen angegeben. Dann wird angegeben, welche Ereignisse geheim und welche Ereignisse korrekt sein sollen um den Minimalanforderungen an diesen Systemtyp zu genügen.

Die geheimzuhaltenden und zu sichernden Ereignisse beziehen sich stets nur auf die Benutzerschnittstelle. Die Angreiferinstanzen der realen Systeme müssen von Fall zu Fall festgelegt werden.

Anmerkung 6.1 *Es werden im folgenden nur die geheimzuhaltenden und zu sichernden Ereignisse, nicht jedoch die Sicherheits– bzw. Geheimhaltungsstufe angegeben.*

Ein einfaches Konzelationssystem muß Datenübertragung gewährleisten.

Eine verschüsselte Nachricht ist an der Benutzerschnittstelle nicht sichtbar, da mit diesem Modell auch mehrstufige, interaktive Konzelationsprotokolle erfasst werden sollen (mit geeigneter Übertragungsfunktion auch Brassards

[1] Für Signatursysteme vgl. [Pfi93b]

„quantum cryptography“ [Bra88, Kap. 6]). Um die Anwendbarkeit auf zukünftige Systeme sicherzustellen, wurden nur minimale Eigenschaften modelliert (u.a. auch keine Adressierung).

6.1 Schnittstelle

Die Benutzerschnittstelle des Kryptosystems besitzt zwei Kanäle `s` (wie send) und `re`[2] (wie receive). Zusätzlich hierzu kann das System noch eine Menge weiterer Kanäle L besitzen (insbes. „für“ den Angreifer).

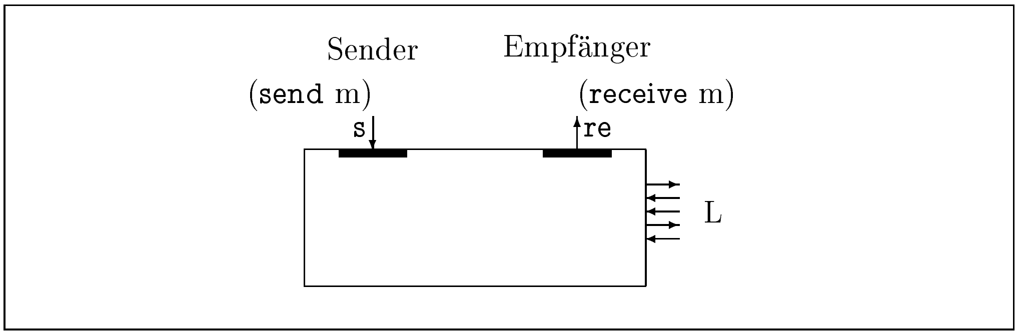

Abbildung 6.1: Schnittstelle eines Konzelationssystems

Def. 6.1 (Schnittstelle eines Konzelationssystems) *Sei $M \neq \emptyset$ abzählbarer Nachrichtenraum. $S = < In, Out, \Gamma, \mathbb{T} >$ mit $s \in In$ und $re \in Out$, $\{\texttt{send}, \texttt{receive}\} \times M \subseteq \Gamma^+$ heißt Schnittstelle eines Konzelationssystems, wobei* `s` *der Sendekanal und* `re` *der Empfangskanal ist. $U \supseteq \{s, re\}$ sei die Benutzerschnittstelle.*

Anmerkung 6.2 *Die Forderung $U \supseteq \{s, re\}$ bezweckt, daß auch Systeme mit einer umfangreicheren Benutzerschnittstelle als Konzelationssysteme bezeichnet werden können, wenn sie Service und Geheimhaltung von Konzelationssystemen bieten.*

Abbildung 6.1 zeigt die Schnittstelle des Konzelationssystems. Instanzen sind Sender $\{s\} \times \mathbb{T}$ und Empfänger $\{re\} \times \mathbb{T}$.

[2]Zur Unterscheidung vom Angreiferantwortkanal `r`.

Als Operationen stehen `send` und `receive` zur Verfügung. Argumente von `send` , `receive` ist eine Nachricht $m \in M$.

Def. 6.2 (Service eines Konzelationssystem) *Sei* $M \neq \emptyset$ *abzählbarer Nachrichtenraum. Sei* $S =< In, Out, \Gamma, \mathbb{T} >$ *Schnittstelle eines Konzelationssystems. Das Kryptosystem* $\Sigma = (S, U, \mathcal{O})$ *mit* $\{s, re\} \times \mathbb{T} \subseteq U$ *bietet den Service eines Konzelationssystem gdw. das folgende Ereignis sicher (vgl. Abschnitt 4.7) ist (p sei Polynom):*

$$[\![\underset{p}{\exists} \underset{t}{\forall} \underset{m \in M}{\forall} i_{s,t} = \texttt{send}\ m \rightarrow \underset{t' < t + p(|m|)}{\exists} o_{re,t'} = \texttt{receive}\ m]\!]$$

Nach dieser Definition muß ein Konzelationssystem eine eingegebene Nachricht nur übertragen. Sie sagt aus, daß falls ein Sender eine Nachricht versendet, diese Nachricht in polynomieller Zeit an den Empfänger weitergeleitet wird. Sie sagt nicht aus, daß eine Nachricht, die empfangen wird, vorher gesendet wurde, da dies bereits ein Dienst eines Authentikationssystems ist. Ein Angreifer dürfte somit beliebige Nachrichten an den Empfänger senden. Da jede `send`–Operation Empfang zur Folge haben muß, wird auch Verfügbarkeit erzwungen.

6.2 Ereignismengen

In der folgenden Definition wird von einem Konzelationssystem nur gefordert, daß es den Nachrichteninhalt geheim hält. Andere Anforderungen, wie zum Beispiel Geheimhaltung der Länge der Nachricht, wurden in Abschnitt 4.6.2 ab Seite 76 ausführlich behandelt.

6.3 Geheimhaltung

Die Geheimhaltung eines Konzelationssystems wird nach dem Muster der modularen Geheimhaltungsdefinition aus Kapitel 4 aufgebaut. Hierbei muß die Art und Verbreitung des Angreifers und die Art der Benutzer bekannt sein. Hierbei ist zu beachten, daß für Konzelationssysteme spezielle Benutzerklassen gebräuchlich sind, die im folgenden Abschnitt aufgezählt werden.

Def. 6.3 (Geheimhaltung eines Konzelationssystems) *Ein Kryptosystem mit einer Schnittstelle eines Konzelationssystems und einem abzählbaren Nachrichtenraum M bietet die Geheimhaltung eines Konzelationssystems, wenn die Ereignisse der folgenden Ereignismengen $\mathcal{E}_{i,t}$; $i \in I\!N$ und $t \in \mathbb{T}$ geheim sind:*

$$\mathcal{E}_{i,t} := \{[\![i_{s,t} = \texttt{send } m]\!] | m \in M \quad und \quad |m| = i\}$$

6.4 Benutzerklassen für Konzelationssysteme

In diesem Abschnitt werden einige in der Literatur vorgeschlagene Angriffsarten auf das hier angegebene (erweiterte) Konzelationssystemmodell übertragen. Sie werden hier als Benutzerklassen modelliert (vgl. auch Abschnitt 3.2.2 ab Seite 41). Die hier definierten Prädikate werden dazu verwendet die Benutzermenge einzuschränken und die Betrachtung somit auf Benutzer mit bestimmten Eigenschaften zu beschränken. Abb. 6.2 gliedert die Angriffe nach ihrer Stärke ([Den82, Pfi92] bzw. [Pfi90, S. 20]). Zu beachten ist hierbei, daß die Klassen nicht in Inklusionsbeziehung zueinander stehen.

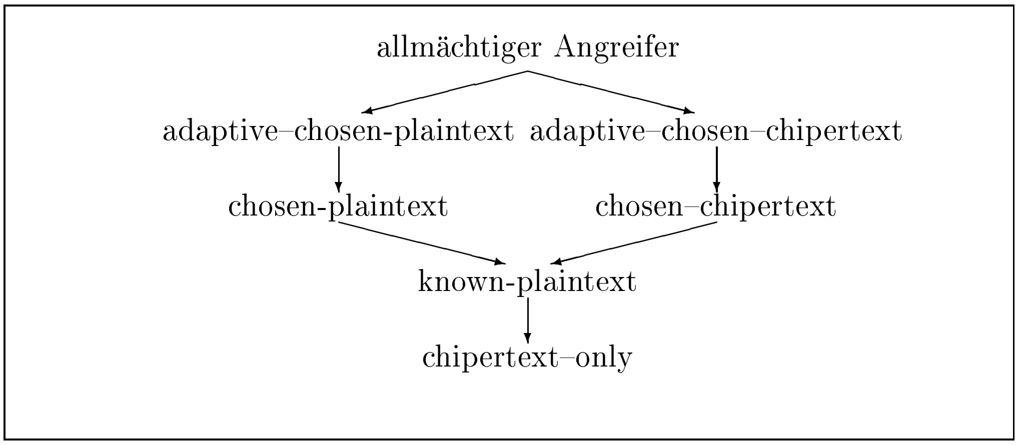

Abbildung 6.2: Vergleich der Angriffsarten bei Konzelationssystemen (A→B, wenn Angriffsart A stärker als Angriffsart B)

Nach Definition von zwei speziellen Benutzerklassen, werden weitere anhand der allgemeinen Benutzerklassen aus Abschnitt 3.2.2 hergeleitet. Die Gültigkeit des Prädikats garantiert, daß dieser und kein stärkerer Angriff möglich ist. Gilt das Prädikat nicht, so kann die entsprechende Angriffsart trotzdem möglich sein. `a` sei der Kanal vom Angreifer zu den Benutzern, `b` der Kanal von den Benutzern zum Angreifer.

6.4.1 Klartext-Schlüsseltext-Angriff

Beim Klartext-Schlüsseltext-Angriff werden dem Angreifer einige Klartext-Schlüsseltextpaare mitgeteilt. Der Angreifer hat außerdem die Möglichkeit, beliebig viel Schlüsseltext zu beobachten. Er kann jedoch nicht bestimmen, welche Klartext-Schlüsseltext-Paare er erhält.

Dies bedeutet, daß der Benutzer den Klartext zu einigen Schlüsseltexten weitergibt. Da jedoch nicht festgelegt ist, wieviel Schlüsseltext weitergegeben wurde, wird hier der Begriff auf einen Benutzer eingeschränkt, der genau n Klartexte an den Angreifer weitergibt. Auf Anweisungen des Angreifers reagiert er nicht.

Def. 6.4 (known-plaintext)

$$
\begin{aligned}
&\texttt{known-plaintext(n)} \Leftrightarrow \\
\underset{t_1<t_2<\ldots<t_n\in T|}{\exists} \quad & o_{s,t_i} = \texttt{send}\ m_i \rightarrow o_{b,t_n+1} = \texttt{send}\ < (t_1, m_1), \ldots, (t_n, m_n) > \\
\wedge \quad & \underset{t' \neq t_n+1}{\forall} o_{b,t'} = \texttt{nop} \\
\wedge \quad & \underset{t}{\forall} P([\![s\lceil(B \cap Out[t+1])]\!] | [\![s\lceil\{a\}[1,t']]\!]) \\
& \quad = P([\![s\lceil(B \cap Out[t+1])]\!])
\end{aligned}
$$

6.4.2 Reiner Schlüsseltextangriff

Beim reinen Schlüsseltextangriff kann der Angreifer nur Schlüsseltexte beobachten. Dies ist gesichert, wenn jegliche Verbindung zwischen Benutzer und Angreifer unterbunden wird. Dies ist die schwächste Angriffsart.

Def. 6.5 (ciphertext-only)

$$\textsf{ciphertext-only} \Leftrightarrow \textsf{passive}$$

6.4.3 Gewählter-Klartext-Schlüsseltext-Angriff

Beim gewählten-Klartext-Schlüsseltext-Angriff kann der Angreifer einige Klartexte verschlüsseln lassen. Vorher und anschließend kann er weitere Schlüsseltexte beobachten.

Def. 6.6 (chosen-plaintext)

$$\texttt{chosen-plaintext} \Leftrightarrow \texttt{chosen-input} \wedge \texttt{non-adaptive} \wedge \texttt{no-chosen-output}$$

6.4.4 Adaptiver Gewählter-Klartext-Schlüsseltext Angriff

Beim adaptiven gewählter-Klartext-Schlüsseltext-Angriff darf der Angreifer mehr als einmal Aufträge an den Benutzer erteilen. Weitere Anweisungen können auch von den bisherigen Beobachtungen abhängen.

Def. 6.7 (adaptive-chosen-plaintext)

$$\texttt{adaptive-chosen-plaintext} \Leftrightarrow \texttt{chosen-input} \wedge \texttt{no-chosen-output}$$

6.4.5 Gewählter-Schlüsseltext-Klartext-Angriff

Der gewählter-Schlüsseltext-Klartext-Angriff entspricht dem gewählten-Klartext-Schlüsseltext-Angriff. Es werden jedoch nun zu vorgegebenen Schlüsseltexten die Klartexte an den Angreifer weitergeleitet. Zu beachten ist hier, daß es beim hier vorgestellten allgemeinen Modell keine „Schlüsseltexte“ im engeren Sinn gibt. Der Angreifer muß also Aktionen durchführen und das Resultat beobachten.

Die folgenden Definition ist somit als einzige nicht unabhängig von der Ereignismenge $\mathcal{E}$.

Def. 6.8 (chosen-ciphertext)

$$\begin{aligned}&\texttt{chosen-ciphertext}(\mathcal{E}) \Leftrightarrow \\ &\quad \texttt{chosen-output}(\mathcal{E}) \wedge \texttt{non-adaptive} \wedge \texttt{no-chosen-input}\end{aligned}$$

6.4.6 Adaptiver Gewählter-Schlüsseltext-Klartext Angriff

Diese Angriffsart entspricht dem Gewählter-Schlüsseltext-Klartext-Angriff. Es können jedoch beliebig viele Abfragen gestartet werden.

Def. 6.9 (adaptive-chosen-ciphertext)

$$\texttt{adaptive-chosen-ciphertext}(\mathcal{E}) \Leftrightarrow \texttt{chosen-output}(\mathcal{E}) \wedge \texttt{no-chosen-input}$$

6.4.7 Gleichverteilte Ereignisse

Folgende Klasse wird für die Nachbildung von bestehenden Definitionen benötigt:

$$\texttt{uniformly-distributed} \Leftrightarrow \underset{E_1, E_2 \in \mathcal{E}}{\forall} P(E_1) = P(E_2)$$

Anmerkung 6.3 *Eine derartige Benutzerklasse existiert nur zu Ereignismengen, deren Einzelwahrscheinlichkeiten nur vom Benutzerverhalten abhängen.*

Ein Beispiel wäre das Senden von Texten fester Länge. Die Benutzer dieser Klasse wählen die Texte gleichverteilt aus.

6.5 Definitionsansätze für zwei spezielle Konzelationssystemtypen

In diesem Abschnitt werden zwei Arten von Konzelationssystemen (Public– und Private–Key–Konzelationssysteme) genauer spezifiziert. Hierzu wird der interne Systemaufbau von Konzelationssystemen exemplarisch behandelt. Die beiden Systemtypen werden sich nur durch einen sicheren Schlüsselübertragungskanal unterscheiden. Abbildung 6.3 zeigt den schematischen Systemaufbau.

6.5.1 Public–Key Konzelationssysteme

Die Idee der folgenden Definition ist, daß das System aus drei Komponenten bestehen muß, die nur über vom Angreifer beobachtete Leitungen verbunden sind.

Anmerkung 6.4 Σ_L *ist Übertragungsfunktion und sollte ggf. folgende Bedingungen erfüllen:*

1. *Um die Diensterbringung (Verfügbarkeit) zu sichern, darf es dem Angreifer durch seine Eingaben über* `g` *nicht möglich sein, die Übertragung völlig zu unterbrechen.*
2. *Um* `chosen-ciphertext` *Angriffe zu ermöglichen muß der Angreifer die Möglichkeit haben, (manchmal) Nachrichten abzuändern bzw. eigene Nachrichten einzuspielen.*

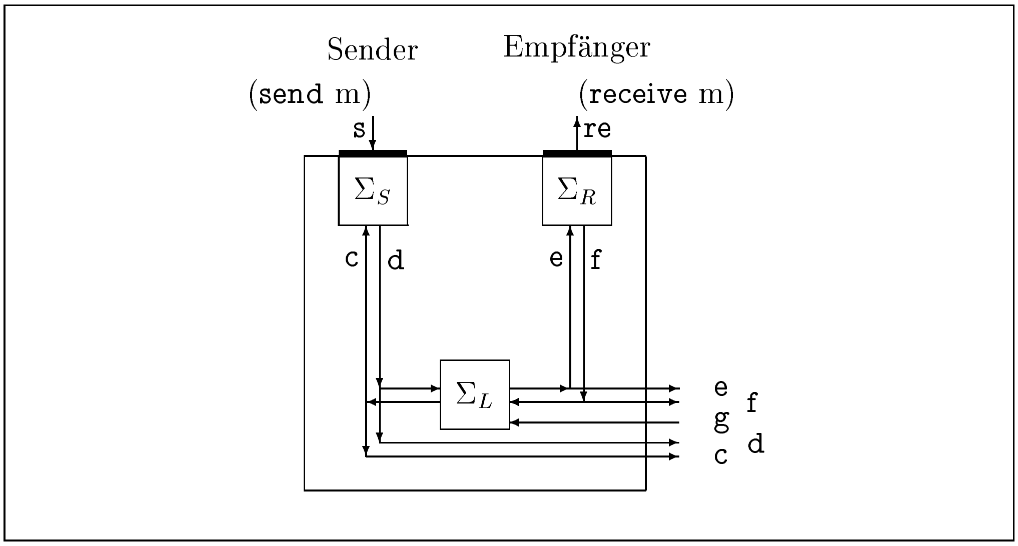

Abbildung 6.3: Schema eines Public–Key Konzelationssystems

Def. 6.10 (Public-Key Konzelationssystem) *Sei M abzählbarer Nachrichtenraum. Ein Kryptosystem (Σ_K, U) mit $\{re, s\} \subseteq U$ heißt Public-Key-Konzelationssystem gdw. es drei Systeme Σ_S, Σ_R und Σ_L gibt mit*

$$S_S = (\{c, s\}, \{d\}, \Gamma, \mathbb{T}) \qquad S := \mathbb{C}_S \times \mathbb{T}$$

mit $(\texttt{send}\ \times M) \subseteq \Gamma^+$,

$$S_R = (\{e\}, \{re, f\}, \Gamma, \mathbb{T}) \qquad R := \mathbb{C}_R \times \mathbb{T}$$

und

$$S_L = (\{d, f, g\}, \{c, e\}, \Gamma, \mathbb{T}) \qquad L := \mathbb{C}_L \times \mathbb{T},$$

so daß für $v(c) = \begin{cases} \uparrow & \text{für } c \in \{s, re\} \\ c & \text{sonst} \end{cases}$ *gilt*

$$\Sigma_K = \Sigma_S \bowtie_v \Sigma_R \bowtie_v \Sigma_L$$

und Σ_K *den Service und die Geheimhaltung eines Konzelationssystems bietet.*

Anmerkung 6.5 *Bei dieser Definition sind auch interaktive Konzelationsprotokolle enthalten.*

6.5.2 Private–Key Konzelationssysteme

Def. 6.11 (Private-Key Konzelationssystem) *Ein Kryptosystem, welches Geheimhaltung und Service eines Konzelationssystems bietet und kein Public–Key Konzelationssystem ist, heißt Private–Key Konzelationssystem.*

Anmerkung 6.6 *Diese Definition ist eventuell zu einfach: selbst ein einfaches Kästchen mit Ein– und Ausgabekanal kann unter Umständen die geforderten Leistungen erbringen.*

6.6 Einordnung der GM–Security

Die in [MRS88] vorgestellte Definition der GM–Security ist schwächer als die relative komplexitätstheoretische Geheimhaltung bei passiven Benutzern, da Geheimhaltung nur für einen bestimmten Benutzer gefordert wird.

Wie hier angedeutet wird, ist die „GM–security“ nicht schwächer als die schwache Geheimhaltung, da analog Geheimhaltung für einen bestimmten nichttrivialen Benutzer gefordert wird.

Ich vermute, daß die GM–Security sogar äquivalent zur hier vorgeschlagenen schwachen komplexitätstheoretischen Sicherheit ist.

6.6.1 Goldwasser-Micali-Security

Def. 6.12 (GM–security aus [MRS88]) *Let $\mathcal{C}$ [which outputs (E_n, D_n) on input 1^n] be a public–key cryptosystem. $\mathcal{C}$ is GM–secure if for all line Tappers T [family of polynomial size circuits] and $c > 0$, for all sufficiently large n, for every $m_o, m_1 \in \{0,1\}^n$*

$$P(m \leftarrow \{m_0, m_1\}; E \leftarrow E(m); \alpha \leftarrow E(m) : T_n(E, m_0, m_1, \alpha) = m) < \frac{1}{2} + n^{-c}.$$

Hierbei bedeutet $\leftarrow$ eine gleichverteilte Auswahl bzw. eine Zuweisung.

Um diese Definition einordnen zu können, wird das dort angegebene System modelliert. Abb. 6.4 zeigt den schematischen Systemaufbau.

Da Definition 5.7 allgemeiner ist, kann hier Äquivalenz nur bezüglich Systemen bewiesen werden, die der Originaldefinition zugrundeliegen. Diese speziellen Teilsysteme werden hier nachgebildet und anhand dieser Nachbildung wird angedeutet, wieso die Definition der „GM–security" (Def. 6.12) schwächer als die hier angegebene schwache Geheimhaltung ist.

Formal wird dieses System wie folgt konstruiert (Skizze):

Beispiel 6.1 (Nachbildung des GM-Systems (Σ_{GM}))

$\Sigma_{S,n}$
- *Wähle $E \leftarrow \mathcal{C}(1^n)$.*
- *Verschlüssele und versende: $i_{s,t} = \texttt{send}\ m \rightarrow o_{c,t+1} = E(m)$.*

Σ_L, Σ_R *Hier nicht nötig.*

$\Sigma_{B,n}$
- *Wähle $\{m_0, m_1\} \in \{0,1\}^n \times \{0,1\}^n$ und $m \in \{m_0, m_1\}$ gleichverteilt aus.*
- $o_{b,1} = \texttt{send}\ \{m_0, m_1\}$
- $o_{s,1} = \texttt{send}\ m$

Ereignisse $\mathcal{E} = \{[\![i_{s,1} = \texttt{send}\ m_1]\!], [\![i_{s,1} = \texttt{send}\ m_2]\!]\}$

A_U *Der „legale" Angreifer erhält nur die beiden Texte m_0 und m_1. Es gilt also:*

$$A_U := \{b\} \times \mathbb{T}$$

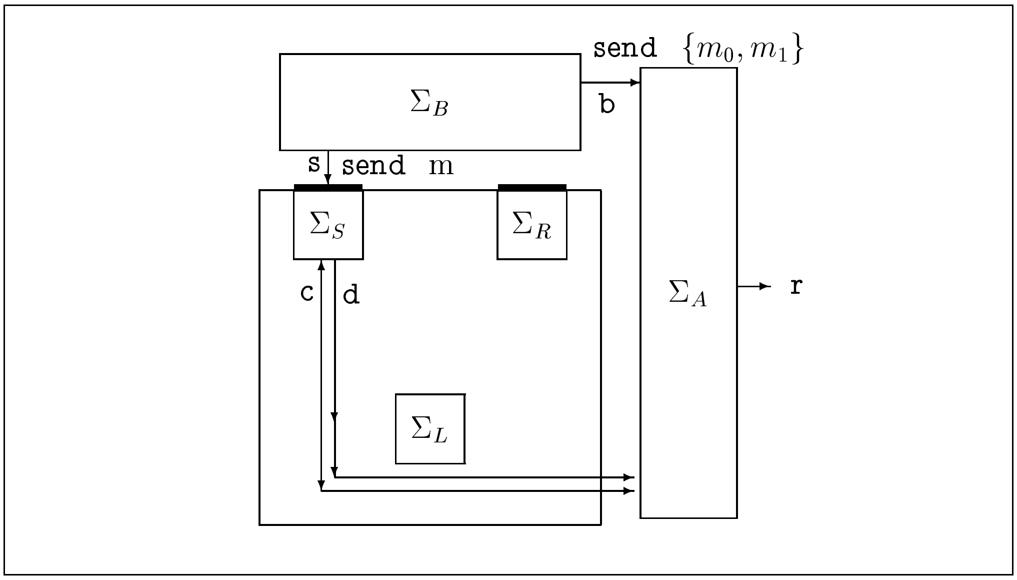

Abbildung 6.4: Systematischer Systemaufbau zu Definition 6.12 (Σ_{GM})

Lemma 6.1 *Ist $\mathcal{E}$ in Σ_{GM} (Nach Beispiel 6.1) geheim nach Definition 5.7 so auch nach Definition 6.12.*

Beweisidee: Sei Σ nicht GM-secure, so gibt es einen Algorithmus T, der die Texte anhand des Schlüsseltextes besser als durch Raten unterscheiden kann. Somit ist auch Def. 5.3 nicht erfüllt, da A' als einzige Information $\{m_1, m_2\}$ erhält und somit für ihn gilt: $P([\![r = 1]\!]|E) = 1/2$. ■

Kapitel 7

Ausblick

Eine Diplomarbeit kann ein solch umfassendes Gebiet natürlich nur auszugsweise bearbeiten. Viele interessante Fragestellungen, deren Grundlagen hier entwickelt wurden, konnten nicht ausreichend behandelt werden, werden aber zum Teil in den folgenden Abschnitten kurz angedeutet:

- Festigung der hier vorgestellten Definitionen durch Vergleich mit weiteren bestehenden Geheimhaltungsdefinitionen und durch Einordnung bestehender Systeme,

- formale Definition weiterer Geheimhaltungsanforderungen, wie z.B. Anonymität,

- Service– und Geheimhaltungsspezifikation weiterer Kryptosysteme wie z.B. anonyme Netzwerke*,

- Erweiterung der Rechenregeln aus Abschnitt 5.1.2 und Übertragen auf andere Geheimhaltungsstufen und eventuell

- formale Definition weiterer Schutzziele, wie Verfügbarkeit und Integrität mit den hier vorgestellten Sprachmitteln.

Hierbei ist der erste Punkt von besonderer Wichtigkeit. Die in dieser Arbeit behandelten Beispiele haben zwar keinen Anhaltspunkt gegeben, daß die Definitionen in dieser Form den Begriff der Geheimhaltung nicht erfassen — dies kann aber in diesem Rahmen nicht sichergestellt werden.

7.1 Festigung der hier vorgestellten Geheimhaltungsdefinitionen

Die hier vorgeschlagenen Definitionen sollten unbedingt weiter gesichert werden.

Hierzu müßten weitere Geheimhaltungsdefinitionen mit den hier vorgestellten verglichen werden (analog zu Abschnitt 6.6). Ein Problem hierbei ist, daß noch keinerlei derart allgemeine Definitionen existieren.

Ergänzend hierzu sollte die Geheimhaltungsstufe bestehender Kryptosysteme bestimmt werden. Existiert ein System, welches intuitiv keine Geheimhaltung bietet und trotzdem einer der hier vorgeschlagenen Definitionen genügt, so muß diese Definition überprüft werden [RSA78, GMT82, BFM90, GM84b].

Analog hierzu sollte eine Definition überprüft werden, falls mehrere Systeme, die Geheimhaltung garantieren, die entsprechende Definition nicht erfüllen.

7.2 Definition weiterer Geheimhaltungsanforderungen

Mit den in dieser Arbeit vorgestellten Geheimhaltungsdefinitionen ist das Werkzeug bereitgestellt worden, beliebige Geheimhaltungsanforderungen für kryptologische Systeme zu präzisieren. Hierzu müssen für die einzelnen Anforderungen nur Ereignismengen und Benutzerklasse angegeben werden.

Exemplarisch skizziere ich hier die Anforderungen für die Gewährleistung von Anonymität.

7.2.1 Anonymität

Zur Gewährleistung von Anonymität soll geheimgehalten werden, wer bei einer Aktion eine gewisse Rolle (wie z.B. Sender, Empfänger, ...) wahrgenommen hat [Pfi90, S. 15]. Es soll nun kurz skizziert werden, wie diese Anforderung formalisiert werden könnte.

Die folgenden Definitionen beziehen sich jeweils auf ein festes System Σ.

7.2.1.1 Rollen und Ablaufeinheiten

Den einzelnen Operationen außer `nop` werden Rollen zugeordnet:

Def. 7.1 (Rolle) *Sei $Op \subset \Gamma^+$ die Menge der Operationsnamen, $Rol \subseteq \Gamma^*$ die Menge der Rollennamen, so ordnet die Rollenfunktion*

$$r : Op - \{\texttt{nop}\} \rightarrow Rol$$

jeder Operation eine Rolle zu [Sne92].

Eine Rollenfunktion eines Zahlungssystems würde z.B. der Operation `pay` die Rolle „Zahler" und der Operation `get` die Rolle „Empfänger" zuordnen.

Systeme laufen oft über einen langen Zeitraum. Da für Eigenschaften wie Anonymität und Geheimhaltung oft nur ein zeitlich zusammenhängender Ausschnitt des Gesamtablaufs betrachtet werden soll, wird der Gesamtablauf in kleinere zusammenhängende Abschnitte sog. Ablaufeinheiten unterteilt.

Def. 7.2 (Ablaufeinheiten) *Ein Ablaufeinheit ist der Durchschnitt einer Folge von Ereignissen E_1, E_2, E_3, … für die für alle i gilt: $E_{i+1} \supseteq E_i$.*

Eine Ablaufeinheit ist selbst wieder ein Ereignis.

Für die Ereignisse einer Ablaufeinheit gilt also, daß ein Ereignis das nächste impliziert. Eine Ablaufeinheit eines Netzes könnte das Senden und Empfangen sein: E_1 ist das Senden einer Nachricht, E_2 der Empfang. Da E_2 aus E_1 folgt, gilt $E_2 \subseteq E_1$. In der Regel werden Ereignisse einer Ablaufeinheit auch in zeitlichem Zusammenhang stehen.

Um den Rollenbegriff auf Ablaufeinheiten zu übertragen, wird nun ein Rollenprädikat definiert. Es dient dazu, einzelnen Kanälen in vorgegebenen Transaktionen eine Rolle zuzuordnen (T ist eine Ablaufeinheit):

Def. 7.3 (Rollenprädikat)

$$R : \mathbb{C} \times \mathbb{E}(S) \times R \rightarrow \{True, False\}$$

$$R(c, T, r) = \begin{cases} True & \textit{falls } c \textit{ in } T \textit{ die Rolle } r \textit{ wahrnimmt} \\ False & \textit{sonst} \end{cases}$$

Diese (noch informelle) Definition sagt aus, daß wenn ein Kanal während einer Ablaufeinheit eine Operation ausgeführt hat, die zu der vorgegebenen Rolle gehört, das Prädikat gilt. Bei einer Formalisierung sollte beachtet werden, daß bei eingeschränkter Sicht alle Operationen möglich sind. Eine Rolle somit nur für die Teile der Ablaufeinheit vergeben werden, die nicht sämtliche Operationen enthalten.

7.2.1.2 Definitionsversuch

Def. 7.4 (Anonymität) *$c \in \mathbb{C}$ heißt in der Rolle r perfekt anonym bezüglich einer Ablaufeinheit T gdw. die folgende Ereignismenge geheim ist:*

$$\mathcal{E} := \{[\![R(c,T,r)]\!] | c \in C\}$$

Anmerkung 7.1 *Bei den hier angegebenen Definitionen werden Personen mit Kanälen gleichgesetzt, was in einem realen Netzwerk nicht sein muß (z.B. in einer Telefonzelle können beliebige Personen telefonieren).*

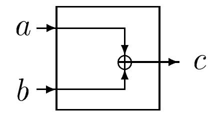

Abbildung 7.1: Beispielsystem zu Def. 7.4

7.2.1.3 Beispiel

Die hier definierte Anonymität soll nun anhand des Systems aus Abbildung 7.1 verdeutlicht werden. Gegeben sei das System aus Bsp. 5.2 auf Seite 85. Intuitiv erhält c beim Empfang einer 1 keine Information darüber, wer 1 und wer 0 eingegeben hat.

Es sei $A = \{(c,1)\}$, $T = [\![o_{c,1} = 1]\!]$, $c = \mathtt{a}$, $Rol(1) = 1$, $Rol(0) = 0$, r=1. Es gilt: $I\!\!E = \{(0,0,0),(0,1,1),(1,0,1),(1,1,0)\}$, $\mathcal{E} = \{E_1, E_2\}$ mit $E_1 = \{(1,0,1)\}$ und $E_2 = \{(1,0,1)\}$.

Zu zeigen (Def. 5.1):

$$P([\![s\lceil A]\!] | [\![R(\mathtt{a}, E, r)]\!]) = P([\![s\lceil A]\!] | [\![R(\mathtt{b}, E, r)]\!]).$$

Seien die Benutzereingabewahrscheinlichkeiten beliebig, und $\mathcal{E}$ möglich.

Für $s\lceil A = 0$ gilt für $i = 1, 2$: $P(E_i) = 0$.

Für $s\lceil A = 1$ gilt:

$$\begin{aligned} & P([\![s\lceil A]\!] | [\![R(a, \{(1,0,1),(0,1,1)\}, 1)]\!]) = \\ = \quad & P(\{(1,0,1),(0,1,1)\} | (1,0,1)) = 1 \end{aligned}$$

Analog gilt dies auch für E_2. ■

7.2.2 Unverkettbarkeit*

Analog kann auch Unverkettbarkeit von zwei Ereignissen formalisiert werden:

Def. 7.5 (Unverkettbarkeit) *Zwei Ereignisse E_1, E_2 sind bezüglich des Merkmals $M : \mathcal{P}(\mathbb{E}) \rightarrow \mathbb{R}$ unverkettbar, wenn die Ereignissmenge*

$$\{[\![M(s) \wedge s \in E_1]\!], [\![M(s) \wedge s \in E_2]\!]\}$$

geheim ist.

7.3 Weitere Systemtypen

Formale Servicedefinitionen für weitere Kryptosystemtypen könnten analog zu der einfachen Spezifikation in Abschnitt 6 erstellt werden.

Für Signatursysteme wurden von B. Pfitzmann bereits Servicedefinitionen vorgestellt [Pfi93a].

Unter anderem wurde Geheimhaltung und Service der folgenden Systemtypen bisher noch nicht allgemein definiert:

- Anonyme Netze
- Zero-Knowledge Proof Systeme
- Zahlungssysteme

7.4 Erweiterung der Rechenregeln

Die Rechenregeln aus Abschnitt 5.1.2 sollten auf die anderen Geheimhaltungsstufen übertragbar sein.

Außerdem sollten auch für folgende Aspekte ähnliche Regeln herleitbar sein:

- Einsetzen mehrerer komplexitätstheoretischer Systeme ineinander und Wahl eines „globalen" Sicherheitsparameters,
- Entwicklung von Logiken mit einer mit den hier angegebenen Mitteln definierten Semantik [AT91, BAN90b, BC91a, BC91b, Syv90, Syv91, Syv92].

Fazit

In dieser Diplomarbeit wurde gezeigt, daß mehrere Stufen von Geheimhaltung unterschieden werden müssen. Eine einzige Geheimhaltungsdefinition reicht nicht aus, um die sehr verschiedenen Sachverhalte zu erfassen.

Daher stellt diese Arbeit die Mittel für eine differenzierte Betrachtung der „Geheimhaltung reaktiver kryptologischer Systeme“ bereit.

Anhang A

Abbildungsverzeichnis

Tabellenverzeichnis

Legende

Die folgende Tabelle gibt einen Überblick über die in den Abbildungen und im Text verwendeten allgemeinen Symbole. Speziellere Symbole sind im Symbolverzeichnis am Anfang des Index aufgeführt.

	Text
wort*	Wort ist im Glossar erklärt
$\blacksquare$	Beweisende
$\square$	Beispielende
$\diamondsuit$	Widerspruch innerhalb eines Beweises
	Formeln
$I\!N$	$\{0, 1, \ldots\}$
$I\!N^+$	$\{1, 2, \ldots\}$
Γ^+	$= \bigcup_{i \in I\!N^+} \Gamma^i$ mit $\Gamma^1 := \Gamma$ und $\Gamma^{i+1} := \{lw \mid l \in \Gamma \wedge w \in \Gamma^i\}$
Λ	$:= \{0, \ldots, 9, A, \ldots, Z, a, \ldots, z, \}$
$\mathcal{P}(M)$	$:= \{M' \mid M' \subseteq M\}$ (Potenzmenge)
$\mathcal{P}$	Menge aller polynomiell zeitbeschränkt Turing-berechenbaren Funktionen.
UM	Gleichverteilung auf der Menge/ dem Intervall M
$A = \dot{\bigcup\limits_i} A_i$	$\iff A = \bigcup\limits_i A_i \wedge \forall\limits_{i \neq j} A_i \cap A_j = \emptyset$
	Abbildungen
`op` m	Ein– oder ausgegebene Operation mit Argument
`a`	Kanalbezeichnung
$\square$	Instanz oder Maschine
▌	Benutzerschnittstelle
ⓕ	Funktionen
$\rightarrow$	Kanäle, Leitungen

Tabelle A.1: Häufig verwendete Symbole

Glossar

adaptive Beim adaptiven Angriff auf ein Kryptosystem darf der Angreifer unbegrenzt viele Anfragen nach dem Ereignis durchführen [Pfi90].

aktiver Angriff Ein *aktiver Angriff* beinhaltet im Gegensatz zum passiven Angriff* auch die Veränderung des Systemablaufs z.B. durch Wahl eigener Eingaben.

Angreifer Pfitzmann verwendet den Begriff *Angreifer* bei Konzelationssystemen als „kurze Bezeichnung für jemanden, der versucht sensitive Daten ohne Einwilligung der Betroffenen zu sammeln“ [Pfi90, S. 14]. Hier wird der Angreiferbegriff weiter gefaßt und umfaßt beliebige Teilsysteme, die eine gewünschte Eigenschaft des Gesamtsystems versuchen zu stören.

Angriffsarten Siehe ciphertext-only, chosen-plaintext, chosen-ciphertext, adaptive und non-adaptive.

Benutzer Hier wird unter einem Benutzer ein Teilsystem verstanden, das Ein– und Ausgaben an der Benutzerschnittstelle vornimmt.

chosen–ciphertext Der Gewählte–Schlüsseltext–Klartext–Angriff auf ein Konzelationssystem erlaubt dem Angreifer, beliebige Schlüsseltexte entschlüsseln zu lassen (Der geheime Text ist hiervon ausgenommen) [ZS93].

chosen-plaintext Der Gewählte–Klartext–Schlüsseltext–Angriff auf ein Konzelationssystem erlaubt dem Angreifer, beliebige Klartexte verschlüsseln zu lassen [Pfi90].

ciphertext-only Beim Nur–Schlüsseltext–Angriff auf ein Konzelationssystem erhält der Angreifer nur Schlüsseltext.

existential-forgery Als existentielles Brechen eines Konzelationssystems wird verstanden, wenn es dem Angreifer gelingt, eine beliebige Nachricht zu verschlüsseln [Pfi90].

Integrität	„Schutz vor unbefugter Modifikation von Informationen“ [Bis93, S. 11].
Klartext	Als Klartext wird die unverschlüsselte Nachricht eines Konzelationssystems bezeichnet.
komplexitäts–theoretische Geheimhaltung	Unter komplexitätstheoretischer Geheimhaltung wird verstanden, daß es einem polynomiell zeitbeschränkten Angreifer nicht möglich ist, das System zu brechen. In der Regel wird dies durch sog. One–Way Funktionen* erreicht, die polynomiell zu berechnen aber nicht polynomiell zu invertieren sind.
Konzelations-system	Ein Konzelationssystem ist ein Kryptosystem*, zur Geheimhaltung eines Ereignisses. Dieses Ereignis ist in der Regel eine übermittelte Nachricht. Es entspricht dem „Kryptosystem zum Zwecke der Konzelation“, das laut PFITZMANN [Pfi90, S. 19] garantieren soll, „[...] daß der Inhalt einer Nachricht nur den Besitzern eines bestimmten Schlüssels zugänglich ist[...]“.
Kryptosystem	Im Gegensatz zu PFITZMANN, der ein Kryptosystem als „Familien von Chiffrierfunktionen zur Ver– und Entschlüsselung“ [Pfi90, S. 19] bezeichnet, wird der Kryptosystembegriff hier weiter gefaßt und umschließt alle kryptologischen Systeme, wie z.B. auch Zahlungs–, Authentikations und Signatursysteme.
non-adaptive	Beim nichtadaptiven Angriff auf ein Kryptosystem darf der Angreifer vor dem Ereignis nur polynomiell viele Anfragen durchführen [Pfi90].
One–Way Function	Eine One–Way Funktion ist eine ehrliche, polynomiell zeitbeschränkt turingberechenbare Funktion deren Inverse nicht polynomiell zeitbeschränkt turingberechenbar ist (Eine Funktion ist ehrlich, wenn zu jedem Bild ein Urbild existiert, dessen Länge polynomiell in der Länge des Bildes beschränkt ist) [BDG88, S. 61 & 63].
passiver Angriff	Ein *passiver Angriff* liegt vor, wenn der Angreifer nur beobachtet und nicht verändert.

Public–Key Ein Public-Key System liegt vor, wenn zwei Schlüssel existieren, von denen ein Schlüssel (Public-Key) öffentlich bekannt (d.h. nicht geheim) ist. Ein Public-Key System wurde erstmals von Diffie und Hellman [DH76] vorgeschlagen.

reaktives System Im Gegensatz zum sequentiellen System, erlauben reaktive Systeme mehrere Eingaben zu verschiedenen Zeitpunkten, die auch von Ausgaben des Systems abhängen dürfen.

Schlüsseltext Als Schlüsseltext wird die verschlüsselte Nachricht eines Konzelationssystems bezeichnet.

selective-forgery Als selektives Brechen eines Konzelationssystems wird verstanden, wenn es dem Angreifer gelingt, einen vorweg von ihm gewählten Text zu verschlüsseln. [Pfi90].

Signatur–system Idee eines Signatursystems ist, daß mit dem geheimen Schlüssel aus einem Klartext eine Signatur errechnet wird. Mit dem öffentlichen Schlüssel kann dann jeder prüfen, ob die Signatur zu dem Text paßt. Paßt sie, so wird davon ausgegangen, daß die Signatur vom besitzer des geheimen Schlüssels stammt, ohne den man keine passende Signatur berechnen kann (vgl. [GMR88, vHPP93])

total-break Unter dem vollständiges Brechen eines Systems wird verstanden, daß der Angreifer den geheimen Schlüssel erfährt und somit alle Möglichkeiten des Benutzers hat. Es ist die stärkste Art, ein System zu brechen [Pfi90].

universal-forgery Unter dem universelles Brechen eines Kryptosystems wird verstanden, daß der Angreifer ein dem geheimen Verfahren äquivalentes Verfahren erhält, ohne den geheimen Schlüssel zu erhalten [Pfi90].

Unverkettbarkeit Zwei Ereignisse sind unverkettbar, wenn für einen Angreifer nicht feststellbar ist, ob sie in einem Merkmal M übereinstimmen [Pfi90, S. 15].

Verbreitung Die Verbreitung eines Angreifers ist die Tatsache, „[...] wie viele und welche Leitungen [der Angreifer] maximal beobachten und/oder gar aktiv kontrollieren [kann]“ [Pfi90, S. 15].

Verfügbarkeit „Schutz vor unbefugter Beeinträchtigung der Funktionalität“ [Bis93, S. 11].

Vertraulichkeit „Schutz vor unbefugtem Informationsgewinn“ [Bis93, S. 11].

Zero-Knowledge Proof Die Idee eines Zero-Knowledge Beweises ist, daß der Beweiser (prover) nur beweist, daß er ein Geheimnis oder Verfahren besitzt, ohne es zu verraten. Zum Beispiel könnte der Beweiser „beweisen“, daß er ein Verfahren zur Faktorisierung besitzt, indem er beliebig viele vorgegebene Zahlen faktorisiert. Das Faktorisierungsverfahren muß hierbei nicht verraten werden.

Literaturverzeichnis

[AT91] MARTIN ABADI and MARK R. TUTTLE: *A semantics for a logic of authentication.* In *Proceedings of the 10th Annual ACM Symposium on Distributed Computing*, pages 201–216. ACM Press, August 1991.

[BAN89] MICHAEL BURROWS, MARTIN ABADI, and ROGER NEEDHAM: *A logic of authentication.* Open Systems Review, 23(5):1–13, 1989. (Orig. in Research Report No. 39 of the Digital Systems Research center).

[BAN90a] MICHAEL BURROWS, MARTÍN ABADI, and ROGER NEEDHAM: *A logic of authentication.* Technical Report SRC-39, DIGITAL Systems Research Center, 130 Lytton Avenue, Palo Alto, California 94301, February 22 1990.

[BAN90b] MICHAEL BURROWS, MARTÍN ABADI, and ROGER NEEDHAM: *A logic of authentication.* ACM Transaction on Computer Systems, 8:18–36, 1990.

[Bau78] HEINZ BAUER: *Wahrscheinlichkeitstheorie und Grundzüge der Maßtheorie.* Walter de Gruyter, Berlin, New-York, 3. Auflage, 1978.

[BC91a] P. BIEBER and F. CUPPENS: *A definition of secure dependencies using the logic of security.* In *Proceedings of the computer security foundations workshop*, pages 2–11, Franconia, 1991.

[BC91b] PIERRE BIEBER and FRÉDÉRIC CUPPENS: *Computer security policies and deontic logic.* In J.-J. CH. MEYER and R. J. WIERINGA (editors): *Proc. 1st Internat. Workshop on Deontic Logic in Computer Science*, pages 127–141, Amsterdam, December 1991.

[BD87] S. Budkowski and P. Dembinski: *An introduction to Estelle: A specification language for distributed systems.* Computer Networks and ISDN Systems, 14:3–23, 1987.

[BDG88] José Luis Balcázar, Josep Diaz, and Joaquim Gabarró: *Structural complexity I*, volume 11 of *EATCS monographs on theoretical computer science.* Springer-Verlag, Berlin, Heidelberg, New-York, 1988.

[Bea91] Donald Beaver: *Secure multiparty protocols and zero-knowledge proof systems tolerating a faulty minority.* Journal of Cryptology, 4(2):75–122, 1991.

[BFM90] M. Blum, P. Feldman, and S. Micali: *Proving security against chosen chipertext attacks.* In *Advances in Cryptology — CRYPTO '88*, pages 256–269, Berlin, Heidelberg, New-York, 1990. Springer-Verlag. Lecture Notes in Computer Science (LNCS) 403.

[Bis93] Joachim Biskup: *Sicherheit von IT-Systemen als „sogar wenn – sonst nichts – Eigenschaft“.* In: *Proceedings der GI–Fachtagung VIS '93*, Seiten 239–254, Wiesbaden, 1993. DuD Fachbeiträge 16, Vieweg.

[BL83] D. E. Bell and L. J. LaPadula: *Secure computer systems: Mathematical foundations.* Technical Report MTR-2547-VOL-1, ESD-TR-73-278-VOL-1, The MITRE Corporation, Bedford, MA, 1983.

[Bra88] Gilles Brassard: *Modern Cryptology: A Tutorial.* Springer-Verlag, Berlin, Heidelberg, 1988.

[Den82] Dorothy Elisabeth Denning: *Cryptography and data security.* Addison-Wessley, Reading, Mass., 1982.

[DH76] Whitfield Diffie and Martin E. Hellmann: *New directions in cryptography.* IEEE Transactions on Information Theory, 22(6):644–654, 1976.

[DoD85] *U. S. department of defense trusted computer system evaluation criteria (TCSEC).* Government Report GpO 008-000-00461-7, DoD 5200.28-STD, Superintendent of Documents, U.S. Government Printing Office, Washington D.C. 20402, December 1985. („orange book“).

[Fel68] WILLIAM FELLER: *An Introduction to Probability and Its Applications*, volume I. John Wiley & Sons, New-York, 3rd edition, 1968.

[Fel71] WILLIAM FELLER: *An Introduction to Probability and Its Applications*, volume II. John Wiley & Sons, New-York, 2nd edition, 1971.

[GHY91] ZVI GALIL, STUART HABER, and MOTI YUNG: *Security against replay chosen-chipertext-attack.* DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 2:183–189, 1991.

[GM82] JOSEPH A. GOGUEN and JOSÉ MESEQUER: *Security policies and security models.* In *Proceedings of the 1982 IEEE Symposium on Security and Privacy*, Oakland, CA, 1982.

[GM84a] JOSEPH A. GOGUEN and JOSÉ MESEQUER: *Unwinding and interference control.* In *Proc. of the IEEE Symposium on Security and Privacy*, pages 75–86, Oakland, 1984.

[GM84b] SHAFI GOLDWASSER and SILVIO MICALI: *Probabilistic encryption.* Journal of Computer and System Sciences, 28(2):270–299, 1984.

[GMR88] SHAFI GOLDWASSER, SILVIO MICALI, and RONALD L. RIVEST: *A digital signature scheme secure against adaptive chosen–message attacks.* SIAM J. Comput., 17(2):281–308, 1988.

[GMT82] SHAFI GOLDWASSER, SILVIO MICALI, and PO TONG: *Why and how to establish a private code on a public network (extended abstract).* In *Proceedings of the 23rd Annual Symposium on Foundations on Computer Science*, pages 134–144. IEEE Press, November 1982.

[Gol93] ODED GOLDREICH: *A uniform-complexity treatment of encryption and zero-knowledge.* Journal of Cryptology, 6:21–53, 1993.

[Gra90] ROBERT M. GRAY: *Entropy and Information Theory.* Springer-Verlag, New-York, Berlin, Heidelberg, 1990.

[Gra92] JAMES W. GRAY III: *Towards a mathematical foundation for information flow security.* Journal of Computer Security, 1:255–294, 1992.

[GS88] JOACHIM GROLLMANN and ALAN L. SELMAN: *Complexity measures for public-key cryptosystems.* SIAM Journal Computing, 17(2):309–335, 1988.

[GS92] JAMES W. GRAY and PAUL F. SYVERSON: *A logical approach to multilevel security of probabilistic systems.* In *Proceedings of the 1992 IEEE Symposium on Research in Security and Privacy*, pages 164–176, Oakland, CA, 1992.

[Hin85] KARL HINDERER: *Grundbegriffe der Wahrscheinlichkeitstheorie.* Springer Hochschultext. Springer-Verlag, Berlin, Heidelberg, New-York, 3. Auflage, 1985.

[Hoa85] CHARLES ANTONY RICHARD HOARE: *Communicating Sequential Processes.* Prentice-Hall, Englewood Cliffs, NJ, 1985.

[Jac88] JEREMY JACOB: *Security specifications.* In *Proceedings of the 1988 IEEE Symposium on Security and Privacy*, pages 14–23, Oakland, CA, 1988.

[Jac92] JEREMY JACOB: *Basic theorems about security.* Journal of Computer Security, 1:385–411, 1992.

[Kah67] DAVID KAHN: *The Codebreakers.* Macmillian Publishing Co., New York, 1967.

[Lea90] JOHN MC LEAN: *Security models and information flow.* In *Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy*, pages 180–187, Oakland, CA, 1990.

[Man91] ROGER MANNING: *The representation of data security standards in SDL.* NPL Report DITC 190/91, National Physical Laboratory, Teddington, Middlesex TW11 OLW, UK, September 1991.

[Mil89] JONATHAN K. MILLEN: *Models of multilevel security.* Advances in Computers, 29:1–45, 1989.

[MP91] ZOHAR MANNA and AMIR PNUELI: *The temporal logic of reactive and concurrent systems.* Springer-Verlag, New-York, Berlin, Heidelberg, 1991.

[MRS88] SILVIO MICALI, CHARLES RACKOFF, and BOB SLOAN: *The notion of security for probabilistic cryptosystems.* SIAM Journal Computing, 17(2):412–426, April 1988.

[NY90] MONI NAOR and MOTI YUNG: *Public-key cryptosystems provably secure against chosen ciphertext attacks.* In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 427–437, Baltimore, MA, May 14-16 1990. ACM Press.

[Pfi90] ANDREAS PFITZMANN: *Diensteintegrierende Kommunikationsnetze mit teilnehmerüberprüfbarem Datenschutz.* Informatik Fachberichte; 234. Springer–Verlag, Berlin, Heidelberg, 1990.

[Pfi92] ANDREAS PFITZMANN: *Sicherheit in Rechnernetzen.* Skript, Universität Hildesheim, Institut für Informatik, 1992.

[Pfi93a] BIRGIT PFITZMANN: *Fail-Stop Signature Schemes.* Dissertation, Universität Hildesheim, 1993. Kapitel 4 & 5; Version: 11.7.93.

[Pfi93b] BIRGIT PFITZMANN: *Sorting out signature schemes and some theory of secure reactive systems.* In *Proceedings of the 1st ACM Conference on Computer and Communication Security*, pages 74–85. Fairfax, ACM Press, 03.–05.11. 1993.

[RG92] DEBORAH RUSSELL and G. T. GANGEMI SR.: *Computer Security Basics.* O'Reilly & Associates Inc., Sebastopol, CA, 1992.

[Ros93] SHELDON M. ROSS: *Introduction to probability models.* Academic Press Inc., San Diego, CA, 5th edition, 1993.

[RS92] CHARLES RACKOFF and DANIEL R. SIMON: *Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack.* In *Advances in Cryptology — CRYPTO '91*, pages 433–444, Berlin, Heidelberg, New-York, 1992. Springer-Verlag. LNCS 576.

[RSA78] R. L. RIVEST, A. SHAMIR, and L. ADLEMAN: *A method for obtaining digital signatures and public key cryptosystems.* Communications of the ACM, 21(2):120–126, Feb. 1978. Reprinted in

Communications of the ACM, 25th Anniversary Issue, 26(1) Jan. 1983.

[Sha48] C. E. Shannon: *A mathematical theory of communication.* The Bell System Techical Journal, 27(3):379–423, 623–656, July 1948.

[Sha49] Claude E. Shannon: *Communication theory of secrecy systems.* The Bell System Technical Journal, 28:656–715, 1949.

[Sne92] Einar Snekkenes: *Roles in cryptographic protocols.* In *Proceedings of the 1992 IEEE Symposium on Security and Privacy*, pages 105–119, Oakland, CA, 1992.

[Syv90] Paul Syverson: *Formal semantics for logics of cryptographic protocols.* In *Proc. of the Computer Security Foundations Workshop*, pages 32–41, Franconia, 1990. IEEE Press.

[Syv91] Paul Syverson: *The use of logics in the analysis of cryptographic protocols.* In *Proc. of the IEEE Symposium on Security and Privacy*, pages 156–170, Oakland, 1991.

[Syv92] Paul F. Syverson: *Knowledge, belief, and semantics in the analysis of cryptographic protocols.* Journal of Computer Security, 1:317–334, 1992.

[Top74] Flemming Topsøe: *Informationstheorie.* B. G. Teubner Verlag, Stuttgart, 1974.

[vHPP93] Eugéne van Heijst, Torben P. Pedersen, and Birgit Pfitzmann: *New constructions of fail–stop signatures and lower bounds.* In *Advances in Cryptology — CRYPTO '92*, pages 15–30, Berlin, Heidelberg, New-York, 1993. Springer-Verlag. Lecture Notes in Computer Science (LNCS) 740.

[Yao82] Andrew C. Yao: *Theory and application of trapdoor functions (extended abstract).* In *Proceedings of the 23rd Annual IEEE Symposium on the Foundations of Computer Science*, pages 80–91, 1982.

[ZS93] Yuliang Zheng and Jennifer Seberry: *Practical approaches to attaining security against adaptively chosen chipertext attacks.* In Ernest F. Brickell (editor): *Advances in Cryptology —*

CRYPTO '92, pages 292–304, Berlin, Heidelberg, New-York, 1993. Springer-Verlag. Lecture Notes in Computer Science (LNCS) 740.

Symbol– & Stichwortverzeichnis