Lehrstuhl für Datenverarbeitung

Entwurf einer Beschreibungssprache zur Verifikation digitaler Signaturen

Dipl.-Ing. Univ. Stephan Spitz

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines Doktor-Ingenieurs genehmigten Dissertation.

Vorsitzender:	Univ.-Prof. Dr.-Ing. Dr.-Ing. E.h. G. Schmidt
Prüfer der Dissertation:	
	1. Univ.-Prof. Dr. techn. J. Swoboda
	2. Univ.-Prof. Dr.-Ing. E. Jessen, em.

Die Dissertation wurde am 28.02.2002 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 05.07.2002 angenommen.

Kurzfassung

Die Bedeutung von digitalen Signaturen und ihre Anwendung in den verschiedensten Bereichen der Gesellschaft, Industrie und Politik nimmt stetig zu. In vielen Fällen wird in Zukunft die digitale Unterschrift einer manuellen gleichgestellt sein. Infolgedessen steigt der Bedarf nach leistungsfähigen Werkzeugen und Methoden für die Qualifizierung und Analyse von Prozessen zur Verarbeitung digitaler Signaturen.

Diese Arbeit befasst sich mit allgemeingültigen Kriterien zur vollständigen Verifikation einer digital geleisteten Unterschrift. Abgesehen von juristischen Aspekten und frei vereinbarten Regeln, gibt es eindeutige Indizien um festzustellen, ob eine digitale Signatur gültig ist. Die technische Gültigkeitsprüfung mit einer eigens dafür entwickelten Logik ist Gegenstand dieser Betrachtungen. Dabei soll mit Hilfe einer formalen Sprache eine vollständige und widerspruchsfreie Aussage zur Gültigkeit einer digitalen Signatur ermöglicht werden. Als Basis für die logische Ableitung dienen die Eigenschaften und Besonderheiten einer Public-Key-Infrastruktur (PKI), die mit den Symbolen der hier vorgestellten Logik formulierbar sind.

Für die Realisierung von Komponenten zur Bearbeitung von digitalen Signaturen ist eine exakte Definition der einzelnen Prüfkriterien notwendig. Insbesondere unterschiedliche Realisierungen müssen auf Basis einer gemeinsamen Entscheidungsgrundlage bei einer Signaturprüfung zum selben Ergebnis kommen. In vielen Fällen ist dafür eine allgemeinsprachlich abgefasste Spezifikation der gesamten PKI zuständig. Allerdings kann, wie in Kapitel 6 gezeigt wird, eine derartige Beschreibungsform mit Lücken behaftet sein und Inkonsistenzen aufweisen. Mit Hilfe einer logischen Sprache wird es einfacher, eine konsequente Umsetzung aller an die Verifikation gestellten Vorgaben zu erreichen. Zusätzlich erzwingen die festen Regeln der Sprache eine in sich schlüssige Darstellung. Aufgrund dieser Erkenntnis wird in der vorliegenden Arbeit eine Logik zur Beschreibung der an der Verifikation beteiligten Prozesse vorgestellt.

Die hier vorgestellte Signatur-BAN (SigBAN) zur Modellierung der Verifikation von digitalen Signaturen beruht auf einer Modifizierung und Erweiterung der BAN-Logik. Die BAN-Logik wurde Ende der 80er Jahre von Burrows, Abadi und Needham zur Untersuchung von Authentisierungsprotokollen entworfen. Als Grundlage nutzen diese Protokolle dieselben kryptographischen Verfahren wie digitale Signaturen. Allerdings gelten für die Überprüfung von Signaturen andere Anforderungen als bei der Protokollanalyse. Diesen veränderten Bedingungen wird in den Axiomen der SigBAN-Logik Rechnung getragen. Basierend auf den neuen Axiomen und einem Teil der BAN-Regeln wird die SigBAN-Syntax definiert.

Im Anschluss an die Definition der SigBAN-Logik werden ihre Einsatzmöglichkeiten an unterschiedlichen Beispielen demonstriert. Anhand mehrerer PKI-Anordnungen wird nicht nur der praktische Einsatz, sondern auch die Vollständigkeit und Widerspruchsfreiheit der logischen Beschreibungsform veranschaulicht. Besonderes Augenmerk liegt dabei auf einer Analyse der Verifikation nach dem deutschen Signaturgesetz (SigG). Mit Hilfe der SigBAN-Logik werden Probleme bei einem SigG-konformen Zeitstempeldienst erkannt und ein Verbesserungsvorschlag erarbeitet. Auch bei den Gültigkeitsmodellen nach dem SigG werden Lücken nachgewiesen, die das SigBAN-Modell vermeidet.

Inhaltsverzeichnis

1. Motivation und Zielsetzung 1

2. Technische Grundlagen 3

2.1 Überblick 3
2.2 PKI-Grundlagen 3
2.2.1 Basisdienste in einer PKI 3
2.2.2 Kryptographische Grundlagen 4
2.3 Komponenten und Prozesse in einer PKI 9
2.3.1 Zentrale Verwaltung 9
2.3.2 Zertifizierungsstellen 10
2.3.3 Registrierstellen 11
2.3.4 Verzeichnisdienst 11
2.3.5 Synthese eines Zertifikates 12
2.3.6 Sperrung eines Zertifikates 12
2.4 Aktuelle Anwendungen 13
2.4.1 Sichere Mail-Nachrichten im S/MIME-Format 13
2.4.2 Sichere Datenübertragung mit dem SSL-Protokoll 14
2.4.3 HBCI - PKI im elektronischen Zahlungsverkehr 14
2.5 Abschließende Kommentare 14

3. Einsatz von digitalen Signaturen 17

3.1 Überblick 17
3.2 Standards und Gesetzgebung 17
3.2.1 Standardisierung in der IETF 18
3.2.2 Der PKCS-Standard 19
3.2.3 Das deutsche Signaturgesetz (SigG) 20
3.2.4 Interoperabilitäts-Spezifikationen zum SigG 21
3.3 Signaturerzeugung und Verifikation 23
3.3.1 Erzeugung von digitalen Signaturen 23
3.3.2 Strukturierung der Prozesse zur Verifikation 24
3.4 Abschließende Kommentare 27

4. Grundlagen für eine logische Sprache zur Verifikation 29

4.1 Überblick 29
4.2 Begriffsdefinitionen 29
4.2.1 Vertrauen 29
4.2.2 Sicherheit 30
4.3 Anforderungen an eine Beschreibungssprache zur Verifikation 31
4.3.1 Praktische Anforderungen 31
4.3.2 Anforderungen an eine logische Sprache 32
4.4 Möglichkeiten zur Analyse sicherheitsrelevanter Systeme und Prozesse 33

4.4.1 Strukturierte Analyse ... 33
4.4.2 Prädikatenlogik ... 33
4.4.3 BAN-Logik ... 34
4.5 Abschließende Kommentare ... 35

5. SigBAN als Beschreibungssprache zur Verifikation ... 37

5.1 Überblick ... 37
5.2 Vorstellung der BAN-Logik ... 37
5.3 BAN-Notation ... 38
5.3.1 Syntax der BAN-Logik ... 38
5.3.2 Regeln der BAN-Notation ... 39
5.4 Möglichkeiten und Grenzen der BAN-Notation ... 40
5.4.1 BAN-Syntax mit PKI Begriffen ... 40
5.4.2 Erweiterung der BAN-Syntax ... 41
5.4.3 Anwendung in einer PKI ... 42
5.5 Ausbau der BAN zur SigBAN-Logik ... 43
5.5.1 BAN-Beschreibung von PKI-Aussagen ... 43
5.5.2 Neue Regeln der SigBAN ... 44
5.5.3 Modellierung zeitlicher Aspekte ... 46
5.5.4 Modellierung quantitativer Aspekte ... 48
5.6 Anwendungsbeispiele zur SigBAN ... 50
5.6.1 Beispielkonstellation ... 50
5.6.2 Logischer Beweis ... 51
5.6.3 Zeitliche Überprüfung mit der SigBAN ... 52
5.6.4 Wahrscheinlichkeitsbetrachtung ... 54
5.7 Abschließende Kommentare ... 55

6. SigBAN-Modell zur Verifikation nach dem Signaturgesetz ... 57

6.1 Überblick ... 57
6.2 Merkmale einer SigG-konformen PKI ... 58
6.2.1 Anwenderinfrastruktur ... 58
6.2.2 Datenformate ... 59
6.2.3 Zeitstempeldienst ... 59
6.2.4 Gültigkeitsmodelle und Signaturprüfung ... 60
6.3 SigBAN-Beschreibung zur einfachen Signaturprüfung ... 61
6.3.1 Beispielanordnung für das SigG-konforme SigBAN-Modell ... 61
6.3.2 Beweisführung ... 63
6.4 SigBAN-Betrachtung zum Zeitstempeldienst ... 64
6.4.1 Anforderungen an den Zeitstempel in Form von SigBAN-Aussagen ... 64
6.4.2 Überprüfung des Zeitstempels bei einer Realisierung nach SigI ... 65
6.4.3 Vorschlag zur zuverlässigen Ermittlung des Signaturzeitpunktes ... 68
6.4.4 Beweis zur zuverlässigen Ermittlung des Signaturzeitpunktes ... 68
6.5 Vergleich der Gültigkeitsmodelle mit den SigBAN-Forderungen ... 70
6.5.1 Schalenmodell ... 71
6.5.2 Kettenmodell ... 72
6.5.3 SigBAN-Modell ... 73
6.6 Abschließender Kommentar und Bewertung der SigBAN Anwendung ... 74

7. Erweiterte Anwendung der SigBAN-Logik 75

7.1 Überblick 75
7.2 SigBAN-Beschreibung zur Verwendung besonderer Zertifikatsformate 76
7.2.1 Mehrfachzertifikate 76
7.2.2 Anwendungsbeispiel zur Cross-Zertifizierung mit Mehrfachzertifikaten .. 77
7.2.3 Attributszertifikate 80
7.3 SigBAN-Betrachtung von Vertrauensmodellen 82
7.3.1 Strikte Hierarchie 82
7.3.2 Verteiltes Vertrauen 84
7.4 Grenzen der SigBAN-Logik 87
7.4.1 Nicht überprüfbare Anordnungen 87
7.5 Abschließende Kommentare 90

8. Abschließende Kommentare und Ausblick 91

8.1 Ausblick 92
8.1.1 Weitere Anwendungsgebiete für eine SigBAN-Analyse 92
8.1.2 Mögliche Erweiterungen der SigBAN-Logik 92

Literaturverzeichnis 95

Abkürzungsverzeichnis 101

Anhang A: Beweise der SigBAN-Theoreme 103

Anhang B: SigBAN-Umformungen 107

1. Motivation und Zielsetzung

Viele Prozesse in der Datenverarbeitung, bei denen Integrität, Authentizität, Nichtabstreitbarkeit und Vertraulichkeit wichtige Aspekte sind, nutzen eine Public-Key-Infrastruktur (PKI). Eine PKI stellt mit Hilfe einer zentral administrierten Verwaltung diese Sicherheitsdienste den einzelnen Teilnehmern in verschiedenen Applikationen zur Verfügung. Dabei bietet eine PKI auf Basis der Public-Key-Kryptographie und entsprechender Standards ein hohes Sicherheitsniveau bei gleichzeitig weitgehendster Interoperabilität. Aufgrund dieser Vorteile haben sich in vielen Bereichen PKIs etabliert. In einer weiteren Ausbaustufe, die derzeit gerade stattfindet, werden diese Infrastrukturen überarbeitet, an die aktuellen Standards und Gesetze angepasst und teilweise zusammengeschlossen. Das Ergebnis dieses Prozesses ist ein komplexes Netzwerk, das auf gegenseitigem Vertrauen beruht.

Ein wichtiger Dienst in einer PKI ist die Erzeugung und die Verifikation von digitalen Signaturen. In vielen Fällen kann erst durch den Wechsel von der manuellen Unterschrift zur digitalen Signatur eine lückenlose elektronische Datenverarbeitung realisiert werden. Allerdings gelten für Anwendungen, die eine manuelle Unterschrift substituieren, hohe Anforderungen an die Qualität der digitalen Signatur. Deshalb sind bei der Überprüfung einer digitalen Signatur einige Schritte notwendig, bis nachgewiesen ist, dass sie zuverlässig von einer bestimmten Instanz zu einem bekannten Zeitpunkt erzeugt wurde und sich eindeutig auf einen bestimmten Inhalt bezieht. Aus diesem Grund befassen sich viele Spezifikationen und die Gesetzgebung mit Verfahren zur Signaturprüfung. Dennoch gibt es wenig Veröffentlichungen, die sich mit der Modellierung des komplexen Prüfprozesses auseinandersetzen. Insbesondere existiert keine abstrakte Beschreibungsform zur Verifikation, die umfassend und allgemeingültig alle Vertrauensbeziehungen in der Sicherheits-Infrastruktur einbezieht. Um eine eindeutige Aussage zur Gültigkeit einer digitalen Signatur unter Berücksichtigung aller Aspekte abzuleiten, bot sich die Definition einer eigenen Logik (griech. „Lehre vom richtigen Denken und Folgern“) an. Deshalb stellt diese Arbeit mit der Signatur-BAN (SigBAN) eine Logik vor, die auf Basis der Beschaffenheit einer PKI die Kriterien zur Gültigkeit einer digitalen Signatur überprüft. Die SigBAN-Logik verwendet die Syntax und teilweise die Regeln der BAN-Logik. Beide Beschreibungsformen, SigBAN und BAN, wurden aufgrund ähnlicher Bedürfnisse entwickelt. Die nach ihren Erfindern Burrows, Abadi und Needham benannte BAN-Logik unterstützt den Entwurf und die Analyse von Authentisierungsprotokollen, während die SigBAN-Logik zur Modellierung der Verifikation von digitalen Signaturen dient. Die SigBAN-Logik nutzt die Eigenschaften der BAN-Logik und fügt neue Elemente auf Basis von Axiomen, die eine PKI beschreiben, hinzu. Beide formalen Sprachen beweisen im praktischen Einsatz, dass sie widerspruchsfrei und vollständig sind. Ein theoretischer Nachweis dieser beiden Eigenschaften erfolgte bis jetzt noch nicht.

Das folgende Kapitel 2 gibt eine Einführung zu dem Thema PKIs und digitale Signaturen. Neben den theoretischen Grundlagen werden auch die Komponenten und Prozesse in einer PKI, sowie aktuelle Anwendungen vorgestellt.

Kapitel 3 behandelt den Einsatz von digitalen Signaturen nach dem heutigen Stand der Technik. Dabei spielen Standards und Gesetze eine wichtige Rolle. Hier werden auch die an der Erzeugung und Verifikation von digitalen Signaturen beteiligten Prozesse detailliert besprochen.

Kapitel 4 schafft die Basis für eine Modellierung der Verifikation von digitalen Signaturen. Zuerst werden grundlegende Begriffe wie Vertrauen und Sicherheit definiert. Unter Verwendung dieser Definitionen werden dann die Anforderungen an eine Beschreibungssprache zur Signaturprüfung formuliert. Im Anschluss werden einige Möglichkeiten zur Beschreibung von Systemen in der Informationsverarbeitung betrachtet. Dabei erweist sich die BAN-Logik als geeignete Basis für eine Sprache zur Modellierung der Verifikation von digitalen Signaturen.

In Kapitel 5 wird mit Hilfe der BAN-Logik die SigBAN-Logik entworfen. Das Augenmerk liegt dabei auf einer formalen Vorgehensweise bei der Definition der SigBAN-Logik. Nach der Erstellung fundamentaler Axiome, werden die neuen Regeln der SigBAN-Logik als Theoreme aus diesen Axiomen und den Regeln der BAN-Logik abgeleitet. Anschließend werden in einem einfachen Anwendungsbeispiel die Konsistenz und grundlegende Eigenschaften der SigBAN-Logik demonstriert.

Die Modellierung der Verifikation von digitalen Signaturen nach dem deutschen Signaturgesetz ist Bestandteil von Kapitel 6. Hier werden die für die Prüfung relevanten Abschnitte aus dem Signaturgesetz, der Signaturverordnung und den Interoperabilitäts-Spezifikationen mit der SigBAN-Logik analysiert. Dabei werden Lücken bei einem signaturgesetzkonformen Zeitstempeldienst erkannt und ein Verbesserungsvorschlag mit der SigBAN-Logik formuliert. Zusätzlich stellen sich weitere Schwachstellen bei den in den Interoperabilitäts-Spezifikationen beschriebenen Gültigkeitsmodellen heraus. Die SigBAN-Logik vermeidet diese Probleme bei der zeitlichen Betrachtung durch feste Vorgaben im zweiten Axiom.

Kapitel 7 befasst sich mit besonderen Zertifikatsformaten und der SigBAN-Modellierung von Vertrauensbeziehungen in unterschiedlich strukturierten PKIs. Dabei werden Attributs- und Mehrfachzertifikate, sowie deren Implikationen auf den Aufbau einer PKI untersucht. Außerdem wird für eine hierarchische Vertrauensstruktur ein Schema für die Verifikation mit der SigBAN-Logik aufgestellt.

Das achte Kapitel schließt die vorliegende Arbeit mit einem Ausblick auf weitere Einsatzmöglichkeiten und Erweiterungen der SigBAN-Logik ab.

2. Technische Grundlagen

2.1 Überblick

Dieses einführende Kapitel befasst sich mit den Grundlagen und dem Aufbau einer Public-Key-Infrastruktur (PKI). Als erstes werden die Basisdienste, die eine PKI bereitstellt, erläutert. Diese Dienste beruhen auf den Eigenschaften von bestimmten Krypto-Algorithmen. Neben den namensgebenden Public-Key-Algorithmen wird auch die Funktionsweise von symmetrischen Verfahren und Hash-Algorithmen beschrieben.

Ein weiterer Abschnitt stellt die technische Umsetzung der kryptographischen Verfahren in einer PKI vor. Hier werden die einzelnen PKI-Komponenten und ihre Rolle aufgeführt. Dabei findet auch eine Betrachtung der Daten statt, die von den einzelnen Instanzen erzeugt oder zwischen ihnen ausgetauscht werden. Erst durch das Zusammenspiel aller Komponenten entsteht eine Sicherheits-Infrastruktur, die eine Umsetzung der am Anfang erwähnten Basisdienste in Applikationen ermöglicht.

Der letzte Abschnitt ist dem Einsatz von PKIs nach dem derzeitigen Stand der Technik gewidmet. Hier wird ein kurzes Bild der vielfältigen Einsatzmöglichkeiten von PKIs skizziert und drei weit verbreitete Anwendungen vorgestellt.

2.2 PKI-Grundlagen

2.2.1 Basisdienste in einer PKI

In einer PKI werden Dienste zur Erhöhung der Sicherheit von Prozessen in der Datenverarbeitung einem begrenzten Teilnehmerkreis von Personen oder technischen Systemen zur Verfügung gestellt (siehe Abbildung 2.1). In Kombination mit der Verwendung von digitalen Signaturen ist eine häufig genutzte PKI-Funktionalität die Verschlüsselung von Daten. Zum Beispiel wird zwischen einzelnen PKI-Teilnehmern durch Ende-zu-Ende-Verschlüsselung eine vertrauliche Datenübertragung unabhängig von dem zugrundeliegenden Transportkanal ermöglicht. Eine unberechtigte Instanz hat somit keinen lesenden oder schreibenden Zugriff auf die übermittelten Daten. In diesem Zusammenhang spielt auch der Schutz der Daten vor Veränderung auf dem Weg zum Empfänger eine wichtige Rolle. Verhindert wird sowohl die Modifikation eines einzelnen Datensatzes (Integrität), als auch die Fälschung durch das Einspielen von fremden Daten (Authentizität).

Ein weiterer Dienst stellt die Verbindlichkeit einer Aussage in einer PKI sicher. Ein Teilnehmer kann eindeutig nachvollziehen, von welcher anderen Instanz in einer PKI ein signierter Datensatz stammt. Insbesondere dieser Basisdienst eignet sich um manuelle Unterschriften zu ersetzen.

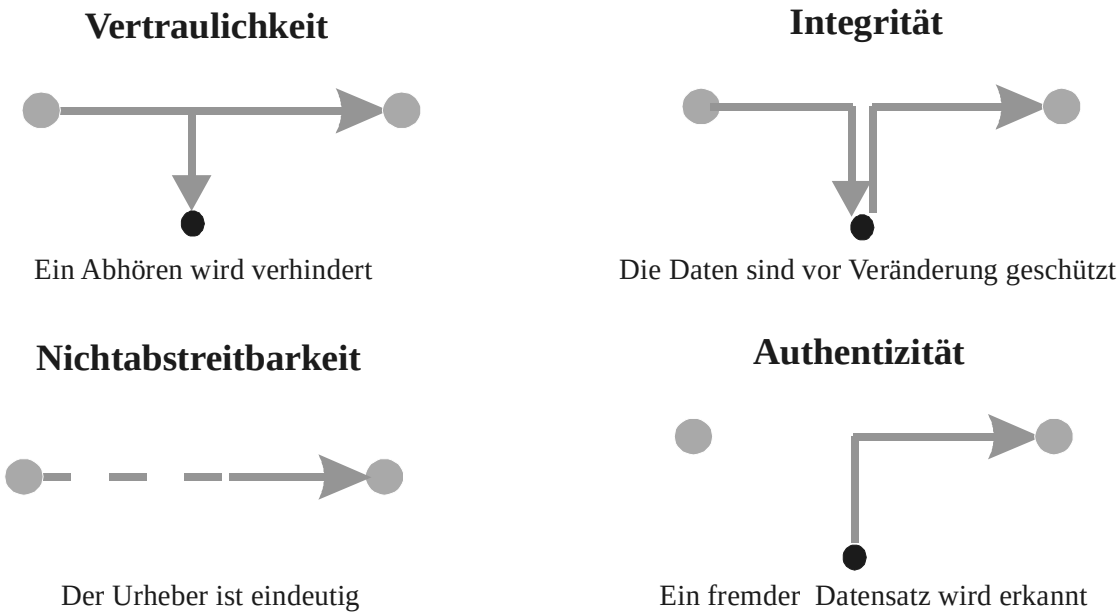

Abbildung 2.1 *Basisdienste in einer PKI*

2.2.2 Kryptographische Grundlagen

Dieser Teil der Arbeit befasst sich mit den in einer PKI verwendeten Arten von Krypto-Algorithmen (siehe [SCH96], [STI01], [MENE97]). Jede Gruppe von Algorithmen hilft auf bestimmte Art, die zuvor vorgestellten Basisdienste zu realisieren. In diesem Teil der Arbeit wird ein Überblick über die spezifischen Eigenschaften jeder Gruppe, die bekanntesten Vertreter und ihr Einsatz im Zusammenhang mit Applikationen gegeben. Ergänzend wird auch kurz auf die mathematischen Grundlagen eingegangen.

Neben den Public-Key-Algorithmen, deren Name bereits auf eine Verwendung in einer PKI hinweist, werden auch symmetrische Verfahren und Hash-Algorithmen in PKI-Anwendungen eingesetzt. Außer den Hash-Algorithmen arbeiten alle Krypto-Algorithmen, wie in Abbildung 2.2 dargestellt, mit Schlüsseln. Während bei symmetrischen Verfahren derselbe Schlüssel zum Ver- und Entschlüsseln verwendet wird, werden im Zusammenhang mit Public-Key-Algorithmen unterschiedliche Schlüssel genutzt.

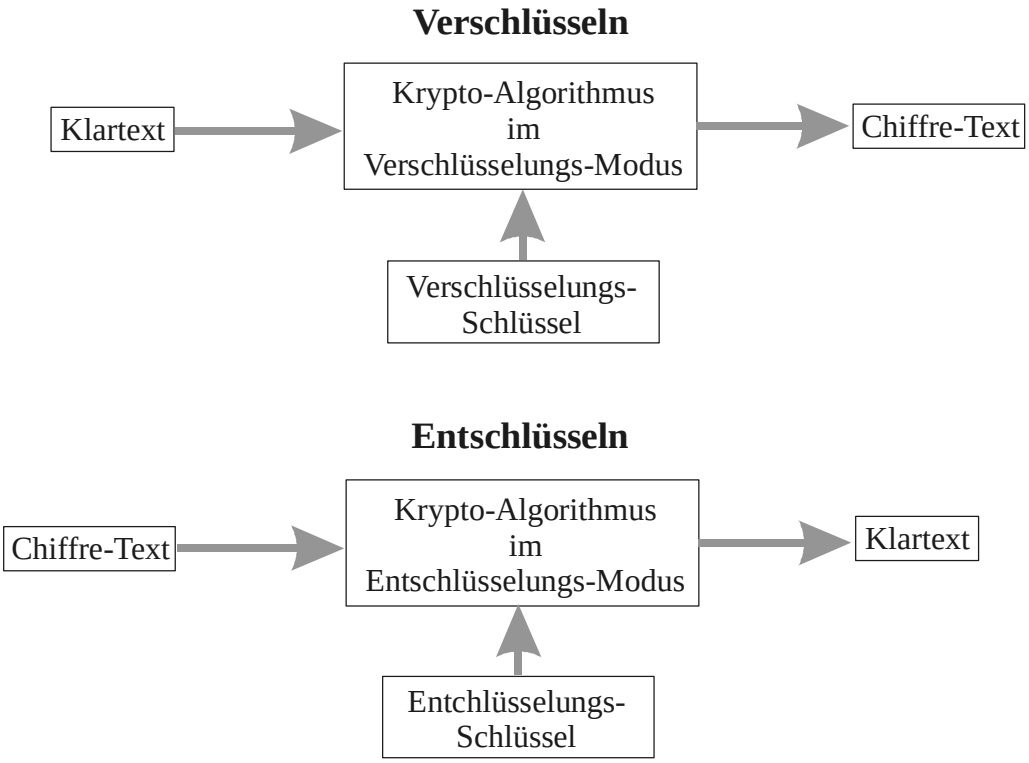

Abbildung 2.2 *Funktionsweise von Krypto-Algorithmen*

a) Public-Key-Algorithmen

Das Herzstück von PKIs sind asymmetrische Algorithmen, die auch Public-Key-Algorithmen genannt werden. Die Grundlagen für asymmetrische Verfahren wurden von Martin Hellman und Whitfield Diffie zuerst 1976 vorgestellt [DH76]. Die Besonderheit dieser Algorithmen ist, dass zum Verschlüsseln ein anderer Schlüssel als zum Entschlüsseln verwendet wird. Beide Schlüssel müssen jedoch korrespondieren und werden immer als ein gemeinsames Schlüsselpaar bestehend aus einem öffentlichen und einem privaten Schlüssel erzeugt. Dabei ist es bei entsprechend hohen Schlüssellängen und dem derzeitigen Stand der Technik nicht möglich, den privaten Schlüssel aus dem öffentlichen in begrenzter Rechenzeit zu ermitteln.

In einer PKI wird der öffentliche Schlüssel (Public Key) in Form eines Zertifikates (siehe Zertifikatsformate in [X50999] oder [GAR95]) für alle anderen PKI-Teilnehmer veröffentlicht, während der private (Private Key) unter allen Umständen geheimgehalten werden muss. Beide Schlüssel werden im Zusammenhang mit einer sogenannten Einwegfunktion benutzt. Diese spezielle Operation ermöglicht die Verschlüsselung von Daten mit einem der beiden Schlüssel. Die Umkehroperation, die wieder den ursprünglichen Datensatz erzeugt, kann ohne Kenntnis des anderen Schlüssels aber nicht durchgeführt werden. Eine Berechnung des Klartextes ohne den korrespondierenden Schlüssel ist deshalb mit den derzeitigen technischen Möglichkeiten nicht realisierbar.

Die Einwegfunktion von Public-Key-Algorithmen basiert auf der NP-Vollständigkeit (Nichtdeterministisch Polynomiell) bestimmter mathematischer Problemstellungen (siehe Kapitel 5.2.4 „Die Klasse NP“ in [REPO99]). NP-vollständig sind Aufgaben, für deren Lösung nur Algorithmen zur Verfügung stehen, bei denen die Rechenzeit t auf einer deterministischen Turing-Maschine exponentiell zur Mächtigkeit der Eingangsparameter n anwächst $t = e^{f(n)}$. Dabei ist n eine frei wählbare Größe, die im Fall von Krypto-Algorithmen mit der Schlüssellänge korreliert. Dies bedeutet, eine Berechnung der Lösung ist in vertretbarer Rechenzeit auf einer deterministischen Turing-Maschine unmöglich, wenn n entsprechend gross ist und bestimmte Informationen (kryptographische Schlüssel) fehlen. Es läßt sich nachweisen, dass alle technisch realisierbaren Systeme mit einer deterministischen Turing-Maschine zu beschreiben sind. Dagegen ist die nichtdeterministische Turing-Maschine ein Gedankenmodell zu einem Automaten, der parallel alle Vermutungen zur Lösung des Problems durchspielt und somit in der Theorie die Aufgabe wieder auf ein polynomielles Problem zurückführt. In der Praxis ist der einzige Weg das NP-Problem auf die polynomielle Klasse P zu reduzieren die Kenntnis eines kryptographischen Schlüssels. Deshalb kann nur derjenige, der die Schlüsselinformation besitzt, die Aufgabenstellung mit vertretbarem Zeitaufwand $t = f(n)$ auf seinem Computer lösen.

Ein Beispiel hierfür ist die Faktorisierung des Produktes zweier grosser Primzahlen, auf dem der bewährte RSA-Algorithmus (benannt nach seinen Erfindern R.Rivest, A.Shamir und L.Adleman) beruht [SCH96]. Die Multiplikation zweier grosser Primzahlen liegt in der Klasse P, während die Umkehrfunktion d.h. die Primfaktorzerlegung nach dem heutigen Stand der Mathematik NP-vollständig ist. Es gibt derzeit keinen Algorithmus der das Problem der Faktorisierung in die Klasse P überführt.

Neuere asymmetrische Krypto-Algorithmen sind die verschiedenen Varianten der ECC (Elliptic Curve Cryptography). Unter anderem wird bei ECC-Verfahren die NP-vollständige Problemstellung im Zusammenhang mit der Berechnung des diskreten Logarithmus, auf der auch das schon länger bekannte ElGamal-Kryptosystem beruht, genutzt (siehe [ROS99]). Bei ECC-Verfahren ist insbesondere die kürzere Schlüssellänge und höhere Performance ein Vorteil gegenüber dem derzeit noch am weitesten verbreiteten RSA-Algorithmus. Aus diesem Grund und der vermutlich höheren Sicherheit (siehe [MAU94]) wird in Zukunft der ECC eine immer höhere Bedeutung zukommen.

Eine Anwendung finden asymmetrische Algorithmen bei der Signaturerzeugung und Verifikation. Der Signierer unterschreibt mit seinem, nur ihm zugänglichen, privaten Schlüssel. Jeder kann jetzt mit dem öffentlichen Schlüssel, aus dem Zertifikat des signierenden PKI-Teilnehmers über eine Funktion aus der Klasse P die Signatur einfach verifizieren. Um aber eine neue Signatur nur mit dem öffentlichen Schlüssel zu erzeugen, müßte ein NP-vollständiges Problem gelöst werden. Nur für den Signierer liegt aufgrund der Kenntnis des privaten Schlüssels die Signaturerzeugung in der Klasse P. Es kann somit angenommen werden, dass nur derjenige, der den privaten Schlüssel kennt, die Unterschrift erzeugt haben kann.

b) Hash-Funktionen

Hash-Funktionen sind Einwegfunktionen, zu denen es keine Umkehrfunktion mit einer eindeutigen Zuordnung geben kann. Ursache dafür ist, dass eine wesentlich größere Ausgangsmenge auf eine viel kleinere Zielmenge abgebildet wird (siehe Abbildung 2.3). Dadurch geht Information unwiederbringlich verloren. Eine Verwendung zur Verschlüsselung von Daten ist deshalb unmöglich.

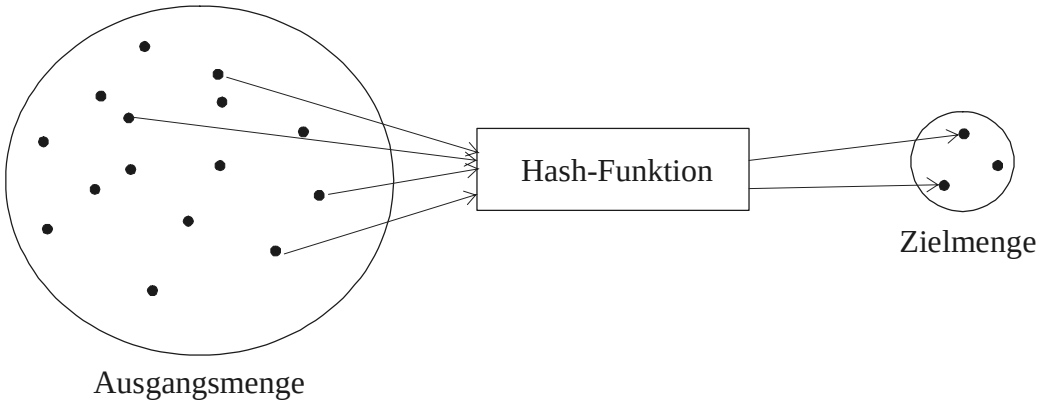

Abbildung 2.3 *Abbildungseigenschaft von Hash-Funktionen*

Die Abbildungseigenschaft der Hash-Funktion wird bei der Signaturerzeugung genutzt. Statt die Signatur direkt über einen Text zu berechnen, wird zuerst der Hash-Wert aus dem Textinhalt (Digest) gebildet und dann dieser signiert. Dabei ist ein Vorteil, dass der Digest eine feste Länge (meistens 128 Bit) aufweist, während der signierte Text beliebige Größe besitzen kann.

Ein wichtiges Merkmal für den Einsatz von Hash-Funktionen im Zusammenhang mit der Erzeugung digitaler Signaturen ist die Kollisionsresistenz. Es muss technisch unmöglich sein, zu einem Wert aus der Zielmenge zwei oder mehrere Werte aus der Ausgangsmenge zu ermitteln. Es darf also mit vertretbaren Rechenaufwand nicht gelingen zwei Nachrichten M_1 und M_2 zu finden, die denselben Hash-Wert $H(M_1) = H(M_2)$ besitzen. Wäre die Kollisionsresistenz nicht gegeben, bestünde die Gefahr, dass zu einer Signatur neben dem Originaltext ein weiterer passender Text ermittelt wird und sich damit die elektronische Unterschrift nicht mehr eindeutig auf ein Dokument bezieht.

Häufig wird bei der Signaturerzeugung der SHA-1 oder RIPE-MD Algorithmus verwendet. Obwohl der MD5 noch in vielen Applikationen eingesetzt wird, gilt er nach dem heutigen Stand der Technik als unsicher. Eine Analyse von Boer und Bosselaers konnte Kollisionen mit dem MD5 Algorithmus produzieren [BB94].

c) Symmetrische Algorithmen

Bei der Erstellung und der Verifikation von digitalen Signaturen spielen symmetrische Algorithmen keine Rolle. In einer PKI sind sie allerdings sehr wichtig, um die Vertraulichkeit von Daten durch Verschlüsselung sicherzustellen, da symmetrische Verfahren eine wesentlich höhere Performance als Public-Key-Algorithmen bei der Ver- und Entschlüsselung besitzen. Aus diesem Grund sind Public-Key-Algorithmen zur Verschlüsselung größerer Datenmengen un-

geeignet. Die vertrauliche Datenübertragung in den heutigen Breitbandnetzen wird deshalb immer über symmetrische Block-Verfahren sichergestellt. Die andere Art von symmetrischen Algorithmen, sogenannte Stream-Verfahren, werden im Zusammenhang mit kommerzieller Verschlüsselung selten eingesetzt und finden in einer PKI keine Anwendung.

Im Unterschied zu Public-Key-Algorithmen nutzen symmetrische Algorithmen meistens denselben Schlüssel für die Ver- und Entschlüsselung. Werden zwei Schlüssel verwendet, besteht immer die Möglichkeit aus dem Entschlüsselungs- den Verschlüsselungsschlüssel zu berechnen. Somit hängt die Absicherung vor unbefugtem Zugriff von der Geheimhaltung eines Schlüssels in einem begrenzten Teilnehmerkreis ab.

Der bekannteste Vertreter von symmetrischen Verfahren ist der Mitte der siebziger Jahre von IBM entwickelte DES (Data Encryption Standard). In der Vergangenheit war jedoch die Sicherheit des DES-Algorithmus aufgrund der Schlüssellänge von nur 56 Bit nicht ausreichend. Deshalb wurde Mitte der achtziger Jahre durch Modifikation der Triple-DES mit 112 Bit konzipiert. Durch eine Hintereinanderschaltung von drei „normalen“ DES-Operationen konnte die Sicherheit erhöht werden (siehe [SCH96] Kapitel 12.6 Varianten des DES) und der Tripel-DES wurde die nächsten Jahre als de-facto Standard für sichere Verschlüsselung genutzt. Seit letztem Jahr hat das amerikanische NIST (National Institute of Standards and Technology) als Nachfolger für den DES den Algorithmus „Rijndael“ als neuen AES (Advanced Encryption Standard) ausgewählt.

In einer PKI werden Daten mit dem in Abbildung 2.4 und 2.5 dargestellten Hybridverfahren, das aus einer Kombination von symmetrischen und asymmetrischen Algorithmen besteht, ver- und entschlüsselt. Dabei wird das zu sichernde Dokument zuerst mit einem frei gewählten symmetrischen Schlüssel verschlüsselt. Anschließend wird der symmetrische Schlüssel mit dem öffentlichen Schlüssel des Empfängers verschlüsselt und zusammen mit dem Dokument übermittelt.

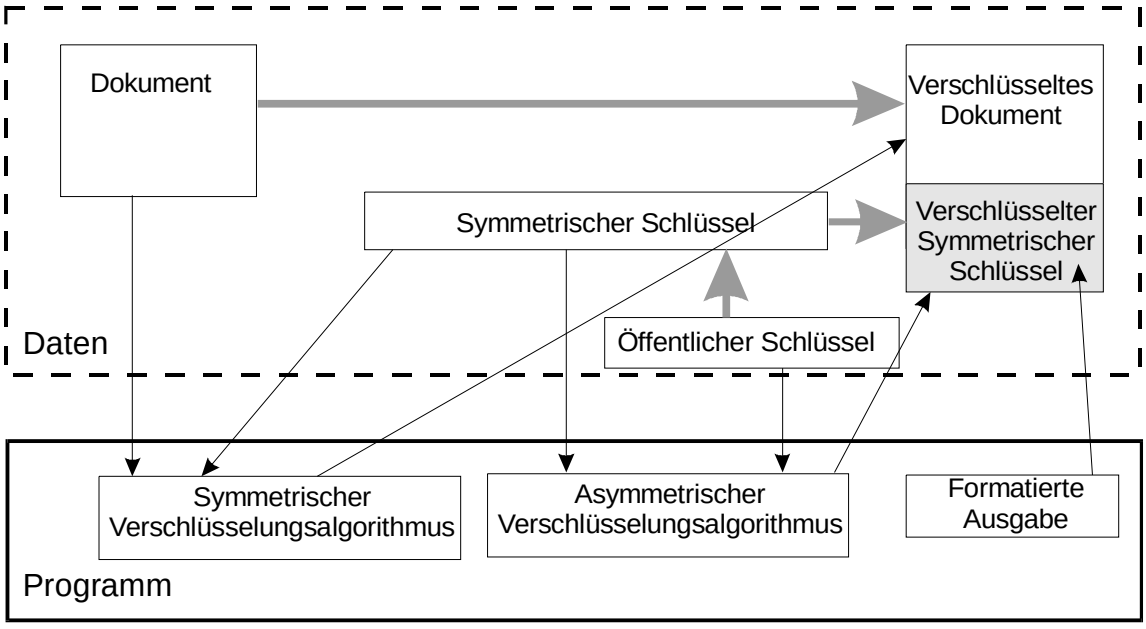

Abbildung 2.4 *Hybridverfahren zur Verschlüsselung*

Nach der vertraulichen Übermittlung, entschlüsselt der Empfänger zuerst den symmetrischen Schlüssel mit seinem privaten Schlüssel. Erst danach kann er mit dem symmetrischen Schlüssel den Originaltext wieder restaurieren.

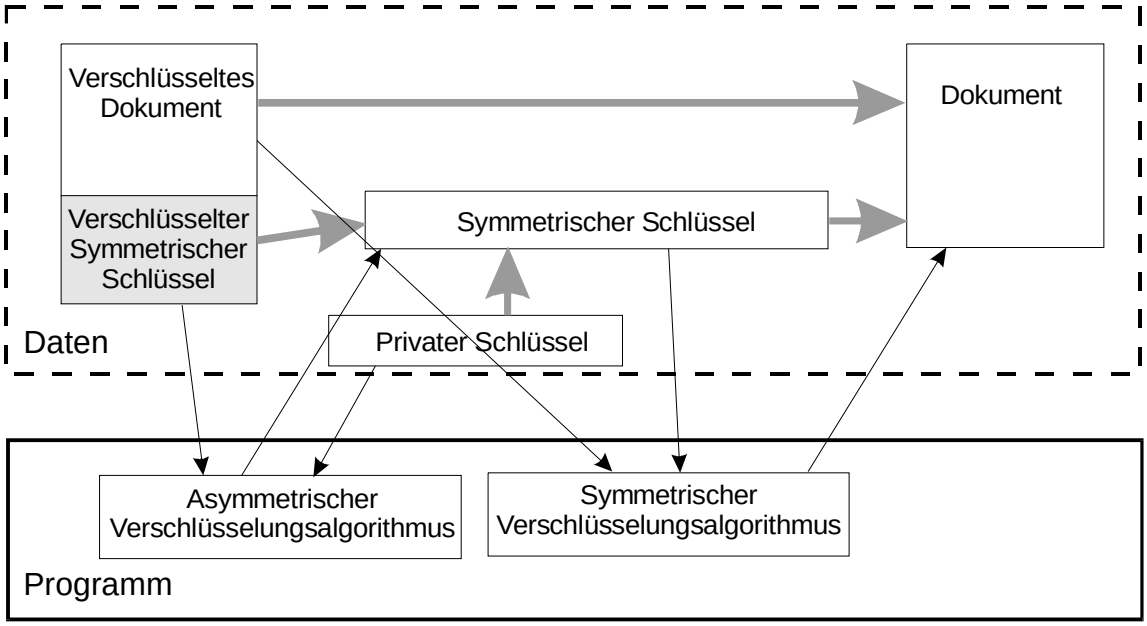

Abbildung 2.5 *Entschlüsselung im Hybridverfahren*

2.3 Komponenten und Prozesse in einer PKI

Wie jede Infrastruktur besteht auch eine PKI aus vielen einzelnen Komponenten, die in ihrer Gesamtheit erst ein funktionsfähiges Gebilde ergeben. Dabei werden alle Vorgänge, die den Aufbau oder eine Strukturänderung in einer PKI betreffen, von einer zentralen Verwaltung gesteuert. Diese zentrale Ausrichtung ist für die später durchgeführten Sicherheitsanalysen mit einer Beschreibungslogik essentiell.

2.3.1 Zentrale Verwaltung

Das Rückgrad einer PKI ist eine zentrale Verwaltung, die neue Benutzer und Dienste registriert und deren Daten, insbesonders Zertifikate, zentral administriert. Neue PKI-Teilnehmer werden durch die Herausgabe eines Zertifikates, das den öffentlichen Schlüssel des neuen Anwenders publiziert, in der PKI etabliert. Der private Schlüssel wird dagegen auf einem PSE (Personal Security Environment) nur für den PKI-Teilnehmer verfügbar gespeichert. Vorhandene PKI-Benutzer können über das Sperrmanagement der zentralen Verwaltung von der Teilnahme ausgeschlossen werden. Ein Grund dafür kann zum Beispiel der Verlust des PSEs sein. Die Prozesse zur zentralen Verwaltung werden in den einzelnen Komponenten wie Zertifizierungsstelle, Registrierstelle und Verzeichnisdienst realisiert.

Die zentrale Verwaltungsinstanz ist auch für die organisatorischen und rechtlichen Aspekte beim Zusammenspiel der einzelnen PKI-Teilnehmer und Dienste verantwortlich. Dabei werden die „Spielregeln“ in Form einer sogenannten CP (Certificate Policy) und eines CPS (Certificate Practice Statement) festgehalten [CP99]. In diesen standardisierten Dokumenten werden die internen Abläufe in einer PKI geregelt. Dazu gehören zum Beispiel die Bedingungen für eine Teilnahme, oder die Gültigkeitsdauer und der Verwendungszweck von Zertifikaten. CP und CPS stellen eine gemeinsame Grundlage für das Vertrauen der PKI-Teilnehmer in die zentrale Verwaltung dar. Deshalb müssen CP und CPS wie die Zertifikate auch, für jeden PKI-Teilnehmer im Verzeichnisdienst einsehbar sein.

2.3.2 Zertifizierungsstellen

Die Autorisierung von neuen Benutzern und Diensten in einer PKI findet immer zentral in einer Zertifizierungsstelle statt. Die Zertifizierungsstelle, auch CA (Certification Authority) genannt, ist eine Basiskomponente beim Aufbau einer PKI. Hier wird das Zertifikat generiert und signiert, das jeden neuen Teilnehmer oder Dienst in die PKI integriert. Umgekehrt können Teilnehmer und Dienst aus der PKI über die Veröffentlichung von Sperrinformationen wieder ausgeschlossen werden. Alternativ zu den unterschiedlichen Formen des Zertifikat-Widerrufs, können von der CA auch Positivlisten mit allen derzeit gültigen Instanzen herausgegeben werden. Erst durch die von der CA generierte Signatur erlangt ein Zertifikat oder eine Sperrinformation Gültigkeit in der PKI. Aus diesem Grund unterliegt das Schlüsselmanagement für den privaten Schlüssel der Zertifizierungsstelle den höchsten Sicherheitsanforderung in einer PKI (siehe deutsches Signaturgesetz §2 Absatz 12a, [SIGG01]).

a) Zertifikat und Sperrinformation

Das Zertifikat ist ein von der CA signierter Datensatz, der Informationen über den Zertifikatsinhaber, dessen Rechte in der PKI, den öffentlichen Schlüssel des Inhabers und eine Gültigkeitsdauer enthält. In den meisten Fällen sind diese Bestandteile in einem von der X509-Spezifikation [X50999] vorgeschriebenen Format angeordnet. Für die Verifikation einer digitalen Signatur ist immer das zugehörige Zertifikat notwendig. Im Zertifikat steht der öffentliche Schlüssel, der mit dem bei der Signaturerzeugung genutzten privaten Schlüssel korrespondiert.

Abhängig von der durch die Signatur autorisierten Instanz und dem Verwendungszweck kann ein Zertifikat unterschiedliche Inhalte aufweisen. Eine wichtige Berechtigung, die in den Zertifikatserweiterungen nach X.509 Version3 [X50999] festgehalten wird, ermöglicht die Erstellung von digitalen Signaturen. Andere Erweiterungsbits im Zertifikat erlauben die Verschlüsselung von statischen Daten oder die Absicherung einer Übertragungsstrecke. Ein Internet-Server kann zum Beispiel ein Server-Zertifikat besitzen, um eine gesicherte Verbindung über das SSL-Protokoll aufzubauen. Auch die CA selbst ist durch ein Zertifikat legitimiert, wieder andere Zertifikate auszustellen. Im Falle einer Wurzel-CA ist dieses Zertifikat mit dem eigenen Schlüssel signiert. Anderenfalls hat eine übergeordnete Zertifizierungsstelle das Zertifikat unterschrieben.

Die Sperrung eines Zertifikates muss ebenfalls durch die CA bestätigt werden. Aus diesem Grund werden neben Zertifikaten auch signierte Sperrinformationen oder Positivlisten von der CA publiziert. Eine Möglichkeit dazu ist die Herausgabe von Sperrlisten nach dem X509-Standard [X50999]. In vielen Fällen, wie dem deutschen Signaturgesetz, darf aus Sicherheitsgründen die Sperrung eines Zertifikates nicht mehr rückgängig gemacht werden.

b) Anordnung von Zertifizierungsstellen

Eine Zertifizierungsstelle kann als eine eigenständige neue Wurzel angelegt, oder in die Hierarchie von bereits vorhandenen CAs eingebunden werden. Im zweiten Fall wird die Zertifizierungsstelle von einer bereits existierenden CA als besonderer Teilnehmer, der autorisiert ist weitere Benutzer oder Dienste zu etablieren, registriert. Wird hingegen eine eigenständige Wurzel-CA angelegt, stellt diese Instanz den Endpunkt in der Zertifizierungshierarchie dar. Diese unterschiedlichen Anordnungen sind von besonderer Bedeutung bei der Verifikation von Signaturen. Bei der Überprüfung einer digitalen Signatur ist der gesamte Zertifizierungspfad vom Signaturzertifikat bis hin zur Wurzel-CA zu betrachten. Eine Signatur kann nur gültig sein, wenn alle Zertifikate im Pfad Gültigkeit besitzen.

2.3.3 Registrierstellen

Bevor die PKI-Teilnehmerdaten im Zertifikat durch die CA festgehalten werden, erfolgt eine Erfassung in der Registrierstelle, die auch als RA (Registration Authority) bezeichnet wird. Neben der Aufnahme neuer Benutzer ist die Registrierstelle auch für die sichere Interaktion mit der Zertifizierungsstelle verantwortlich. Nach einer erfolgreichen Registrierung und Zertifizierung verfügt der neue PKI-Teilnehmer über ein in der RA erzeugtes PSE. Meistens wird das PSE in Form einer Chipkarte [RAEF99] oder einer Datei nach dem „Personal Information Exchange Syntax Standard“ (siehe Kapitel 3.2.2) [PKCS12] von der Registrierstelle ausgehändigt.

2.3.4 Verzeichnisdienst

Im Verzeichnisdienst, der oft auch als DS (Directory Service) bezeichnet wird**,** werden Zertifikate und Sperrinformationen, sowie die in 2.3.1 erwähnte CP (Certificate Policy) und das CPS (Certificate Practice Statement) für alle PKI-Teilnehmer zugänglich veröffentlicht. Neben diesen standardisierten Datenstrukturen finden sich auch weitere, von der einzelnen PKI-Realisierung abhängige Informationen im Verzeichnisdienst. Der DS erhält die Daten entweder von der Zertifizierungsstelle oder der Registrierstelle. Während des Betriebs einer PKI ist eine hohe Verfügbarkeit des DS zu garantieren, da PKI-Teilnehmer ständig aktualisierte Informationen vom Verzeichnisdienst beziehen. Insbesondere wird der größte Teil, der für eine Verifikation notwendigen Daten, vom DS zur Verfügung gestellt. Häufig erfolgt der Zugriff auf den Verzeichnisdienst durch das standardisierte LDAP-Protokoll (Lightweight Directory Access Protocol) [HOW99].

2.3.5 Synthese eines Zertifikates

Nach der Erfassung der Benutzerdaten, wird in der Registrierstelle ein Schlüsselpaar für den in der PKI verwendeten asymmetrischen Algorithmus generiert. Mit dem privaten Schlüssel wird ein Zertifikatsantrag, der die Teilnehmerdaten enthält, an die Zertifizierungsstelle übermittelt. Nach der erfolgreichen Prüfung des Antrags liefert die CA das erstellte Teilnehmerzertifikat an die RA zurück. Das Zertifikat wird auch gleichzeitig im Verzeichnisdienst veröffentlicht und von der Registrierstelle zusammen mit dem privaten Schlüssel im PSE an den neuen PKI-Teilnehmer ausgehändigt.

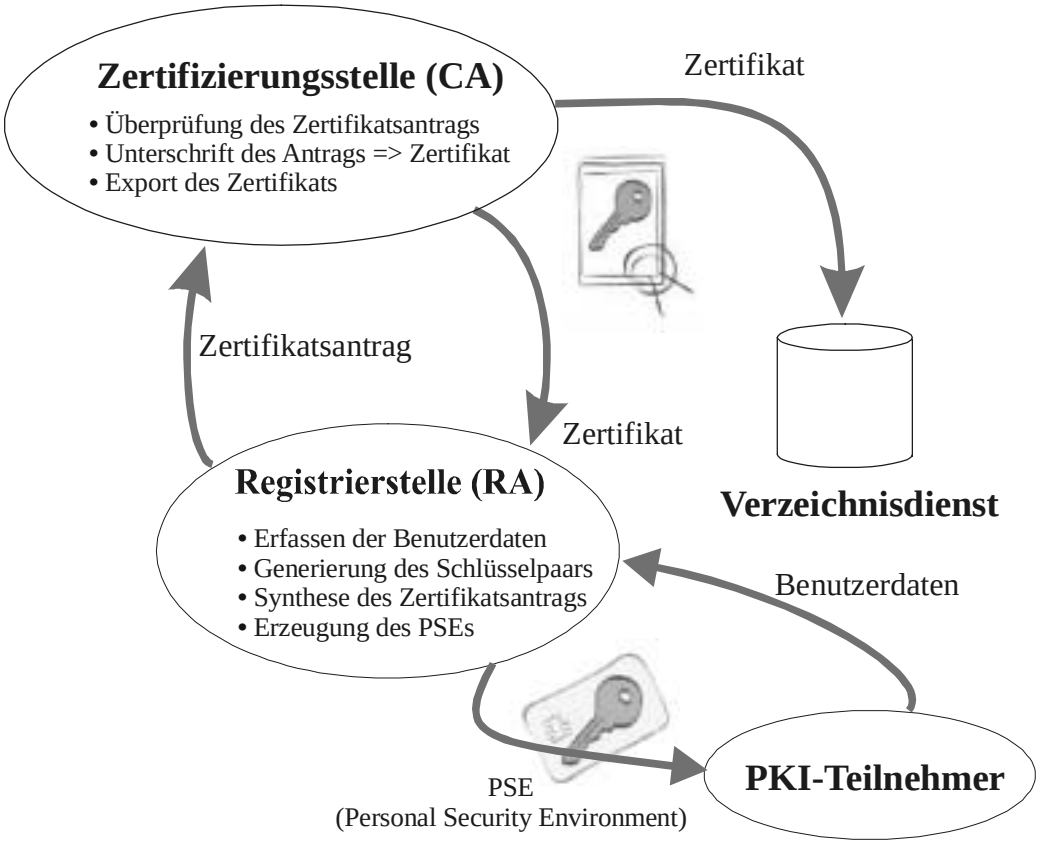

Abbildung 2.6 *An der Erfassung eines neuen PKI-Teilnehmers beteiligte Instanzen*

2.3.6 Sperrung eines Zertifikates

Die Sperrung eines Zertifikates wird meistens durch den PKI-Teilnehmer selbst ausgelöst. In vielen Fällen besitzt er ein Passwort, die sogenannte CRN (Certificate Revocation Number), mit der er seine Anforderung zur Sperrung autorisiert. Nachdem die Berechtigung des Teilnehmers zur Sperrung von der Registrierstelle überprüft wurde, sendet diese einen Antrag zur Zertifikatssperrung an die CA. Besitzt dieser Antrag Gültigkeit, wird von der CA eine neue CRL (Certificate Revocation List) [X50999], die einen eindeutigen Hinweis auf das gerade gesperrte Zertifikat enthält, im Verzeichnisdienst veröffentlicht. Gleichzeitig wird die RA von der Sperrung benachrichtigt. Diese informiert wiederum den PKI-Teilnehmer über den erfolgreichen Zertifikatswiderruf.

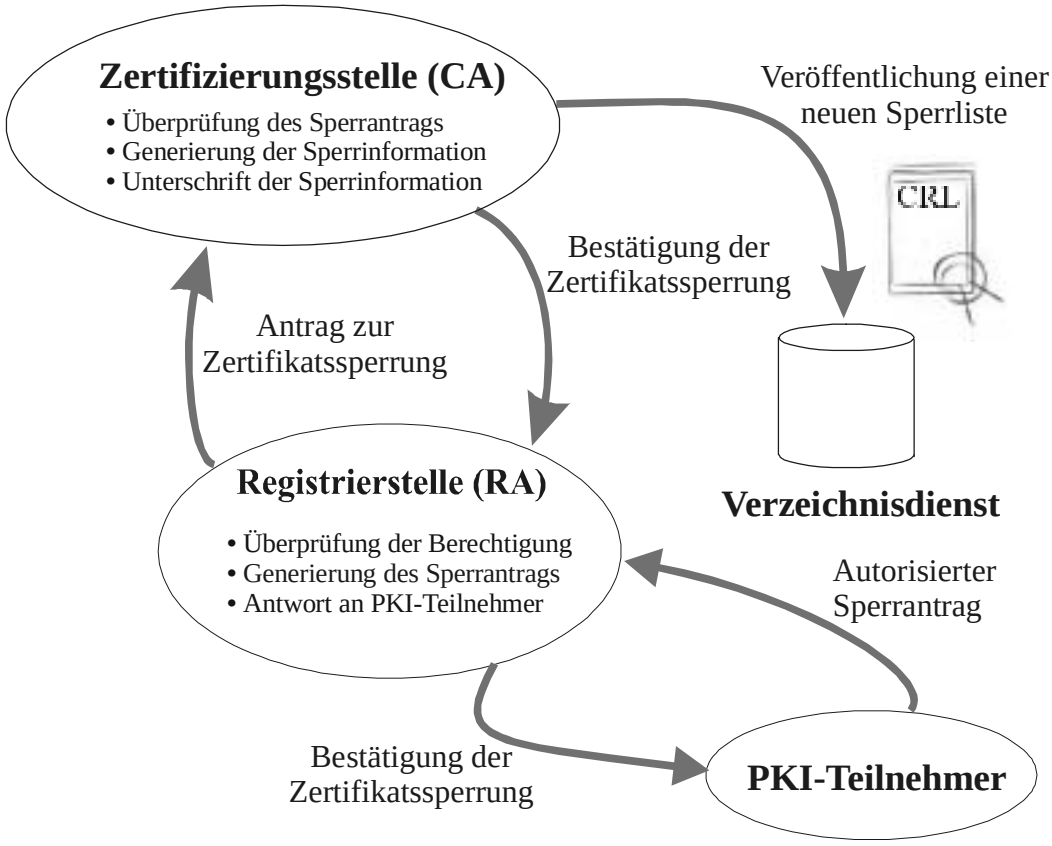

Abbildung 2.7 *Am Zertifikatswiderruf beteiligte Instanzen*

2.4 Aktuelle Anwendungen

Hier werden exemplarisch einige der verbreitesten PKI-Anwendungen kurz vorgestellt. Alle Beispiele befassen sich mit der Absicherung des per se unsicheren Datenverkehrs im Internet. Zuerst werden zwei Anwendungen vorgestellt, die in jedem aktuellen Betriebssystem zur Verfügung stehen. Das letzte Beispiel betrachtet eine PKI, die im Zusammenhang mit sicheren Finanztransaktionen eingesetzt wird.

2.4.1 Sichere Mail-Nachrichten im S/MIME-Format

Zur sicheren Übermittlung der elektronischen Post im Internet gibt es den S/MIME-Standard (Secure Multipurpose Internet Mail Extensions) [SMIME98]. S/MIME nutzt die „Cryptographic Message Syntax“ (siehe PKCS#7-Standard [PKCS7]), um das ungesicherte MIME-Protokoll mit den Diensten einer PKI zu erweitern. Nachrichten können digital signiert und verschlüsselt in der PKCS#7-Struktur abgelegt werden. Zusätzlich befindet sich das Zertifikat des Absenders der Nachricht ebenfalls im PKCS#7-Format. Der Empfänger kann anhand des beigefügten Zertifikates eine einfache Verifikation der digitalen Signatur durchführen.

Für die Verschlüsselung von Nachrichten muss immer das Zertifikat des anderen Kommunikationspartners vorliegen, da das bereits in Kapitel 2.2.2c vorgestellte Hybridverfahren genutzt wird. Das Zertifikat des Empfängers kann vom Absender entweder über den Verzeichnisdienst oder aus einer vorherigen Nachricht des Empfängers gelesen werden.

2.4.2 Sichere Datenübertragung mit dem SSL-Protokoll

Das SSL-Protokoll (Secure Socket Layer) wurde von Netscape zur Absicherung der Datenübertragung zwischen Browsern und Servern im Internet entwickelt. Obwohl die SSL-Spezifikation nicht von einer IETF-Arbeitsgruppe (Internet Engineering Task Force) erstellt ist, wurde SSL aufgrund der weiten Verbreitung von Netscape Browsern ein de facto Internet-Standard. SSL verwendet nach dem ISO/OSI-Referenzmodell die Applikationsschicht zur sicheren Kommunikation.

Eine SSL-Anbindung kann durch die flexible Gestaltung des SSL-Handshakes in zwei verschiedenen Varianten erfolgen. Eine einseitige Authentisierung des Servers ist genauso möglich, wie eine gegenseitige Authentisierung zwischen Server und Client. In vielen Fällen wird nur die erste Möglichkeit genutzt, um dem Internet-Anwender die Sicherheit zu geben, dass er sich auf der richtigen Internetseite befindet.

Mit der in dieser Arbeit später vorgestellten BAN-Logik konnten in der ersten SSL-Version bereits Sicherheitslücken nachgewiesen werden. Seitdem wurde SSL bis zur aktuellen Version 3.0 bzw. TLS 1.0 (Transport Layer Security) weiterentwickelt [FRI96].

2.4.3 HBCI - PKI im elektronischen Zahlungsverkehr

Der HBCI-Standard (Home Banking Computer Interface) wurde von den deutschen Banken spezifiziert, um Finanztransaktionen zwischen Kunden und Banken im Internet zu ermöglichen [HBCI99]. Der HBCI-Standard soll das derzeit noch häufiger verwendete, aber unsichere und unkomfortable PIN-TAN-Verfahren durch die Verwendung einer PKI ablösen.

Das Ende-zu-Ende Sicherheitskonzept des HBCI-Standards definiert ein eigenes Nachrichtenformat, in dem verschlüsselte und signierte Daten zwischen dem Rechner der Bank und dem des Anwenders über das nicht gesicherte TCP/IP-Protokoll (Transmission Control Protocol/ Internet Protocol) ausgetauscht werden. Als PSE kann dabei zukünftig die EC-Karte genutzt werden. Auf dieser EC-Chipkarte sind dann zusätzlich ein Zertifikat und der Signaturschlüssel des Inhabers sicher abgelegt. Durch eine Signatur mit seinem privaten Schlüssel auf der EC-Karte, kann der Bankkunde die zur Kontoführung notwendigen Aktionen legitimieren.

2.5 Abschließende Kommentare

Kapitel 2 gibt einen Überblick über die Grundlagen, den Aufbau und die Anwendung von PKIs. Auf die hier vermittelten Informationen greifen die folgenden Kapitel, die schrittweise zur Definition der SigBAN-Logik hinführen, zurück. Viele der hier vorgestellten Abläufe haben weitreichende Implikationen für die Verifikation von digitalen Signaturen. Neben den mathematischen Grundlagen, die in der Syntax der BAN-Logik wiederzufinden sind, spielen

auch die Prozesse in einer PKI eine wichtige Rolle für die SigBAN-Logik. Ein Beispiel hierfür ist die Abbildung des bei der Zertifizierung entstehenden Pfades vom PKI-Teilnehmer zur Wurzel-CA in der SigBAN-Logik. Drei kurze Anwendungsbeispiele skizzieren abschließend den praktischen Einsatz von PKIs in alltäglichen Anwendungen.

3. Einsatz von digitalen Signaturen

3.1 Überblick

Der erste Abschnitt befasst sich mit den für eine PKI relevanten Standards und Gesetzen. An der Standardisierung arbeiten weltweit unterschiedliche Firmen und Organisationen, während bei der Gesetzgebung in der vorliegenden Arbeit speziell das deutsche Signaturgesetz betrachtet wird. Deutschland war im Jahr 1997 in Europa Vorreiter mit dem ersten Signaturgesetz [SIGG97] und der zugehörigen Signaturverordnung [SIGV97]. An der Standardisierung ist am längsten die amerikanische Firma RSA Laboratories Inc. tätig, die auch bis zum Jahr 2000 das Patent für den RSA-Algorithmus innehatte. Der erste PKCS-Standard (Public Key Cryptographic Standard) wurde bereits Anfang der neunziger Jahre von RSA Laboratories Inc. veröffentlicht. Eine weitere wichtige Rolle bei der Normierung von PKI-Diensten für den Einsatz im Internet übernimmt die IETF (Internet Engineering Task Force), die im Gegensatz zu RSA Laboratories Inc. nicht kommerziell arbeitet.

Im zweiten Abschnitt werden die zur Signaturerzeugung und -prüfung notwendigen Prozesse analysiert. Eine detaillierte Betrachtung dieser beiden Vorgänge ist eine Voraussetzung für den Entwurf einer Beschreibungssprache zur Verifikation von digitalen Signaturen. Hier wird auch darauf eingegangen, welche Auswirkungen die Gesetzgebung und die Standards auf die Generierung und Prüfung einer Signatur haben.

3.2 Standards und Gesetzgebung

Die im folgenden vorgestellten Standards spielen für die Verbreitung von PKIs eine wichtige Rolle. Sie gewährleisten die Interoperabilität von Applikationen unterschiedlicher Hersteller, den plattformunabhängigen Datenaustausch und die kontinuierliche Weiterentwicklung der Technologie. Die Standardisierung im PKI-Umfeld ist auf allen Ebenen bereits sehr weit fortgeschritten. Neben Standards zum Aufbau und zur Administration einer PKI, existieren auch viele Spezifikationen, die eine Anwendung der einzelnen PKI-Dienste in Applikationen definieren.

Im Gegensatz zur Standardisierung wurden PKIs in der Gesetzgebung erst später berücksichtigt. Das weltweit erste Gesetz über digitale Signaturen wurde in den USA Mitte der 90er Jahre als „Utah Digital Signature Act“ [SIGUT95] veröffentlicht. Zu dieser Zeit wurde gerade in Deutschland mit der Arbeit an einem Signaturgesetz begonnen, das von dem amerikanischen Vorbild im Bundesstaat Utah beeinflusst war. Das deutsche Signaturgesetz existiert derzeit in einer überarbeiteten zweiten Fassung, die an die europäischen Richtlinien zu digitalen Unterschriften angepasst wurde.

Staatliche Vorgaben spielen für die Verbreitung von digitalen Signaturen eine wichtige Rolle, da nur der Staat die Regeln für eine juristisch abgesicherte Anwendung schaffen kann. Deshalb wird in einem eigenen Abschnitt die historische Entwicklung und der aktuelle Stand der deutschen Gesetzgebung aufgezeigt und in Kapitel 6 die Verifikation nach den technischen Spezifikationen zum Signaturgesetz analysiert.

3.2.1 Standardisierung in der IETF

Die Organisation IETF (Internet Engineering Task Force) befasst sich in einzelnen Arbeitsgruppen mit der Standardisierung von Anwendungen für das Internet. Dazu gehören die bekannten Protokolle TCP/IP (Transmission Control Protocol/Internet Protocol), SMTP (Secure Mail Transfer Protocol) oder HTTP (Hyper Text Transfer Protocol). Im Bereich Sicherheitstechnik erstellt die PKIX-Arbeitsgruppe (Public Key Infrastructure X.509 Standard) innerhalb der IETF die Dokumente für den X.509-Standard.

Die bekannteste X.509-Spezifikation „Internet X.509 Public Key Infrastructure Certificate and CRL Profile“ [X50999] beinhaltet die Beschreibung eines Formates für Zertifikate und Widerrufslisten in ASN.1-Syntax (Abstract Syntax Notation). ASN.1 ist eine Beschreibungssprache für Datenformate [LAR00]. Auch die Bestandteile von „Certificate Policies“ (CP) und „Certificate Practice Statements“ (CPS) richten sich nach einem X.509-Dokument [CP99], in dem sie mit ASN.1 bitgenau festgelegt sind. Neben dieser für eine PKI elementaren Datenstrukturen, definiert die PKIX-Arbeitsgruppe auch die Protokolle zur Administration und den Betrieb einer PKI. Zum Beispiel wurde für die einfachere Verifikation von digitalen Signaturen von der IETF das OCSP-Protokoll (Online Certificate Status Protocol) [OCSP99] standardisiert. Über dieses Protokoll können vertrauenswürdig und sicher Zustandsinformationen über Zertifikate bei einem OCSP-Server abgefragt werden. Damit entfällt das aufwendige Auswerten der Sperrinformationen oder der Positivlisten durch die prüfende Instanz. Der OCSP-Server wird dazu von einer CA ermächtigt, über alle von dieser CA erstellten Zertifikate Auskunft zu erteilen. Ein anderes von der IETF definiertes Protokoll ist das CMP (Certificate Management Protocol). Es dient zur Standardisierung der Kommunikation innerhalb der zentralen Verwaltungsinstanz und beinhaltet unter anderem einen authentisierten Zertifikatsantrag von der RA an die CA. Andere aktuelle Arbeitsgebiete der PKIX-Gruppe sind Zeitstempeldienste und Attributszertifikate.

Durch die Bemühungen der IETF werden nicht nur die Möglichkeiten zur Absicherung des Internets, sondern auch die Grundlage für die weitere Verbreitung von PKIs geschaffen. Um die Basisdienste einer PKI sinnvoll nutzen zu können, ist die Spezifikation von Datenformaten und Protokollen notwendig. Sämtliche Arbeiten der IETF finden sich in Form von RFCs (Request for Comments) im Internet auf der Seite http://www.ietf.org/rfc.html.

3.2.2 Der PKCS-Standard

Die Firma RSA Laboratories Inc. rief bereits 1993 die Standardisierung der Public-Key-Kryptographie ins Leben, um die Verbreitung von PKIs und die Realisierung von interoperablen Anwendungen zu fördern. Ein Grund dafür ist sicherlich das Patent dieser Firma auf den RSA-Algorithmus, das im Jahr 2000 auslief. In Zusammenarbeit mit anderen Firmen werden von RSA Lab. die sogenannten PKCS-Standards (Public Key Cryptographic Standards) verabschiedet.

Standard	Inhalt
PKCS#1	Der Standard zum RSA-Algorithmus in dem alle für die Signaturerzeugung und Verschlüsselung notwendigen Prozesse definiert sind.
PKCS#3	Hier wird das Diffie-Hellmann Protokoll für den sicheren Schlüsselaustausch beschrieben.
PKCS#5	Dieses Dokument spezifiziert das PBE-Verfahren (Password Based Encryption), um aus einem Passwort einen Schlüssel zu generieren.
PKCS#6	Ein Standard für Zertifikatsformate, der aufgrund des bereits existierenden X.509 Version3 Standards nicht mehr weiter verfolgt wird.
PKCS#7	Die bereits vorgestellte Cryptographic Message Syntax.
PKCS#8	Ein Verfahren um den privaten Schlüssel sicher aufzubewahren.
PKCS#9	Dieser Standard erläutert bestimmte Attribute die in den anderen PKCS-Standards verwendet werden.
PKCS#10	Ein mögliches Format für einen Zertifikatsantrag von der Registrierstelle an die Zertifizierungsstelle.
PKCS#11	Hier wird die CRYPTOKI API für den Zugriff auf kryptographische „Token“ spezifiziert. Ein „Token“ ist zum Beispiel eine Chipkarte.
PKCS#12	Die bereits vorgestellte „Personal Information Exchange Syntax“.
PKCS#13	Hier wird der Einsatz von ECC-Verfahren in einer PKI definiert.
PKCS#14	Dieser in Arbeit befindliche Standard befasst sich mit der Generierung von Zufallszahlen PRNG (Pseudo Random Number Generation).
PKCS#15	Ein in Zusammenhang mit Chipkarten genutztes Format, um Schlüssel und Zertifikate auf einem „Token“ abzuspeichern.

Tab. 3.1 *Übersicht zu den PKCS-Standards*

Hohe Bedeutung kommt in den PKCS-Standards dem Datenaustausch zwischen Applikationen zu. Deshalb sind zwei häufig in Realisierungen umgesetzte PKCS-Standards die „Cryptographic Message Syntax“ (PKCS#7, siehe [PKCS7]) und die „Personal Information Exchange Syntax“ (PKCS#12, siehe [PKCS12]). Der PKCS#7-Standard wird von allen S/MIME-fähigen „E-Mail-Clients“ genutzt, da S/MIME auf Basis von PKCS#7 ein Format zur Absicherung von elektronischen Nachrichten definiert. Der PKCS#12-Standard dient dagegen zum Transport von vertraulichen Daten, wie privaten Schlüsseln. In vielen Fällen werden deshalb PKCS#12-Dateien als ein im Vergleich zur Chipkarte kostengünstigeres, aber wesentlich unsichereres PSE (Personal Security Environment), verwendet. Neben diesen und anderen Formaten (Message Syntax), beschreiben die PKCS-Standards auch die Anwendung bestimmter Krypto-Algorithmen (Specific Algorithms).

Es gibt derzeit die in Tabelle 3.1 aufgelisteten Standards PKCS#1 bis PKCS#15 ohne PKCS#2 und PKCS#4, die mittlerweile im PKCS#1-Standard aufgegangen sind.

3.2.3 Das deutsche Signaturgesetz (SigG)

Das erste deutsche Signaturgesetz (SigG) trat als Bestandteil des „Gesetzes zur Regelung der Rahmenbedingungen für Informations- und Kommunikationsdienste (IuKDG)“ am 1. August 1997 in Kraft [SIGG97]. Ergänzend zum SigG wurde die Signaturverordnung (SigV) mit den Ausführungsbestimmungen zum Einsatz von rechtlich abgesicherten digitalen Unterschriften am 1. November 1997 verabschiedet [SIGV97]. Die technische Umsetzung wird auf der nächsten Stufe, wie in Abbildung 3.1 dargestellt, durch den Massnahmenkatalog von der Regulierungsbehörde für Telekommunikation und Post (RegTP) vorgegeben. Auf der untersten Ebene in der Hierarchie befinden sich die Interoperabilitäts-Spezifikationen mit der Beschreibung konkreter Implementierungen zur Realisierung von rechtsverbindlichen digitalen Signaturen.

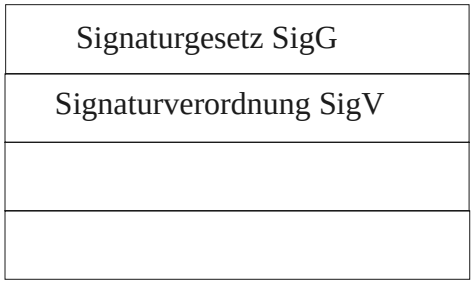

Abbildung 3.1 *Gesetzliche Rahmenbedingungen für den Einsatz digitaler Signaturen in Deutschland*

Am 22. März 2001 löste das „Gesetz über Rahmenbedingungen für elektronische Signaturen“ [SIGG01] das Signaturgesetz von 1997 ab. Dabei ist zu beachten, dass die SigV von 1997 bis zum 16. November 2001 Gültigkeit besaß und erst ungefähr ein halbes Jahr nach dem neuen SigG in einer aktualisierten Fassung [SIGV01] erschien. Ein Hauptgrund für das neue SigG waren die Anpassungen an die Forderungen der EU-Signaturrichtlinie [SIGEU99] vom Dezember 1999. Durch das neue Gesetz soll ein Austausch von digitalen Signaturen und den Leistungen von Zertifizierungsdienstanbietern (Trustcentern) innerhalb der EU gefördert wer-

den. Die im alten Gesetz vorgeschriebene, sehr aufwendige Akkreditierung von Trustcentern besitzt aus diesem Grund in der neuen Fassung nur freiwilligen Charakter. Deshalb werden auch die Signaturverfahren nach „sonstige“, „qualifizierte“ und „akkreditierte“ unterschieden. Die beiden letzteren Varianten beinhalten digitale Signaturen auf Basis einer PKI.

3.2.4 Interoperabilitäts-Spezifikationen zum SigG

Die Überprüfung von digitalen Signaturen ist ein wichtiger Aspekt in den technischen Spezifikationen zur Umsetzung der Vorgaben aus SigG und SigV (Interoperabilitäts-Spezifikationen). Nur wenn es technisch einwandfrei möglich ist, die digitale Signatur einem Urheber und einem Dokument zuzuweisen, kann die Beweiskraft einer manuellen Unterschrift erreicht werden. Bei dem Einsatz von digitalen Signaturen mit einer juristischen Bedeutung gelten die höchsten Anforderungen an die Sicherheit und Unverfälschbarkeit. Aus diesem Grund wird der Einsatz der in dieser Arbeit entworfenen SigBAN-Logik an einem SigG-konformen Beispiel demonstriert. Bei der Modellierung der Verifikation nach dem SigG in Kapitel 6 wird auf die technischen Beschreibungen der Prozesse aus den im Folgenden vorgestellten Spezifikationen zurückgegriffen.

a) SigI

Abschnitt	Inhalt
A1 Zertifikat	Das hier beschriebene Zertifikatsformat richtet sich nach dem PKIX Standard.
A2 Signatur	In diesem Teil werden die nach dem SigG erlaubten Krypto-Algorithmen und Austauschformate für verschlüsselte und signierte Daten festgelegt.
A3 Anwenderinfrastruktur	Der dritte Abschnitt definiert Prozesse in Applikationen, die von den PKI-Teilnehmern zur Verifikation und Signaturerzeugung genutzt werden.
A4 Zeitstempeldienst	Dieser Teil befasst sich mit der sicheren Datierung des Erstellungszeitpunktes von digitalen Signaturen.
A5 Verzeichnisdienst	Für einen SigG-konformen Verzeichnisdienst gelten die hier erwähnten, besonderen Anforderungen.
A6 Gültigkeitsmodell	Die Verifikation von digitalen Signaturen nach dem SigG erfolgt nach bestimmten Gültigkeitsmodellen.

Tab. 3.2 *Abschnitte der SigI*

Die vom BSI (Bundesamt für Sicherheit in der Informationstechnik) herausgegebene „Schnittstellenspezifikation zur Entwicklung interoperabler Verfahren und Komponenten nach SigG/SigV“ (SigI) ist die derzeit umfassendste technische Beschreibung zur Realisierung einer SigG-konformen PKI. Obwohl sie sich auf das Signaturgesetz von 1997 [SIGI-XXX] bezieht, gibt es derzeit keine andere Interoperabilitätsspezifikation, die endgültig verabschiedet ist. Die SigI gliedert sich in die in Tabelle 3.2 dargestellten Abschnitte.

b) ISIS

Die ebenfalls zum Signaturgesetz von 1997 erschienene ISIS (Industrial Signature Interoperability Specification) wurde parallel zur SigI von den Betreibern der deutschen Trustcenter erstellt [ISIS99]. Deshalb werden darin besonders die für den Betrieb einer Zertifizierungsstelle notwendigen Prozesse und Datenformate in Übereinstimmung mit der SigI normiert.

In vielen Fällen wird in der ISIS auch auf internationale Standards, wie X.509 [X50999], zurückgegriffen. Neben der Verwendung von X.509-Zertifikaten, kommt auch ein hoher Stellenwert der einheitlichen Veröffentlichung von Sperrinformationen in Form von X509-CRLs (Certificate Revocation Lists) zu. Der Grund dafür ist, dass jeder PKI-Teilnehmer einer Zertifizierungsstelle in der Lage sein muss die Sperrlisten der anderen akkreditierten CAs auszuwerten. Insbesondere für die Verifikation spielt dieser Aspekt der Interoperabilität eine wichtige Rolle. Wie in der SigI werden auch hier die Anforderungen aus dem „alten“ SigG im Hinblick auf die Realisierung eines Trustcenters umgesetzt.

c) ISIS-MTT

Ähnlich wie die SigV vom November 2001 hinkt auch die Spezifizierung der technischen Umsetzung dem neuen SigG hinterher. Auf Basis der beiden vorher erwähnten Spezifikationen wird derzeit von den Zertifizierungsdienstanbietern und dem TeleTrusT-Verein die ISIS-MTT (Industrial Signature Interoperability Specification-MailTrusT) zum aktuellen SigG erarbeitet.

Die ISIS-MTT ist im derzeitigen Stand (Version 1.0, [ISIS01]) der SigI in vielen Bereichen sehr ähnlich. An einigen Stellen bezieht sie jedoch die derzeitigen PKI-Entwicklungen und neuen Versionen der internationalen Standards mit ein und schafft damit eine technisch aktuellere Basis. Bei den Krypto-Algorithmen in Teil 6 werden neue Verfahren wie elliptische Kurven berücksichtigt. In Teil 4 „Operational Protocols“ wird das neue TSP-Protokoll (Time Stamp Protocol) für den SigG-konformen Einsatz zugelassen. Die Arbeit an der ISIS-MTT gliedert sich in die folgenden Abschnitte:

Teil	Inhalt
Part 1: Certificate and CRL Profiles	Hier werden wie in der SigI die Formate für Zertifikate und Sperrlisten nach X.509 definiert.
Part 2: PKI Management	Dieser Teil befasst sich auf Basis von PKIX-Standards mit Protokollen zur Administration einer PKI.
Part 3: Message Format	Darin wird das S/MIME-Format zur Kommunikation zwischen PKI-Komponenten spezifiziert.
Part 4: Operational Protocols	Für die Verifikation von digitalen Signaturen sind die hier behandelten Protokolle OCSP, LDAP und TSP von besonderer Bedeutung.
Part 5: Certificate Path Validation	Derzeit liegt noch kein Dokument vor.
Part 6: Cryptographic Algorithms	Ähnlich dem Abschnitt A2 der SigI werden hier die für eine SigG-konforme PKI zugelassenen Krypto-Algorithmen unter Berücksichtigung der aktuelleren ECC-Verfahren festgelegt.
Part 7: Cryptographic Token Interface	Dieser Abschnitt entspricht weitgehenst dem PKCS#11-Standard.

Tab. 3.3 *Bestandteile der ISIS-MTT*

3.3 Signaturerzeugung und Verifikation

In Vorbereitung auf eine Modellierung der Verifikation werden im Folgenden die elementaren Prozesse bei der Überprüfung und Erzeugung von digitalen Signaturen betrachtet.

3.3.1 Erzeugung von digitalen Signaturen

In Abbildung 3.2 wird die einfache Signaturerzeugung zu einem Dokument, nach Daten und Programm aufgeteilt veranschaulicht. Zusatzdienste werden dabei nicht berücksichtigt. Wie später in Kapitel 5 und 6 gezeigt wird, ist für die vollständige Überprüfbarkeit der Signatur mehr als nur die Signaturberechnung über den Dokumenteninhalt notwendig, da auch der Signaturzeitpunkt verläßlich mit der Signatur und dem Dokumenteninhalt verbunden sein muss. Zur Generierung einer Signatur sind die folgenden Schritte notwendig:

1. Über den Inhalt des Dokumentes wird ein Hash-Wert mit einer fixen Länge berechnet.

2. Der erzeugte Hash-Wert wird mit dem privaten Schlüssel des Signierers verschlüsselt.

3. Abschließend wird die Signatur mit dem Dokument in eine dem Verifizierer bekannte Struktur gebracht. Dafür eignet sich zum Beispiel das im PKCS#7-Standard definierte Format. Darin sind neben der Signatur und dem Dokument auch das Zertifikat des Signieres und Informationen über den verwendeten Hash- und Signaturalgorithmus enthalten.

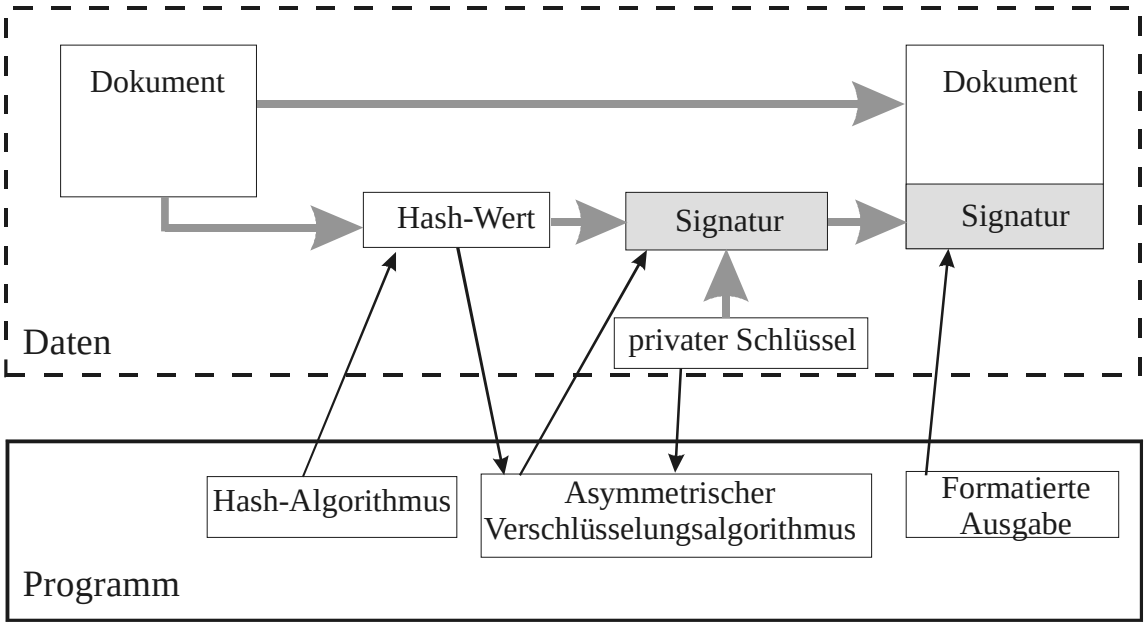

Abbildung 3.2 *Signaturerzeugung*

Um eine Qualität, ähnlich der einer manuellen Unterschrift sicherzustellen, hat eine Software bei der Realisierung der in Abbildung 3.2 dargestellten Prozesse mehrere Punkte zu beachten. Zum einem muss sie den Inhalt des zu signierenden Dokumentes nachweislich vertrauenswürdig darstellen. Zum anderen darf der zur Signatur benutzte private Schlüssel die geschützte Umgebung des PSEs nicht verlassen. Zusätzlich hat sich die signierende Person vor einer Signaturerzeugung an ihrem PSE entsprechend zu authentisieren, um den Zugriff auf den privaten Schlüssel freizuschalten. Diese Punkte sind unter anderem auch Forderungen des SigG/ SigV, die in den technischen Umsetzungen der Interoperabilitäts-Spezifikationen berücksichtigt sind.

3.3.2 Strukturierung der Prozesse zur Verifikation

Für den Nachweis der kryptographischen Korrektheit einer digitalen Signatur sind die folgenden Einzelschritte notwendig:

1. Aufteilung der übermittelten Daten in die eigentliche Signatur, den signierten Text und eventuell weitere Informationen.

2. Mit dem öffentlichen Schlüssel des Signieres kann jetzt aus der Signatur der Hash-Wert restauriert werden. Der öffentliche Schlüssel kann zum Beispiel aus dem Signaturzertifikat stammen, das beim PKCS#7 bzw. S/MIME-Standard mit der Signatur und dem Dokument kombiniert ist.

3. Der Hash-Wert wird neu aus dem Inhalt des signierten Dokumentes berechnet.

4. Sind die beiden unabhängig voneinander ermittelten Hash-Werte identisch, ist die Signatur mathematisch korrekt.

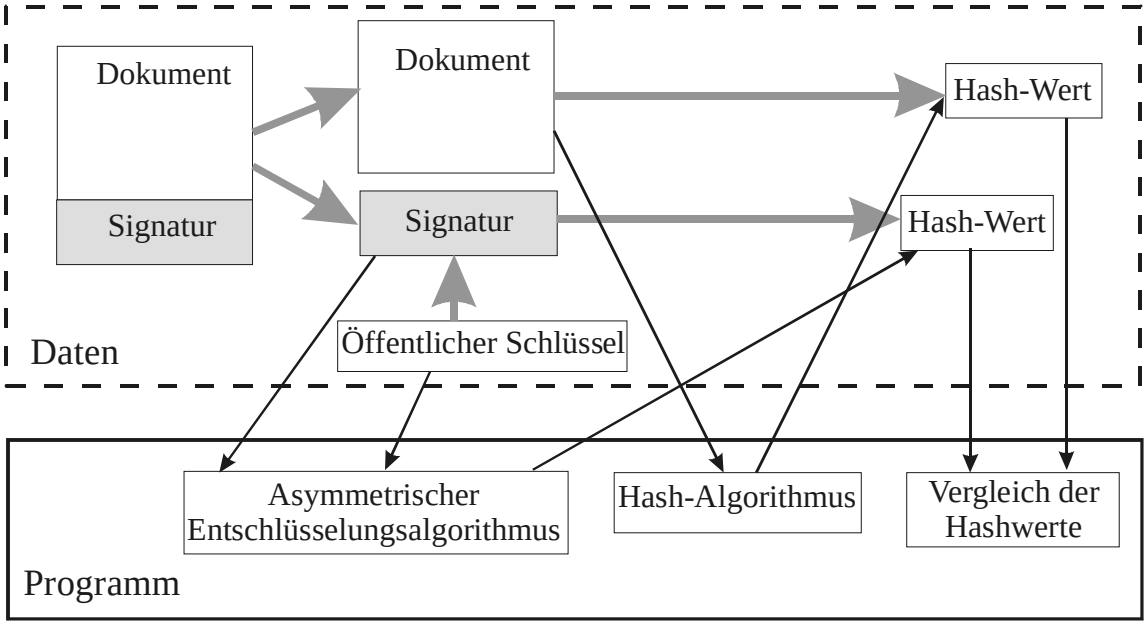

Abbildung 3.3 *Überprüfung der kryptographischen Korrektheit einer digitalen Signatur*

Mit dem in Abbildung 3.3 dargestellten Prozess müssen auch alle Zertifikate im Pfad bis zur Wurzel-CA überprüft werden. Jedes Zertifikat kann als signierter Datensatz betrachtet werden, dessen Signatur mit dem öffentlichen Schlüssel der übergeordneten CA zu verifizieren ist (siehe Abbildung 3.4). Der dafür benötigte öffentliche Schlüssel findet sich immer im Zertifikat der entsprechenden Aussteller-CA. Erst das Zertifikat der Wurzel-CA ist selbstsigniert. Zur Signatur des Wurzel-Zertifikates wird der private Schlüssel der Wurzel-CA, der zu dem im Zertifikat vorhandenen öffentlichen Schlüssel korrespondiert, verwendet.

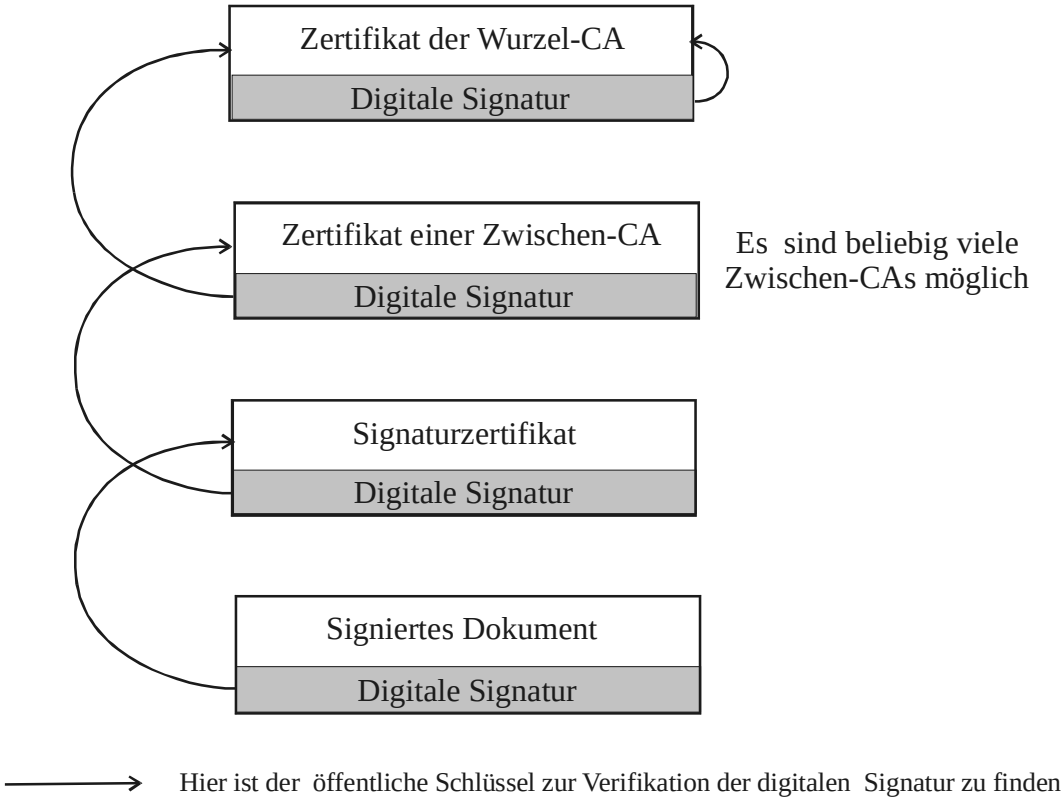

Abbildung 3.4 *Überprüfung aller Signaturen im Pfad bis zur Wurzel-CA*

Neben der mathematischen Überprüfung unter der Verwendung von Hash-Funktionen und Krypto-Algorithmen müssen auch zeitliche Aspekte bei der Verifikation der digitalen Signatur beachtet werden. Grundlage dafür sind die Gültigkeitszeiträume aller Zertifikate im Pfad bis zur Wurzel-CA. Zusätzlich wird der Zeitpunkt der Signaturerzeugung oder die aktuelle Zeit bei der Signaturprüfung benötigt. Abhängig von dem zugrundeliegenden Gültigkeitsmodell muss der Prüf- oder Signaturzeitpunkt in einem bestimmten Zeitraster, das von den Gültigkeitszeiträumen der Zertifikate vorgegeben wird, liegen. Gültigkeitsmodelle werden zum Beispiel in der SigI Abschnitt A6 [SIGI-GÜM99] definiert (siehe Kapitel 6.5) oder in vielen Fällen implizit durch die Implementierung des Verifikationsprozesses vorgegeben.

Mit der zeitlichen Betrachtung ist die Verifikation noch nicht abgeschlossen. Abschließend muss noch sichergestellt werden, dass keines der Zertifikate im Pfad zur Wurzel-CA von der übergeordneten CA gesperrt ist. Dieser Prüfschritt kann durch die Verwendung eines OCSP-Servers wesentlich vereinfacht werden. OCSP [OCSP99] ist eine Alternative zur Auswertung der Sperrinformationen vom Verzeichnisdienst. Statt nach dem verwendeten Zertifikat in einer unter Umständen sehr grossen Sperrliste zu suchen, kann über eine OCSP-Protokollanfrage der Status einfacher ermittelt werden.

3.4 Abschließende Kommentare

Im Hinblick auf die Modellierung der Verifikation mit der Beschreibungssprache SigBAN wurde in diesem Kapitel die Erzeugung und Prüfung von digitalen Signaturen besprochen. Wie gezeigt wurde, sind dabei neben kryptographischen Algorithmen wesentlich mehr Prozesse zu berücksichtigen. Insbesonders kommt den erläuterten Standards eine wichtige Aufgabe zu, um eine technischen Umsetzung der Public-Key-Kryptographie in Anwendungen zu ermöglichen. Neben technischen Aspekten spielen auch die juristischen Bedingungen für den Einsatz von rechtskräftigen digitalen Signaturen eine tragende Rolle. Aus diesem Grund wird hier das SigG mit der zugehörigen SigV und den Interoperabilitäts-Spezifikationen erläutert. In vielen Fällen greifen die technischen Spezifikationen zum SigG auch auf internationale Standards zurück. Daraus resultiert eine enge Verbindung zwischen Standardisierung und Gesetzgebung.

4. Grundlagen für eine logische Sprache zur Verifikation

4.1 Überblick

In den vorherigen Kapiteln wurden bereits die Begriffe „Vertrauen“ und „Sicherheit“ im Zusammenhang mit PKIs und digitalen Signaturen verwendet. Eine Logik erfordert jedoch eine präzise Festlegung der Bedeutung dieser beiden Worte. Deshalb wird hier versucht eine Beschreibung unter der Berücksichtigung bereits vorhandener Definitionen aus den einzelnen Standards zu erstellen.

Im zweiten Abschnitt werden die Anforderungen an eine logische Sprache zur Signaturprüfung aufgestellt. Dabei wird zwischen den speziellen Bedürfnissen zur Beschreibung der Verifikation und den allgemeinen Anforderungen, die jede Logik erfüllen muss, unterschieden.

Im Anschluss daran werden die während dieser Arbeit betrachteten Modellierungsmöglichkeiten kurz vorgestellt und auf ihre Eignung eingegangen. Als erste Beschreibungsform wurde die strukturierte Analyse (SA) verwendet, um eine übersichtliche Darstellung aller an der Verifikation beteiligten Prozesse zu erreichen. Die SA ist keine formale Sprache, sondern nutzt eine spezielle Graphik, um Prozesse in der Datenverarbeitung übersichtlich darzustellen. Dagegen sind die Prädikatenlogik (PL) und die BAN-Logik formale Sprachen, die alle zuvor ermittelten Anforderungen an eine Logik zur Modellierung der Verifikation erfüllen.

4.2 Begriffsdefinitionen

Im Folgenden werden die für eine logische Sprache zur Verifikation von digitalen Signaturen notwendigen Begriffe definiert. Soweit es möglich ist, wird dabei auf bereits vorhandene Beschreibungen aus den in Kapitel 3.2 vorgestellten PKI-Standards zurückgegriffen. Bei der Erstellung der Definition wird insbesondere die spätere Verwendung in der SigBAN-Logik berücksichtigt.

4.2.1 Vertrauen

Hier wird die Beschreibung des Begriffs „Vertrauen“ aus der ITUT-T Recommondation X.509 [ITU97] als Grundlage für eine eigene Definition genutzt. Laut dieser Spezifikation vertraut eine Instanz einer anderen Instanz, wenn diese Instanz sich so verhält, wie es die erste Instanz erwartet:

„Generally, an entity can be said to „trust“ a second entity when it (the first entity) makes the assumption that the second entity will behave exactly as the first entity expects. This trust may apply only for some specific function. ...“ (ITUT-T Recommondation X.509, Abschnitt 3.3.23) [ITU97]

Im Hinblick auf eine Quantifizierung und die Einschränkung auf nur eine Aussage wird die Definition von „Vertrauen“ neu formuliert. Diese Ergänzung findet später bei der Modellierung quantitativer Aspekte mit der SigBAN-Logik in Kapitel 5.5.4 Anwendung.

> **Vertrauen**: Eine Instanz vertraut der Aussage einer anderen Instanz mit einem gewissen Wahrscheinlichkeitswert X, wenn sich die andere Instanz mit der Wahrscheinlichkeit X nach der von ihr gemachten Aussage richtet. Die Aussage trifft somit mit der Wahrscheinlichkeit X zu (wahr) bzw. mit 1-X nicht zu (falsch).

Die Besonderheit an der hierarchischen Struktur in einer PKI ist, dass nicht jeder Teilnehmer eigene Regeln für das Vertrauen in die Aussagen anderer Instanzen aufstellt. Die Bedingungen für den Vertrauensfluss werden von den Zertifizierungsstellen festgelegt. Nur diese sind in der Lage, das Vertrauen in Form von Zertifikaten zu delegieren. Die Wurzel-CA fungiert dabei als Vertrauensquelle, während die Signatur eines PKI-Teilnehmers eine Vertrauenssenke darstellt.

4.2.2 Sicherheit

In der englischen Sprache wird der deutsche Begriff „Sicherheit“ nach Ausfallsicherheit (eng. „reliability“) und der Sicherheit vor Angriffen (eng. „security“) unterschieden. Die ISO (International Organization for Standardization) [ISO88] definiert die hier betrachtete Sicherheit vor Angriffen als die „Minimierung der Verwundbarkeit von Werten und Ressourcen“. Die dafür benötigten Sicherheitsdienste können von PKI-Applikationen zur Verfügung gestellt werden. In den einzelnen Anwendungen beruht dann die Sicherheit auf der Stärke der verwendeten Public-Key-Algorithmen und der Vertraulichkeit und Integrität der verwendeten Software. Insbesondere Letzteres ist mit sehr viel Entwicklungsaufwand verbunden, wie aus den verschiedenen Beschreibungen zur Evaluierung sicherheitsrelevanter Software deutlich wird ([DD85], [ITSEC91], [CC98]). Diese Dokumente fordern alle übereinstimmend, dass ein Programm nur dann als sicher betrachtet werden kann, wenn das korrekte Verhalten unter allen möglichen Eingangsbedingungen getestet wurde. Eine Überprüfung aller möglichen Eingangsparameter und internen Zustände von Programmen wird in den wenigsten Fällen möglich sein. Aus diesem Grund betrifft die Überprüfung der Korrektheit meistens den nachweislichen Ausschluß sämtlicher Manipulationsmöglichkeiten, die abhängig von bestimmten Bedrohungszenarien ermittelt wurden.

> **Sicherheit**: Ein bestimmtes Ausmaß an Vertraulichkeit und Integrität, das die Verwundbarkeit von Werten und Ressourcen zu einem gewissen Grad verhindert.

Sicherheit spielt eine wichtige Rolle bei der Betrachtung von Vertrauen im Zusammenhang mit Prozessen der Datenverarbeitung. Zum Beispiel kann eine anfangs richtige Aussage nur dann noch vertrauenswürdig sein, wenn sie über eine gesicherte Datenstrecke übermittelt wurde.

Für die idealisierende Betrachtungsweise wird in dieser Arbeit angenommen, dass alle Prozesse bei der Verifikation von digitalen Signaturen sicher implementiert sind. Eine Möglichkeit die unterschiedliche „Sicherheit“ der an der Verifikation beteiligten technischen Komponenten im Modell dennoch zu berücksichtigen, ist indirekt durch die Quantifizierung des Vertrauens gegeben.

4.3 Anforderungen an eine Beschreibungssprache zur Verifikation

Die hier vorgestellten Anforderungen an eine Beschreibungssprache zur Verifikation von digitalen Signaturen können in zwei Kategorien aufgeteilt werden. Zum einen werden Anforderungen, die den praktischen Einsatz der Logik im Zusammenhang mit digitalen Signaturen betreffen, aufgestellt. Zum anderen werden allgemeine Bedingungen an die logische Überprüfbarkeit, die unabhängig vom Verwendungszweck sind, formuliert.

4.3.1 Praktische Anforderungen

a) Einfache Handhabung

Für die Beschreibungssprache ist eine einfache Anwendbarkeit wichtig. Deshalb muss die Anzahl der Regeln überschaubar gehalten und das Alphabet auf eine minimale Menge notwendiger Symbole beschränkt werden. Eine Darstellung der Verifikation mit den Formeln der Sprache soll auch Vorteile im Hinblick auf die Übersichtlichkeit gegenüber einer allgemeinsprachlichen Form bieten und gegebenenfalls in Software implementierbar sein.

b) Unabhängigkeit von bestimmten Standards und Krypto-Algorithmen

Die Beschreibung sollte nicht von bestimmten Standards und Algorithmen abhängen. Zum Beispiel darf eine Modellierung der Verifikation nicht nur mit Zertifikaten und Sperrlisten nach dem X.509-Standard [X50999] möglich sein. Um ein universelles Einsatzgebiet zu erschließen, ist eine Verknüpfung mit PKI-Standards, wie PKIX oder PKCS, zu vermeiden. Genauso wenig darf die Sprache an bestimmte Krypto-Algorithmen gebunden sein. Dennoch ist es unumgänglich, bei der Definition die grundlegenden Eigenschaften von Public-Key-Algorithmen oder Hash-Funktionen in der Syntax zu berücksichtigen.

c) Modellierung des Vertrauens

Bei der Verifikation spielen die Vertrauensbeziehungen in einer PKI eine wichtige Rolle. Die Regeln der Sprache müssen den Vertrauensfluss unter Berücksichtigung der in Kapitel 4.2.1 aufgestellten Definition für „Vertrauen“ widerspiegeln.

d) Berücksichtigung von zeitlichen und quantitativen Aspekten

Die meisten Aussagen besitzen nur einen begrenzten Gültigkeitszeitraum. Somit ist auch das Vertrauen in Aussagen zeitlich beschränkt. Die gewählte Beschreibungslogik muss deshalb die Zeit als Parameter berücksichtigen. Zusätzlich soll die Sprache auch in der Lage sein, Aussagen mit einem Wahrscheinlichkeitswert zu gewichten. Dies kann hilfreich sein, wenn eine Betrachtung der Sicherheit von den, an der Signaturprüfung beteiligten Komponenten gefordert wird. Eine quantitative Wertung eines positiven Verifikationsergebnisses soll dadurch möglich sein.

4.3.2 Anforderungen an eine logische Sprache

a) Eindeutige Syntax

Sämtliche Regeln und Operationen zur Beschreibung des Verifikationsprozesses müssen mit einer fest vorgegebenen Syntax formulierbar sein. Diese Syntax ist unveränderbar und beruht auf grundlegenden Axiomen und daraus abgeleiteten Theoremen. Die Zahl der Axiome sollte möglxilte

4.4 Möglichkeiten zur Analyse sicherheitsrelevanter Systeme und Prozesse

4.4.1 Strukturierte Analyse

Die strukturierte Analyse (SA) verwendet zur Beschreibung von Prozessen in der Datenverarbeitung eine spezielle Graphik und ist keine formale Sprache [ROB96]. In der SA werden die Prozesse, Daten- und Steuerflüsse in einem System graphisch veranschaulicht. Dabei kann durch die Aufteilung auf mehrere Ebenen ein sehr hoher Detailierungsgrad erreicht werden, ohne den Überblick zu verlieren. Diese Methode ist für eine Analyse eine sehr hilfreiche Vorgehensweise. Für ein logisches Modell zur Verifikation, das eindeutige Aussagen zur Gültigkeit einer Signatur ermöglicht, ist die SA allerdings nicht geeignet.

Im Hinblick auf diese Arbeit wurde mit der erweiterten strukturierten Analyse (eSA) eine spezielle Variante, die sicherheitskritische Prozesse berücksichtigt, entworfen. Dabei war ein Ziel dieser Vorarbeiten, eine Gliederung und übersichtliche Darstellung aller an der Verifikation beteiligten Vorgänge mit der eSA zu erstellen. Diese eSA-Modellierung der Verifikation kann in dem Artikel „Verifikation von digitalen Signaturen“ aus den DuD-Fachbeiträgen nachgelesen werden [SPI01].

4.4.2 Prädikatenlogik

Die Prädikatenlogik (PL) ist eine formale Sprache, deren Syntax durch eigene Bildungsregeln (Grammatik) beschrieben wird. Die Syntax der PL erster Stufe (PL1) teilt sich in die Signatur und die Bildungsregeln für Terme und Formeln auf (Kapitel 1.6 [REPO99]). Die Signatur der PL1 hat nichts mit den hier besprochenen digitalen Signaturen gemeinsam, sondern gibt das Alphabet an, aus dem die einfachsten Sätze, die sogenannten atomaren Formeln bestehen. In der PL1 werden Verknüpfungen über Variablen-, Funktions- und Prädikatensymbole gebildet. Die Anzahl der Argumente, die auf das betreffende Funktions- oder Prädikatensymbol folgt, heißt „Stelligkeit“. Mit Hilfe von frei definierbaren Symbolen werden in der PL1 Terme und Formeln zur Parametrisierung der Prädikate definiert. Mit einer genauen Semantik wird festgelegt, wie die unterschiedlichen Symbole zu interpretieren sind.

Die PL ist eine für die Modellierung der Signaturprüfung geeignete Möglichkeit. Sie hat allerdings den Nachteil, dass eine Beschreibungssprache zur Verifikation sich aufwendig gestaltet und dadurch die Übersichtlichkeit und einfache Anwendbarkeit verloren gehen. Alles Wissen, das im Zusammenhang mit der Verifikation von digitalen Signaturen eingesetzt wird, muss bei den grundlegenden Prädikaten berücksichtigt werden. Basierend auf den Eigenschaften von Public-Key-Algorithmen und den Merkmalen von PKIs werden die elementaren Prädikate formuliert. Diese atomaren Formeln können dann in der Beschreibung zur Verifikation von digitalen Signaturen verwendet werden.

Eine vollständige Darstellung der PL würde hier zu weit führen. Deshalb wird im Anschluss ein kurzes Anwendungsbeispiel aufgeführt, das bei der Eignungsuntersuchung zur Beschreibung der Verifikation erstellt wurde.

a) Beispiel zur Modellierung der NP-Vollständigkeit beim RSA-Algorithmus

Grundlage für die Modellierung dieses Beispiels mit der Prädikatenlogik ist das Wissen über den grossen Unterschied zwischen der Kenntnis, dass die Zahl n das Produkt zweier Primzahlen ist, und einer Möglichkeit zur Faktorisierung von n. Auf diesem Axiom beruht der in Kapitel 2.2.2a erwähnte RSA-Algorithmus. Als Semantik werden hier die natürlichen Zahlen n,p,q,x,y mit den Funktionen pt und fr, sowie boolsche und mathematische Operatoren genutzt. Der RSA-Algorithmus verwendet als öffentlichen Schlüssel das Produkt zweier grosser Primzahlen, hier p und q genannt. Dabei ist eine weitere Annahme die Eigenschaft, dass eine grosse Primzahl relativ leicht zu ermitteln ist.

$$K^{pt(p)}\overline{\exists}\ x,y\ (p=x*y\ \&\ x>1\ \&\ y>1)$$

Es kann innerhalb von pt(p) Zeiteinheiten festgestellt werden, dass keine weitere Zerlegung von p möglich ist. Dabei besitzt die Funktion zum Primzahlentest pt(p) die Komplexität P.

$$\exists\, p,q\ K^{fr(n)}(n=p*q\ \&\ p>1\ \&\ q>1)$$

Die hier dargestellte mögliche Faktorisierung von n in p und q benötigt fr(n) Zeiteinheiten. Die Funktion fr liegt dabei in der Klasse NP.

Die für eine Modellierung der Verifikation mit der Prädikatenlogik notwendige Formulierung des Axioms zur NP-Vollständigkeit kann folgendermassen lauten:

$$NP(p,q,n) \text{ mit } 2 * pt(q) << fr(n)$$

Die Sicherheit des RSA-Algorithmus beruht darauf, dass es wesentlich länger dauert eine Produkt von zwei grossen Primzahlen zu faktorisieren, als zwei grosse Primzahlen zu ermitteln. Somit benötigt fr(n) wesentlich mehr Zeiteinheiten als 2* pt(q).

Das Prädikat NP besitzt die Stelligkeit 3. Dabei stellt das Symbol „<<“ einen nicht näher quantifizierten Vergleich der Rechenzeiten dar. Die negierte Aussage $\neg$NP würde bedeuten, dass ein „schneller“ Algorithmus zur Faktorisierung gefunden wurde und ein grundlegender Algorithmus der Public-Key-Kryptographie hinfällig ist.

4.4.3 BAN-Logik

Die BAN-Logik wurde zur Analyse von Authentisierungsprotokollen entworfen. Sie besitzt ebenfalls die Merkmale einer formalen Sprache, berücksichtigt aber in ihrer Syntax bereits die besonderen Gegebenheiten der Public-Key-Kryptographie. Die semantischen Basisgrössen der BAN-Logik sind Protokollinstanzen, Schlüssel für Krypto-Algorithmen und Formeln, die

für ein Authentisierungsprotokoll typische Operationen abbilden. Die Basissyntax wird um Regeln, die eine Transformation der atomaren Formeln untereinander ermöglichen, erweitert. Diese Regeln werden in der Veröffentlichung von Burrows, Abadi und Needham als Axiome („Logical postulates“) betrachtet, die nicht weiter zu beweisen sind [BAN90].

Die BAN-Logik bietet bei einer Beschreibung von Prozessen in einer PKI dieselbe Mächtigkeit wie die wesentlich aufwendigere PL. Die Modellierung bestimmter Eigenschaften von Krypto-Algorithmen und digitalen Signaturen existieren bereits in der BAN-Logik. Aus diesem Grund wird im nächsten Kapitel die BAN-Logik als Grundlage für den Entwurf einer Beschreibungssprache zur Verifikation von digitalen Signaturen genutzt. Ergänzungen beim Ausbau zur Signatur-BAN (SigBAN) werden dabei nach den Vorschriften zur Definition neuer Terme als Axiome formuliert. Die SigBAN-Regeln werden dann als Theoreme aus diesen Axiomen und den bereits bekannten BAN-Regeln abgeleitet.

4.5 Abschließende Kommentare

Zuerst wurden in diesem Kapitel die für die Signaturprüfung elementaren Begriffe „Vertrauen“ und „Sicherheit“ definiert. Bei der Definition dieser Begriffe ist bereits ein Teil der Anforderungen an eine Sprache zur Beschreibung der Verifikation berücksichtigt, die im Anschluss daran aufgelistet sind.

Im letzten Teil dieses Kapitels sind die Ergebnisse der im Zuge dieser Arbeit betrachteten Möglichkeiten zur Modellierung der Überprüfung von digitalen Signaturen zusammengestellt. Dabei soll vermittelt werden, warum die BAN-Logik statt der Prädikatenlogik als Basis für eine Beschreibungssprache gewählt wurde.

5. SigBAN als Beschreibungssprache zur Verifikation

5.1 Überblick

In diesem Kapitel wird die Beschreibungssprache Signatur-BAN (SigBAN) für die Verifikation von digitalen Signaturen in einer PKI entworfen. Die SigBAN-Logik basiert auf der BAN-Logik und berücksichtigt die in Kapitel 4.3 erwähnten Anforderungen an eine Beschreibungssprache zur Signaturprüfung.

Nach einer Vorstellung von Syntax, Semantik und Regeln der BAN-Logik, werden ihre Möglichkeiten für den Einsatz bei der Signaturprüfung aufgezeigt. Daraus wird in den folgenden Kapiteln die SigBAN-Logik, die speziell auf die Signaturprüfung angepasst wurde, stufenweise entwickelt. Neben neuen Regeln besitzt die SigBAN-Logik zusätzlich die Möglichkeit, Aussagen aus der BAN-Logik in zeitliche Zusammenhänge zu setzen und mit Wahrscheinlichkeitswerten zu gewichten. Im letzten Abschnitt werden an einem einfachen Anwendungsbeispiel die Möglichkeiten der SigBAN demonstriert.

5.2 Vorstellung der BAN-Logik

Die BAN-Logik wurde von Burrows, Abadi und Needham zur Überprüfung der Korrektheit von Authentisierungsprotokollen entworfen [BAN89][BAN90]. Die BAN-Logik eignet sich sowohl für die Analyse von Protokollen, die auf asymmetrischen als auch auf symmetrischen Verfahren beruhen. Asymmetrische Kryptoalgorithmen stellen in einem Authentisierungsprotokoll wie IPSec [DOR99] sicher, dass die in die Kommunikation einbezogenen Instanzen (Benutzer, Rechensysteme, Prozesse) eindeutig definiert sind und ein Angreifer nicht die Identität einer autorisierten Instanz annehmen kann. Diese Anforderung ist für digitale Signaturen identisch [CAR99]. Jede digitale Unterschrift ist über ein Zertifikat eindeutig einem Teilnehmer in der PKI zugeordnet. Dennoch gibt es Unterschiede, ob ein Authentisierungsprotokoll oder die Signaturprüfung verifiziert werden. Ein wesentlicher Differenzpunkt ist, dass die Gültigkeit bei einer digitalen Signatur in den meisten Fällen wesentlich länger erforderlich ist, als bei einem Authentisierungsprotokoll der Nachweis der Identität. Bei einem Authentisierungsprotokoll ist der Nachweis der Gültigkeit von Aussagen auf die Zeit der Kommunikation beschränkt. Ein weiterer Unterschied ist, dass bei Protokollen nur die an der Kommunikation beteiligten Instanzen betrachtet werden und nicht alle in den Vertrauensfluss involvierten. In beiden Fällen wird ein Teilnehmer aufgrund der Kenntnis eines Geheimnisses identifiziert. Bei der Protokollbetrachtung berücksichtigt die BAN-Notation jedoch nur bedingt, wie dieses Geheimnis erlangt wurde und welche übergeordneten Instanzen involviert sind.

5.3 BAN-Notation

Die Syntax der BAN-Notation besteht aus Basisrelationen und darauf aufbauenden Regeln. Die Basisrelationen definieren elementare Verknüpfungen zwischen Teilnehmerinstanzen, Aussagen und Schlüsseln. Dabei werden alle Zuordnungen von Daten zu Teilnehmern, die bei einem Authentisierungsprotokoll eine Rolle spielen, abgebildet. Die Regeln stellen semantische Folgerungen zu Überführungen von Basisrelationen dar. Die BAN-Regeln wurden aufgrund praktischer Anforderungen entwickelt, weshalb auch ihre Widerspruchsfreiheit und Vollständigkeit nicht formal bewiesen sind [BAN89][BAN90]. Dennoch hat sich die Anwendung der BAN-Logik bewährt [JAC00] und es gibt keine Hinweise auf Widersprüche oder Unvollständigkeiten in den Regeln.

5.3.1 Syntax der BAN-Logik

Die Syntax der BAN-Notation umfasst die folgenden Elemente, die nach Teilnehmern, Aussagen, Schlüsseln und Funktionen unterschieden werden:

P, Q, R sind Bezeichner für die Teilnehmerinstanzen, die Aussagen wie **X, Y, Z** treffen.

K ist ein kryptographischer Schlüssel und **H** eine Hash-Funktion.

Relation	Bedeutung
$P \models X$	P glaubt an die Richtigkeit von Aussage X.
$P \triangleleft X$	P hat die Aussage X empfangen und sieht X.
$P \mid\sim X$	P hat die Aussage X gemacht.
$P \Rightarrow X$	P hat Kontrolle über den Wahrheitsgehalt der Aussage X.
#(X)	Die Aussage X ist unbenutzt (Nonce).
$P \overset{K}{\leftrightarrow} Q$	P und Q benutzen den gemeinsamen, geheimen Schlüssel K.
$\overset{K}{\rightarrow} P$	P besitzt ein asymmetrisches Schlüsselpaar, wovon K öffentlich ist.
$P \overset{X}{\Rightarrow} Q$	P und Q teilen das Geheimnis X.
$\{X\}_K$	X wurde mit dem Schlüssel K chiffriert.
H(X)	Hash-Wert der Aussage X.

Tab. 5.1 *BAN-Notation*

5.3.2 Regeln der BAN-Notation

Die BAN-Logik beinhaltet Regeln zur Umformung von Aussagen, die mit den Basisrelationen aufgestellt wurden. Ein Authentisierungsprotokoll wird durch die Anwendung dieser Regeln, nachdem es mit Basisrelationen beschrieben wurde, verifiziert. Manche Regeln beziehen sich deshalb auf bestimmte Fälle, die nur bei der Protokollbetrachtung vorkommen, und bei der Verifikation von digitalen Signaturen nicht relevant sind. Deswegen werden hier nur die Regeln vorgestellt, die auch in der SigBAN-Logik Anwendung finden.

a) Regel zur Bedeutung von Nachrichten für öffentliche Schlüssel

Hier wird absichtlich nicht die allgemeinere Regel zur Bedeutung von Nachrichten für gemeinsame geheime Schlüssel betrachtet, da bei der Betrachtung von digitalen Signaturen die Geheimhaltung von Daten durch Verschlüsselung nicht relevant ist.

$$\frac{P \models \xrightarrow{K} Q,\ P \triangleleft \{X\}_K^{-1}}{P \models Q \mid\sim X}$$

P glaubt, K ist der öffentliche Schlüssel von Q. Außerdem hat P eine mit dem zugehörigen privaten Schlüssel K^{-1} signierte Nachricht X erhalten. In diesem Fall glaubt P, dass die Nachricht von Q stammt.

b) Regel zur Autorität

$$\frac{P \models Q \Rightarrow X,\ P \models Q \models X}{P \models X}$$

P glaubt, Q ist in der Lage, die Aussage X zu treffen. Zusätzlich nimmt P an, dass Q an die Aussage X glaubt. Die Schlussfolgerung ist, dass P auch an die Aussage X glaubt.

c) Regel für Hash-Funktionen

$$\frac{P \models Q \mid\sim H(X),\ P \triangleleft X}{P \models Q \mid\sim X}$$

P glaubt, Q hat H(X) übermittelt und sieht X. Somit weiß P auch, dass Q die Aussage X übermittelt hat.

d) Regel für das Sehen einer Komponente

$$\frac{P \triangleleft (X, Y)}{P \triangleleft X} \qquad \frac{P \triangleleft (X, Y)}{P \triangleleft Y}$$

P sieht X in Kombination mit Y. Dann sieht P auch X bzw. Y. Ein Umkehrschluss, dass P die Kombination sieht, wenn er X und Y vorliegen hat, ist nicht gültig.

e) Regel zur Zusammensetzung und Aufteilung vertrauenswürdiger Aussagen

$$\frac{P \models X,\ P \models Y}{P \models (X, Y)} \qquad \frac{P \models (X,Y)}{P \models X}$$

P glaubt an X und Y. Somit kann P auch der Verknüpfung (X, Y) vertrauen.

P vertraut der Verknüpfung (X, Y). Dann vertraut P auch der Aussage X.

$$\frac{P \models Q \models (X,Y)}{P \models Q \models X}$$

P glaubt, dass Q an die Verknüpfung (X,Y) glaubt. Dann glaubt P, dass Q auch die Komponente X der Aussage für richtig hält.

f) Regel zum Aufteilen einer übermittelten Aussage

$$\frac{P \models Q \mid\sim (X,Y)}{P \models Q \mid\sim X}$$

P glaubt, dass Q die Verknüpfung (X,Y) übermittelt hat. Dann glaubt P, dass Q auch die Komponente X übermittelt hat.

5.4 Möglichkeiten und Grenzen der BAN-Notation

5.4.1 BAN-Syntax mit PKI Begriffen

Die Relationssymbole der BAN-Logik eignen sich sehr gut für die Beschreibung von Beziehungen in einer PKI. Hier wird der entsprechende BAN-Ausdruck in seiner Bedeutung mit den für eine Signaturprüfung notwendigen Begriffen belegt. In einer PKI werden die Basiselemente, die weiterhin Teilnehmer, Aussagen, Schlüssel und Funktionen beinhalten, mit den folgenden neuen Bezeichnern verwendet:

ROOT, CA, IM_CA, CA_X, VER, SIG sind Bezeichner für die Teilnehmerinstanzen in einer PKI.

ZERT, REV, DATA, X, T sind Aussagen der Teilnehmerinstanzen.

$\mathbf{K^{-1}}$ ist der private Schlüssel.

K ist der öffentliche Schlüssel.

H ist eine Hash-Funktion.

Relation	Beispiel zur Signaturprüfung
VER $\models$ ZERT	Der Verifier VER glaubt an die Richtigkeit des Zertifikates ZERT.
VER $\triangleleft$ DATA	Der Verifier VER hat die Nachricht DATA vorliegen.
SIG $\mid\sim$ DATA	Der Signierer SIG hat eindeutig die Aussage DATA gemacht.
ROOT $\Rightarrow$ ZERT	Die Zertifizierungsstelle ROOT hat Kontrolle über das Zertifikat ZERT.
$\xrightarrow{K}$ SIG	K ist der öffentliche Schlüssel von SIG; d.h. SIG besitzt auch den privaten Schlüssel K^{-1}.
H(DATA)	Hash-Wert über den Inhalt von DATA.
$\{H(DATA)\}_K{}^{-1}$	Der Hash-Wert von DATA wurde mit dem privaten Schlüssel K^{-1} signiert.

Tab. 5.2 *BAN-Relationen mit PKI-Begriffen belegt*

5.4.2 Erweiterung der BAN-Syntax

In den Kapiteln 5.5.3 und 5.5.4 wird die Verknüpfung von BAN-Aussagen mit einem Gültigkeitszeitraum und einem Wahrscheinlichkeitswert eingeführt. Ist der über das Symbol „*“ mit einer (Sig)BAN-Aussage verbundene Wahrscheinlichkeitswert kleiner als *0,5*, kann diese Aussage nicht als „wahr“ verwendet werden. Der Gültigkeitszeitraum zu einer (Sig)BAN-Aussage wird dabei über eine Zeitfunktion, die nur in einem bestimmten Bereich einen Wert ungleich 0 aufweist, dargestellt.

Um die hierarchische Struktur in einer PKI besser beschreiben zu können, wird ein weiteres Relationssymbol definiert. In einer PKI sind einzelne Instanzen in der Lage ähnliche Aussagen zu treffen. Alle Aussagen eines bestimmten Typs werden von den anderen PKI-Teilnehmern im Bezug auf die Vertrauenswürdigkeit als gleichwertig angesehen. Deshalb können sie mit dem neuen SigBAN-Symbol „$\forall X$“ zusammengefasst werden. Dies ist zum Beispiel bei den Aussagen der Instanz „Zertifizierungsstelle“ der Fall. Die Zertifizierungsstelle veröffentlicht beliebige Zertifikate unterschiedlichen Inhaltes. Dennoch wird von den PKI-Teilnehmern der Aussage „Zertifikat“ allgemein und unabhängig vom Inhalt vertraut. In diesem Zusammenhang wird diese Relation auch bei dem ersten Axiom der SigBAN-Logik in Kapitel 5.5.2a verwendet.

Symbol	Bedeutung
ZERT * $\propto$	Die Aussage ZERT trifft mit der Wahrscheinlichkeit $\propto$ zu.
$(\forall X)_{CA}$	Alle Aussagen der Instanz CA werden als eine gemeinsame Menge X betrachtet.

Tab. 5.3 *Neue Symbole der SigBAN-Logik*

5.4.3 Anwendung in einer PKI

Die BAN-Logik ist allgemein anerkannt für die Analyse und Spezifikation von Protokollen. Da viele Protokolle wie IPSec [DOR99], SSL oder TLS [FRI96] eine PKI als Grundlage nutzen, gibt es in der BAN-Logik bereits Möglichkeiten, die Vertrauensmodelle in einer PKI zu beschreiben [BAN89][BAN90]. Für die Verifikation von digitalen Signaturen bietet es sich an, die BAN-Notation um neue Definitionen und Regeln zu erweitern. Erst dadurch wird eine vollständige Beschreibung aller zur Signaturprüfung notwendigen Vorgänge möglich. Desweiteren wird eine präzisere Beschreibung von PKI-spezifischen Abläufen erreicht und der Umformungsaufwand durch die Anwendung neuer SigBAN-Regeln reduziert.

Eine Einschränkung der BAN-Logik, die nicht nur die Signaturprüfung betrifft, sind die fehlenden Möglichkeiten, zeitliche Aspekte zu modellieren. Bei den Authentisierungsprotokollen gibt es aus diesem Grund Weiterentwicklungen, wie die von Gaarder und Snekkenes [GAA91]. In der vorliegenden Arbeit wird eine Ergänzung zur BAN-Notation, die für die Signaturprüfung relevante Zeitintervalle berücksichtigt, vorgestellt.

Aussagen sind in vielen Fällen nicht eindeutig als wahr oder falsch kategorisiert, sondern ihnen wird mit einer gewissen Wahrscheinlichkeit vertraut. Dieses Vertrauen kann nach der Definition aus Kapitel 4.2.1 mit einer Wahrscheinlichkeit gewichtet sein. Im Gegensatz zur klassischen Logik wird deshalb in der SigBAN-Logik eine Wahrscheinlickeitsbetrachtung von Aussagen eingeführt, wie sie bereits in anderen Modellen zur Beschreibung einer PKI verwendet wird [MAU96]. Ziel dieser Wahrscheinlichkeitsbetrachtung ist es, Aussagen gleichen Inhaltes zu quantifizieren und dadurch einen Vergleich zu ermöglichen. Ein Beispiel hierfür, kann die Bewertung mehrerer erfolgreich verifizierter Pfade vom Signierer bis zur Wurzel-CA sein. Es ist nicht möglich, das Vertrauen in eine Aussage als absolut quantifizierbare Einheit zu definieren. Da die BAN-Logik aufgrund ihres ursprünglichen Verwendungszwecks nur wahr oder falsch kennt, bot es sich allerdings an, eine Erweiterung einzuführen, die eine weitere Differenzierung ermöglicht.

5.5 Ausbau der BAN zur SigBAN-Logik

Das Ziel ist, die BAN-Logik für die Signaturprüfung aufzubereiten. Die SigBAN-Logik entsteht aus der BAN-Logik durch einen erweiterten Regelsatz und Ergänzungen um zeitliche und quantitative Aspekte. Dazu werden in einem ersten Schritt die elementaren Aussagen „Zertifikat“ und der „Widerruf von Zertifikaten“ mit der BAN-Notation beschrieben.

5.5.1 BAN-Beschreibung von PKI-Aussagen

Hier werden Zertifikat und Sperrinformation mit der BAN-Notation beschrieben. Dabei ist eine Besonderheit für die BAN-Notation die Betrachtung von Zeitintervallen in diesen elementaren PKI-Datenstrukturen. Dies ist jedoch keine Erweiterung der BAN-Notation, wenn die Zeiten (T_{begin}, T_{end}) als einfache Aussagen, die als Inhalt einen Zeitpunkt haben, angesehen werden. Eine genauere Betrachtung der Zeiten findet später in Kapitel 5.5.3 statt.

a) Zertifikat

$$ZERT = ZERTINFO_{OWNER}, K_{OWNER}, T_{begin}, T_{end}, ZERTSIG_{CA}$$

$$ZERTINFO_{OWNER} = ZERT_{ID}, ISSUER_{CA}, MISC_{PKI}$$

$$ZERTSIG_{CA} = \{H(ZERTINFO_{OWNER}, K_{OWNER}, T_{begin}, T_{end})\}_{K^{-1}_{CA}}$$

Ein Zertifikat besteht immer aus Informationen zur Authentisierung des Inhabers $ZERTINFO_{OWNER}$, seinem öffentlichen Schlüssel K_{OWNER}, einem Gültigkeitszeitraum (T_{begin}, T_{end}) und der Signatur eines Herausgebers $ZERTSIG_{CA}$. In den Authentisierungs-Informationen sind meistens eine eindeutige Nummer $ZERT_{ID}$, ein Verweis auf die Zertifizierungsinstanz $ISSUER_{CA}$ und vom Verwendungszweck des Zertifikates in der PKI abhängige Daten $MISC_{PKI}$ festgehalten. Im wesentlichen werden in $MISC_{PKI}$ die Rolle und Rechte des Zertifikatsinhabers in der PKI beschrieben. Das Zertifikat erlangt erst durch eine Signatur $ZERTSIG_{CA}$ der vertrauenswürdigen Instanz CA Gültigkeit. Die Signatur besteht aus einem Hash-Wert über den Inhalt $H(ZERTINFO_{OWNER}, K_{OWNER}, T_{begin}, T_{end})$, der mit dem privaten Schlüssel der Zertifizierungsstelle K^{-1}_{CA} unterschrieben wurde. Aus der obigen Darstellung eines Zertifikates kann mit der BAN-Regel für das Sehen einer Komponente aus 5.3.2d die folgende Behauptung abgeleitet werden:

> Hat ein PKI-Teilnehmer ein Zertifikat vorliegen, dann kennt er auch die mit dem Zertifikat verbundene Rolle und die Zertifizierungsinstanz in der PKI, sowie die Gültigkeitsdauer und den öffentlichen Schlüssel des Zertifikatsinhabers.

b) Sperrinformation

$$REV = ISSUER_{CA}, ZERT_{ID1} \ldots ZERT_{IDn}, T_{begin}, T_{end}, REVSIG_{CA}$$

$$REVSIG_{CA} = \{H(ISSUER_{CA}, ZERT_{ID1} \ldots ZERT_{IDn}, T_{begin}, T_{end})\}_{K^{-1}_{CA}}$$

Der Widerruf von Zertifikaten kann in unterschiedlichen Formaten (Sperrlisten, Speerbäumen, Delta-Sperrlisten, Positivlisten [CAR99]) erfolgen oder über das OCSP-Protokoll (Online Certificate Status Protocol) [OCSP99] abgefragt werden. Die Aussage „Widerruf“ lässt sich wie die Aussage „Zertifikat“ in einen Datenbereich und die dazugehörende Signatur $REVSIG_{CA}$ aufteilen. Bei der Betrachtung des „Widerrufs“ als BAN-Aussage werden nur die für einen Widerruf essentiellen Informationen, die in allen Formaten gemeinsam ist, aufgeführt. Dazu gehören Informationen zum Herausgeber $ISSUER_{CA}$, eine eindeutige Kennung $ZERT_{IDx}$ für jedes gesperrte Zertifikat und die Gültigkeitsdauer (T_{begin}, T_{end}) der Sperrinformation. Wie ein Zertifikat erlangt auch die Sperrinformation erst durch die Unterschrift mit dem privaten Schlüssel der CA Gültigkeit in einer PKI.

5.5.2 Neue Regeln der SigBAN

Die SigBAN besitzt im Vergleich zur BAN-Notation einen anderen Regelsatz. Viele Regeln, wie zum Beispiel Umformungen, die eine symmetrische Verschlüsselung betreffen, sind in der SigBAN nicht relevant. Dafür werden neue Regeln benötigt, um den Vertrauensfluss zu beschreiben.

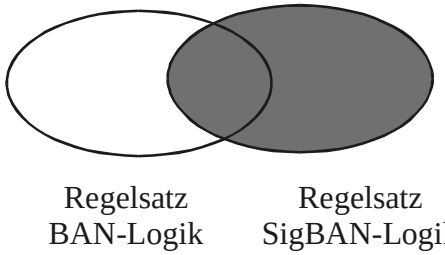

Abbildung 5.1 *Überschneidung der Regelsätze von BAN und SigBAN-Logik*

Für die Signaturprüfung müssen die Möglichkeiten der BAN-Logik, den Vertrauensfluss zu modellieren, präzisiert werden. Dabei wird das für die BAN-Logik neue Axiom, das die gesamten logischen Umformungen auf eine gemeinsame Quelle des Vertrauens zurückführt, definiert.

a) Allgemeingültiges Axiom für die Wurzel-CA

Die Wurzel-CA ist die einzige Instanz, die eine Gültigkeit ihrer Aussagen selbst bestimmen kann und muss. Für die Verifikation wird eine gemeinsame Vertrauensquelle, der sowohl der Signierer als auch der Verifizierer vertrauen, gefordert. Deshalb wird das folgende allgemeine Axiom auf die Wurzel-CA angewandt:

> **Erstes Axiom der SigBAN-Logik**: Die gesamte Verifikation lässt sich auf eine Vertrauensquelle ROOT, deren Aussagen $(\forall X)_{ROOT}$ alle anderen Instanzen X vertrauen, zurückführen:
>
> $$X \models ROOT \Rightarrow (\forall X)_{ROOT}$$

b) Erstes Theorem der SigBAN-Logik

Das erste Axiom hat insbesondere Auswirkungen auf die Zuordnung des asymmetrischen Schlüsselpaars zu der Wurzel-CA. Das erste Theorem der SigBAN-Logik lässt sich folgern, wenn die allgemeine Relation $(\forall X)_{ROOT}$ durch eine spezielle Aussage über den privaten Schlüssel $\xrightarrow{K_{ROOT}}$ ROOT ersetzt wird.

> **Erstes Theorem der SigBAN-Logik**: Die Aussage, dass der Schlüssel K_{ROOT} zur Instanz ROOT gehört, wird von der Instanz ROOT selbst bestätigt.
>
> $$ROOT \Rightarrow \xrightarrow{K_{ROOT}} ROOT$$

Mit dem ersten Theorem sind auch Konsequenzen für das Wurzelzertifikat verbunden. Da nur die Wurzel-CA ihre eigenen Aussagen bestätigen kann, ist das Wurzelzertifikat das einzige selbstsignierte Zertifikat in einer PKI und besitzt somit den folgenden Aufbau:

$$ZERT_{ROOT} = ZERTINFO_{ROOT}, T_{begin}, T_{end}, K_{ROOT}, ZERTSIG_{ROOT}$$

$$ZERTSIG_{ROOT} = \{H(ZERTINFO_{ROOT}, T_{begin}, T_{end}, K_{ROOT})\}_{K^{-1}_{ROOT}}$$

c) Wurzelregel (Zweites Theorem der SigBAN-Logik)

Aus dem ersten Axiom für die Wurzel-CA lässt sich für einen PKI-Teilnehmer, der das Wurzelzertifikat besitzt, die Wurzelregel der SigBAN-Logik formulieren. Die dafür notwendigen Schlussfolgerungen sind im Anhang A1 aufgeführt.

$$\frac{VER \models ROOT \Rightarrow (\forall X)_{ROOT},\ VER \triangleleft ZERT_{ROOT}}{VER \models \xrightarrow{K_{ROOT}} ROOT}$$

> Der PKI-Teilnehmer VER weiß, die Instanz ROOT kann die Gültigkeit aller eigenen Aussagen bestätigen. Außerdem besitzt VER das Zertifikat von ROOT. Somit weiß VER, dass die Instanz ROOT über das Schlüsselpaar $(K_{ROOT}, K^{-1}_{ROOT})$ verfügt.

d) Regel zur Zertifizierung (Drittes Theorem der SigBAN-Logik)

Ein PKI-Teilnehmer kann nur dann seinen privaten Schlüssel zur Signaturerzeugung verwenden, wenn die anderen PKI-Teilnehmer auch verifizieren können, dass er berechtigt ist, über diesen Schlüssel zu verfügen. Alle PKI-Teilnehmer vertrauen deshalb einer übergeordneten Zertifizierungsstelle, die jeden PKI-Teilnehmer legitimiert. Diese Legitimation erhält er in Form eines Zertifikates, das mit dem privaten Schlüssel der Zertifizierungsstelle unterschrieben wurde. Sein öffentlicher Schlüssel ist in diesem Zertifikat enthalten. Dieser Zusammenhang wird durch die folgende Regel beschrieben und im Anhang A2 bewiesen:

$$\frac{VER \models \xrightarrow{K_{CA}} CA,\ VER \triangleleft ZERT_{SIG}}{VER \models \xrightarrow{K_{SIG}} SIG}$$

VER ist bekannt, dass K_{CA} der öffentliche Schlüssel von der Zertifizierungsstelle CA ist und VER besitzt das Zertifikat von SIG. Somit lässt sich folgern, dass VER weiß, dass SIG über das Schlüsselpaar (K_{SIG}, K^{-1}_{SIG}) verfügt.

5.5.3 Modellierung zeitlicher Aspekte

Hier wird die BAN-Logik um spezielle Zeitfunktionen, die im Zusammenhang mit einer Signaturprüfung sinnvoll sind, erweitert. Die SigBAN betrachtet bei der Verifizierung von Signaturen die Gültigkeitszeiträume von Zertifikaten und Sperrinformationen. Aus diesem Grund werden bestimmte Aussagen (ZERT, REV) mit einer Zeitfunktion verknüpft. Die Eingangsparameter dieser Zeitfunktion sind:

t: frei wählbarer Prüfzeitpunkt

T_{begin}: Beginn der Gültigkeit einer Aussage

T_{end}: Ende der Gültigkeit einer Aussage

Dabei können T_{begin} und T_{end} über die Regel für das Sehen einer Komponente aus dem Zertifikat bzw. der Sperrinformation abgeleitet werden. Die Verknüpfung mit der Aussage wird durch ein Multiplikationssymbol dargestellt. Die Aussage ist nur in einem begrenzten Zeitraum, in dem die Funktion f(t) den Wert 1 annimmt, gültig. Ausserhalb dieses Intervalls ist die Funktion f(t) = 0. Unter der Beachtung zeitlicher Aspekte lässt sich jetzt ein Zertifikat bzw. die Sperrinformation als Kombination aus einer BAN Aussage und einer Zeitfunktion modellieren.

Zertifikat:

$ZERT_{OWNER} = (ZERTINFO_{OWNER}, T_{begin}, T_{end}, K_{OWNER}, ZERTSIG_{CA}) * f(t, T_{begin}, T_{end})$

Sperrinformation:

$REV = (ZERT_{ID1} ... ZERT_{IDn}, T_{begin}, T_{end}, REVSIG_{CA}) * f(t, T_{begin}, T_{end})$

Als $f(t, T_{begin}, T_{end})$ kommt im einfachsten Fall die folgende Funktion in Frage:

$f(t) = 1$ für $T_{begin} \leq t \leq T_{end}$ und $f(t) = 0$ für $t < T_{begin}$ oder $t > T_{end}$

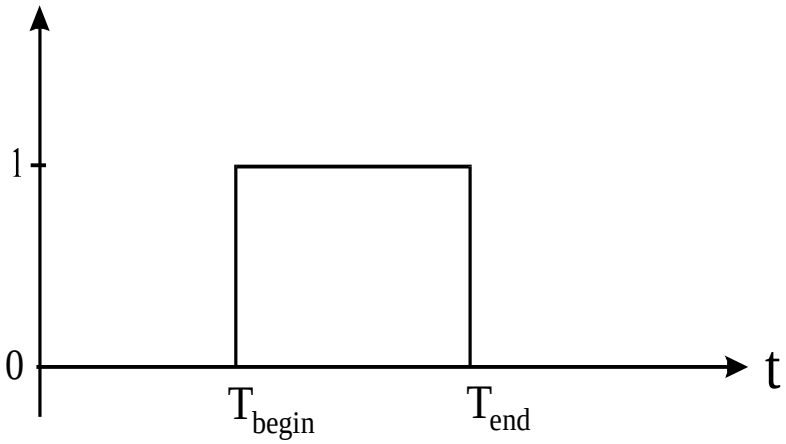

Abbildung 5.2 *Einfache Gültigkeit einer Aussage*

a) Normierung der Zeitachse

Bei der zeitlichen Betrachtung von Aussagen muss eine gemeinsame Zeitbasis festgelegt werden. In einer PKI bietet es sich an, den Nullzeitpunkt folgendermassen zu definieren:

> Als Ursprung der Zeitachse soll der Gültigkeitsbeginn der ersten Aussage der Vertrauensquelle angenommen werden. Die erste Aussage ist die mit dem am weitesten zurückliegenden Gültigkeitsbeginn.

Die Granularität der zeitlichen Auflösung ist abhängig von der Implementierung und wird nicht in der SigBAN spezifiziert. Eine höhere Auflösung, als die Verifikation (als atomarer Prozess gesehen) in Anspruch nimmt, macht jedoch keinen Sinn.

b) Zweites Axiom der SigBAN-Logik

Bei der Signaturprüfung ergibt sich eine Überlagerung der Zeitfunktionen f(t) aller an der Verifikation beteiligten Aussagen. Dies führt zu einem weiteren Axiom:

> **Zweites Axiom der SigBAN-Logik**: Die gesamte Aussage ist gültig, wenn alle an der Schlussfolgerung beteiligten Aussagen, zu dem für die Überprüfung relevanten Zeitpunkt ebenfalls Gültigkeit besitzen.

Die Zeitfunktion für die gesamte Anzahl N der Aussagen lautet somit wie folgt:

$$f_{gesamt}(t) = \prod_{X=1}^{N} f_{Aussage_X}(t)$$

Mit der Funktion $f_{gesamt}(t)$ muss in der SigBAN das Ergebnis der BAN-Umformungen noch gewichtet werden, um Gewissheit über die Gültigkeit unter Berücksichtigung der zeitlichen Aspekte zu bekommen. Dabei kann die Verknüpfungsfunktion $f_{gesamt}(t)$ immer losgelöst von den BAN-Umformungen betrachtet werden. Dies hat den Vorteil, dass die zeitliche Betrachtung in der SigBAN die Komplexität der BAN-Logik nicht unnötig erweitert. Hierzu ein einfaches Beispiel:

$$P \triangleleft \mathbf{(}\ (X,Y) * f_{AussageX}(t) * f_{AussageY}(t)\ \mathbf{)}$$

Im obigen Beispiel sieht P die Kombination der Aussagen X und Y, wenn $f_{AussageX}(t) * f_{AussageY}(t)$ nicht 0 wird. Dabei kann die Zeitfunktion $f_{gesamt}(t) = f_{AussageX}(t) * f_{AussageY}(t)$ unabhängig von der Relation der BAN-Notation berechnet werden.

Eine Überlagerung von mehreren Funktionen führt zu einem Gültigkeitsbereich, der mit der Gültigkeit aller Aussagen beginnt und mit dem Ende der Gültigkeit einer Aussage endet. In Abbildung 5.3 ist eine Überlagerung von mehreren Aussagen mit einfachen Gültigkeitsfunktionen dargestellt und dieser Gültigkeitsbereich markiert.

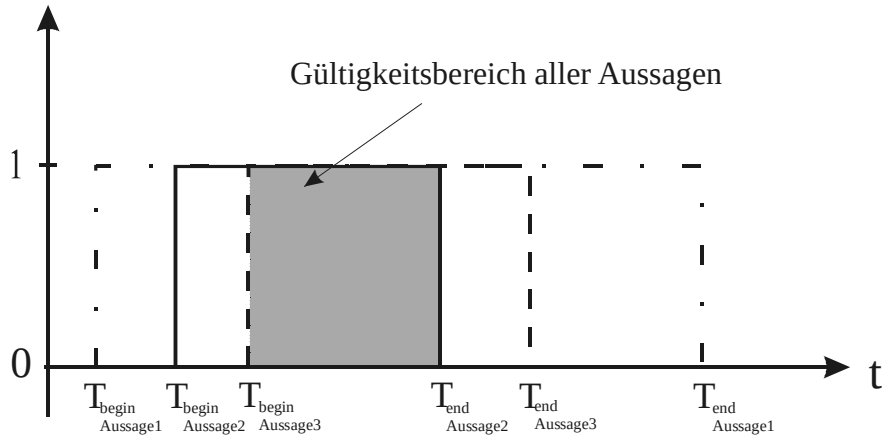

Abbildung 5.3 *Überlagerte Zeitfunktionen mehrerer Aussagen*

5.5.4 Modellierung quantitativer Aspekte

Häufig kann eine Aussage nicht eindeutig als wahr oder falsch eingestuft werden. Dies wird auch in Kapitel 4.2.1 bei der Definition von Vertrauen deutlich. Hier wird eine Wertung des Vertrauens in eine Aussage vorgenommen. Deshalb sind auch SigBAN-Aussagen mit einer Wahrscheinlichkeit verknüpft, die einen relativen Wert für das Vertrauen in den Wahrheitsgehalt angibt. Dabei wird eine SigBAN-Aussage als zu einem bestimmten Grad, der zwischen 0 und 1 liegt, wahrscheinlich eingestuft. Die Aussagen der SigBAN-Logik unterscheiden sich jetzt von denen der BAN-Logik durch die Ergänzung eines Gültigkeitszeitraums und eines Wahrscheinlichkeitswertes. Die Aussage in der SigBAN-Notation kann auf BAN-Notation zurückgeführt werden, wenn die Überprüfung in den Gültigkeitszeitraum fällt und die Wahrscheinlichkeit für die Richtigkeit 100% (d.h. $\propto$ =1) ist.

Aussage in der BAN-Logik: DATA

SigBAN-Aussage mit Gültigkeitszeitraum und Wahrscheinlichkeitswert: DATA * f(t) * $\propto$

Ein Beispiel für die Verwendung einer Quantifizierung, ist die Berücksichtigung der Sicherheit unterschiedlicher kryptographischer Verfahren bei der Signaturerzeugung. Der Grad des Vertrauens kann von dem zur Generierung des eindeutigen Fingerabdrucks verwendeten Hash-Algorithmus abhängig sein. Einer Signatur auf Basis der „SHA-1“ Hash-Funktion kann relativ zur Signatur über „MD5“ mit einer höheren Wahrscheinlichkeit vertraut werden [BB95]. Aufgrund der Erläuterung aus Kapitel 2.2.2 sollte allerdings deutlich sein, dass bei der Bewertung von kryptographischen Verfahren keine absoluten Messwerte möglich sind. Ein anderes Beispiel für die Verknüpfung von Aussagen mit Wahrscheinlichkeitswerten, kann die Verarbeitung der Information einer Aussage in bedingt vertrauenswürdigen Umgebungen sein [SPI01]. Hier lassen sich wiederum keine absoluten Werte definieren. Die Normierung der Wahrscheinlichkeitswerte wird in beiden Fällen von Annahmen zur untersuchten Infrastruktur vorgegeben.

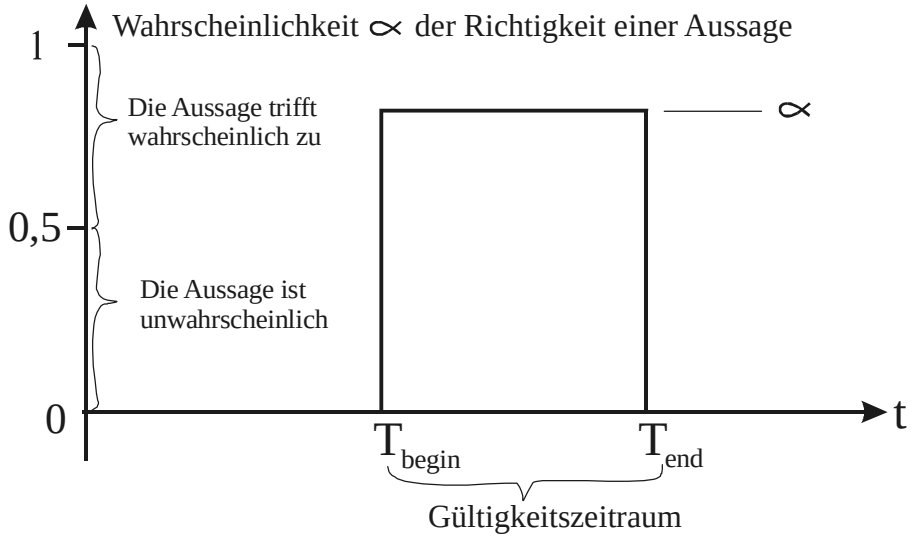

Abbildung 5.4 *Gewichtung des Gültigkeitszeitraums mit einem Wahrscheinlichkeitswert*

a) Drittes Axiom der SigBAN-Logik

Die Berücksichtigung der Wahrscheinlichkeitswerte bei einer Kombination von mehreren Aussagen führt zum dritten Axiom der SigBAN-Logik. Dabei wird angenommen, dass alle SigBAN-Aussagen voneinander statistisch unabhängig sind.

> **Drittes Axiom der SigBAN-Logik**: Der Wahrscheinlichkeitswert einer gefolgerten Aussage, ergibt sich aus der Multiplikation der Wahrscheinlichkeiten aller an der Herleitung beteiligten Aussagen.

Somit ergibt sich die Gesamtwahrscheinlichkeit für die Gültigkeit einer Signatur, durch die Multiplikation der Wahrscheinlichkeitswerte aller an der Prüfung beteiligten Aussagen:

$$\propto_{gesamt} = \prod_{X=1}^{N} \propto_{Aussage_X}$$

b) Besonderheit bei der Vertrauensquelle in einer PKI

Die Wahrscheinlichkeit für alle Aussagen der Wurzel-CA in einer PKI-Hierarchie wird in der SigBAN immer als 1 angenommen. Der Grund dafür ist, dass die Vertrauensquelle Wurzel-CA in einer PKI die Gültigkeit ihrer Aussagen selbst bestätigen kann und muss. Dies ist notwendig, weil das erste Axiom der SigBAN-Logik fordert, dass das Vertrauen immer auf einen gemeinsamen Punkt für alle Instanzen zurückgeführt wird. Außerdem ist es naheliegend, dass eine Instanz ihren eigenen Aussagen zu 100% vertraut.

Bei Aussagen der Wurzel-CA gilt immer: $\propto_{Root} = 1$

Diese Besonderheit führt dazu, dass Zertifikate oder Sperrinformationen der Wurzel-CA immer mit dem Wahrscheinlichkeitswert 1 gewichtet sind.

5.6 Anwendungsbeispiele zur SigBAN

Um den Einsatz der SigBAN an einem einfachen Beispiel zu demonstrieren, wird hier die Verifikation einer digitalen Signatur in einer einstufigen Zertifizierungshierarchie mit einer Zertifizierungsstelle, die auch gleichzeitig Wurzel-CA ist, durchgeführt. Die dabei verwendeten Umformungen sind in einer Übersicht im Anhang B1 zusammengestellt.

5.6.1 Beispielkonstellation

In der Anordnung aus Abbildung 5.5 ist nachzuweisen, dass VER$\models$SIG$\sim$DATA gilt, d.h. der Verifizierer VER glaubt, der Signierer SIG hat eindeutig die Aussage DATA gemacht.

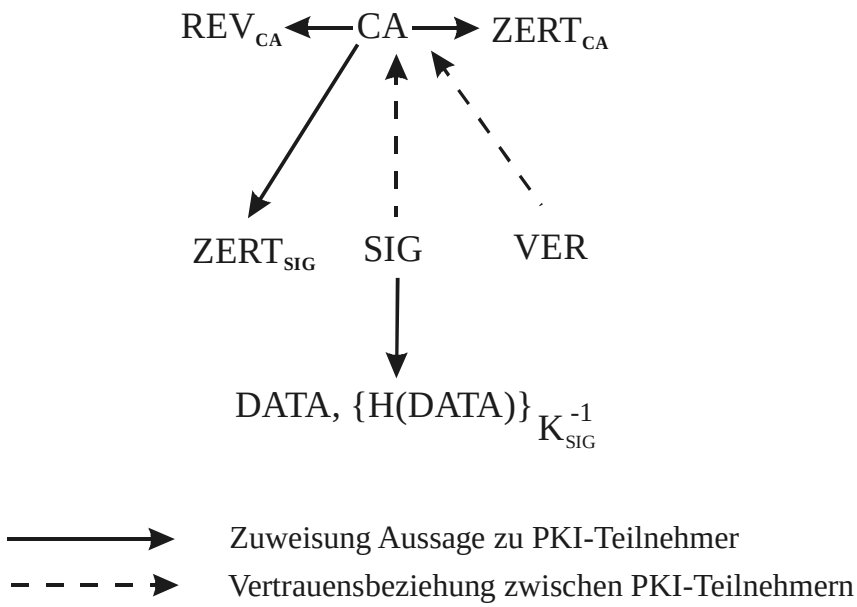

Abbildung 5.5 *Aussagen und Teilnehmer bei der Verifikation in einem einfachen Beispiel*

Wie bei einem Authentisierungsprotokoll müssen auch für die Verifikation digitaler Signaturen Annahmen aufgrund einer Idealisierung, der daran beteiligten Prozesse getroffen werden. Die hier vorgestellte Prüfung in der einfachen Anordnung nach Abbildung 5.5 ermöglicht es, die für jede Verifikation notwendigen Basisannahmen vorzustellen. Dazu gehört neben den signierten Daten, die Verfügbarkeit der zur Verifikation benötigten Zertifikate (hier $ZERT_{CA}$ und $ZERT_{SIG}$) und der Sperrinformationen (REV_{CA}) für den Verifizierer. Die Sperrinformation REV_{CA} beinhaltet alle widerrufenen Zertifikate und einen Gültigkeitszeitraum. In einer ersten Überprüfung muss festgestellt werden, ob eines der verwendeten Zertifikate (oder eine ID) Bestandteil der Sperrinformationen ist. Ist dies der Fall, muss die Gültigkeit der Sperrinformation mit SigBAN-Umformungen bestätigt werden und die Aussage des betreffenden Zertifikates kann nicht für die Gültigkeitsprüfung verwendet werden. Nach der Suche von verwendeten Zertifikaten kann der Inhalt der Sperrinformation verworfen werden. Nur die

Zeitfunktion und der Wahrscheinlichkeitswert werden noch für die endgültige Signaturprüfung benötigt. Eine Voraussetzung für die Überprüfung ist, dass sowohl der Verifizierer VER als auch der Signierer SIG Teil derselben Infrastruktur sind. Andernfalls ließe sich der Vertrauensfluss nicht auf die in der SigBAN geforderte gemeinsame Quelle CA zurückführen.

An der Verifikation sind die folgenden Instanzen und Aussagen beteiligt:

- CA: Die Wurzel-CA, die auch die Zertifizierungsstelle für SIG und VER ist.
- SIG: Ein PKI-Teilnehmer, der die Signatur zur Aussage DATA erzeugt hat.
- VER: Ein PKI-Teilnehmer, der die Signatur von SIG zur Aussage DATA verifiziert.

Die Sperrinformation und die Zertifikate können folgendermassen als SigBAN-Aussagen dargestellt werden:

$REV_{CA} = (ZERT_{ID1} \ldots ZERT_{IDn}, T_{begin}, T_{end}, REVSIG_{CA}) * f(t, T_{begin}, T_{end}) * \propto_{Root}$

$ZERT_{CA} = (ZERTINFO_{CA}, K_{CA}, T_{begin}, T_{end}, ZERTSIG_{CA}) * f(t, T_{begin}, T_{end}) * \propto_{Root}$

$ZERT_{SIG} = (ZERTINFO_{SIG}, K_{SIG}, T_{begin}, T_{end}, ZERTSIG_{CA}) * f(t, T_{begin}, T_{end}) * \propto_{SIG}$

5.6.2 Logischer Beweis

Die Verifikation mit der SigBAN-Logik startet immer mit dem Vertrauen des Verifizierers in die Wurzelinstanz. In dieser Startaussage findet sich das erste Axiom der SigBAN-Logik wieder. Zusätzlich muss der Verifizierer über alle für die Verifikation notwendigen Zertifikate und Sperrinformationen verfügen.

Startbedingungen

$VER \models CA \Rightarrow (\forall X)_{CA}$

Der Verifizierer VER vertraut der Zertifizierungsstelle CA als zentraler Vertrauensquelle, d.h. die Instanz CA kann aus Sicht von VER die Richtigkeit aller ihrer Aussagen selbst bestätigen[1].

Neben der signierten Aussage DATA mit der dazugehörigen Signatur $(\{H(DATA)\}_{K_{SIG}^{-1}})$ hat der Verifizierer VER auf die notwendigen Zertifikate und Sperrinformationen Zugriff:

$VER \triangleleft REV_{CA}$, $VER \triangleleft ZERT_{CA}$ und $VER \triangleleft ZERT_{SIG}$

1. Umformung

Mit der Wurzelregel kann jetzt die Kenntnis des öffentlichen Schlüssels der Zertifizierungsstelle CA für den Verifizierer VER nachgewiesen werden.

$$\frac{VER \models CA \Rightarrow (\forall X)_{CA},\ VER \triangleleft ZERT_{CA}}{VER \models \xrightarrow{K_{CA}} CA}$$

1. *Der Signierer SIG vertraut ebenfalls der Instanz CA. Diese Aussage ist jedoch für die Verifikation der Signatur nicht explizit notwendig bzw. bereits implizit durch das Zertifikat des Signierers gegeben.*

2. Umformung

Mit der Regel zur Zertifizierung kann im zweiten Schritt die Kenntnis des öffentlichen Schlüssels des Signierers SIG für den Verifizierer VER abgeleitet werden.

$$\frac{VER \models \xrightarrow{K_{CA}} CA,\ VER \triangleleft ZERT_{SIG}}{VER \models \xrightarrow{K_{SIG}} SIG}$$

3. Umformung

Mit der Regel zur Bedeutung von Nachrichten in Kombination mit der Regel für Hash-Funktionen aus der BAN-Logik kann jetzt die Richtigkeit der Aussage VER $\models$ SIG $\mid\sim$ DATA bestätigt werden und die logischen Umformungen sind abgeschlossen.

$$\frac{VER \models \xrightarrow{K_{SIG}} SIG,\ VER \triangleleft \{DATA\}_{K_{SIG}^{-1}}}{VER \models SIG \quad DATA}$$

5.6.3 Zeitliche Überprüfung mit der SigBAN

In der SigBAN kann die Zeitbetrachtung unabhängig von den logischen Umformungen durchgeführt werden. Obwohl die Gültigkeit der Signatur logisch bewiesen wurde, müssen im Anschluss daran zeitliche Zusammenhänge überprüft werden. In dem einfachen Fall einer einstufigen Zertifizierungshierarchie sind die Gültigkeitszeiträume der drei Aussagen Signaturzertifikat, Ausstellerzertifikat und Sperrinformation der CA zu betrachten. Die Aussage DATA ist mit keinem Zeitintervall verbunden, da die Gültigkeitsdauer von DATA für die Signaturprüfung nicht relevant ist. Stattdessen spielt der Zeitpunkt, zu dem die Aussage DATA signiert wurde, eine wichtige Rolle. Für eine vollständige Verifikation ist es notwendig, dass der Signaturzeitpunkt explizit und vertrauenswürdig bei der Signaturerzeugung festgehalten wurde. Diese Aufgabe übernimmt ein Zeitstempeldienst, der über einen eigenen Pfad zur Vertrauensquelle autorisiert ist. In dem vorliegenden Beispiel wird die Aussage DATA um den Zeitstempel $TIMESTAMP_{CA}$, der von der Wurzel-CA signiert wurde, ergänzt:

$$DATA = CONTENT, TIMESTAMP_{CA}$$

$$TIMESTAMP_{CA} = TIME, \{H(H(CONTENT), TIME)\}$$

Der Zeitstempel $TIMESTAMP_{CA}$ besteht hier aus einem unabhängig ermittelten Zeitpunkt TIME, der eineindeutig mit den signierten Daten CONTENT von der vertrauenswürdigen Instanz Wurzel-CA verbunden wurde. Diese eindeutige Verknüpfung wird über den Hash-Wert H(CONTENT), der mit dem festgestellten Zeitpunkt TIME signiert wird, erreicht. Dabei ist zu bemerken, dass die Zeitaussage TIME hier mit dem Signaturzeitpunkt gleichgesetzt wird, um dieses erste SigBAN-Anwendungsbeispiel nicht zu verkomplizieren. Tatsächlich sagt diese Verknüpfung nur aus, dass der Inhalt CONTENT zum Zeitpunkt TIME, der vor dem Signaturzeitpunkt liegt, bereits existierte. Die genaue und vertrauenswürdige Ermittlung des Signaturzeitpunktes mit einem Zeitstempel ist Bestandteil von Kapitel 6.3.

Der Signierer verwendet den signierten Zeitstempel wiederum bei seiner Signaturerzeugung. Die endgültige Signatur kann als SigBAN-Aussage folgendermassen formuliert werden:

$\{H(CONTENT, TIME, TIMESTAMP_{CA})\}_{K_{SIG}^{-1}}$ bzw.

$\{H(CONTENT, TIME, \{H(H(CONTENT), TIME)\}_{K_{CA}^{-1}})\}_{K_{SIG}^{-1}}$

Der mit dem Zeitstempel zusammenhängende Vertrauensfluss muss ebenfalls mit SigBAN-Umformungen überprüft werden und nach dem ersten Axiom auf eine gemeinsame Vertrauensquelle zurückgeführt werden. Dazu werden unterschiedliche SigBAN-Behauptungen aufgestellt, die durch Umformungen zu beweisen sind. Ein wichtiger Aspekt ist dabei der Nachweis des Vertrauens des Verifizierers VER in die Aussage TIME. Andere mit dem Zeitstempel im Zusammenhang stehende SigBAN-Behauptungen werden in Kapitel 6.3 betrachtet. Hier wird nur das Vertrauen in die Zeitaussage ($VER \models CA \mid\sim TIME$) bewiesen:

Die Kenntnis des öffentlichen Schlüssels der Instanz CA für den Verifizierer VER wurde bereits mit den vorherigen Umformungen nachgewiesen. Somit läßt sich mit der BAN-Regel zur Bedeutung von Nachrichten beweisen, dass der Verifizierer VER weiß, dass die Aussage $TIMESTAMP_{CA}$ verläßlich von der Instanz CA stammt:

$$\frac{VER \models \xrightarrow{K_{CA}} CA,\ VER \triangleleft TIMESTAMP_{CA}}{VER \models CA \mid\sim TIMESTAMP_{CA}}$$

Mit der Anwendung der Regel zur Aufteilung von Nachrichten, kann der Beweis abgeschlossen werden:

$$\frac{VER \models CA \mid\sim (TIME,\ H(CONTENT))}{VER \models CA \mid\sim TIME}$$

Nachdem die Vertrauenswürdigkeit der Aussage TIME sichergestellt ist, und hier der Zeitpunkt TIME mit dem Signaturzeitpunkt gleichgesetzt wird, kann überprüft werden, ob zum Zeitpunkt TIME alle an der Verifikation beteiligten Aussagen gültig waren. Nach Abbildung 5.6 muss dabei der Verifizierer VER feststellen, ob der Zeitpunkt TIME im Zeitintervall $[T_{begin\ Sperrinfo}, T_{end\ Sperrinfo}]$ liegt:

$$T_{begin\ Sperrinfo} < TIME < T_{end\ Sperrinfo}$$

Als Nullpunkt der Zeitachse wird der Beginn der Gültigkeit des Wurzelzertifikates $ZERT_{CA}$ gewählt, wie in Kapitel 5.5.3 bei der Normierung der Zeitachse beschrieben.

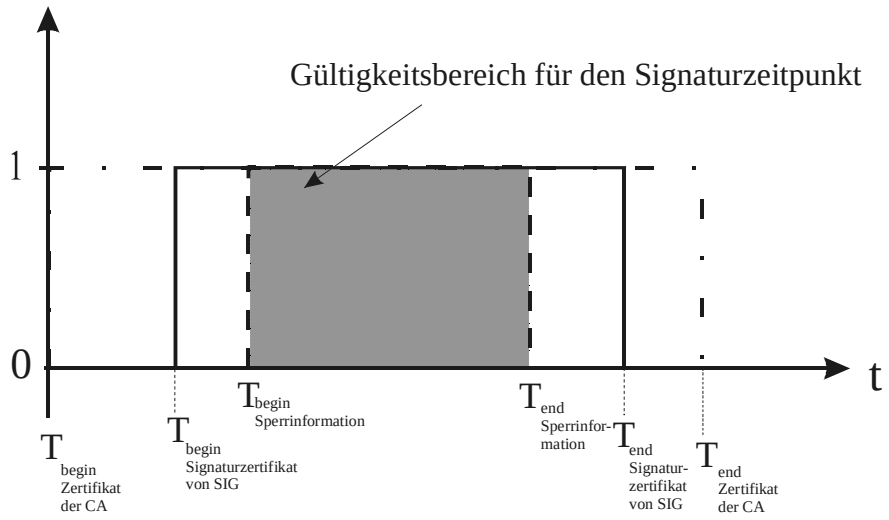

Abbildung 5.6 *Überlagerte Zeitfunktionen aller an der Verifikation in einer einstufigen PKI Hierarchie beteiligten Aussagen*

5.6.4 Wahrscheinlichkeitsbetrachtung

Analog der Zeit kann in der SigBAN auch die Wahrscheinlichkeit unabhängig von den logischen Umformungen betrachtet werden. Nachdem die Signatur logisch und zeitlich überprüft wurde, wird die Wahrscheinlichkeit für die Gültigkeit der Signatur ermittelt. Dies kann durch die Multiplikation der Wahrscheinlichkeitswerte aller an der Verifikation beteiligten Aussagen erreicht werden.

$$\propto_{gesamt} = \prod_{X=1}^{N} \propto_{AussageN} = \propto_{ZertCA} * \propto_{SperrCA} * \propto_{ZertSIG}$$

Da hier die Zertifizierungsinstanz auch gleichzeitig als Wurzel-CA fungiert, trifft die Regel für den Vertrauensendpunkt zu und es gilt:

$$\propto_{ZertCA} = \propto_{SperrCA} = 1$$

Somit ist bei der einfachen Zertifizierungshierarchie die gesamte Wahrscheinlichkeit gleich der Vertrauenswürdigkeit des Signaturzertifikates:

$$\propto_{gesamt} = \propto_{ZertSIG}$$

Der Wahrheitsgehalt der signierten Daten spielt für die Signaturprüfung keine Rolle. Deshalb braucht die Aussage DATA auch nicht mit einem Wahrsheinlichkeitswert verknüpft werden.

5.7 Abschließende Kommentare

In diesem Kapitel wurde die SigBAN-Logik als Beschreibungssprache zur Verifikation von digitalen Signaturen in einer PKI vorgestellt. Die SigBAN-Notation ist eine Weiterentwicklung der BAN-Logik, in der die speziellen Bedürfnisse einer vollständigen Signaturprüfung berücksichtigt wurden. Die wesentlichen Änderungen zur BAN-Logik bestehen aus einem neuen Regelsatz und der Möglichkeit, zeitliche und quantitative Zusammenhänge zu modellieren. Die SigBAN-Logik beruht auf den Anforderungen an eine Beschreibungssprache zur Verifikation, die in Kapitel 4.3 festgelegt wurden. Insbesondere wird durch die neuen Axiome der SigBAN-Logik die zentrale Struktur in einer PKI modelliert.

Ein wichtiger Aspekt bei der Entwicklung der SigBAN-Logik war eine einfach anzuwendende Beschreibungssprache zu definieren. Deshalb ist ein Merkmal, dass logische Umformungen, sowie zeitliche und quantitative Betrachtung unabhängig voneinander durchführbar sind. Um die Möglichkeiten der SigBAN-Logik zu demonstrieren, wurde in Kapitel 5.6 die vollständige Verifikation in einer einstufigen Zertifizierungshierarchie mit der SigBAN beschrieben. Bei der Validierung wird zuerst die Gültigkeit auf logischer Ebene durch Umformungen mit dem neu definierten Regelsatz bewiesen. Im nächsten Schritt findet eine zeitliche Überprüfung der an der Verifikation beteiligten Aussagen statt. Die anschließende quantitative Betrachtung ist nur dann sinnvoll, wenn die Gültigkeit der Signatur feststeht, und eine Gewichtung der Glaubwürdigkeit erforderlich ist.

6. SigBAN-Modell zur Verifikation nach dem Signaturgesetz

6.1 Überblick

In diesem Kapitel wurde eine PKI nach dem deutschen Signaturgesetz (SigG) [SIGG01] aufgrund der Aktualität und der detaillierten und öffentlichen Spezifikationen für eine weiterführende Demonstration der SigBAN-Logik gewählt. Wie hier gezeigt wird, ist bereits bei einer überschaubaren Komplexität der PKI eine formale Beschreibungssprache hilfreich, um eine umfassende Analyse aller an der Verifikation beteiligten Prozesse zu erreichen und Lücken zu erkennen. Die technischen Anforderungen für eine SigG-konforme PKI werden aus den drei Interoperabilitäts-Spezifikationen SigI (Abschnitt A1 bis A6, siehe [SIGI-*XXX*]), der ISIS [ISIS99] und der ISIS-MTT [ISIS01] entnommen. Die Beschreibungen der für die Signaturprüfung relevanten Datenstrukturen und Prozesse aus diesen Spezifikationen dienen als Ausgangspunkt für die logische SigBAN-Darstellung.

Zuerst werden die Merkmale einer SigG-konformen PKI für die SigBAN-Analyse aufbereitet. Dabei spielen für das SigBAN-Modell der Aufbau der PKI-Hierarchie, die für die Verifikation relevanten Datenformate, sowie die zur Signaturprüfung verwendeten Gültigkeitsmodelle eine wichtige Rolle. In einer Beispielkonstellation wird dann eine Sig-konforme Signaturprüfung mit der SigBAN-Logik durchgeführt. Dabei werden alle an eine digitale Signatur gestellten Anforderungen in Form von SigBAN-Aussagen formuliert und es wird versucht, diese Behauptungen durch anschließende Umformungen mit den SigBAN-Regeln zu beweisen. Dabei erwiesen sich die Anforderungen aus der SigI-Spezifikation zum Zeitstempeldienst als unvollständig, um den Zeitpunkt der Signaturerzeugung eindeutig festzuhalten. Deshalb wird mit der SigBAN ein eigener Vorschlag entworfen, der die Signaturerzeugung einem Zeitintervall eindeutig zuordnen läßt.

Bei der Untersuchung der Gültigkeitsmodelle im nächsten Abschnitt spielt weiterhin der Signaturzeitpunkt eine wichtige Rolle. Hier findet ein Vergleich und eine Bewertung dieser Gültigkeitsmodelle statt. Dabei kann nachgewiesen werden, dass Probleme, die beim SigBAN-Modell ausgeschlossen sind, mit den derzeitigen SigG-kompatiblen Gültigkeitsmodellen auftreten. Insbesondere bei den zeitlichen Aspekten wird deutlich, wie notwendig formale Methoden bei der Betrachtung der Verifikation sind.

Dieses Kapitel veranschaulicht, in welchen Bereichen eine logische Sprache einer allgemeinsprachlichen Form überlegen ist, und zeigt die Vorteile einer formalen Beschreibungssprache, wie der SigBAN-Logik, auf.

6.2 Merkmale einer SigG-konformen PKI

Die SigI ([SIGI-AIS99], [SIGI-GUM99], [SIGI-SIG99], [SIGI-VER99], [SIGI-ZE99], [SIGI-ZS99]) ist die derzeit umfassendste Spezifikation für Signaturdienstleistungen. Sie wurde vom BSI (Bundesamt für Sicherheit in der Informationstechnik) zum Signaturgesetz von 1997 [SIGG97] entworfen. Parallel zur SigI erstellten die Betreiber der einzelnen Trustcenter die weniger umfangreiche ISIS in der „T7“-Arbeitsgruppe. Zum aktuellen Signaturgesetz [SIGG00] wird gerade die ISIS-MTT gemeinsam von der „T7“ und dem TeleTrusT-Verein erarbeitet. Die derzeit noch nicht endgültige ISIS-MTT basiert auf der SigI und ISIS. Alle drei Interoperabilitäts-Spezifikationen wurden im Hinblick auf Kompatibilität zum PKIX-Standard (Public Key Infrastructure X.509 Standard, siehe [X50999]) entworfen.

Als Referenz wird hier die ISIS-MTT in der Version 1.0 und ergänzend die SigI verwendet, da die erste Version der ISIS-MTT erst vor kurzem veröffentlicht wurde und noch nicht vollständig ist. Die Zertifizierungshierarchie, sowie die Formate und Prozesse zur Verwaltung von Zertifikaten und Sperrlisten, wurden der ISIS-MTT entnommen. Die in den beiden Spezifikationen ISIS-MTT (Teil SigG-conforming systems and applications) und SigI (Abschnitt A6 [SIGI-GÜM99]) definierten Gültigkeitsmodelle sind identisch. In der ISIS-MTT wird allerdings das Schalenmodell bereits als nur bedingt geeignet erkannt. Für die Modellierung eines SigG-konformen Zeitstempeldienstes werden die aus der SigI (Abschnitt A4 [SIGI-ZS99]) definierten Anforderungen und Datenformate herangezogen, da es dazu noch keine Informationen in der ISIS-MTT gibt.

6.2.1 Anwenderinfrastruktur

Das Signaturgesetz basiert auf der in Abbildung 6.1 dargestellten zweistufigen Infrastruktur. Diese besteht aus der übergeordneten, staatlichen Wurzelinstanz „RegTP“ und den Zwischen-CAs der privatwirtschaftlich organisierten, akkreditierten Trustcenter, wie „D-Trust“ oder „TC Trustcenter“.

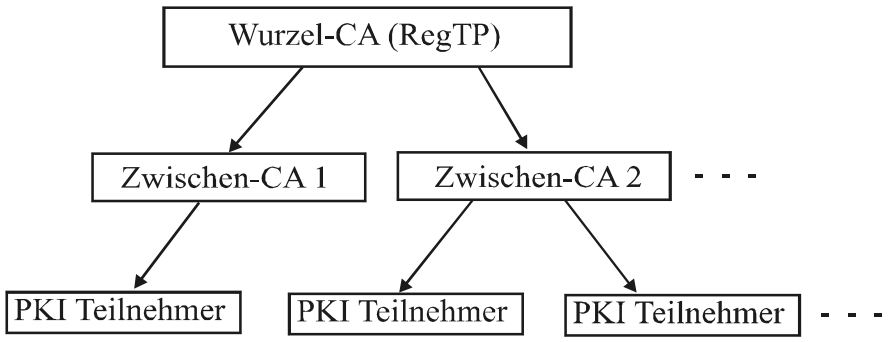

Abbildung 6.1 *Zertifizierungshierarchie nach SigG/SigV*

SigG-konforme Trustcenter können nur Teilnehmerzertifikate ausstellen. Eine Legitimierung von weiteren Zertifizierungsstellen über die Zwischen-CAs ist in einem SigG-konformen System nicht vorgesehen. Deshalb wird eine Crosszertifizierung oder Erweiterung der Zertifizierungshierarchie von den Interoperabilitäts-Spezifikationen zum SigG nicht berücksichtigt. Allerdings kann eine Zwischen-CA neben PKI-Teilnehmern auch Zusatzdienste, wie einen Zeitstempeldienst oder OCSP-Server (Online Certificate Status Protocoll), zertifizieren.

6.2.2 Datenformate

Für das SigBAN-Modell ist der Aufbau der Aussagen, die an der Verifikation beteiligt sind, entscheidend. Dabei spielt eine „bitgenaue“ Repräsentation, wie sie für die Implementierung und Interoperabilität notwendig ist, keine Rolle. Es ist nur wichtig zu modellieren, welche essentiellen Bestandteile in Zertifikaten und Sperrlisten vorhanden sind und wie diese Aussagen durch eine Signatur legitimiert wurden. Da alle drei Interoperabilitäts-Spezifikationen ([SIGI-AIS99], [ISIS99], [ISIS01]) kompatibel zum PKIX-Standard [X50999] sind, kann die Struktur nach dem X.509-Format als Grundlage für die Formulierung von SigBAN-Aussagen genutzt werden. Dieses Format ist in vereinfachter Form identisch mit dem bereits in Kapitel 5.5.1 verwendeten. In der Komponente $ZERTINFO_{OWNER}$ werden alle, für die hier durchgeführten SigBAN-Beweise nicht relevanten Informationen zusammengefasst. Dazu gehören zum Beispiel der Name der Zertifizierungsstelle oder die Rolle des Zertifikatsinhabers in der PKI.

Zertifikat:

$$ZERRT_{OWNER}=(ZERTINFO_{OWNER},K_{OWNER},T_{begin},T_{end},ZERTSIG_{CA}) * f(t,T_{begin},T_{end}) * \propto_{OWNER}$$

$$ZERTSIG_{CA}= \{H(ZERTINFO_{OWNER}, K_{OWNER}, T_{begin}, T_{end})\}_{K_{CA}^{-1}}$$

Sperrinformation:

$$REV_{CA} = (ZERT_{ID1} ... ZERT_{IDn}, T_{begin}, T_{end}, REVSIG_{CA}) * f(t, T_{begin}, T_{end}) * \propto_{CA}$$

$$REVSIG_{CA} = \{H(ZERT_{ID1} ... ZERT_{IDn}, T_{begin}, T_{end}) \}_{K_{CA}^{-1}}$$

6.2.3 Zeitstempeldienst

Ein autorisierter Zeitstempeldienst soll nach §4 Abs.1 Nr. 5 der Signaturverordnung (SigV, siehe [SIGV97]) zur Verfügung stehen, wenn „für die Verwendung signierter Daten ein Zeitpunkt von erheblicher Bedeutung sein kann“. Dies ist immer der Fall, wenn die digitale Signatur gleichwertig zu einer manuellen Unterschrift sein soll. Somit ist ein Zeitstempeldienst eine unverzichtbare Instanz in einer PKI. Im neuen Signaturgesetz (siehe §9 [SIGG01]) ist deshalb ein qualifizierter Zeitstempel ein Pflichtdienst einer akkreditierten Zertifizierungsstelle. Die Eigenschaften des Zeitstempeldienstes werden derzeit allerdings nur im Abschnitt A4 der SigI [SIGI-ZS99] im Bezug auf das Signaturgesetz von 1997 [SIGG97] spezifiziert.

Ein SigG-konformer Zeitstempel wird durch die zwei Instanzen Zeitstempeldienst und Zeitdatendienst in Abbildung 6.2 generiert. Der Zeitstempeldienst ist mit einem Pfad zur Vertrauensquelle (Wurzel-CA) autorisiert. Der Zeitdatendienst liefert dem Zeitstempeldienst zur Generierung des Zeitstempels eine verläßliche Zeitauskunft, die zusätzlich mit für den Zeitpunkt charakteristischen Daten versehen sein kann.

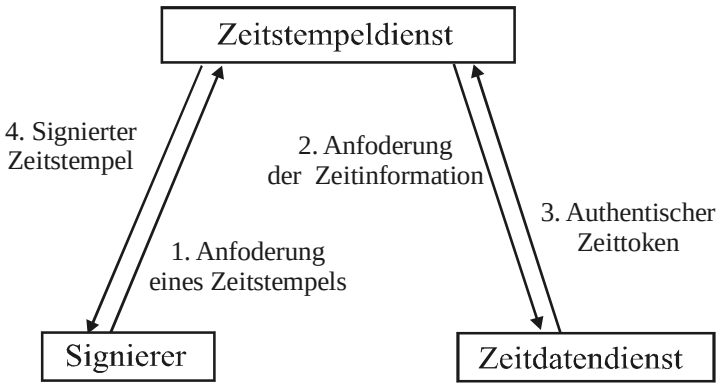

Abbildung 6.2 *Zeitstempeldienst nach SigI Abschnitt A4*

Für eine Verifikation des über den Zeitstempel festgehaltenen Signaturzeitpunktes sind die folgenden Sachverhalte zu überprüfen. Bei dieser Überprüfung sind nur die Aussagen der Instanz Zeitstempeldienst relevant, da für die Verifikation einer digitalen Signatur das Protokoll zwischen Zeitstempeldienst und Zeitdatendienst keine Rolle spielt.

- **Der Pfad von der Vertrauenssenke Zeitstempel zur Vertrauensquelle Wurzel-CA** ist analog einer digitalen Signatur zu überprüfen.
- **Der Signaturzeitpunkt** muss jederzeit nachvollziehbar und vertrauenswürdig mit den signierten Daten und der Signatur festgehalten werden.
- **Die Verbindung zwischen dem Zeitstempel, der Signatur und dem Inhalt** des signierten Dokumentes muss eineindeutig sein.

6.2.4 Gültigkeitsmodelle und Signaturprüfung

Momentan werden für die Zeitbetrachtung bei einer SigG-konformen Signaturprüfung das Schalenmodell (SM) und das Kettenmodell (KM) verwendet. Beide Modelle basieren auf einem Abgleich der Gültigkeitszeiträume von Zertifikaten im Pfad, vom signierenden PKI-Teilnehmer bis zur Wurzel-CA.

SM und KM führen in bestimmten Fällen zu widersprüchlichen Aussagen über die Gültigkeit einer digitalen Signatur. Dieser Widerspruch wurde in der SigI Abschnitt A6 [SIGI-GÜM99] nicht berücksichtigt und führt dazu, dass die Gültigkeit einer Signatur in einer PKI nach dem SigG vom gewählten Gültigkeitsmodell abhängt. Die ISIS-MTT [ISIS01] versucht dieses Problem zu beheben und läßt deshalb nur das KM für die Prüfung von SigG-konformen Langzeitsignaturen zu. Dagegen wird in der ISIS-MTT die Anwendung des SM auf Signaturen, deren Gültigkeit nur kurze Zeit relevant ist, beschränkt.

Das SM wurde somit bereits in der ISIS-MTT als unzureichend erkannt, um eine manuelle Unterschrift zu ersetzen. Dies wird nocheinmal bei einem Vergleich von SM, KM und SigBAN-Forderungen in Kapitel 6.5 deutlich. Zusätzlich werden in diesem Kapitel weitere Probleme, die SM und KM gemeinsam besitzen, herausgestellt und mit der in sich konsistenten SigBAN-Zeitbetrachtung verglichen.

6.3 SigBAN-Beschreibung zur einfachen Signaturprüfung

Die zweistufige Zertifizierungshierarchie nach dem Signaturgesetz (identisch in [SIGG97] und [SIGG01]) eignet sich aufgrund ihrer Übersichtlichkeit gut für ein weiterführendes Beispiel zur SigBAN-Anwendung. Eine Herausforderung für das SigBAN-Modell im Zusammenhang mit dem Signaturgesetz ist die Berücksichtigung aller zeitlichen Aspekte. Der dafür notwendige Zeitstempeldienst wird bereits hier mit aufgeführt, während die damit verbundenen Aussagen und Beweise in den nächsten Kapiteln 6.4 und 6.5 gesondert behandelt werden.

Das SigBAN-Modell zur Verifikation nach dem SigG wird stufenweise aufgestellt. Zuerst wird eine für den Anwendungsfall SiG exemplarische Konstellation von PKI-Instanzen ermittelt. Die gewählte Anordnung muss alle Komponenten, die an einer Verifikation nach dem SigG beteiligt sind, einbeziehen und möglichst allgemeingültig sein. In dieser ersten Stufe werden auch die zu beweisenden Aussagen in der BAN-Notation formuliert. Dabei ist es wichtig, alle an die digitale Signatur gestellten Anforderungen zu berücksichtigen. Andernfalls ist auch mit der SigBAN-Logik keine vollständige Verifikation zu erreichen.

Im nächsten Schritt sind die an der Verifikation beteiligten Aussagen in der Beispielanordnung zu ermitteln. Dies ist bei der zweistufigen Hierarchie nach dem Signaturgesetz leicht möglich. Allerdings kann in komplexen PKIs die Anzahl dieser Aussagen erheblich steigen. Dies ist insbesondere der Fall, wenn verschiedene Zertifizierungspfade von der Vertrauensquelle (Wurzel-CA) bis zur Vertrauenssenke (Signatur eines PKI-Teilnehmers) existieren.

Zuletzt wird versucht, mit den Regeln der SigBAN-Logik die Richtigkeit der im ersten Schritt aufgestellten Aussagen zu beweisen. Dabei ist der Ausgangspunkt für die Umformungen immer das erste Axiom der SigBAN-Logik (siehe Kapitel 5.5.2.).

6.3.1 Beispielanordnung für das SigG-konforme SigBAN-Modell

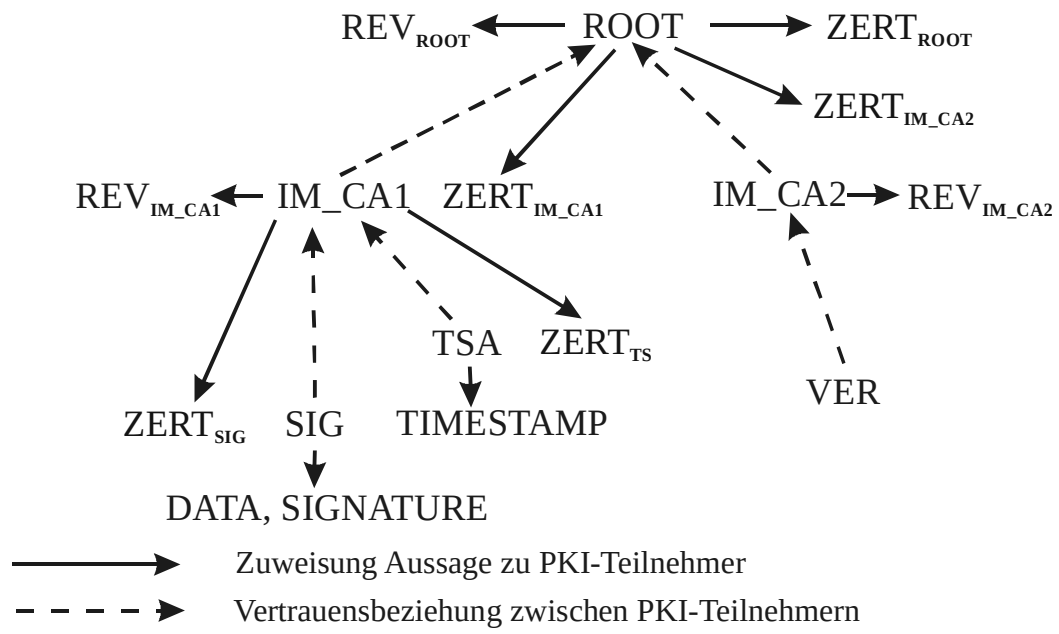

Abbildung 6.3 *Beispielanordnung für ein SigBAN-Modell zur Verifikation nach dem SigG*

Zu beweisende Aussage nach Abbildung 6.3:

Es ist zu beweisen, dass der Verifizierer VER glaubt, der Signierer SIG hat eindeutig dem Inhalt eines Dokumentes CONTENT durch seine elektronische Unterschrift SIGNATURE zugestimmt. Dies ist gleichbedeutend mit der SigBAN-Behauptung VER SIG CONTENT. Die tatsächlich signierten Daten, hier DATA, können neben dem Dokumenteninhalt CONTENT weitere Bestandteile, wie einen Zeitstempel enthalten. Dies spielt aufgrund der BAN-Regel zum Aufteilen einer übermittelten Aussage (5.3.2.f,) für den weiteren Beweis keine Rolle.

Vorhandene Aussagen nach Abbildung 6.3:

Die für die Verifikation notwendigen Aussagen sind an bestimmte Instanzen gebunden. Der Verifizierer VER liegt in der Beispielanordnung aus Abbildung 6.3 in einem anderen Zweig als der signierende PKI-Teilnehmer SIG. Somit müssen die Aussagen der folgenden Instanzen bei der Signaturprüfung berücksichtigt werden:

- ROOT: Die Wurzel-CA ist gemeinsame Vertrauensquelle für SIG und VER.
- IM_CA1/2: Zwischen-CAs, die SIG bzw. VER als PKI-Teilnehmer zertifiziert haben.
- TSA: Ein Zeitstempeldienst, der von der CA IM_CA1 autorisiert wurde.
- SIG: Ein PKI-Teilnehmer, der die Signatur zur Aussage DATA erzeugt hat.
- VER: Ein PKI-Teilnehmer, der die Signatur von SIG zur Aussage DATA verifiziert.

Nach SigI Abschnitt A2 [SIGI-SIG99] sind bei einer Verifikation Prüfobjekte erster, zweiter und dritter Ordnung zu berücksichtigen. Die signierten Daten DATA mit der dazugehörigen Signatur SIGNATURE gelten als Prüfobjekt erster Ordnung. Bei der Verwendung eines Zeitstempels nach SigI teilt sich DATA in den Inhalt des signierten Dokumentes CONTENT und den Zeitstempel TIMESTAMP auf:

DATA = CONTENT, TIMESTAMP

SIGNATURE = {H(DATA) }

Die Kette der Zertifikate und eventuell vorhandene Attributszertifikate gelten als Prüfobjekte zweiter Ordnung [SIGI-ZE99]. In diesem Beispiel zum Signaturgesetz sind die folgenden Aussagen Prüfobjekte zweiter Ordnung:

$ZERT_{ROOT}=(ZERTINFO_{ROOT},K_{ROOT},T_{begin},T_{end},ZERTSIG_{ROOT}) * f(t,T_{begin},T_{end}) *$ Root

$ZERT_{IM_CA1}=(ZERTINFO_{IM_CA1},K_{IM_CA1},T_{begin},T_{end},ZERTSIG_{ROOT}) * f(t,T_{begin},T_{end}) *$ IM_CA1

$ZERT_{SIG} = (ZERTINFO_{SIG}, K_{SIG}, T_{begin}, T_{end}, ZERTSIG_{IM_CA1}) * f(t,T_{begin},T_{end}) *$ SIG

Bei den Sperrlisten handelt es sich um Prüfobjekte dritter Ordnung. Diese Auskünfte werden von einem Verzeichnisdienst nach SigI [SIGI-VER99] bzw. ISIS Abschnitt AII [ISIS99] zur Verfügung gestellt. Hier sind die Sperrinformationen der Wurzel-CA ROOT und der Zertifizierungsstelle IM_CA1, die das Signaturzertifikat ausgestellt hat, zu untersuchen:

$REV_{ROOT} = (ZERT_{ID1} ... ZERT_{IDn}, T_{begin}, T_{end}, REVSIG_{ROOT}) * f(t, T_{begin}, T_{end}) *$ Root

REV

6.3.2 Beweisführung

Grundlegende Annahmen:

Die Annahmen für eine Verifikation mit der SigBAN beinhalten immer das Vertrauen von Verifizierer und Signierer in eine gemeinsame Quelle nach dem ersten Axiom. Desweiteren wird die Verfügbarkeit der Zertifikate und Sperrinformationen mindestens eines Pfades von dem signierten Dokument (Vertrauenssenke) zu der Wurzel-CA (Vertrauensquelle) angenommen.

$$\text{VER} \models \text{ROOT} \Rightarrow (\forall \text{X})_{\text{ROOT}}$$

Der Verifizierer VER vertraut der Wurzel-CA als zentrale Vertrauensquelle.

$\text{VER} \triangleleft \text{ZERT}_{\text{ROOT}}$, $\text{VER} \triangleleft \text{REV}_{\text{ROOT}}$, $\text{VER} \triangleleft \text{ZERT}_{\text{IM_CA1}}$,

$\text{VER} \triangleleft \text{REV}_{\text{IM_CA1}}$, $\text{VER} \triangleleft \text{ZERT}_{\text{TS}}$, $\text{VER} \triangleleft \text{ZERT}_{\text{SIG}}$

Der Verifizierer VER verfügt über die notwendigen Zertifikate und Sperrinformationen mindestens eines Pfades von den signierten Daten DATA und SIGNATURE bis zur Wurzel-CA ROOT.

Eine Überprüfung, die hier aufgrund der unterschiedlichsten Ausprägungen nicht modelliert wird, ist die Suche nach verwendeten Zertifikaten (oder den Zertifikats-IDs) in den Sperrinformationen. Alternativ dazu können auch alle gültigen Zertifikate in einer Positivliste eingetragen sein. Erst, wenn die Zertifikate als nicht gesperrt nachgewiesen wurden, können sie als gültige SigBAN-Aussagen für weitere logische Umformungen genutzt werden.

a) Beweis[1] der Aussage $\text{VER} \models \text{SIG} \vdash \text{CONTENT}$

Dieser Beweis ist ähnlich wie in dem Beispiel zur SigBAN-Verifikation aus Kapitel 5.6. Hier muss nur noch eine weitere Ebene in der Zertifizierungshierarchie berücksichtigt werden.

1. Umformung

Die erste Umformung leitet immer mit der Wurzelregel die Kenntnis des öffentlichen Schlüssels der Wurzel-CA für den Verifizierer VER ab.

$$\frac{\text{VER} \models \text{ROOT} \Rightarrow (\forall \text{X})_{\text{ROOT}},\ \text{VER} \triangleleft \text{ZERT}_{\text{ROOT}}}{\text{VER} \models \xrightarrow{\text{K}_{\text{ROOT}}} \text{ROOT}}$$

2a. Umformung

Mit den nächsten Umformungen wird durch mehrfache Anwendung der SigBAN-Regel zur Zertifizierung, entsprechend den Stufen der Zertifizierungshierarchie, die Kenntnis des öffentlichen Schlüssels des Signaturerzeugers SIG für den Verifizierer VER abgeleitet.

1. *Eine Aufstellung aller für den Beweis notwendigen Umformungen unter Berücksichtigung der eingehenden und resultierenden SigBAN-Aussagen ist im Anhang B2.1 zu finden.*

$$\frac{VER \mid\equiv \xrightarrow{K_{ROOT}} ROOT,\ VER \triangleleft ZERT_{IM_CA1}}{VER \mid\equiv \xrightarrow{K_{IM_CA1}} IM_CA1}$$

2b. Umformung

$$\frac{VER \mid\equiv \xrightarrow{K_{IM_CA1}} IM_CA1,\ VER \triangleleft ZERT_{SIG}}{VER \mid\equiv \xrightarrow{K_{SIG}} SIG}$$

3. Umformung

Mit der Regel zur Bedeutung von Nachrichten in Kombination mit der Regel für Hash-Funktionen aus der BAN-Logik kann jetzt die Richtigkeit der Aussage VER$\mid\equiv$SIG$\mid\sim$DATA bestätigt werden.

$$\frac{VER \mid\equiv \xrightarrow{K_{SIG}} SIG,\ VER \triangleleft \{DATA\}_{K_{SIG}^{-1}}}{VER \mid\equiv SIG \mid\sim DATA}$$

4. Umformung

Mit der Regel zur Aufteilung von einer übermittelten Aussage und der Kenntnis, dass DATA aus CONTENT und TIMESTAMP besteht, kann der Beweis abgeschlossen werden.

$$\frac{VER \mid\equiv SIG \mid\sim (CONTENT,\ TIMESTAMP)}{VER \mid\equiv SIG \mid\sim CONTENT}$$

6.4 SigBAN-Betrachtung zum Zeitstempeldienst

Eine Herausforderung an die SigBAN-Logik im Zusammenhang mit dem Signaturgesetz ist die Berücksichtigung aller zeitlichen Aspekte. Dazu werden hier die mit dem Zeitstempeldienst assoziierten Eigenschaften aus Kapitel 6.2.3 überprüft.

6.4.1 Anforderungen an den Zeitstempel in Form von SigBAN-Aussagen

Um die Signatur im Zusammenhang mit dem Zeitstempel zu verifizieren, sind die folgenden SigBAN-Behauptungen zu überprüfen. Bei einer Betrachtung mit der SigBAN-Logik fällt auf, dass auch der Zeitstempel TIMESTAMP wie die signierte Aussage DATA eine Vertrauenssenke darstellt. Der Zeitstempel kann deshalb wieder als eine signierte Aussage betrachtet werden, die analog einem Prüfobjekt erster Ordnung zu verifizieren ist.

Erste Aussage: VER $\models$ TSA $\sim$ TS_CONTENT

Der Verifizierer VER glaubt, der Inhalt der Aussage Zeitstempel TS_CONTENT stammt vom autorisierten Zeitstempeldienst TSA.

Da eine eindeutige Verknüpfung der signierten Daten mit dem Zeitstempel erforderlich ist, hat der Verifizierer VER zu überprüfen, ob der Zeitstempeldienst TSA eineindeutig die Aussage TIME mit der Signatur SIGNATURE und den signierten Daten CONTENT verbunden hat.

Zweite Aussage: VER $\models$ TSA $\sim$ (TIME, CONTENT)

Dritte Aussage: VER $\models$ TSA $\sim$ (TIME, SIGNATURE)

Der Verifizierer VER glaubt, die Aussagen CONTENT bzw. SIGNATURE lagen zum Zeitpunkt TIME dem Zeitstempeldienst TSA vor.

Weiterhin ist das Vertrauen des Verifizierers in die Zeitauskunft des Zeitstempeldienstes, die dieser vom Zeitdatendienst bezieht, nachzuweisen. Diese Anforderung an den Zeitstempel aus Kapitel 6.2.3 lässt sich aus den beiden obenstehenden Aussagen mit der Regel zur Aufteilung von übermittelten Aussagen ableiten.

Vierte Aussage: VER $\models$ TSA $\sim$ TIME

Mit der Regel zur Aufteilung von übermittelten Aussagen ist diese Forderung bereits in VER$\models$TSA$\sim$(TIME,SIGNATURE) oder in VER$\models$TSA$\sim$(TIME,CONTENT) enthalten.

6.4.2 Überprüfung des Zeitstempels bei einer Realisierung nach SigI

Vorhandene Aussagen:

Wie bei der Modellierung der einfachen Signaturprüfung aus Kapitel 6.3 müssen auch vor der Beweisführung zum Zeitstempeldienst alle vorhandenen SigBAN-Aussagen ermittelt werden. Als Grundlage wird hier ebenfalls die Beispielkonstellation aus Abbildung 6.3 verwendet.

Bei einer Realisierung eines Zeitstempeldienstes auf Basis der SigI-Spezifikation (siehe Abschnitt A4, Kapitel 6 Nachrichtenformate) kann ein qualifizierter Zeitstempel folgendermaßen als SigBAN-Aussage formuliert werden:

$$\text{TIMESTAMP} = \text{TS_CONTENT}, \{H(\text{TS_CONTENT})\}_{K_{TSA}^{-1}}$$

mit TS_CONTENT = H(CONTENT), TIME

Dabei ist der Hash-Wert des Inhaltes der signierten Daten H(CONTENT) Bestandteil des Zeitstempels. Dadurch wird eine eindeutige Verknüpfung zwischen den signierten Daten CONTENT und der Zeitinformation TIME erreicht. Der Zeitpunkt TIME ist eine Aussage, die vom Zeitdatendienst nach Abbildung 6.2 getroffen wird. Für die Verifikation des Zeit-

[redacted]

[redacted]sammenhang m[redacted]

[redacted].3). Durch Anwendung der Regel zur [redacted]entlichen Schlüssels des Zeitstempeldien-[redacted]den.

[redacted], VER $ZERT_{TSA}$

TSA

[redacted] Nachrichten und der Regel für Hash-Funktionen kann nach-[redacted]erifizierer VER sicher sein kann, der Inhalt des Zeitstempels [redacted]n Zeitstempeldienst TSA:

[redacted] TSA, VER {H(TS_CONTENT)}

VER TSA H(TS_CONTENT)

VER TSA H(TS_CONTENT), VER TS_CONTENT

VER TSA TS_CONTENT

b) Beweis der zweiten Aussage VER ∈ TSA ⊢ (TIME, CONTENT)

Diese Aussage kann auch auf Basis von VER∈TSA⊢TS_CONTENT bewiesen werden. Dazu wird die Regel für Hash-Funktionen für den Nachweis verwendet, dass der Zeitstempeldienst TSA über den Inhalt des signierten Dokumentes CONTENT zum Zeitpunkt TIME verfügt hat. Tatsächlich lag dem Zeitstempel nur der Hashwert des Inhaltes H(CONTENT) vor. Dies spielt aber aufgrund folgender Umformung mit der Regel für Hash-Funktionen keine Rolle, wenn der Verifizierer VER Kenntniss über den Inhalt CONTENT besitzt.

$$\frac{VER \in TSA \vdash H(CONTENT),\ VER \triangleleft CONTENT}{VER \in TSA \vdash CONTENT}$$

c) Beweis der vierten Aussage VER ∈ TSA ⊢ TIME

Mit dem Beweis von VER∈TSA⊢TS_CONTENT und der Regel zum Aufteilen einer übermittelten Aussage, kann der Verifizierer VER sicherstellen, dass die Information TIME vom Zeitstempeldienst TSA stammt.

$$\frac{VER \in TSA \vdash (H(CONTENT), TIME)}{VER \in TSA \vdash TIME}$$

d) Versuch die dritte Aussage VER ∈ TSA ⊢ (TIME, SIGNATURE) zu beweisen

Mit TIMESTAMP = H(CONTENT), TIME, $\{H(H(CONTENT),TIME)\}_{K_{TSA}^{-1}}$ ist ein Beweis dieser Forderung unmöglich, da die Aussage SIGNATURE kein Bestandteil des Zeitstempels TIMESTAMP ist. Der Verifizierer VER kann deshalb nicht sicher sein, dass die Signaturerzeugung auch tatsächlich zum Zeitpunkt TIME stattgefunden hat und nicht später.

In Kombination mit der SigBAN-Aussage VER∈TSA⊢(TIME, CONTENT) lässt sich folgern, dass der Verifizierer VER nur die Existenz der signierten Daten CONTENT zum Zeitpunkt TIME als gesichert ansehen kann.

Die Untersuchung mit der SigBAN-Logik hat hier den Hinweis auf einen Schwachpunkt bei einem Zeitstempeldienst nach SigI Abschnitt A4 [SIGI-ZS99] ergeben. Durch den darin beschriebenen Zeitstempeldienst wird nicht der tatsächliche Signaturzeitpunkt festgehalten, sondern nur die Existenz der signierten Daten zu diesem Zeitpunkt bewiesen. Damit kann der für eine Verifikation der Signatur wichtige Signaturzeitpunkt nicht verläßlich ermittelt werden.

6.4.3 Vorschlag zur zuverlässigen Ermittlung des Signaturzeitpunktes

Aufgrund der zuvor beschriebenen Nachteile eines Zeitstempels nach SigI Abschnitt A4 wird hier mit Hilfe der SigBAN-Logik ein Vorschlag für ein Zeitstempelformat, das eine verlässliche Auskunft über den Signaturzeitpunkt beinhaltet, formuliert. Die Aussage für einen Zeitstempel VER$\models$TSA$\mid\sim$(TIME, SIGNATURE) soll damit beweisbar werden. Durch eine Kombination von zwei Zeitstempeln kann ein eindeutiges Intervall [TIME1, TIME2], in dem der Signaturzeitpunkt liegt, bestimmt werden. Im Folgenden wird dieses Verfahren, das aus mehreren Schritten besteht, beschrieben:

1. Schritt: Der erste Zeitstempel TIMESTAMP1 wird über den Hash-Wert des Dokumenteninhaltes H(CONTENT) und die erste Zeitinformation TIME1 erzeugt.

$$\text{TIMESTAMP1} = \text{H(CONTENT), TIME1, } \{\text{H(H(CONTENT), TIME1)}\}_{K_{TSA}^{-1}}$$

2. Schritt: Dieser Zeitstempel TIMESTAMP1 wird wie bei dem Verfahren nach SigI mit dem Inhalt des Dokumentes CONTENT vom Signierer SIG unterschrieben.

$$\text{SIGNATURE} = \{\text{H(CONTENT, TIMESTAMP1)}\}_{K_{SIG}^{-1}}$$

Der **3. Schritt** wird ergänzend zu dem Zeitstempel nach SigI eingeführt, um die Signatur mit einer Zeitinformation zu verbinden. Der endgültige Zeitstempel TIMESTAMP wird jetzt über die Signatur SIGNATURE und einen erneut ermittelten Zeitpunkt TIME2 vom Zeitstempeldienst TSA berechnet.

$$\text{TIMESTAMP} = \text{TIME2, SIGNATURE, TIMESTAMP1, } \{\text{H(SIGNATURE, TIME2)}\}_{K_{TSA}^{-1}}$$

6.4.4 Beweis zur zuverlässigen Ermittlung des Signaturzeitpunktes

Hier soll die Aussage VER$\models$TS$\mid\sim$(TIME, SIGNATURE) bewiesen werden. Da sich das Format des Zeitstempels geändert hat, müssen auch alle anderen, von der Formatänderung betroffenen Aussagen, erneut betrachtet werden. Der Nachweis, dass die Aussage TIMESTAMP von der autorisierten Instanz TSA stammt (VER$\models$TSA$\mid\sim$TIMESTAMP), wird davon nicht beeinflusst, da hier die Bestandteile von TIMESTAMP bzw. TIMESTAMP1 keine Rolle spielen. Das gleiche gilt für den Beweis von VER$\models$SIG$\mid\sim$CONTENT.

a) Skizze zum Beweis[1] der Aussagen VER$\models$TSA$\mid\sim$TIME2

Mit der Regel für Hash-Funktionen und den Bestandteilen H(SIGNATURE, TIME2), SIGNATURE und TIME2 der Aussage TIMESTAMP kann der Verifizierer VER sicherstellen, dass die Information (SIGNATURE, TIME2) vom Zeitstempeldienst TSA übermittelt wurde.

$$\frac{\text{VER} \models \text{TSA} \mid\sim \text{H(SIGNATURE, TIME2)}, \text{VER} \triangleleft \text{(SIGNATURE, TIME2)}}{\text{VER} \models \text{TSA} \mid\sim \text{(SIGNATURE, TIME2)}}$$

1. *Die Skizzen zu den beiden Beweisen verwenden bereits Ergebnisse aus anderen Beweisketten. Eine vollständige Herleitung über die Ausgangsrelationen befindet sich in Anhang B2.3.*

Daraus kann mit der Regel zur Zusammensetzung und Aufteilung vertrauenswürdiger Aussagen das Vertrauen des Verifizierers VER in die Richtigkeit der Aussage TIME2 durch den Zeitstempeldienst TSA nachgewiesen werden.

$$\frac{\text{VER} \mid\equiv \text{TSA} \mid\sim (\text{SIGNATURE}, \text{TIME2})}{\text{VER} \mid\equiv \text{TSA} \mid\sim \text{TIME2}}$$

b) Skizze zum Beweis[1] von VER$\mid\equiv$TSA$\mid\sim$TIME1:

Die Beweisführung zu VER$\mid\equiv$TSA$\mid\sim$TIME1 ist analog zu der von VER$\mid\equiv$TSA$\mid\sim$TIME2 mit dem einzigen Unterschied, dass der Inhalt von TIMESTAMP1 d.h. (H(CONTENT), TIME1) statt (SIGNATURE, TIME2) in der Regel für Hash-Funktionen verwendet wird:

$$\frac{\text{VER} \mid\equiv \text{TSA} \mid\sim (\text{H(CONTENT)}, \text{TIME1})}{\text{VER} \mid\equiv \text{TSA} \mid\sim \text{TIME1}}$$

c) Beweisskizze zu VER $\mid\equiv$ TSA $\mid\sim$ (TIME, SIGNATURE)

Im Gegensatz zu einer SigG-konformen Realisierung des Zeitstempels ist jetzt der Nachweis möglich, dass der Zeitstempeldienst TSA einen bestimmten Zeitpunkt TIME eindeutig mit der Signatur SIGNATURE verbunden hat. Das Zeitintervall [TIME1, TIME2] kann dabei als atomarer Zeitpunkt TIME betrachtet werden, da die Signaturerzeugung, bei der die beiden Zeitstempel angefordert werden, als ein Gesamtprozess angesehen wird. Die Zeitdauer der Signaturerzeugung ist dabei gegenüber den Gültigkeitszeiträumen der anderen Aussagen vernachlässigbar. Somit gilt:

$$\text{TIME} = \text{TIME1} = \text{TIME2}$$

Nach der Herleitung der Kenntniss des öffentlichen Schlüssels des Zeitstempeldienstes für den Verifizierer VER $\mid\equiv \xrightarrow{K_{TSA}}$ TSA, kann mit der Regel zur Bedeutung von Nachrichten bewiesen werden, dass dem Zeitstempeldienst TSA zum Zeitpunkt TIME2 = TIME die Aussage SIGNATURE vorlag:

$$\frac{\text{VER} \mid\equiv \xrightarrow{K_{TSA}} \text{TSA}, \text{VER} \triangleleft \{\text{H(TIME2, SIGNTURE)}\}_{K_{TSA}^{-1}}}{\text{VER} \mid\equiv \text{TSA} \mid\sim (\text{TIME}, \text{SIGNATURE})}$$

d) Skizze des Beweises[1] der Aussage VER TSA (TIME, CONTENT)

Ähnlich wie in der vorherigen Beweisführung wird hier mit der Regel zur Bedeutung von Nachrichten gefolgert, dass dem Zeitstempeldienst TSA zum Zeitpunkt TIME1 = TIME der Hashwert des Dokumenteninhaltes H(CONTENT) zur Verfügung stand. Zusätzlich wird die Regel für Hash-Funktionen benutzt, um den Zugriff auf den eigentlichen Inhalt CONTENT für den Zeitstempel nachzuweisen VER TSA CONTENT:

VER TSA,VER {H(TIME1, H(CONTENT))}

VER TSA (TIME, H(CONTENT))

VER TSA H(CONTENT), VER CONTENT

VER TSA CONTENT

Wird das Ergebnis der Regel zur Bedeutung von Nachrichten und der Regel für Hash-Funktionen zusammengeführt, gilt VER TSA (TIME, CONTENT).

6.5 Vergleich der Gültigkeitsmodelle mit den SigBAN-Forderungen

Hier wird zuerst das Schalenmodell (SM) und das Kettenmodell (KM) aus der SigI [SIGI-GÜM99] bzw. ISIS-MTT Version 1.0 [ISIS01] analysiert. Dabei werden Probleme beim Einsatz des SM und des KM erkannt. Zusätzlich stellt sich heraus, dass SM und KM untereinander widersprüchlich sind. Eine Ursache für die in SM und KM unter bestimmten Bedingungen

6.5.1 Schalenmodell

Das Schalenmodell (SM) entspricht dem SigBAN-Ansatz am weitesten. Hier wird wie im zweiten Axiom der SigBAN-Logik die Gültigkeit aller Zertifikate, zum für die Überprüfung relevanten Zeitpunkt gefordert. Allerdings berücksichtigt das SM nur Zertifikate und nicht konsequent alle an der Verifikation beteiligten Aussagen. Die zeitliche Gültigkeit der Prüfobjekte dritter Ordnung und des Zertifizierungspfades zum Zeitstempeldienst werden nicht in die zeitliche Gültigkeitsbetrachtung einbezogen. Dies führt unter Umständen zu einer falschen Einstufung der Gültigkeit:

- Trotz eines abgelaufenen Zertifikates des Zeitstempeldienstes, kann nach dem SM die digitale Unterschrift gültig sein.
- Ein positives Verifikationsergebnis ist auch möglich, wenn eine verwendete Sperrliste zum für die Überprüfung relevanten Zeitpunkt bereits abgelaufen oder noch nicht gültig ist.

Eine Besonderheit beim SM ist, dass bei einer Überprüfung alternativ der Signaturzeitpunkt oder der Prüfzeitpunkt als relevant angesehen wird. Durch Verwendung des Prüfzeitpunktes ist es möglich, den aufwendigen und problematischen Zeitstempeldienst zu umgehen, wenn die Signaturprüfung kurz nach der Erstellung erfolgt. In einer SigBAN-Beschreibung würde das bedeuten, dass statt dem Signaturzeitpunkt die aktuelle Zeit mit den Gültigkeitszeiträumen aller Aussagen abgeglichen wird und eine logische Überprüfung der Aussagen des Zeitstempeldienstes nicht mehr erforderlich ist. Eine Überprüfung ist allerdings dann nicht mehr möglich, wenn eine an der Verifikation beteiligte Aussage ungültig wird. Deshalb ist dieses Verfahren besonders für eine PKI, in der die Kommunikation zwischen verschiedenen Parteien abgesichert wird, geeignet (siehe RFC 1442 [PEM]). Für Langzeitsignaturen, die eine manuelle Unterschrift ersetzen sollen, kann das SM nicht verwendet werden. Dies wurde auch in der neuen Interoperabilitätsspezifikation ISIS-MTT [ISIS01] erkannt[1].

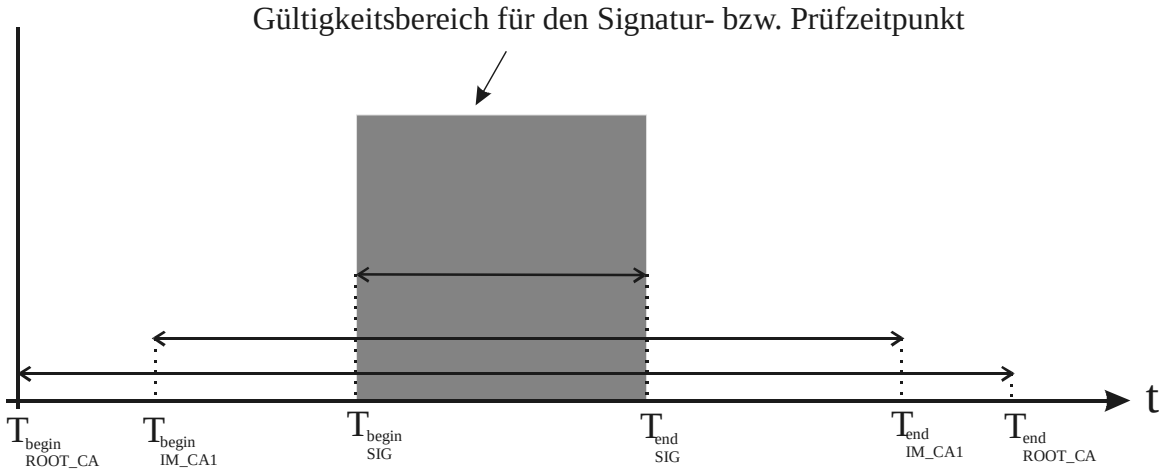

Abbildung 6.4 *Beispiel zur zeitlichen Betrachtung aller Zertifikate nach dem Schalenmodell*

1. *Im Teil „SigG-conforming systems and applications“ (Kapitel 2.1 zum Schalenmodell) ist festgehalten: „Signatures are therefore required to be valid only during a short period of time, ...This fact is refelcted by the validity model“*

6.5.2 Kettenmodell

Das Kettenmodell (KM) verwendet zur Überprüfung im Gegensatz zum SM nur den Signaturzeitpunkt. Es ist also ein Zeitstempeldienst im KM zur verläßlichen Ermittlung des Signaturzeitpunktes erforderlich. Wie im SM werden auch im KM nicht die Gültigkeitszeiträume aller Aussagen, die an der Verifikation beteiligt sind, berücksichtigt. Dies ist insbesondere kritisch, da das KM den Zertifizierungspfad des zwingend erforderlichen Zeitstempeldienstes nicht beachtet. Ein weiterer wesentlicher Kritikpunkt am KM ist die Gültigkeit einer Signatur, obwohl an der Verifikation beteiligte Zertifikate bereits nicht mehr gültig sind. Im KM wird nur eine Verkettung der Gültigkeitszeiträume (siehe dunkel grauer Bereich in Abbildung 6.5) gefordert. Es muss also nur überprüft werden, ob zum Erstellungszeitpunkt eines Zertifikates das Ausstellerzertifikat gültig war. Dies vereinfacht zwar den Überprüfungsprozess wesentlich, verhindert aber aber den Widerruf von Ausstellerzertifikaten:

- Einer einmal getroffenen Aussage einer Ausstellerinstanz wird im KM eine Gültigkeit für alle Ewigkeit eingeräumt.

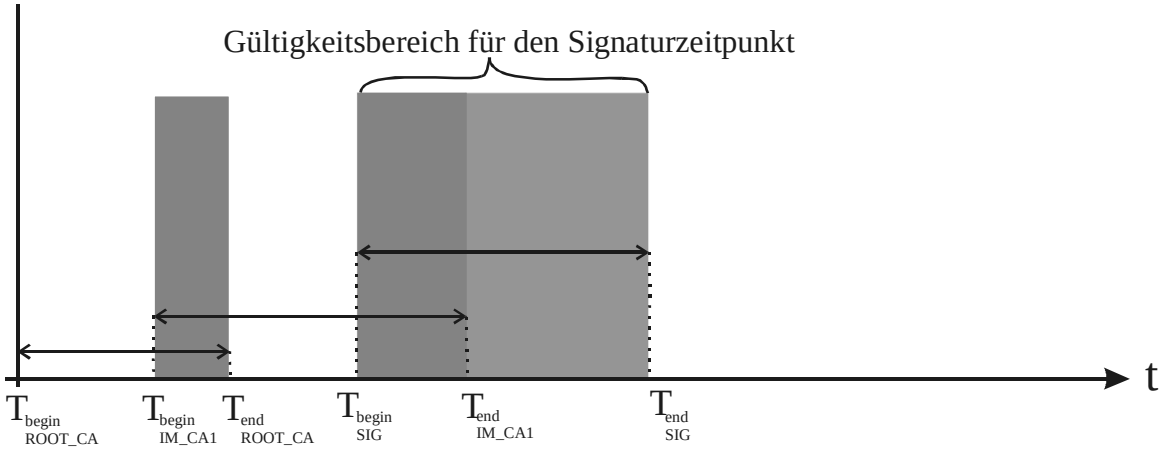

Abbildung 6.5 *Zeitliche Betrachtung nach dem Kettenmodell*

Wird im KM ein Teil der an der Verifikation beteiligten Aussagen (Wurzelzertifikat oder Ausstellerzertifikat) ungültig, hat dies keine Auswirkung auf das Verifikationsergebnis. Dies steht im Widerspruch zum zweiten Axiom der SigBAN-Logik. Unabhängig von der SigBAN-Betrachtung gibt es Fälle, in denen die Gültigkeit einer digitalen Signatur im KM problematisch ist. Hier einige Beispiele:

- Wird ein Angriff auf einen bei der Implementierung verwendeten Algorithmus bekannt, und die Zertifizierungsstelle deshalb auf einen neuen Algorithmus umgestellt, besteht noch immer die Möglichkeit, dass ein Angreifer über die bereits erkannte Sicherheitslücke Daten in die PKI einschleusst. Bei der Verifikation würde die „alte“ Zertifizierungshierarchie noch akzeptiert werden.

- Stellt eine Zertifizierungsinstanz den Dienst ein und kann ihren Verpflichtungen nicht mehr nachkommen, ist es möglich, das herausgegebene Teilnehmerzertifikat ohne Deckung durch eine übergeordnete Instanz weiterzuverwenden.

Sowohl das Kettenmodell als auch das Schalenmodell berücksichtigen nicht die Gültigkeit aller an der Verifikation beteiligten Informationen. Darüberhinaus lassen sich Schwachstellen bei der Gültigkeitsbetrachtung mit verketteten Zeiträumen nachweisen. Dies führt im Kettenmodel dazu, dass Ausstellerzertifikate nicht gesperrt werden können.

6.5.3 SigBAN-Modell

Im SigBAN-Modell ist nach dem zweiten Axiom eine zeitliche Gültigkeit aller an der Verifikation beteiligten Aussagen erforderlich (siehe 5.5.3b). Es müssen also die Gültigkeitszeiträume aller Prüfobjekte zweiter und dritter Ordnung unter Berücksichtigung des Pfades für den Zeitstempeldienst betrachtet werden. Dazu gehören die Zertifikate des Signaturerzeugers, des Zeitstempeldienstes, der Aussteller-CA, der Wurzel-CA, sowie die Sperrinformationen der Aussteller und Wurzel-CA. In dem Beispiel zur Verifikation aus Kapitel 6.3 nach dem SigG sind das hier die folgenden SigBAN-Aussagen: REV_{ROOT_CA}, $ZERT_{ROOT_CA}$, REV_{IM_CA1}, $ZERT_{IM_CA1}$, $ZERT_{TS}$, $ZERT_{SIG}$

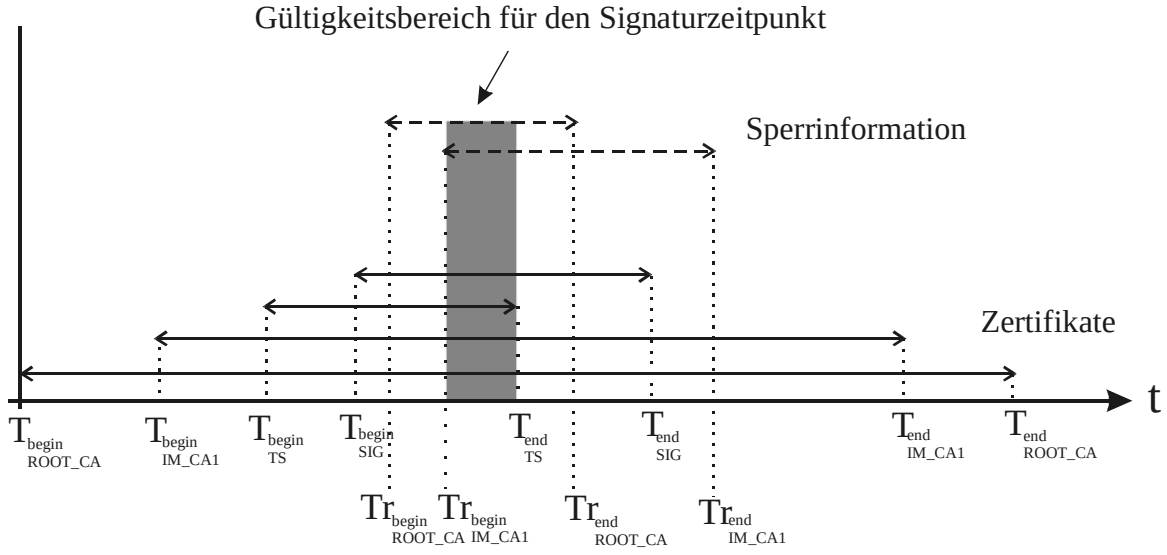

Abbildung 6.6 *Zeitliche Betrachtung nach dem SigBAN-Modell*

Nach dem zweiten Axiom der SigBAN-Notation, ist für die vollständige Signaturprüfung die zum Signaturzeitpunkt gültige Sperrinformation notwendig (siehe Sperrinformation in Abbildung 6.3). Die Verwendung einer später ausgestellten Sperrliste ist zwar möglich, da ein einmal widerrufenes Zertifikat nach dem SigG auch in allen weiteren Sperraussagen ei-

ner Zertifizierungsstelle zu finden ist. Allerdings kann ein kurz nach der Signaturerzeugung und noch vor der Prüfung widerrufenes Zertifikat zu einem falschen Ergebnis führen, wenn nicht die zum Zeitpunkt der Signaturerzeugung gültige Sperraussage bei der Verifikation der Signatur verwendet wird.

Wie gerade gezeigt, werden vom SigBAN-Modell im Unterschied zu den Gültigkeitsmodellen nach dem SigG alle an der Verifikation beteiligten Aussagen berücksichtigt. Zusätzlich gelten für die Betrachtung von Gültigkeitszeiträumen mit dem zweiten SigBAN-Axiom feste Vorgaben, die nicht wie im KM oder SM den Bedürfnissen einer einfacheren Realisierbarkeit angepasst sind. Dadurch wird in diesem Abschnitt besonders deutlich, dass eine vertrauenswürdige Signaturprüfung die Eigenschaften einer formalen Sprache benötigt.

6.6 Abschließender Kommentar und Bewertung der SigBAN Anwendung

Mit der Anwendung der SigBAN-Logik auf die Verifikation nach dem deutschen Signaturgesetz wurde gezeigt, dass bereits bei einer einfacheren PKI eine vollständige Überprüfung ein aufwendiger Vorgang ist. Der logische Beweis wird durch die konsequente Beachtung aller Aussagen, die mit einer qualifizierten elektronischen Unterschrift verbunden sind, bereits sehr umfangreich.

Im Zusammenhang mit der Verwendung eines SigG-konformen Zeitstempeldienstes konnte eine Sicherheitslücke durch eine nicht beweisbare SigBAN-Aussage erkannt werden. Deshalb wurde mit der SigBAN-Logik ein Verbesserungsvorschlag für die zuverlässige Ermittlung des Signaturzeitpunktes formuliert. Weiterhin wurden auch Probleme bei den zum SigG gehörenden Gültigkeitsmodellen herausgestellt, die eine SigBAN-Beschreibung vermeidet.

Der Hauptvorteil einer formalen Beschreibungssprache ist die konsequente Anwendung von Regeln bis eine gewünschte Aussage bewiesen oder widerlegt ist. Eine allgemeinsprachliche Form kann eine Analyse nur schwer mit derselben Konsequenz vorantreiben. Insbesondere wird im Fließtext ein Bruch von vorangestellten Prämissen nicht so leicht erkannt. Mit der SigBAN-Logik konnte die technische Verifikation von qualifizierten elektronischen Unterschriften nach dem Signaturgesetz auf Lücken untersucht und präzisiert werden.

Eine konventionelle Analyse von softwaretechnischen Prozessen ist in vielen Fällen nicht ausreichend, sondern muss durch ein passendes logisches Modell unterstützt werden. Das ist auch der Grund, warum die ursprüngliche BAN-Logik für die Implementierung von Authentisierungsprotokollen entwickelt wurde. Hier wurde die Eignung der SigBAN-Logik für die Modellierung der Verifikation anhand des deutschen SigG demonstriert.

7. Erweiterte Anwendung der SigBAN-Logik

7.1 Überblick

Die Analyse der Verifikation nach dem Signaturgesetz aus dem vorherigen Kapitel war das erste größere Beispiel zur Demonstration der Beschreibungssprache SigBAN. Da das SigG nur eine bestimmte Ausprägung einer PKI darstellt, werden in diesem Kapitel weitere Einsatzmöglichkeiten der SigBAN-Logik vorgestellt. Dabei soll der Entwurf einer Beschreibungssprache zur Verifikation abgerundet und die Flexibilität der Analyse mit SigBAN aufgezeigt werden.

Zuerst werden besondere Zertifikatstypen und ihr Einsatz mit der SigBAN-Logik analysiert. Insbesondere die vorgestellten Cross-Zertifikate haben eine wichtige Aufgabe bei der zunehmenden Verbreitung von PKIs. In vielen Fällen werden früher eigenständige Insellösungen über Cross-Zertifizierung zusammengeführt, wodurch eine sichere Kommunikation über vorhandene Domänengrenzen hinweg möglich wird. Attributszertifikate erweitern dagegen den Funktionsumfang in einer existierenden PKI. Mit Attributszertifikaten können bestimmte Daten oder Prozesse mit unterschiedlichen Rollen zur Nutzung bestimmter PKI-Funktionalitäten versehen werden. Dadurch wird eine stärkere Differenzierung innerhalb einer PKI erreicht.

Ein wichtiger Aspekt bei der Verifikation ist das zugrundeliegende Vertrauensmodell. Deshalb werden in einem eigenem Kapitel die verschiedenen Möglichkeiten für den Vertrauensfluss in einer PKI vorgestellt. Neben einem Schema für die SigBAN-Analyse in einer strikten Vertrauenshierarchie, werden auch die Möglichkeiten zur Verbindung unterschiedlicher Vertrauens-Domänen (eng. Trust Domains) betrachtet. Mit der SigBAN-Logik kann dieser Vertrauensübergang modelliert werden. Zusätzlich ist eine Differenzierung zwischen den alternativen Pfaden von der Vertrauenssenke zur Vertrauensquelle möglich. Als Anwendungsbeispiel für einen verteilten Vertrauensfluss wird die Realisierung der Bridge-CA Initiative mit der SigBAN-Logik analysiert.

Der Entwurf einer Beschreibungssprache zur Verifikation wird mit Fällen, die für eine SigBAN-Beschreibung nicht geeignet sind, abgerundet. Eine generelle Sicherheitsbetrachtung aller Prozesse, die auf Public-Key-Kryptographie beruhen, ist mit der SigBAN-Logik nicht möglich. Die Beschreibungssprache SigBAN wurde entworfen, um die Gestaltung von Verfahren zur Anwendung von digitalen Signaturen, die hohen Ansprüchen genügen sollen, zu unterstützen. Dies sind insbesondere Applikationen bei denen eine digitale Signatur eine manuelle Unterschrift ersetzen soll.

7.2 SigBAN-Beschreibung zur Verwendung besonderer Zertifikatsformate

Ein Zertifikat dient grundsätzlich zur Legitimation für die Ausübung einer bestimmten Tätigkeit in einer PKI. Es ist dabei nicht von Bedeutung, ob eine Person oder ein Prozess (z.B. OCSP-Server) legitimiert wird. Das gilt insbesondere für die in diesem Kapitel betrachteten Zertifikate, deren Zweck die Erweiterung und flexiblere Gestaltung von PKIs ist. In den hier vorgestellten Zertifikatsformaten wird auch deutlich, dass eine PKI keine statische Struktur ist, sondern einer ständigen Veränderung unterworfen ist. Bestimmte Zertifikate dienen dabei zur Erweiterung der PKI (Cross-Zertifizierung) oder zur stärkeren Differenzierung einzelner Aufgaben innerhalb der Infrastruktur (Attributszertifikate). Deshalb spielt neben der Beschreibung dieser besonderen Zertifikate als SigBAN-Aussagen die Modellierung der damit verbundenen Implikationen auf die Infrastruktur eine wichtige Rolle.

7.2.1 Mehrfachzertifikate

Mehrfachzertifikate beinhalten wie Teilnehmer- und Ausstellerzertifikate einen öffentlichen Schlüssel und verbinden dadurch mit dem zugehörigen privaten Schlüssel einen bestimmten Verwendungszweck. Sie entsprechen im Aufbau dem X.509 Standard [X50999]. Die einzige Besonderheit bei Mehrfachzertifikaten ist, dass ein bereits existierendes Schlüsselpaar erneut in einem Zertifikat bestätigt wird. Ein Anwendungsbeispiel für Mehrfachzertifikate ist die Verlängerung eines existierenden Zertifikates. Ein Mehrfachzertifikat kann nicht nur einen veränderten Gültigkeitszeitraum, sondern auch einen anderen Inhalt als das korrespondierende Zertifikat aufweisen. Durch eine Änderung des Zertifikatsinhaltes können andere Rechte mit dem bereits vorhandenen Schlüsselpaar verknüpft werden.

In der SigBAN-Notation würde ein Mehrfachzertifikat dadurch auffallen, dass ein bereits existierender öffentlicher Schlüssel erneut in einem anderen Zertifikat bestätigt wird. Bei einem Verlängerungszertifikat ändert sich dabei nur der Gültigkeitszeitraum [T_{begin}, T_{end}], während ein Austauschzertifikat auch einen anderen Inhalt ZERTINFO besitzt.

Ursprüngliches Zertifikat: $ZERT_{OLD}$= $ZERTINFO_{OLD}$, T_{old_begin}, T_{old_end}, **K**, $ZERTSIG_{CA_X}$

Mehrfachzertifikat: $ZERT_{NEW}$ = $ZERTINFO_{NEW}$, T_{new_begin}, T_{new_end}, **K**, $ZERTSIG_{CA_Y}$

Wie anhand obiger SigBAN-Aussagen deutlich wird, kann sich in einem Mehrfachzertifikat alles bis auf den öffentlichen Schlüssel K ändern. Die Signatur der Aussteller-CA wird jedoch nur in den seltenen Fällen einer Cross-Zertifizierung wechseln. Allerdings ist die Cross-Zertifizierung aufgrund der damit verbundenen Strukturänderung einer PKI interessant für eine weitere SigBAN-Betrachtung. Eine „klassische“ PKI besitzt den Aufbau einer Baumstruktur mit einer Vertrauensquelle. Mit Hilfe von Cross-Zertifizierung kann ein Vertrauensnetzwerk durch Querverbindungen zwischen Vertrauensquellen geschaffen werden. Neben der Zusammenführung früher eigenständiger PKIs, ist auch die Verkürzung des Vertrauenspfades ein wichtiges Anwendungsgebiet für Cross-Zertifikate. Aus diesem Grund wird die Cross-Zertifizierung mit Mehrfachzertifikaten in dem folgenden Anwendungsbeispiel vorgestellt.

Bei der Cross-Zertifizierung ist zu beachten, dass mehrere Vertrauensquellen in einer PKI nicht im Widerspruch zum ersten Axiom der SigBAN-Logik stehen. Die darin festgehaltene Einschränkung auf nur eine Vertrauensquelle bezieht sich immer auf die Überprüfung eines Pfades von der Vertrauenssenke (Zertifikat) zur Vertrauensquelle (Wurzel-CA). Gibt es aufgrund von Cross-Zertifikaten mehrere verschiedene Pfade, ist jeder einzelne mit SigBAN-Umformungen unter Anwendung des ersten Axioms zu überprüfen.

7.2.2 Anwendungsbeispiel zur Cross-Zertifizierung mit Mehrfachzertifikaten

Beim Zusammenschluss von zuvor eigenständigen PKIs stellt eine Zertifizierungsstelle für eine andere ein Ausstellerzertifikat aus (siehe Abbildung 7.1). Dadurch werden alle Teilnehmer der einen PKI-Hierarchie auch in der anderen legitimiert. Jede CA legt dabei fest, mit welchen Rechten die PKI-Teilnehmer der anderen CA in der eigenen PKI agieren dürfen. Die Vergabe dieser Rechte ist Bestandteil des Zertifikatsinhaltes, der von der herausgebenden CA festgelegt wird.

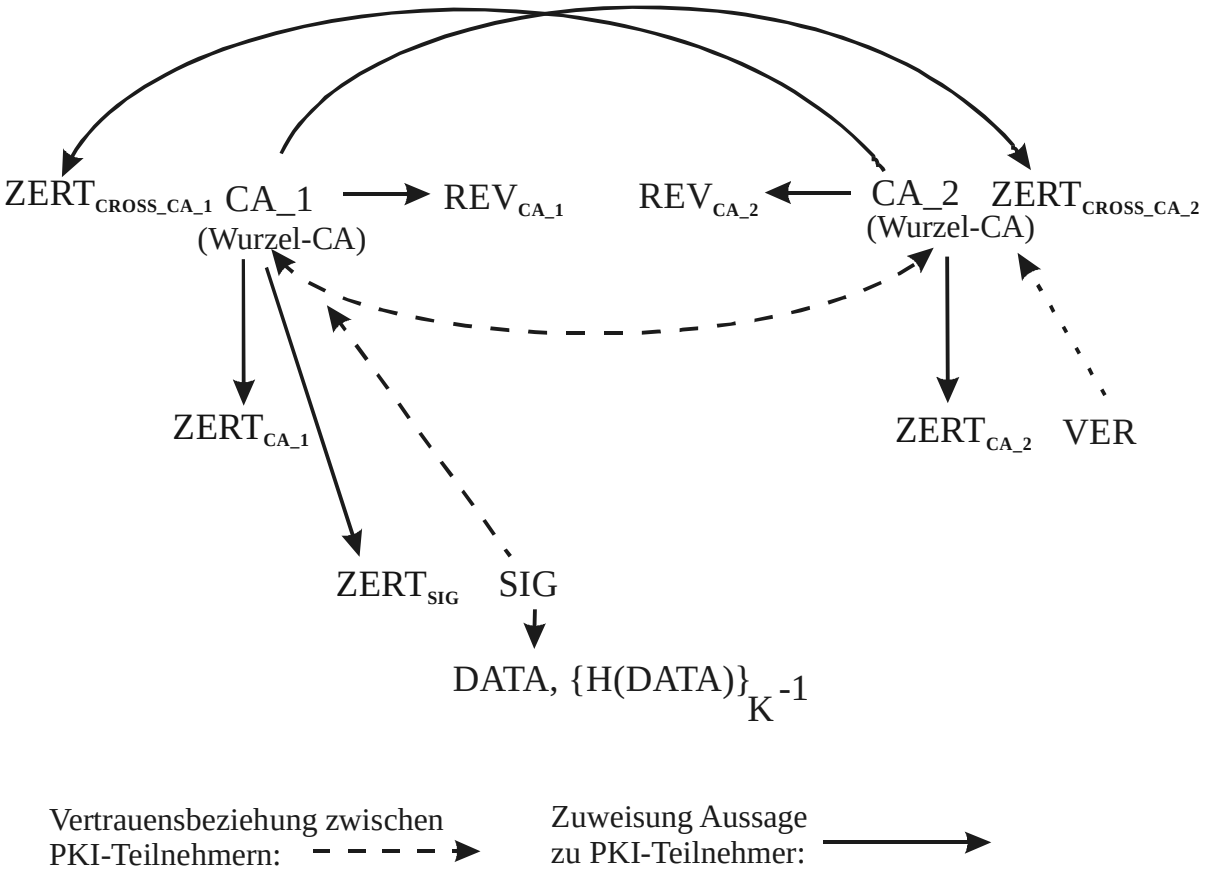

Abbildung 7.1 *Cross-Zertifizierung durch Mehrfachzertifikate*

In der Abbildung 7.1 wird ein Beispiel zur Cross-Zertifizierung dargestellt, in dem der Verifizierer VER einer anderen Wurzel vertraut als der Signierer SIG. Es gibt also nur einen Pfad über ein Cross-Zertifikat vom signierten Datensatz DATA zu der Wurzel-CA CA_2.

Die dafür notwendigen Cross-Zertifikate besitzen den folgenden Aufbau:

$ZERT_{CROSS_CA_1} = ZERTINFO_{CA1}, T_{begin}, T_{end}, K_{CA1}, ZERTSIG_{CA_2}$

$ZERT_{CROSS_CA_2} = ZERTINFO_{CA2}, T_{begin}, T_{end}, K_{CA2}, ZERTSIG_{CA_1}$

Vertrauenspfad zwischen den Instanzen:

1. Pfad: VER - - - - -► CA_2 - - - - -► CA_1 ◄- - - - - SIG

Vertrauenspfad zwischen den Aussagen:

$DATA, \{DATA\}_{K^{-1}_{SIG}}$ - -► $ZERT_{SIG}$ - -► $ZERT_{CROSS_CA_1}$ - -► $ZERT_{CA_2}$

Die Verifikation beginnt mit dem Vertrauen in die Wurzelinstanz CA_2 des Verifizierers VER:

$VER \models CA_2 \Rightarrow (\forall X)_{CA_2}$

Neben einem Zertifikat mehr ($ZERT_{CROSS_CA_1}$), ist im Pfad über Cross-Zertifizierung auch eine Sperrliste (REV_{CA_2}) zusätzlich zu überprüfen. Ein möglicher Widerruf des Cross-Zertifikates wird über die Zertifizierungsstelle CA_2 bekanntgegeben. Deshalb muss bei der Überprüfung der Zertifikate auf Sperrung auch die Aussage REV_{CA_2} ausgewertet werden:

$ZERT_{CA_2} = (ZERTINFO_{CA_2}, K_{CA_2}, T_{begin}, T_{end}, ZERTSIG_{CA_2}) * f(t, T_{begin}, T_{end}) * \propto_{CA_2}$

$ZERT_{CROSS_CA_1} = (ZERTINFO_{CA_2}, K_{CA_1}, T_{begin}, T_{end}, ZERTSIG_{CA_2}) * f(t, T_{begin}, T_{end}) * \propto_{CA_1}$

$ZERT_{SIG} = (ZERTINFO_{SIG}, K_{SIG}, T_{begin}, T_{end}, ZERTSIG_{CA_1}) * f(t, T_{begin}, T_{end}) * \propto_{SIG}$

$REV_{CA_1} = (ZERT_{ID1} \ldots ZERT_{IDn}, T_{begin}, T_{end}, REVSIG_{CA_1}) * f(t, T_{begin}, T_{end}) * \propto_{CA_1}$

$REV_{CA_2} = (ZERT_{ID1} \ldots ZERT_{IDn}, T_{begin}, T_{end}, REVSIG_{CA_2}) * f(t, T_{begin}, T_{end}) * \propto_{CA_2}$

a) Logischer Beweis von VER $\models$ SIG $\vdash$ DATA über Cross-Zertifizierung

1. Umformung

Mit dem Vertrauen in die cross-zertifizierte Wurzel-CA $VER \models CA_2 \Rightarrow (\forall X)_{CA_2}$ und der Wurzelregel wird die Kenntnis des öffentlichen Schlüssels der Vertrauensquelle CA_2 abgeleitet.

$$\frac{VER \models CA_2 \Rightarrow (\forall X)_{CA_2},\ VER \lhd ZERT_{CA_2}}{VER \models \xrightarrow{K_{CA_2}} CA_2}$$

2. Umformung

Die Ableitung der Kenntnis des öffentlichen Schlüssels von CA_1 erfolgt über das Cross-Zertifikat in Verbindung mit der Regel zur Zertifizierung. Die Regel zur Zertifizierung gilt für Aussteller-, Teilnehmer- oder Mehrfachzertifikate.

$$\frac{VER \models \xrightarrow{K_{CA_2}} CA_2,\ VER \lhd ZERT_{CROSS_CA_1}}{VER \models \xrightarrow{K_{CA_1}} CA_1}$$

Die weiteren Umformungen mit der Regel zur Bedeutung von Nachrichten und der Regel für Hash-Funktionen sind identisch mit dem Beweis von VER∈SIG∼DATA ohne Cross-Zertifizierung. Aus diesem Grund wird für den vollständigen Beweis auf Anhang B3 verwiesen.

b) Zeitliche Betrachtung zur Cross-Zertifizierung

Die zeitliche Betrachtung für den Pfad über Cross-Zertifizierung gestaltet sich analog den Beispielen aus Kapitel 5 und 6. Es ist allerdings darauf zu achten, dass die Gültigkeitszeiträume aufeinander abgestimmt sind. Das Cross-Zertifikat sollte nicht länger gültig sein, als das Austellerzertifikat der herausgebenden Instanz. Zusätzlich können auch die Gültigkeitszeiträume in alternativen Pfaden berücksichtigt werden. Dabei wird der Gültigkeitszeitraum des Cross-Zertifikates stark von dem gewünschten Zweck der Cross-Zertifizierung abhängen. Ein Cross-Zertifikat kann zum Beispiel die Funktion eines Verlängerungszertifikates für eine CA übernehmen, wenn eine Zertifizierungsstelle beabsichtigt, ihre Dienste nach einer gewissen Zeit einzustellen. Eine andere Variante ist die Vordatierung eines Cross-Zertifikates, um für die Überprüfung von bereits erstellten Signaturen einen neuen Pfad zu ermöglichen. Eine mögliche Überlagerung der Gültigkeitszeiträume zu dem hier vorgestellten Beispiel ist in Abbildung 7.2 dargestellt.

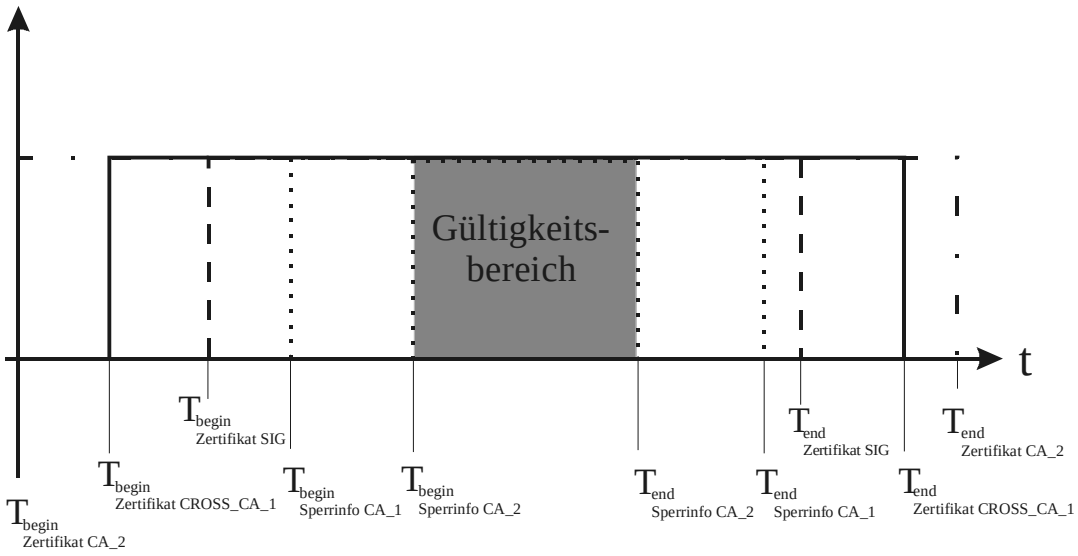

Abbildung 7.2 *Mögliche Gültigkeitsbereiche im Beispiel zur Cross-Zertifizierung*

c) Qualifikation unterschiedlicher Pfade

Die Gewichtung von SigBAN-Aussagen mit Wahrscheinlichkeitswerten eignet sich für den Vergleich der verschiedenen Pfade bei der Cross-Zertifizierung. Aufgrund der unterschiedlichen Wege für den Vertrauensfluss entstehen verschiedene Sicherheitsklassen, die bei der Bewertung des Verifikationsergebnisses zu berücksichtigen sind. Die SigBAN-Logik liefert dafür die technische Entscheidungshilfe. Die praktische Umsetzung dieser Sicherheitsklassen beinhaltet neben technischen auch viele organisatorische Aspekte, die nicht Bestandteil einer Beschreibungssprache zur Verifikation sein können.

Wahrscheinlichkeitsbetrachtung zur Cross-Zertifizierung:

$$\propto_{gesamt} = \propto_{SIG} * \propto_{CROSS_CA_1} * \propto_{CA_2} * \propto_{REV_CA_1} * \propto_{REV_CA_2}$$

Im hier besprochenen Fall sind einige Wahrscheinlichkeitswerte „1“, da es sich bei den cross-zertifizierten CAs um Wurzel-CAs handelt, deren Aussagen zu 100% vertraut wird (siehe Kapitel 5.5.4a).

$$\propto_{CA_1} = \propto_{REV_CA_1} = \propto_{CA_2} = \propto_{REV_CA_2} = \propto_{CROSS_CA_1} = 1$$

somit gilt hier: $\propto_{gesamt} = \propto_{SIG}$

In dem hier vorgestellten Beispiel von cross-zertifizierten Wurzelinstanzen ist bemerkenswert, dass der Wahrscheinlichkeitswerte $\propto_{gesamt}$ unabhängig vom Pfad über das Cross-Zertifikat ist. Dies ist immer bei einer einstufigen Zertifizierungshierarchie mit cross-zertifizierten Wurzelinstanzen der Fall, da nach dem ersten Axiom der SigBAN-Logik Wurzelinstanzen generell vertraut wird und ihren Aussagen der Wahrscheinlichkeitswert „1“ zugewiesen ist.

Eine Cross-Zertifizierung kann auch zwischen CAs, die keine Wurzelinstanzen sind, stattfinden. Mit obiger Wahrscheinlichkeitsbetrachtung wird deutlich, dass in diesem Fall ein Weg über die Cross-Zertifizierung andere Wahrscheinlichkeitswerte berücksichtigen muss. Dies wird in den meisten Anwendungen zu einer anderen Einstufung des Pfades über Cross-Zertifikate als beim „normalen“ Zertifizierungspfad führen. Es kann zum Beispiel ein höherer Wahrscheinlichkeitswert erreicht werden, wenn durch ein Cross-Zertifikat der Weg zur Wurzel-CA verkürzt wird. Abhängig von dem gewählten Vertrauenspfad können durch die Gewichtung mit der SigBAN-Logik unterschiedliche Rollen in einer PKI definiert werden.

Ein Beispiel für den praktischen Einsatz der Wahrscheinlichkeitsbetrachtung kann eine länderübergreifende PKI mit cross-zertifizierten Sub-CAs in den einzelnen Ländern sein. Wird die Unterschrift eines Teilnehmers über Cross-Zertifizierung in einem anderen Land legitimiert, besitzt sie einen geringeren Stellenwert als bei einer Legitimation über die landeseigene Zertifizierungsstelle. Dies kann zum Beispiel dazu führen, dass der Einsatz der eigenen Signaturchipkarte im Inland mehr Möglichkeiten als im Ausland zuläßt.

7.2.3 Attributszertifikate

Durch Attributszertifikate werden rollen- oder anwendungsspezifische Datenelemente (Attribute) mit einem Signaturzertifikat verbunden (siehe X.509 Spezifikation [X50999]). Statt die Attribute direkt im Teilnehmerzertifikat festzuhalten, wird zum Zertifikat oder der Signatur das gerade benötigte Attributszertifikat von einer dazu autorisierten Instanz erstellt. Durch diese Aufteilung wird eine höhere Flexibilität erreicht, als wenn die Attribute direkt im Signaturzertifikat festgehalten werden. Das Attributszertifikat beinhaltet deshalb auch keinen öffentlichen Schlüssel, da dieser sich im zugehörigen Signaturzertifikat befindet. Aus diesem Grund gilt auch die SigBAN-Regel zur Zertifizierung nicht für Attributszertifikate. In der X.509 Spezifikation wird die Struktur von Attributszertifikaten beschrieben, die als Grundlage für die folgende SigBAN-Beschreibung dient.

Signaturzertifikat mit zugehörigem Attributszertifikat:

$ZERT_{ENTITY} = ID_{ENTITY}, CONTENT_{ENTITY}, T_{ENT_begin}, T_{ENT_end}, K_{ENTITY}, ZERTSIG_{CA}$

$ZERT_{ATTRIB}$ = ID_{ENTITY}, $ATTRIBUTES_{ENTITY}$, T_{ATTR_begin}, T_{ATTR_end}, $ZERTSIG_{ENTITY\ oder\ CA}$

Alternatives Signaturzertifikat mit den zusätzlichen Attributen im Zertifikat:

$ZERT_{ENTITY}$ = ID_{ENTITY}, $CONTENT_{ENTITY}$, $ATTRIB_{ENTITY}$, T_{begin}, T_{end}, K_{ENTITY}, $ZERTSIG_{CA}$

Zur oben stehenden SigBAN-Beschreibung des Attributszertifikates ist zu bemerken, das ein öffentlicher Schlüssel kein Bestandteil der Aussage $ZERT_{ATTRIB}$ ist und ein anderer Gültigkeitszeitraum [T_{ATTR_begin}, T_{ATTR_end}] wie im zugehörigen Signaturzertifikat gelten kann. Eine Autorisierung des Attributszertifikates kann durch eine Signatur des Inhabers des Signaturzertifikates $ZERTSIG_{ENTITY}$ oder einer Zertifizierungsstelle $ZERTSIG_{CA}$ im Pfad vom zugehörigen Signaturzertifikat zur Wurzel-CA erfolgen.

Mit Attributszertifikaten kann eine rollenbasierte Rechtevergabe realisiert werden. Dadurch wird die Flexibilität der PKI bei der Anpassung an Betriebsabläufe erhöht. Ein Anwendungsbeispiel ist die Ergänzung von Signaturzertifikaten in einer Firmen-PKI um tätigkeitsbezogene Attributszertifikate. Eine andere Anwendung ist die Erweiterung einer digitalen Signatur um ein Attributszertifikat zur Autorisierung bestimmter Aufgaben. Ein Bankkunde könnte einer signierten Überweisung ein Attributszertifikat beifügen, worin die Verfügungsgewalt für sein Konto bestätigt wird. Dadurch kann eine automatische Verarbeitung auf Seiten der Bank gesteuert werden.

Aus Sichtweise der SigBAN-Logik wird durch Attributszertifikate eine weitere Vertrauenssenke eingeführt, deren Pfad zur Vertrauensquelle zu überprüfen ist. Da das Attributszertifikat an ein anderes Zertifikat gebunden ist, wird ein Teil des Beweises bereits bei der Überprüfung dieses Zertifikates durchgeführt. Lediglich der Vertrauenspfad des Attributszertifikates bis zur Ausstellerinstanz ist zu verifizieren (siehe Abbildung 7.3).

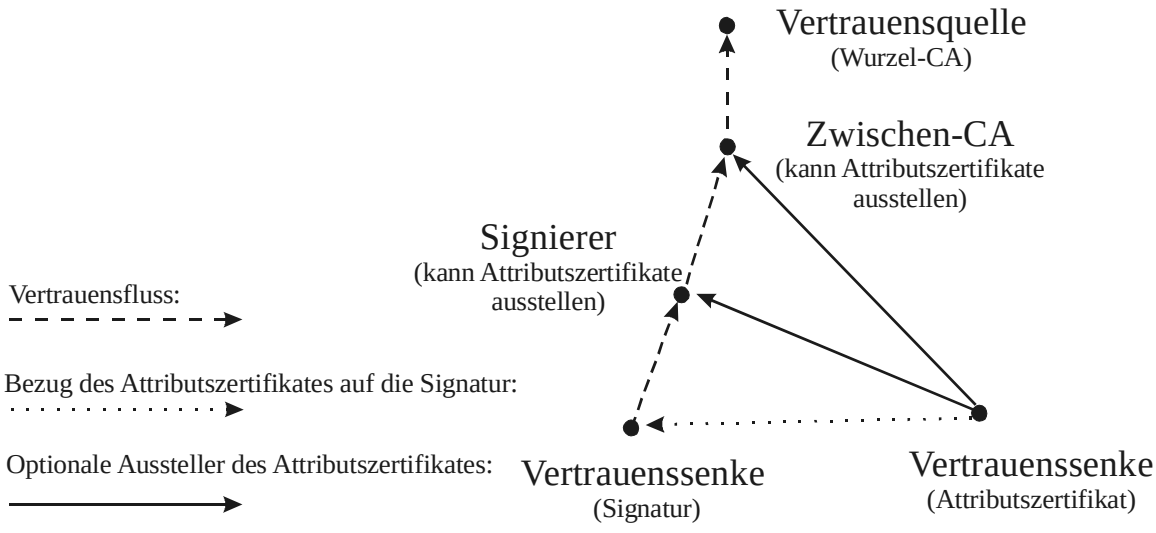

Abbildung 7.3 *Zusammenführung des Vertrauenspfades bei Attributszertifikaten*

Für ein Attributszertifikat ist nur ein Gültigkeitszeitraum, der kürzer oder gleich dem Gültigkeitszeitraum des im Bezug stehenden Zertifikates ist, sinnvoll. Deshalb ist nach einer erfolgreichen zeitlichen Überprüfung des zugehörigen Signaturzertifikates beim Attributszertifikat nur mehr der eigene Gültigkeitszeitraum zu betrachten.

7.3 SigBAN-Betrachtung von Vertrauensmodellen

Eine wichtige Aufgabe bei der Modellierung der Verifikation von digitalen Signaturen ist die Beschreibung des Vertrauensflusses. In der SigBAN-Logik existieren deshalb eigene Regeln, die eine Delegation des Vertrauens beschreiben. In diesem Kapitel wird ein Überblick über die einer PKI zugrundeliegenden Vertrauensmodelle gegeben. Die hier vorgestellten Modelle haben weitreichende Implikationen auf die Zuverlässigkeit, Sicherheit und Verfügbarkeit der PKI-Dienste. Bevor eine PKI näher beschrieben werden kann, ist die Festlegung auf ein Vertrauensmodell notwendig.

7.3.1 Strikte Hierarchie

Eine strikt hierarchische Baumstruktur mit einer Vertrauensquelle als Wurzel ist die häufigste Form einer PKI. Wie in Kapitel 6 erläutert, beruht das deutsche Signaturgesetz [SIG01] auf einer solchen Anordnung mit einer zweistufigen Hierarchie. Auch die meisten lokal begrenzten PKIs in Firmen weisen einen derartigen Aufbau auf.

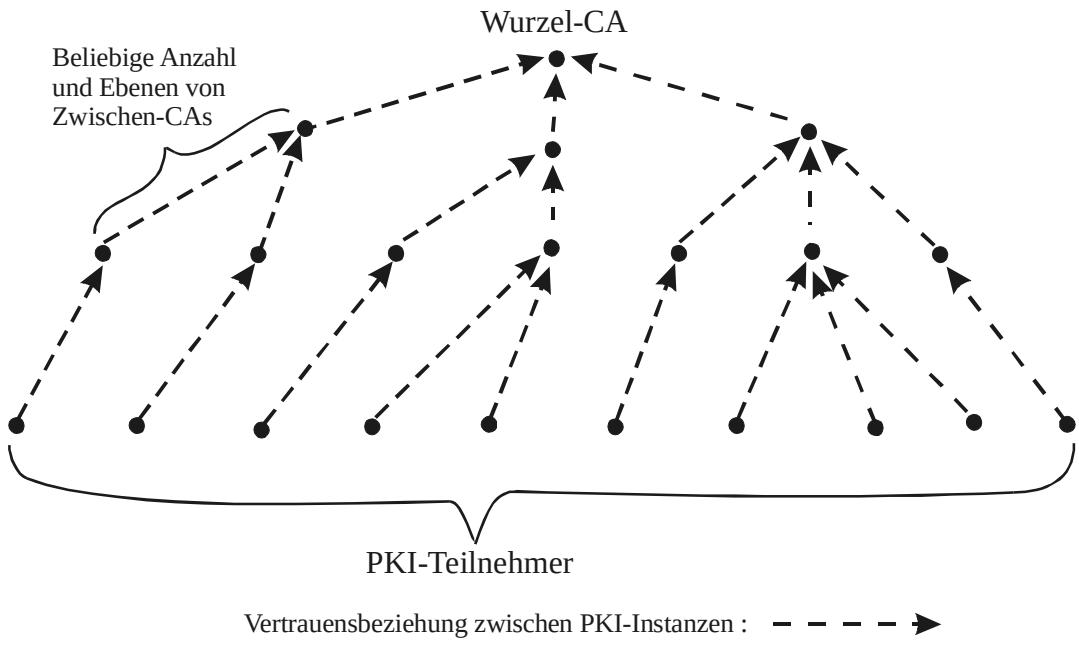

Abbildung 7.4 *Strikte Hierarchie von Vertrauensbeziehungen*

Die PKI-Teilnehmer müssen nicht unbedingt, wie in Abbildung 7.4 dargestellt, alle auf derselben Ebene liegen. Es besteht auch die Möglichkeit, dass jede Zwischen-CA oder sogar Wurzel-CA End-Entitäten autorisiert.

a) Schema zur Verifikation mit SigBAN in einer Baumstruktur

Die Verifikation einer digitalen Signatur mit der SigBAN-Logik wird in einer Baumstruktur immer nach folgendem Schema ablaufen:

1. Der Wurzel-CA wird nach dem ersten Axiom der SigBAN-Logik uneingeschränkt vertraut. Dieser Vertrauensanker wird üblicherweise über einen Weg, der außerhalb der SigBAN-Betrachtung liegt, etabliert. Dazu eignet sich zum Beispiel eine Veröffentlichung des Hash-Wertes des Wurzel-Zertifikates in einer Zeitung oder eine telefonische Übermittlung des öffentlichen Root-Schlüssels.

2. Mit dem Nachwies der Kenntnis des öffentlichen Root-Schlüssels über die Wurzelregel beginnen die SigBAN-Umformungen. Die Voraussetzungen für diese Ableitung sind das uneingeschränkte Vertrauen in die Aussagen der Wurzelinstanz und die Verfügbarkeit des Wurzelzertifikates.

3. Im nächsten Schritt ist der Vertrauensfluss bis zum Zertifikat der End-Entität zu überprüfen. Dies geschieht durch wiederholte Anwendung der Regel zur Zertifizierung. Dafür ist das Vorliegen aller Zertifikate und Sperrinformationen im Vertrauenspfad notwendig.

4. Für die endgültige Überprüfung der Signatur sind abhängig von der Struktur der signierten Daten weitere SigBAN-Regeln, wie die Regel zur Bedeutung von Nachrichten oder die Regel für Hash-Funktionen anzuwenden.

Wie im Beispiel zum Signaturgesetz aus Kapitel 5 gezeigt, ist der Nachweis der logischen Gültigkeit der Signatur bei einer strikten Hierarchie analog dem oben vorgestellten Schema durchzuführen. Nicht schematisieren lässt sich die vollständige Formulierung aller Anforderungen an die digitale Signatur in Form von SigBAN-Aussagen. Der Umfang der Startbedingung kann durch die Berücksichtigung aller praktischen Anforderungen, die an eine digitale Signatur gestellt werden, erheblich zunehmen. Dazu gehört neben dem bereits vorgestellten qualifizierten Zeitstempel, die Berücksichtigung einer signierten CP (Certificate Policy) bzw. CPS (Certificate Practice Statement) oder eine vollständige Modellierung des Zertifikatswiderrufs auf allen Ebenen.

7.3.2 Verteiltes Vertrauen

Neben der strikten Hierarchie mit einer Wurzelinstanz gibt es auch die Möglichkeit von mehreren Vertrauensquellen, die untereinander in Verbindung stehen. Wie solche Konstellationen mit verteiltem Vertrauen aufgebaut werden und welche Eigenschaften sie aufweisen, wird im Folgenden beschrieben.

a) Cross-Zertifizierung von unabhängigen Wurzelinstanzen

Eine Möglichkeit Verbindungen zwischen unabhängigen Wurzelinstanzen zu schaffen, ist eine gegenseitige Cross-Zertifizierung zwischen den einzelnen Wurzel-CAs. Die Cross-Zertifizierung zwischen zwei CAs wurde bereits in Kapitel 7.2.1 besprochen. Hier wird allgemein die Verknüpfung beliebig vieler Wurzelinstanzen betrachtet. Dabei können entweder alle Wurzel-CAs untereinander verbunden sein (volle Vermaschung), oder nur einzelne Wurzel-CAs (teilweise Vermaschung) interagieren. Eine sinnvolle Alternative ist auch eine ringförmige Cross-Zertifizierung. Dadurch wird mit einer minimalen Anzahl an Verbindungen bereits ein Weg in jede Vertrauens-Domäne geschaffen. Wie aus Abbildung 7.5 ersichtlich, werden zur Verbindung von n Wurzel-CAs $s = n^2-n$ Cross-Zertifikate bei voller Vermaschung benötigt.

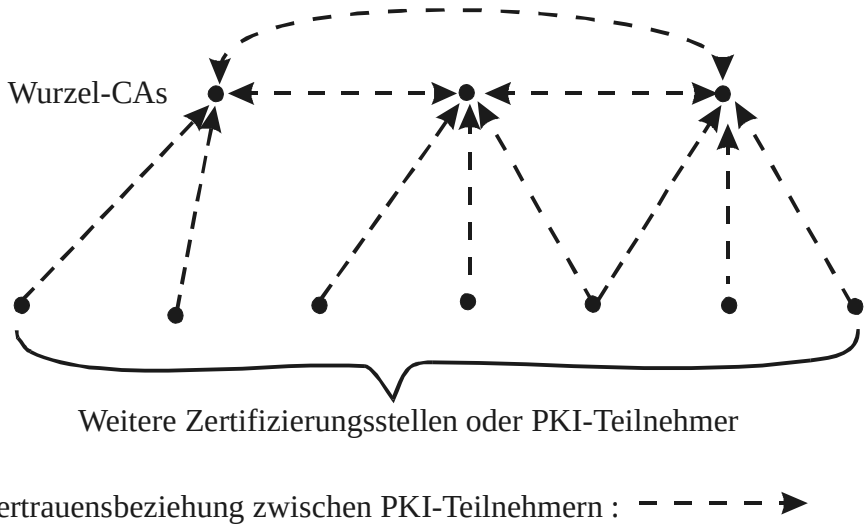

Abbildung 7.5 *Volle Vermaschung von drei unabhängigen Wurzel-CAs*

Abhängig vom Grad der Vermaschung und der ausgewählten Vertrauensquelle ist auch der Wahrscheinlichkeitswert, der dem Verifikationsergebnis mit der SigBAN-Logik zugewiesen wird. Durch die Gewichtung der einzelnen Vertrauensbeziehungen mit Wahrscheinlichkeitswerten wird in einem Vertrauensnetzwerk eine Differenzierung mehrerer logisch richtiger Ergebnisse erst möglich.

In dem Fall einer direkten Cross-Zertifizierung besteht eine höhere Flexibilität als bei der im Anschluss betrachteten Alternative über eine Bridge-CA. Die Regeln für eine gegenseitige Zertifizierung können bei der gerade vorgestellten Lösung für jede Verbindung zwischen den Wurzel-CAs frei definiert werden. Während bei der folgenden Konfiguration die Regeln von der Bridge-CA fest vorgegeben werden.

b) Konfiguration mit einer Bridge-CA

Ein Vertrauensnetzwerk kann auch mit Hilfe einer zentralen Bridge-CA realisiert werden. Statt einer gegenseitigen Zertifizierung der Wurzelinstanzen untereinander, wird jede Wurzel-CA mit der Bridge-CA verbunden (siehe Abbildung 7.6). Dadurch wird die Anzahl der Verbindungen s auf s = 2n (n ist die Anzahl der Wurzelinstanzen) reduziert. Diese Reduzierung geht allerdings auf Kosten der Redundanz und somit der Verfügbarkeit des Vertrauensnetzwerks. Bei einem Ausfall der Bridge-CA ist ein Übergang in eine andere Trust Domain nicht mehr möglich. Obwohl für eine Zertifizierungsstelle eine ständige Verfügbarkeit nicht unbedingt erforderlich ist, sind bestimmte Aufgaben in regelmäßigen Abständen durchzuführen. Eine abgelaufene Sperrliste der Bridge-CA würde zu einem Ausfall der Überbrückungsinstanz führen und somit einen Wechsel in andere Vertrauens-Domänen verhindern.

Die Anbindung der Wurzel-CAs an die Bridge-CA kann auch auf Cross-Zertifikaten beruhen. Ein anderer Weg ist die Verwendung von Vertrauenslisten. Für die Verwendung von Cross-Zertifikaten wurde der Beweis mit der SigBAN-Logik in Kapitel 7.2.2 geführt. Abhängig von der Gestaltung der Vertrauenslisten ist der SigBAN-Beweis ähnlich zu gestalten. Auf alle Fälle ist ein gegenseitiger Vertrauensfluss zwischen Bridge-CA und Wurzel-CA nachzuweisen. Dies ist nicht immer möglich, wie im nächsten Beispiel gezeigt wird.

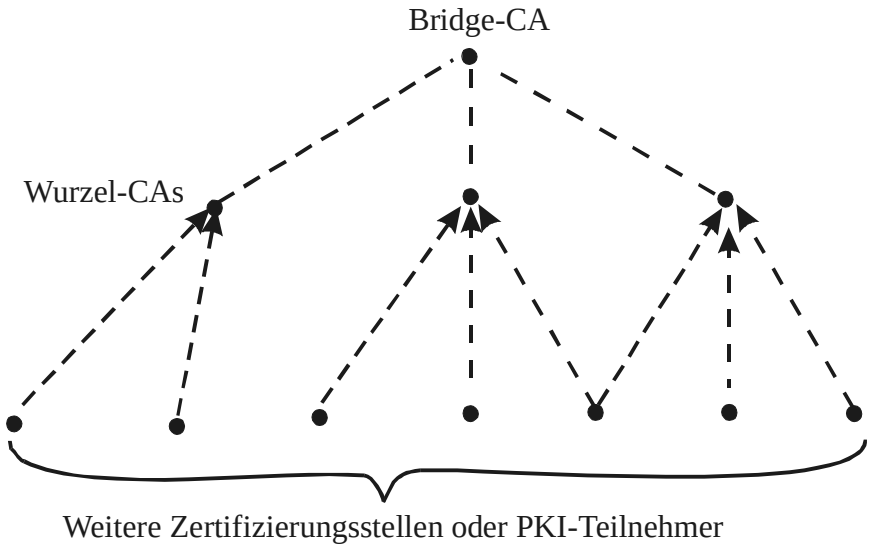

Abbildung 7.6 *Vertrauensbrücke durch eine Bridge-CA*

c) Analyse der Realisierung der Bridge-CA Initiative

Die Realisierung der Bridge-CA Initiative von Wirtschaftsunternehmen, die eigene PKIs betreiben ([BRIBR01],[BRIRO01]), basiert auf einer sogenannten „Trust List“. In dieser Vertrauensliste werden die Zertifikate der einzelnen Wurzel-CAs mit dem Bridge-CA Zertifikat zusammengefasst, von der Bridge-CA signiert und veröffentlicht. Eine Beschreibung dieser Vertrauensliste als SigBAN-Aussagen lässt sich folgendermassen formulieren:

$TLIST_{BRIDGE} = CONTENT_{TLIST}, SIG_{BRIDGE}$

$SIG_{BRIDGE} = \{H(CONTENT_{TLIST})\}_{K^{-1}_{BRIDGE}}$

$CONTENT_{TLIST} = ZERT_{BRIDGE}, ZERT_{ROOT_CA_1}, ZERT_{ROOT_CA_2}, ZERT_{ROOT_CA_3}, ...$

Eine Aussage, die hier im Zusammenhang mit der Bridge-CA BRIDGE bewiesen wird, ist das Vertrauen der teilnehmenden Wurzel-CAs ROOT_CA_X in den Inhalt der Vertrauensliste $CONTENT_{TLIST}$ der Bridge-CA BRIDGE:

$$ROOT_CA_X \models BRIDGE \mid\sim TLIST_{BRIDGE}$$

Die Kenntnis des öffentlichen Schlüssels der Bridge-CA K_{BRIDGE} wird dabei als bekannt angesehen, da den teilnehmenden Instanzen das Zertifikat der Bridge-CA $ZERT_{BRIDGE}$ vorliegt und über einen auf der Homepage veröffentlichen elektronischen Fingerabdruck (http://www.bridge-ca.org) überprüfbar ist. Mit der Signatur der „Trust List“ durch die Bridge-CA BRIDGE und der Regel zur Bedeutung von Nachrichten kann das Vertrauen in den Inhalt der „Trust List“ $CONTENT_{TLIST}$ für die Wurzel-CA ROOT_CA_X nachgewiesen werden:

$$\frac{ROOT_CA_X \models \xrightarrow{K_{BRIDGE}} BRIDGE,\ ROOT_CA_X \lhd \{H(CONTENT_{TLIST})\}_{K^{-1}_{BRIDGE}}}{ROOT_CA_X \models BRIDGE \mid\sim CONTENT_{TLIST}}$$

Ein weitaus wichtigerer Nachweis für die Verifikation einer digitalen Signatur ist das Vertrauen des PKI-Teilnehmers VER in die Aussage $TLIST_{BRIDGE}$ der Bridge-CA. Dies ist eine Voraussetzung, um in einen anderen Zweig zu gelangen und mit den PKI-Teilnehmern in einer anderen Wurzel-CA interagieren zu können:

$$VER \models BRIDGE \mid\sim TLIST_{BRIDGE}$$

Mit dem ersten Axiom der SigBAN-Logik vertraut der PKI-Teilnehmer nur seiner Wurzel-CA ROOT_CA_X. Es gilt also die Startbedingung:

$$VER \models ROOT_CA_X \Rightarrow (\forall X)_{ROOT_CA_X}$$

Da die Trust List nicht von der eigenen Wurzel-CA durch eine Signatur legitimiert ist, gelingt der Nachweis des Vertrauens für den PKI-Teilnehmer in diese Aussage der Bridge-CA nicht. Ein Vertrauensfluss ist somit über die Bridge-CA nicht möglich. Es kann zwar angenommen werden, dass die Bridge-CA nur Zertifikate von Wurzel-CAs, die dies wünschen, veröffentlicht, ein mit der SigBAN logisch überprüfbarer Nachweis existiert dafür jedoch nicht.

Die einzige Möglichkeit den gegenseitigen Vertrauensfluss zu etablieren, wäre eine Signatur der „Trust List“ durch alle Wurzel-CAs.

Die TeleTrust Bridge-CA signiert die Zertifikate der teilnehmenden Wurzel-CAs in Form einer „Trust List“. Eine Autorisierung der Bridge-CA durch die Wurzel-CAs findet aber nicht statt. Deshalb fehlt eine notwendige „Gegenverbindung“ um einen Vertrauensfluss zu etablieren, der den Übergang von einer Vertrauens-Domäne in eine andere ermöglicht.

7.4 Grenzen der SigBAN-Logik

Die Grenzen der SigBAN-Logik werden im wesentlichen durch das erste und zweite Axiom definiert. Konstellationen, die Sicherheit auf Basis von Public-Key-Kryptographie bieten, und sich nicht an diese beiden Axiome halten, können nicht mit der SigBAN-Logik untersucht werden. Es ist auch nicht Ziel des SigBAN-Entwurfs, eine Logik für eine allgemeine Sicherheitsanalyse bereitzustellen. Die SigBAN-Logik soll eine Beschreibungssprache zur Verifikation von digitalen Signaturen definieren. Dabei ist ein wichtiges Kriterium die Möglichkeit durch eine vollständige SigBAN-Analyse die Gleichwertigkeit von digitaler Signatur und manueller Unterschrift herzustellen.

7.4.1 Nicht überprüfbare Anordnungen

Um den Entwurf der SigBAN-Logik abzurunden, werden hier einige Konstellationen vorgestellt, die sich nicht mit der SigBAN-Logik beschreiben lassen. In beiden Beispielen wird auf unterschiedliche Weise ein Vertrauensmodell umgesetzt, das den Zielsetzungen der SigBAN-Logik widerspricht. Im zweiten Fall ist dies ein konkreter Widerspruch zum ersten Axiom, während im ersten Beispiel unterschiedliche Sicherheitsphilosophien vorliegen.

a) Web-Modell

Die Bezeichnung „Web Modell“ leitet sich vom Einsatz dieses PKI-Modells in Internet-Browsern, wie dem Netscape Navigator oder Microsoft Internet Explorer ab (siehe [MAR01], [MIC99]). Es handelt sich dabei um voneinander unabhängige strikte Vertrauenshierarchien, die wie in Abbildung 7.7 dargestellt, über den Import von Wurzel-Zertifikaten in dem Browser etabliert werden. Die Verifikation einer digitalen Signatur wird dann über einen der im Browser verankerten Vertrauenspfade durchgeführt.

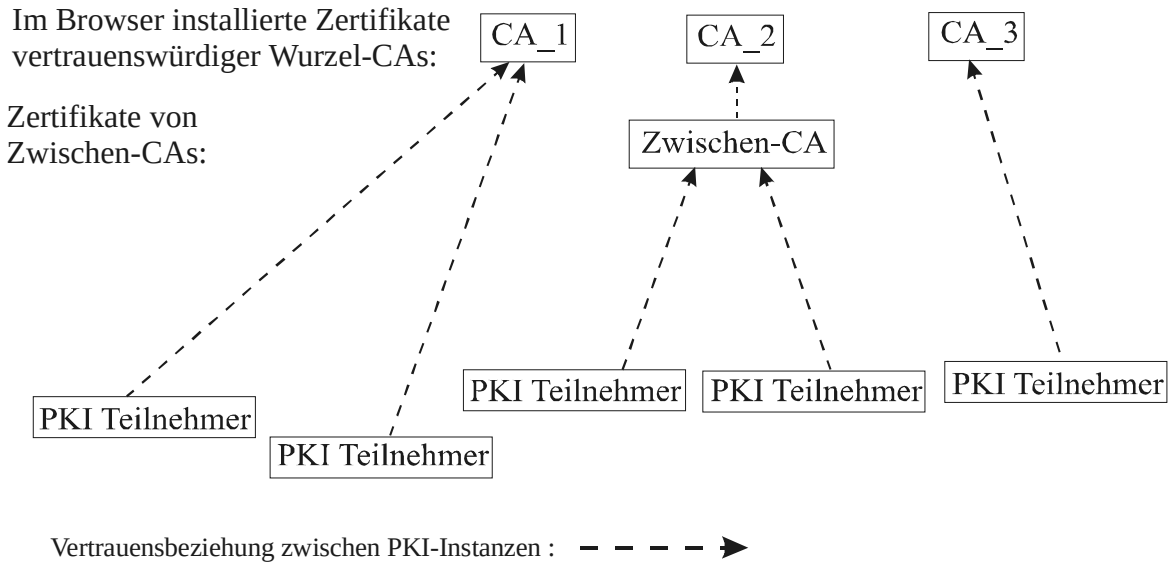

Abbildung 7.7 *Beliebige Wurzelinstanzen können im Browser etabliert werden*

Bei dem Einsatz von digitalen Signaturen nach dem Web-Modell wird die Integrität von Internetseiten und eine gesicherte Übertragung im Internet erreicht. Die häufigste Anwendung ist dabei die gesicherte Kommunikation zwischen Browser und Internet-Server über das HTTPS-Protokoll. Es ist nicht die Intention dieses Ansatzes digitale Signaturen auf dem Niveau einer manuellen Unterschrift zu realisieren. Aufgrund der unterschiedlichen Sicherheitsphilosophien eignet sich die SigBAN-Logik deshalb auch nur bedingt für die Analyse von PKIs nach dem Web-Modell. Dabei sind wesentliche Differenzen:

- **Eine Langzeitüberprüfbarkeit ist im Web-Modell nicht gefordert.** Die SigBAN-Logik berücksichtigt insbesondere Aspekte bei der Verifikation von digitalen Signaturen lange nach deren Erstellung. Für das Web-Modell ist die Gültigkeit einer digitalen Signatur nur für die aktuelle Browser Sitzung relevant.
- **Bei allen derzeitigen Realisierungen nach dem Web-Modell werden bei der Signaturprüfung nicht alle notwendigen Informationen berücksichtigt.** Bei einer SigBAN-Beschreibung werden erst die Erwartungen an eine digitale Signatur formuliert. Anschließend erfolgt ein Beweis mit SigBAN-Umformungen unter Berücksichtigung aller an der Verifikation beteiligten Aussagen. Diese Einbettung der Signaturprüfung in einen gesamten Kontext unter Berücksichtigung aller für die Signaturprüfung notwendigen Informationen ist in keinem Browser realisiert. In vielen Fällen findet nicht einmal eine Auswertung der notwendigen Sperrinformationen einer Zertifizierungsstelle statt.
- **Im Web-Modell erfolgt keine explizite Zuweisung des Vertrauens.** Der Hauptkritikpunkt am Web-Modell aus SigBAN-Sicht ist die fehlende explizite Zuweisung des Vertrauens in eine Wurzelinstanz. Dies ist zwar kein direkter Widerspruch zum ersten Axiom der SigBAN-Logik, aber für das darin festgelegte Vertrauen in die Wurzel-CA ist eine von allen PKI-Teilnehmern akzeptierte Autorisierung der Vertrauensquellen

notwendig. Um eine verläßliche Vertrauensquelle für digitale Signaturen zu etablieren, muss der derzeit nicht autorisierte Import von Wurzel-Zertifikaten durch den Browser-Hersteller technisch und rechtlich abgesichert werden.

Obwohl das Web-Modell nicht direkt im Widerspruch zur SigBAN-Logik steht, ist die SigBAN-Analyse einer Verifikation nach dem Web-Modell aufgrund der unterschiedlichen sicherheitstechnischen Anforderungen nicht sinnvoll. Wird der Einsatz des Web-Modells zur Bearbeitung von digitalen Signaturen für Dokumente erweitert, kommen alle oben genannten Kritikpunkte zum tragen. Dies ist zum Beispiel beim CSP (Cryptographic Service Provider) im Betriebssystem Windows 2000 der Fall (siehe [WIN98]). Um PKI-Dienste in Betriebssystemen nach dem Web-Modell zu etablieren, sollte eine einheitliche, herstellerunabhängige Vorgehensweise bei der Verifikation von digitalen Signaturen festgelegt werden. Ein allgemeingültiges Verifikationsmodell ließe sich bei den Herstellern der verschiedenen Betriebssysteme und Browser allerdings kaum durchsetzen.

b) Gegenseitiges Vertrauen (PGP)

Bei dieser Konstellation gibt es keine zentrale Vertrauensquelle. Stattdessen entscheidet jeder PKI-Teilnehmer selbst, welchem anderen PKI-Teilnehmer er vertrauen möchte. Er etabliert somit seine eigenen Vertrauensquellen aufgrund seiner unter Umständen subjektiven Kriterien. Dieser offene Ansatz bietet dann Vorteile, wenn ein pragmatisches Sicherheitskonzept in einer kleinen Benutzergruppe gefordert ist. Mit relativ wenig Aufwand kann zum Beispiel eine schnelle und flexible Absicherung der Kommunikation per E-Mail erreicht werden. Als technische Grundlage für eine rechtsgültige Signatur ist diese PKI-Variante jedoch ungeeignet.

Das bekannteste Beispiel zu diesem Vertrauensmodell ist PGP (Pretty Good Privacy) von Phil Zimmermann (siehe [ZIM95,[GAR96]). PGP-Implementierungen werden in vielen Firmen lokal begrenzt eingesetzt, um eine Absicherung des E-Mail Verkehrs mit hochwertigen Krypto-Algorithmen zu erreichen.

Es gibt bei der Anordnung aus Abbildung 7.8 keine Wurzelinstanz auf die das erste Axiom der SigBAN-Logik anwendbar ist. Deshalb kann in einem System, das auf gegenseitigem Vertrauen beruht, die SigBAN-Logik nicht eingesetzt werden.

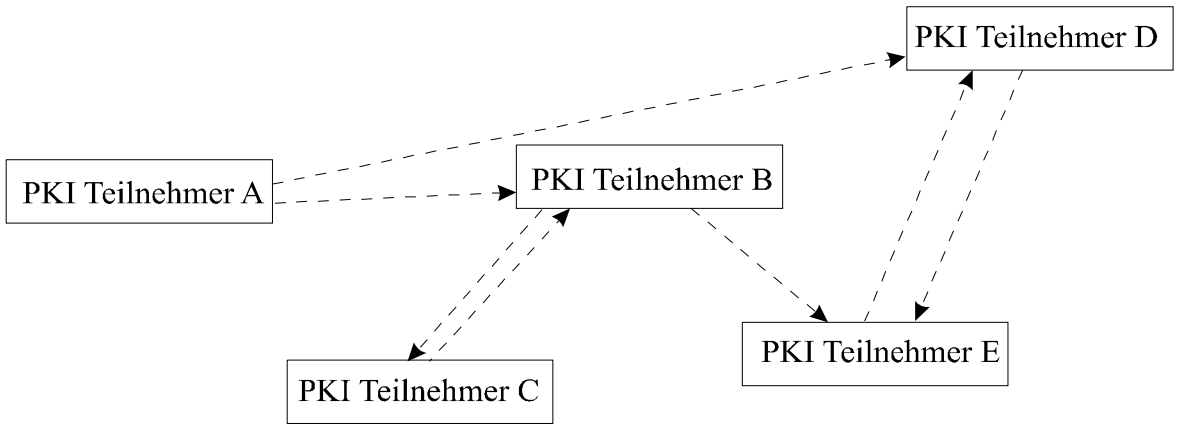

Abbildung 7.8 *Gegenseitiges Vertrauen*

7.5 Abschließende Kommentare

In diesem Kapitel wurden die unterschiedlichsten Varianten von PKIs mit der SigBAN-Logik modelliert oder auf die Möglichkeiten einer SigBAN-Analyse hin untersucht. Dabei wurde festgestellt, dass unabhängig von der Ausprägung einer PKI die Verifikation einer digitalen Signatur in den meisten Fällen mit der SigBAN-Logik formulierbar ist. Die SigBAN-Beschreibung ist dabei nicht an Datenformate oder Prozesse gebunden, sondern kann flexibel angepasst werden, wie anhand von Mehrfach- und Attributszertifikaten gezeigt wurde. Die Voraussetzung dafür ist, dass der fundamentale Aufbau einer PKI die Anwendung der zwei Axiome der SigBAN-Logik zuläßt.

Bei der Betrachtung der einzelnen Modelle zu Verteilung des Vertrauens wurde deutlich, dass ein wichtiger Aspekt bei der Überprüfung einer digitalen Signatur der Vertrauensfluss zur Wurzelinstanz ist. Jede Beschreibung einer Verifikation muss diesen Vertrauensfluss berücksichtigen. In der SigBAN-Logik geschieht dies durch spezielle Regeln (siehe erstes bis drittes Theorem aus Kapitel 5.5.2), die ergänzend zur BAN-Logik definiert wurden. Deshalb wurde in einem eigenen Abschnitt der Einsatz der SigBAN-Regeln im Zusammenhang mit den einzelnen Vertrauensmodellen vorgestellt.

Abschliessend wurde anhand von Beispielkonstellationen gezeigt, dass die Grenze der SigBAN-Logik bei PKIs liegt, die eine zu den SigBAN-Axiomen widersprüchliche Sicherheitsphilosophie verfolgen. Ziel der SigBAN-Logik ist es, die Verifikation einer digitalen Signatur auf hohem Niveau äquivalent zu einer manuellen Unterschrift zu modellieren. Bei Systemen, die eine Datenübertragung absichern (Web-Modell) oder in denen jeder Anwender seine eigenen Regeln aufstellen kann (Gegenseitiges Vertrauen), ist dies nicht der Fall.

8. Abschließende Kommentare und Ausblick

In der vorliegenden Arbeit wurde die SigBAN-Logik als Beschreibungssprache zur Verifikation von digitalen Signaturen vorgestellt. Die Anwendung der SigBAN-Logik wurde anhand mehrerer Beispiele in unterschiedlichen PKIs demonstriert. Dabei lag besonderes Augenmerk auf einem SigBAN-Modell zur Verifikation von digitalen Signaturen nach dem deutschen Signaturgesetz. Mit Hilfe der SigBAN-Logik konnte in diesem Modell ein verbesserter Vorschlag für einen Zeitstempeldienst zur sicheren Datierung des Erstellungszeitpunktes der digitalen Signatur erarbeitet werden. Darüber hinaus wurden durch die zeitliche Betrachtung mit der SigBAN-Logik Probleme bei den Gültigkeitsmodellen des Signaturgesetzes erkannt.

Der Entwurf der SigBAN-Logik teilt sich in mehrere Schritte auf. Zuerst mussten die Anforderungen einer Beschreibungssprache für die Verifikation von digitalen Signaturen ermittelt werden. Im nächsten Schritt wurden dann die Möglichkeiten der BAN-Logik für diesen Zweck untersucht. Da sich die BAN-Logik als nur bedingt geeignet herausstellte, wurde die SigBAN-Logik entwickelt. Die Basis der SigBAN-Logik beruht auf drei Axiomen, die komplett unabhängig von der BAN-Logik sind. Auf Basis dieser Axiome wurden mit der Syntax und einem Teil der Regeln der BAN-Logik die Theoreme der SigBAN-Logik entwickelt. Wesentliche Unterschiede zwischen den Sprachen BAN und SigBAN sind:

- Die SigBAN-Logik führt aufgrund der Forderung des ersten Axioms die Überprüfung immer auf eine Vertrauensquelle zurück.
- Die SigBAN-Logik berücksichtigt zeitliche Aspekte auf Basis des zweiten Axioms, während die BAN-Logik keine zeitliche Betrachtung kennt.
- SigBAN definiert eine Möglichkeit zur Quantifizierung von Aussagen, während es bei BAN nur wahr oder falsch gibt.
- BAN berücksichtigt auch die Verschlüsselung von Daten, die bei der Verifikation von digitalen Signaturen keine Rolle spielen.

Die Vorgehensweisen bei der Überprüfung eines Authentisierungsprotokolls mit BAN oder der Verifikation einer digitalen Signatur mit SigBAN sind identisch. Zuerst werden die Ausgangsbedingungen und die zu beweisenden Behauptungen in Form von Aussagen in der Syntax der BAN-Notation formuliert. Durch die Anwendung des BAN- bzw. SigBAN-Regelsatzes können dann die Zielaussagen bewiesen oder widerlegt werden. Die SigBAN-Logik bietet im Anschluss an diese logische Überprüfung noch die Möglichkeit einer zeitlichen Betrachtung und einer Quantifizierung des Ergebnisses. Durch eine Trennung von logischer, zeitlicher und quantitativer Anschauung in der SigBAN-Logik, wird eine einfachere Anwendbarkeit und bessere Transparenz bei der Überprüfung einer digitalen Signatur erreicht.

Die Idee einer Beschreibungssprache basiert auf der praktischen Erfahrung, dass es derzeit sehr wenige Anwendungen gibt, die eine digitale Signatur vollständig verifizieren. Außerdem existieren in den verschiedenen Applikationen unterschiedliche Vorgehensweisen bei der Signaturprüfung. Um die digitale Signatur mit den gleichen Merkmalen wie eine manuelle Un-

terschrift auszustatten, ist jedoch eine einheitliche Sichtweise bei der Verifikation unbedingt notwendig. Mit einer allgemeinsprachlichen Form kann dies allerdings nur schwer erreicht werden. Die SigBAN-Logik ist deshalb als Unterstützung beim Entwurf oder der Analyse der für die Verifikation notwendigen Prozesse gedacht.

8.1 Ausblick

Abschließend werden hier noch einige weitere interessante Anwendungsgebiete und Ausbaumöglichkeiten der SigBAN-Logik kurz vorgestellt.

8.1.1 Weitere Anwendungsgebiete für eine SigBAN-Analyse

In dieser Arbeit wurde versucht, den Einsatz der SigBAN-Logik möglichst umfassend zu demonstrieren. Jedoch würde eine Untersuchung aller möglichen Konstellationen, in denen eine digitale Signatur verifizierbar ist, den Rahmen sprengen.

Ein Bereich, der nur am Rande angesprochen wurde, ist die Modellierung des Widerrufs von Zertifikaten mit der SigBAN-Logik. Für diesen Zweck gibt es die unterschiedlichsten Formate, wie Widerrufslisten, Delta-Widerrufslisten, indirekte Widerrufslisten, Widerrufsbäume oder Positivlisten (siehe Kapitel 8 „Certificate Revocation“ in [CAR99]). Jedes dieser Formate läßt sich als SigBAN-Aussage formulieren und in den Verifikationsprozess integrieren. Eine SigBAN-Analyse zur Betrachtung aller Formen des Widerrufs im Zusammenhang mit der Verifikation von digitalen Signaturen, wäre ein interessantes Thema für eine eigene ausführliche Betrachtung.

Eine Zertifizierungsstelle veröffentlicht die „Spielregeln“ der PKI in Form einer CP (Certificate Policy) und eines CPS (Certificate Practice Statements) nach RFC2527 [CP99]. Eine Modellierung dieser signierten Aussagen im Zusammenhang mit der Verifikation würde auch ein weites Feld für eine Untersuchung mit der SigBAN-Logik ergeben.

Die hier vorgeführten Beispiele verwenden immer Zertifikate nach dem X.509-Format, da dies die verbreiteteste Zertifikatsform ist. Eine Betrachtung der Verifikation von digitalen Signaturen mit anderen Zertifikatsformaten wäre ein neues Anwendungsgebiet für die SigBAN-Logik. Hierzu würde sich unter anderem das von PGP definierte Format anbieten, da PGP-Zertifikate mittlerweile auch in hierarchischen PKI-Modellen eingesetzt werden.

8.1.2 Mögliche Erweiterungen der SigBAN-Logik

Eine Erweiterungsmöglichkeit, die ein sehr weites Spektrum an Anwendungen eröffnet, ist die Kombination von BAN- und SigBAN-Logik zu einer umfassenden Beschreibungssprache für sämtliche Prozesse in einer PKI wie in der folgenden Abbildung 8.1 dargestellt.

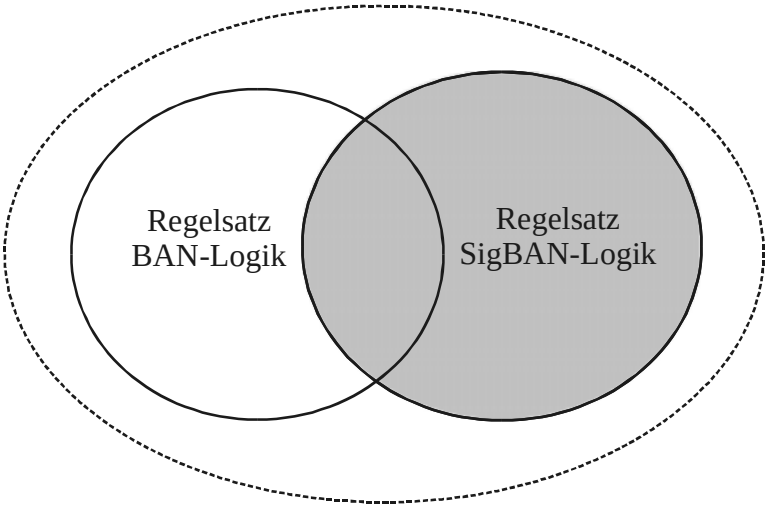

Abbildung 8.1 *Neue Beschreibungssprache für Prozesse in einer PKI*

In der vorliegenden Arbeit wurde ausschließlich die Verifikation von digitalen Signaturen betrachtet, da dies für die Interoperabilität und die weitere Verbreitung von hochwertigen PKIs der zentrale Ansatzpunkt ist. Viele Prozesse, die eine PKI organisieren und administrieren, könnten mit einer solchen Beschreibungssprache modelliert werden. Ein Anwendungsfall wäre zum Beispiel eine komplette Analyse des CMP-Protokolls aus RFC 2510.

Literaturverzeichnis

[BAL00] Baltimore Technologies, *PKI-Plus Developer Guide*, Version 2.2, 2000

[BAN89] M. Burrows, M. Abadin, R. Needham, *A Logic of Authentication*, Digital Equiptment Corp. Systems Research Center, 1989

[BAN90] M. Burrows, M. Abadin, R. Needham, *A Logic of Authentication*, ACM Transactions on Computer Systems, 1990

[BAU99] M. Baum, *Das Gültigkeitsmodell des Signaturgesetzes*, DuD Fachbeiträge Ausgabe 4/1999

[BB94] B. den Boer, A. Bosselaers, *Collisions for the Compression Function of MD5*, Advances in Cryptology-EUROCRYPT ´93, Springer Verlag, 1994

[BRIBR01] TeleTrust, *Bridge-CA Broschüre*, http://www.bridge-ca.org

[BRIRO01] TeleTrust, *Bridge-CA Roadmap*, http://www.bridge-ca.org

[CAR99] Adams Carlisle and Steve Loyd, *Understanding Public Key Infrastructure*, New Riders Publishing, 1999

[CC98] ISO/IEC 15408 *Common Criteria for Information Technology Security Evaluation Part 1-3*, ISO/IEC 1998

[CMP99] RFC 2510, *X.509 Internet Public Key Infrastructure Certificate Management Protocols*, 1999

[CMS00] RFC 2797, *Certificate Management Messages over CMS Policy*, 2000

[CP99] RFC 2527, *X.509 Internet Public Key Infrastructure Certificate Policy and Certification Practices Framework*, 1999

[DAV82] G.I. Davida, *Chosen Signature Cryptoanalysis of the RSA (MIT) Public Key Cryptosystems*, Technical Report of the TR-CS-82-2 Department of EECS, University of Wisconsin 1982

[DD85] DOD 5200.28-STD, *Trusted Computer System Evaluation Criteria*, Department of Defense Library No. S225, 711, Washington 1985

[DH76] W. Diffie, M.E. Hellman, *Multiuser Cryptographic Techniques*, Proceedings of AFIPS National Computer Conference, 1976

[DOR99] N. Doraswamy, D. Harkins, *IPSec-The New Security Standard for the Internet, Intranets and Virtual Private Networks*, Prentice Hall Inc., 1999

[FRI96] Frier, Karlton, Kocher, *The SSL 3.0 Protocol*, Netscape Communications Corp., 1996

[FORD97] Ford Baum, *Secure Electronic Commerce*, Prentice Hall Inc., 1997

[GAA91] Gaarder, Snekkenes, *Applying a formal analysis technique to the CCITT X.509 strong two-way authentication protocol*, Journal of Cryptology, 1991

[GAR95] S. Garfinkel, *PGP: Pretty Good Privacy*, O'Reilly and Associates, 1995

[GER98] Ed Gerck, MCG, *Toward Real-World Models of Trust: Reliance on Received Information*, http://mcg.org.br/trustdef.htm, 1998

[HA00] V. Hammer, Holger Petersen, *Aspekte der Cross-Zertifizierung*, DuD Fachbeiträge Ausgabe 5/2000

[HBCI99] Kurt Haubner, SIX SIGMA EDV Konzepte , *Das HBCI Kompendium*, http://www.hbci.de/download/HBCI_Kompendium_V2.1.pdf, 3/1999

[HOW99] T.A.Howes, M.C. Smith, *LDAP Programming Directory-Enabled Applications with LDAP*, Macmillan Technology Series, 1999.

[ID298] iD2 Technologies, *Technical Description for ID2 Certificate Manager*, Version 1.2, 1998

[ISIS01] ISIS Technical Working Group T7 & TeleTrust, *Common ISIS-MTT Specification for PKI Applications*, Version 1.0, 30. September 2001

[ISIS99] Arbeitsgemeinschaft Trust-Center für digitale Signaturen, *Industrial Signature Interoperability Specification ISIS*, Version 1.2, 1999

[ISO88] ISO IS 7498/2, *Security Architecture*, 1988

[ITU97] ITU-T Recommendation X.509 ,*Information Technology - Open Systems Interconnection - The Directory: Authentication Framework*, 1997

[ITSEC91] *Information Technology Security Evaluation Criteria ITSEC*, Department of Trade and Industry, London 1991

[JAC00] G. Jacobson, *Formaler Entwurf effizienter Authentisierungsprotokolle mit Schlüsselabsprache*, Lincom Europe, München 2000

[LAI00] Laing, Baltimore Technologies, *Attribute certificates - a new initiative in PKI technology*, www.baltimore.com/library/whitepapers/acswp-hm.htm, 2000

[LAR00] J. Larmouth, *ASN.1 Complete*, Morgan Kaufmann Publishers, 2000

[MAU94] Ueli Maurer, ETH Zürich, *Towards the Equivalence of Breaking the Diffie-Hellmann Protocol and Computing Discrete Logarithms*, Lecture Notes in Computer Science vol. 839, Springer Verlag, 1994

[MAU96] Ueli Maurer, ETH Zürich, *Modelling a Public-Key Infrastructure*, Lecture Notes in Computer Science vol. 1146, Springer Verlag, 1996

[MAR01] E.G. Martin, *Netscape Navigator 6 : Mastering Today's Software*, Holt Rinehart and Winston, 2001

[MENE97] Menezes, Oorschot, Vanstone, *Handbook of Applied Cryptography*, CRC Press LLC, 1997

[MIC99] Microsoft, *Microsoft Internet Explorer 5 At a Glance*, Microsoft Press, 1999

[NER93] A. Nerode and R.A. Shore, *Logic for Applications*, Springer Verlag, 1993

[OCSP99] RFC 2560, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol OCSP*, 1999

[OSSL00] DFN-Projekt, *Das OpenSSL Handbuch*, Version 2.03, Universität Hamburg FB Informatik RZ, 2000

[PEM] PEM working group, *Privacy Enhancement for Internet Electronic Mail Part I-VI*, RFC 1421-1424, www.ietf.org

[PKCS1] RSA Laboratories, *PKCS#1 Standard Version 2.1 RSA Laboratories Technical Note*

[PKCS7] RSA Laboratories, *PKCS#7 Standard Version 2.1 RSA Laboratories Technical Note*

[PKCS10] RSA Laboratories, *PKCS#10 Standard Version 1.7 RSA Laboratories Technical Note*

[PKCS11] RSA Laboratories, *PKCS#11 Standard Version 2.10 RSA Laboratories Technical Note*

[PKCS12] RSA Laboratories, *PKCS#12 Standard Version 1.0 RSA Laboratories Technical Note*

[RAEF99] W. Rankl, W. Effing, *Handbuch der Chipkarten*, 3. Auflage, Carl Hanser Verlag, 1999

[REPO99] Rechenberg, Pomberger, *Informatik Handbuch*, 2. Auflage, Carl Hanser Verlag, 1999

[ROB96] J. Robertson, *Vollständige Systemanalyse*, Carl Hanser Verlag, 1996

[ROS99] M. Rosing, *Implementing Elliptic Curve Cryptography*, Manning Publications Co., 1999

[SCH96] B. Schneier, *Angewandte Kryptographie*, Addison-Wesley Deutschland GmbH, 1999

[SIGEU99] Europäisches Parlament und Rat, *Richtlinie 1999/93/EG des europäischen Rates und Paralaments über gemeinschaftliche Rahmenbedingungen für elektronische Signaturen*, EU-Server http:// europa.eu.int, 1999

[SIGI-AIS99] BSI, *Spezifikation zur Entwicklung interoperabler Verfahren und Komponenten nach SigG/SigV -SigI Abschnitt A3 Anwenderinfrastruktur, BSI Bonn, 1999*

[SIGI-GÜM99] BSI, *Spezifikation zur Entwicklung interoperabler Verfahren und Komponenten nach SigG/SigV -SigI Abschnitt A6 Gültigkeitsmodell, BSI Bonn, 1999*

[SIGI-SIG99] BSI, *Spezifikation zur Entwicklung interoperabler Verfahren und Komponenten nach SigG/SigV -SigI Abschnitt A2 Signatur, BSI Bonn, 1999*

[SIGI-VER99] BSI, *Spezifikation zur Entwicklung interoperabler Verfahren und Komponenten nach SigG/SigV -SigI Abschnitt A5 Verzeichnisdienst, BSI Bonn, 1999*

[SIGI-ZE99] BSI, *Spezifikation zur Entwicklung interoperabler Verfahren und Komponenten nach SigG/SigV -SigI Abschnitt A1 Zertifikate, BSI Bonn, 1999*

[SIGI-ZS99] BSI, *Spezifikation zur Entwicklung interoperabler Verfahren und Komponenten nach SigG/SigV -SigI Abschnitt A4 Zeitstempeldienst, BSI Bonn, 1999*

[SIGG01] Bundesanzeiger, *Gesetz zur Rahmenbedingungen für elektronische Signaturen und zur Änderung weiterer Vorschriften*, Bundesanzeiger Nr.22, 21.Mai 2001

[SIGG97] Bundesanzeiger, *Gesetz zur Regelung der Rahmenbedingungen für Informations- und Kommunikationsdienste IuKDG*, BT_Drs. 13/7934, 13. Juni 1997

[SIGUT95] Bundesstaat Utah, *Digital Signature Act*, http://www.le.state.ut.us/~code/TITLE46/46_03.htm, 1. Mai 1995

[SIGV01] Bundeskabinett, *Verordnung zur elektronischen Signatur*, Entwurfs-Fassung zur Notifizierung aufgrund der Richtlinie 98/43/EG, August 2001

[SIGV97] Bundeskabinett, *Signaturverordnung -SigV*, Fassung vom 08.10.1997 zum Signaturgesetz -SigG vom 22.07.97, 1997

[SMIME98] RFC2311, *S/MIME Version 2 Message Specification*, 1998

[STI01] D. R. Stinson, *Cryptography Theorie and Practice*, CRC Press, 1995

[SPI01] S. Spitz, *Verifikation von digitalen Signaturen*, DuD Ausgabe 8/2001

[TLS99] RFC 2246, *The TLS Protocol Version 1.0*, RFC 2246, 1999

[VA00] ValiCert Inc., *ValiCert Enterprise VA - Installaton and Administration Guide*, Version 3.2 2000

[WES00] M. Welschenbach, *Rijndael - Nachfolger des DES - Der zukünftige Advanced Encryption Standard*, DuD Ausgabe 6/2001

[WIN98] Microsoft Cor., *Windows NT Server 5.0 - Public Key Infrastruktur White Paper*, 1998

[X50999] RFC 2459, *Internet X.509 Public Key Infrastructure Certificate and CRL Profile*, 1999

[ZIM95] Phil Zimmermann, *The Official PGP User´s Guide*, MIT Press, 1995

Abkürzungsverzeichnis

AES	Advanced Encryption Standard (-Algorithmus)
BAN	Burrows Abadi Needham (-Logik)
CA	Certification Authority
CMP	Certificate Management Protocol
CMS	Certificate Management Services
CP	Certification Policy
CPS	Certification Practice Statement
DES	Data Encryption Standard (-Algorithmus)
DN	Distinguished Name
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard
ECC	Elliptic Curve Cryptography (-Algorithmus)
eSA	Erweiterte Strukturierte Analyse
HBCI	Home Banking Computer Interface
HTTP	Hypertext Transfer Protocol
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
ISIS	Industrial Signature Interoperability Specification
ISIS-MTT	Industrial Signature Interoperability Specification Mail TrusT
ISO	International Organization for Standardization
LDAP	Lightweight Directory Access Protocol
NIST	National Institute of Standards and Technology
MIME	Multipurpose Internet Mail Extensions
OCSP	Online Certification Status Protocol
OSI	Open System Interconnection
PGP	Pretty Good Privacy
PIN	Personal Identification Number
PKCS	Public Key Cryptographic Standard
PKI	Public Key Infrastructure
PKIX	Public Key Infrastructure X.509 (-Standard)

PL	Prädikatenlogik
PSE	Personal Security Environment
RA	Registration Authority
RFC	Request for Comments
RSA	Rivest Shamir Adleman (-Algorithmus)
SA	Strukturierte Analyse
SigBAN	Signatur-BAN (-Logik)
SigG	Signatur-Gesetz
SigI	Signatur-Interoperabilitäts-Spezifikation
SigV	Signatur-Verordnung
S/MIME	Secure Multipurpose Internet Mail Extensions
SSL	Secure Socket Layer (-Protokoll)
TC	Trust Center
TS	Time Stamping
TSA	Time Stamping Authority
WWW	World Wide Web

Anhang A: Beweise der SigBAN-Theoreme

A1 Beweis der Wurzelregel

A1.1 Ausgangsbedingungen

Der Verifizierer vertraut als PKI-Teilnehmer allen Aussagen der Wurzelinstanz und hat das Wurzelzertifikat vertrauenswürdig übermittelt bekommen.

$$\mathrm{VER} \models \mathrm{ROOT} \Rightarrow (\forall \mathrm{X})_{\mathrm{ROOT}}$$

$$\mathrm{VER} \triangleleft \mathrm{ZERT}_{\mathrm{ROOT}}$$

Die BAN-Regel zur Bedeutung von Nachrichten dient als Ausgangspunkt um die Wurzelregel zu beweisen.

$$\frac{\mathrm{VER} \models \xrightarrow{K} \mathrm{ROOT},\ \ \mathrm{VER} \triangleleft \{\mathrm{Y}\}_{K_{\mathrm{ROOT}}^{-1}}}{\mathrm{VER} \models \mathrm{ROOT} \mid\sim \mathrm{Y}}$$

A1.2 Herleitung

1. Schritt

Zuerst wird die Annahme, dass VER den öffentlichen Schlüssel der Instanz $\xrightarrow{K}$ROOT kennt, durch das Sehen des Wurzelzertifikates $\mathrm{ZERT}_{\mathrm{ROOT}}$, das den öffentlichen Schlüssel enthält, ersetzt. Gleichzeitig wird statt der speziellen Aussage Y ebenfalls das Wurzelzertifikat bzw. dessen Signatur $\mathrm{ZERTSIG}_{\mathrm{ROOT}}$ verwendet.

$$\frac{\mathrm{VER} \triangleleft \mathrm{ZERT}_{\mathrm{ROOT}}, \mathrm{VER} \triangleleft \mathrm{ZERTSIG}_{\mathrm{ROOT}}}{\mathrm{VER} \models \mathrm{ROOT} \mid\sim \mathrm{ZERT}_{\mathrm{ROOT}}}$$

2. Schritt

Die Aussage $\mathrm{VER} \triangleleft \mathrm{ZERT}_{\mathrm{ROOT}}$ beinhaltet $\mathrm{VER} \triangleleft \mathrm{ZERTSIG}_{\mathrm{ROOT}}$. Mit der Regel für das Sehen einer Komponente können also beide obenstehenden Aussagen zusammengefasst werden:

$$\frac{\mathrm{P} \triangleleft \mathrm{ZERT}_{\mathrm{ROOT}}}{\mathrm{VER} \triangleleft \mathrm{ZERTCONTENT}_{\mathrm{ROOT}}} \qquad \frac{\mathrm{P} \triangleleft \mathrm{ZERT}_{\mathrm{ROOT}}}{\mathrm{VER} \models \mathrm{ROOT} \mid\sim \mathrm{ZERT}_{\mathrm{ROOT}}}$$

3. Schritt

Zuerst wird im letzten Schritt die Aussage der Instanz ROOT vom Zertifikat auf den darin enthaltenen öffentlichen Schlüssel eingeschränkt. Grundlage für diese Umformung ist die Regel zur Aufteilung einer gesendeten Komponente:

$$\frac{VER \models ROOT \mid\sim ZERT_{ROOT} \qquad P \lhd ZERT_{ROOT}}{VER \models ROOT \mid\sim K_{ROOT} \qquad VER \models ROOT \mid\sim K_{ROOT}}$$

Unter Berücksichtigung des ersten SigBAN-Axioms wird der Verifizierer auch dem im Zertifikat enthaltenen öffentlichen Schlüssel vertrauen. Die beiden folgenden Aussagen sind somit gleichwertig, wenn es sich bei den Daten K_{ROOT} um einen öffentlichen Schlüssel handelt:

$$VER \models ROOT \mid\sim K_{ROOT} \text{ entspricht } VER \models \xrightarrow{K_{ROOT}} ROOT$$

$$\frac{VER \models ROOT \Rightarrow (\forall X)_{ROOT},\ VER \lhd ZERT_{ROOT}}{VER \models \xrightarrow{K_{ROOT}} ROOT}$$

A2 Beweis der Regel zur Zertifizierung

A2.1 Startbedingungen

Die BAN-Regel zur Bedeutung von Nachrichten in der besonderen Form für asymmetrische Kryptosysteme, dient auch als Ausgangspunkt um die Regel zur Zertifizierung zu beweisen. Da diese Regel ebenfalls für den Beweis der Wurzelregel verwendet wurde, ist die folgende Beweisführung der aus Teil A1 ähnlich und es wird stellenweise auf schon in A1 bewiesene Aussagen zurückgegriffen.

$$\frac{P \models \xrightarrow{K} Q,\ P \lhd \{X\}_K^{-1}}{P \models Q \mid\sim X}$$

A2.2 Herleitung

1. Schritt

Im ersten Schritt wird die allgemeine Aussage X durch den Inhalt des Signaturzertifikates $ZERTCONTENT_{SIG}$ bzw. dessen Signatur $\{ZERTCONTENT_{SIG}\}_{K^{-1}_{CA}}$, ersetzt:

$$\frac{VER \models \xrightarrow{K_{CA}} CA,\ VER \lhd \{ZERTCONTENT_{SIG}\}_{K^{-1}_{CA}}}{VER \models CA \mid\sim ZERTCONTENT_{SIG}}$$

2. Schritt

Aufgrund der Regel für das Sehen einer Komponente kann bei der zweiten Ausgangsrelation das komplette Zertifikat statt dem Zertifikatsinhalt eingesetzt werden. Die Schlussfolgerung kann mit der Regel zur Aufteilung einer gesendeten Aussage vom Zertifikatsinhalt auf den öffentlichen Schlüssel der Instanz CA eingesschränkt werden.

$$\frac{\text{VER} \models \xrightarrow{K_{CA}} \text{CA}, \text{VER} \triangleleft \text{ZERT}_{SIG}}{\text{VER} \models \text{CA} \mid\sim K_{SIG}}$$

3. Schritt

Wird der öffentliche Schlüssel nicht mehr als eine Aussage betrachtet, sondern als Teil eines Schlüsselpaars kann mit der Hilfsregel aus dem Beweis zur Wurzelregel die folgende Aussage getroffen werden:

Die Behauptung $\text{VER} \models \text{SIG} \mid\sim K_{SIG}$ ist äquivalent zu $\text{VER} \models \xrightarrow{K_{SIG}} \text{SIG}$, falls die Instanz VER über das Zertifikat von SIG verfügt. Damit ist die Regel zur Zertifizierung bewiesen:

$$\frac{\text{VER} \models \xrightarrow{K_{CA}} \text{CA}, \text{VER} \triangleleft \text{ZERT}_{SIG},}{\text{VER} \models \xrightarrow{K_{SIG}} \text{SIG}}$$

Anhang B: SigBAN-Umformungen

B1 Einfaches Anwendungsbeispiel aus Kapitel 5.6

B1.1 Beweis zur Gültigkeit der Signatur (VER ⊨ SIG ⊢ DATA)

Ausgangsrelationen:	**SigBAN-Regel:**	**Ergebnisrelation:**
Startbedingungen:		
$VER \models CA \Rightarrow (\forall X)_{CA}$	Wurzelregel:	
$VER \lhd ZERT_{CA}$	$\dfrac{VER \models CA \Rightarrow (\forall X)_{CA},\ VER \lhd ZERT_{CA}}{VER \models \xrightarrow{K_{CA}} CA}$	
$VER \models \xrightarrow{K_{CA}} CA$		$VER \models \xrightarrow{K_{CA}} CA$
$VER \lhd ZERT_{SIG}$	Regel zur Zertifizierung:	
	$\dfrac{VER \models \xrightarrow{K_{CA}} CA,\ VER \lhd ZERT_{SIG}}{VER \models \xrightarrow{K_{SIG}} SIG}$	
		$VER \models \xrightarrow{K_{SIG}} SIG$
	Regel zur Bedeutung von Nachrichten:	
$VER \models \xrightarrow{K_{SIG}} SIG$		
$VER \lhd \{H(DATA)\}_{K_{SIG}^{-1}}$	$\dfrac{VER \models \xrightarrow{K_{SIG}} SIG,\ VER \lhd \{H(DATA)\}_{K_{SIG}^{-1}}}{VER \models SIG \vdash H(DATA)}$	
		$VER \models SIG \vdash H(DATA)$
	Regel für Hash-Funktionen:	
$VER \models SIG \vdash H(DATA)$		
$VER \lhd DATA$	$\dfrac{VER \models SIG \vdash H(DATA),\ VER \lhd DATA}{VER \models SIG \vdash DATA}$	
		Ergebnis:
		$VER \models SIG \vdash DATA$

B1.2 Beweis zur Zeitaussage (VER ∈ CA ⊢ TIME)

Ausgangsrelationen:	**SigBAN-Regel:**	**Ergebnisrelation:**
Startbedingungen: $VER \in CA \Rightarrow (\forall X)_{CA}$ $VER \triangleleft ZERT_{CA}$	Wurzelregel: $\frac{VER \in CA \Rightarrow (\forall X)_{CA},\ VER \triangleleft ZERT_{CA}}{VER \in \xrightarrow{K_{CA}} CA}$	$VER \in \xrightarrow{K_{CA}} CA$
$VER \in \xrightarrow{K_{CA}} CA$ $VER \triangleleft TIMESTAMP_{CA}$ mit: $TIMESTAMP_{CA} = (TDATA, \{H(TDATA)\}_{K^{-1}_{CA}})$ und	Regel zur Bedeutung von Nachrichten: $\frac{VER \in \xrightarrow{K_{CA}} CA, VER \triangleleft \{H(TDATA)\}_{K^{-1}_{CA}}}{VER \in CA \vdash H(TDATA)}$	$VER \in CA \vdash H(TDATA)$
$VER \in CA \vdash H(TDATA)$ $VER \triangleleft TDATA$	Regel für Hash-Funktionen: $\frac{VER \in CA \vdash H(TDATA),\ VER \triangleleft TDATA}{VER \in CA \vdash TDATA}$	$VER \in CA \vdash TDATA$
$TDATA = (TIME, H(CONTENT))$	Regel zum Aufteilen einer übermittelten Aussage: $\frac{VER \in CA \vdash (TIME,\ H(CONTENT))}{VER \in CA \vdash TIME}$	***Ergebnis:*** **$VER \in CA \vdash TIME$**

B2 Beispiel zum SigG aus Kapitel 6.3 und Kapitel 6.4

B2.1 Beweis zur Gültigkeit der Signatur (VER ∈ SIG ⊢ CONTENT)

Ausgangsrelationen:	**SigBAN-Regel:**	**Ergebnisrelation:**
Startbedingungen: $VER \in ROOT \Rightarrow (\forall X)_{ROOT}$ $VER \triangleleft ZERT_{ROOT}$	Wurzelregel: $\dfrac{VER \in ROOT \Rightarrow (\forall X)_{ROOT},\ VER \triangleleft ZERT_{ROOT}}{VER \in \xrightarrow{K_{ROOT}} ROOT}$	$VER \in \xrightarrow{K_{ROOT}} ROOT$
$VER \in \xrightarrow{K_{ROOT}} ROOT$ $VER \triangleleft ZERT_{IM_CA1}$	Regel zur Zertifizierung: $\dfrac{VER \in \xrightarrow{K_{ROOT}} ROOT,\ VER \triangleleft ZERT_{IM_CA1}}{VER \in \xrightarrow{K_{IM_CA1}} IM_CA1}$	$VER \in \xrightarrow{K_{IM_CA1}} IM_CA1$
$VER \in \xrightarrow{K_{IM_CA1}} IM_CA1$ $VER \triangleleft ZERT_{SIG}$	Regel zur Zertifizierung: $\dfrac{VER \in \xrightarrow{K_{IM_CA1}} CA,\ VER \triangleleft ZERT_{SIG}}{VER \in \xrightarrow{K_{SIG}} SIG}$	$VER \in \xrightarrow{K_{SIG}} SIG$
$VER \in \xrightarrow{K_{SIG}} SIG$ $VER \triangleleft \{H(DATA)\}_{K_{SIG}^{-1}}$	Regel zur Bedeutung von Nachrichten: $\dfrac{VER \in \xrightarrow{K_{SIG}} SIG,\ VER \triangleleft \{H(DATA)\}_{K_{SIG}^{-1}}}{VER \in SIG \vdash H(DATA)}$	$VER \in SIG \vdash H(DATA)$
$VER \in SIG \vdash H(DATA)$ $VER \triangleleft DATA$	Regel für Hash-Funktionen: $\dfrac{VER \in SIG \vdash H(DATA),\ VER \triangleleft DATA}{VER \in SIG \vdash DATA}$	$VER \in SIG \vdash DATA$
DATA = (CONTENT, TIMESTAMP)	Regel zum Aufteilen einer übermittelten Aussage: $\dfrac{VER \in SIG \vdash (CONTENT,\ TIMESTAMP)}{VER \in SIG \vdash CONTENT}$	***Ergebnis:*** **$VER \in SIG \vdash CONTENT$**

formen Zeitstempeldienst

usammenhang mit dem SigG-konformen Zeitstempel- ΞTSA$\vdash\sim$TIME und VERΞTSA$\vdash\sim$ (TIME, CONTENT) bewie-

SigBAN-Regel:	Ergebnisrelation:
$\Rightarrow (\forall X)_{ROOT}$, VER $\triangleleft$ZERT$_{ROOT}$	
R $\Xi \xrightarrow{K_{ROOT}}$ ROOT	VER ROOT
zierung:	
$\xrightarrow{OT}$ ROOT, VER $\triangleleft$ ZERT$_{IM_CA1}$	
$\Xi \xrightarrow{K_{IM_CA1}}$ IM_CA1	VER IM_CA1
zierung:	
$\xrightarrow{IM_CA1}$ IM_CA1, VER $\triangleleft$ ZERT$_{TSA}$	
ER $\Xi \xrightarrow{K_{TSA}}$ TSA	VER TSA
tung von Nachrichten:	
$\xrightarrow{TSA}$ TSA, VER $\triangleleft$ TIMESTAMP	***1. Ergebnis:***
R Ξ TSA $\vdash$ TS_CONTENT	**VER TSA TS_CONTENT**
ilen einer übermittelten Aussage:	
TSA $\vdash$ (TIME, H(CONTENT))	***2. Ergebnis:***
VER Ξ TSA $\vdash$ TIME	**VER TSA TIME**
unktion	
H(CON	
TSA CONTENT	**(TIME, CONTENT)**

B2.3 Beweise zur exakten Ermittlung des Signaturzeitpunktes

Zuerst wird die Aussage $VER \models TSA \mid\sim TIME2$ bewiesen.

.

Ausgangsrelationen:	**SigBAN-Regel:**	**Ergebnisrelation:**
Startbedingungen: $VER \models ROOT \Rightarrow (\forall X)_{ROOT}$ $VER \triangleleft ZERT_{ROOT}$	Wurzelregel: $\dfrac{VER \models ROOT \Rightarrow (\forall X)_{ROOT},\ VER \triangleleft ZERT_{ROOT}}{VER \models \xrightarrow{K_{ROOT}} ROOT}$	$VER \models \xrightarrow{K_{ROOT}} ROOT$
$VER \models \xrightarrow{K_{ROOT}} ROOT$ $VER \triangleleft ZERT_{IM_CA1}$	Regel zur Zertifizierung: $\dfrac{VER \models \xrightarrow{K_{ROOT}} ROOT,\ VER \triangleleft ZERT_{IM_CA1}}{VER \models \xrightarrow{K_{IM_CA1}} IM_CA1}$	$VER \models \xrightarrow{K_{IM_CA1}} IM_CA1$
$VER \models \xrightarrow{K_{IM_CA1}} IM_CA1$ $VER \triangleleft ZERT_{TSA}$	Regel zur Zertifizierung: $\dfrac{VER \models \xrightarrow{K_{IM_CA1}} IM_CA1,\ VER \triangleleft ZERT_{TSA}}{VER \models \xrightarrow{K_{TSA}} TSA}$	$VER \models \xrightarrow{K_{TSA}} TSA$
$VER \models \xrightarrow{K_{TSA}} TSA$ $VER \triangleleft TIMESTAMP$ mit TIMESTAMP = TIMESTAMP1 TS_CONT, $\{H(TS_CONT)\}_{K_{TSA}^{-1}}$	Regel zur Bedeutung von Nachrichten: $\dfrac{VER \models \xrightarrow{K_{TSA}} TSA,\ VER \triangleleft \{H(TS_CONT)\}_{K_{TSA}^{-1}}}{VER \models TSA \mid\sim H(TS_CONT)}$	$VER \models TSA \mid\sim H(TS_CONT)$
$VER \models TSA \mid\sim H(TS_CONT)$ $VER \triangleleft TIMESTAMP$	Regel für Hash-Funktionen: $\dfrac{VER \models TSA \mid\sim H(TS_CONT),\ VER \triangleleft TS_CONT}{VER \models TSA \mid\sim TS_CONT}$	$VER \models TSA \mid\sim TS_CONT$
TS_CONT = SIGNATURE, TIME2 $VER \models TSA \mid\sim TS_CONT$	Regel zum Aufteilen einer übermittelten Aussage: $\dfrac{VER \models TSA \mid\sim (SIGNATURE, TIME2)}{VER \models TSA \mid\sim TIME2}$	***Ergebnis:*** **$VER \models TSA \mid\sim TIME2$**

Im Beweis von $VER \models TSA \vdash TIME1$ sind die ersten drei Umformungen identisch zum Beweis von $VER \models TSA \vdash TIME2$.

Ausgangsrelationen:	**SigBAN-Regel:**	**Ergebnisrelation:**
Startbedingungen: $VER \models ROOT \Rightarrow (\forall X)_{ROOT}$ $VER \lhd ZERT_{ROOT}$	Wurzelregel: $\dfrac{VER \models ROOT \Rightarrow (\forall X)_{ROOT},\ VER \lhd ZERT_{ROOT}}{VER \models \xrightarrow{K_{ROOT}} ROOT}$	$VER \models \xrightarrow{K_{ROOT}} ROOT$
$VER \models \xrightarrow{K_{ROOT}} ROOT$ $VER \lhd ZERT_{IM_CA1}$	Regel zur Zertifizierung: $\dfrac{VER \models \xrightarrow{K_{ROOT}} ROOT,\ VER \lhd ZERT_{IM_CA1}}{VER \models \xrightarrow{K_{IM_CA1}} IM_CA1}$	$VER \models \xrightarrow{K_{IM_CA1}} IM_CA1$
$VER \models \xrightarrow{K_{IM_CA1}} IM_CA1$ $VER \lhd ZERT_{TSA}$	Regel zur Zertifizierung: $\dfrac{VER \models \xrightarrow{K_{IM_CA1}} IM_CA1,\ VER \lhd ZERT_{TSA}}{VER \models \xrightarrow{K_{TSA}} TSA}$	$VER \models \xrightarrow{K_{TSA}} TSA$
$VER \models \xrightarrow{K_{TSA}} TSA$ $VER \lhd TIMESTAMP$ mit TIMESTAMP = TIMESTAMP1 TS_CONT, $\{H(TS_CONT)\}_{K^{-1}_{TSA}}$ und TIMESTAMP1 = TS1_CONT, $\{H(TS1_CONT)\}_{K^{-1}_{TSA}}$	Regel zur Bedeutung von Nachrichten: $\dfrac{VER \models \xrightarrow{K_{TSA}} TSA,\ VER \lhd \{H(TS1_CONT)\}_{K^{-1}_{TSA}}}{VER \models TSA \vdash H(TS1_CONT)}$ Regel für Hash-Funktionen: $\dfrac{VER \models TSA \vdash H(TS1_CONT), VER \lhd TS1_CONT}{VER \models TSA \vdash TS1_CONT}$	$VER \models TSA \vdash H(TS1_CONT)$ $VER \models TSA \vdash TS1_CONT$
$VER \models TSA \vdash TS1_CONT$ TS1_CONT = H(CONTENT), TIME1	Regel zum Aufteilen einer übermittelten Aussage: $\dfrac{VER \models TSA \vdash (H(CONTENT), TIME1)}{VER \models TSA \vdash TIME1}$	***Ergebnis:*** **$VER \models TSA \vdash TIME1$**

Hier erfolgt der Nachweis der für die Ermittlung des exakten Signaturzeitpunktes notwendigen Aussage VER $\models$ TSA $\vdash\sim$ (TIME, SIGN).

Ausgangsrelationen:	**SigBAN-Regel:**	**Ergebnisrelation:**
Startbedingungen: VER $\models$ ROOT $\Rightarrow (\forall X)_{ROOT}$ VER $\triangleleft$ $ZERT_{ROOT}$	Wurzelregel: VER ROOT ($)_{ROOT}$, VER $ZERT_{ROOT}$ VER ROOT	VER ROOT
VER $\models \xrightarrow{K_{ROOT}}$ ROOT VER $ZERT_{IM_CA1}$	Regel zur Zertifizierung: VER ROOT, VER $ZERT_{IM_CA1}$ VER IM_CA1	VER IM_CA1
VER IM_CA1 VER $ZERT_{TSA}$	Regel zur Zertifizierung: VER IM_CA1, VER $ZERT_{TSA}$ VER TSA	VER TSA
VER TSA VER TIMESTAMP mit TIMESTAMP = TIMESTAMP1, TIME2, SIGN),{H(TIME2, SIGN)}	Regel zur Bedeutung von Nachrichten: VER TSA, VER {H(TIME2, SIGN)} VER TSA H(TIME2, SIGN)	VER TSA H(TIME2, SIGN)
VER TSA H(TIME2, SIGN) TIME = TIME1 = TIME2	Regel für Hash-Funktionen: VER TSA H(TIME, SIGN), VER (TIME, SIGN) VER TSA (TIME, SIGN)	***Ergebnis:*** **VER TSA (TIME, SIGN)**

B3 Verwendung besonderer Zertifikatsformate aus Kapitel 7.2

B3.1 Beweis von VER ∈ SIG ⊢ DATA über Cross-Zertifizierung

Ausgangsrelationen:	**SigBAN-Regel:**	**Ergebnisrelation:**
Startbedingungen:		
$VER \in CA_2 \Rightarrow (\forall X)_{CA_2}$	Wurzelregel:	
$VER \triangleleft ZERT_{CA_2}$	$\dfrac{VER \in CA_2 \Rightarrow (\forall X)_{CA_2}, VER \triangleleft ZERT_{CA_2}}{VER \in \xrightarrow{K_{CA_2}} CA}$	$VER \in \xrightarrow{K_{CA_2}} CA_2$
$VER \in \xrightarrow{K_{CA_2}} CA_2$		
$VER \triangleleft ZERT_{CROSS_CA_1}$	Regel zur Zertifizierung:	
	$\dfrac{VER \in \xrightarrow{K_{CA_2}} CA_2, VER \triangleleft ZERT_{CROSS_CA_1}}{VER \in \xrightarrow{K_{CA_1}} CA_1}$	$VER \in \xrightarrow{K_{CA_1}} CA_1$
	Regel zur Zertifizierung:	
$VER \in \xrightarrow{K_{CA_1}} CA_1$	$\dfrac{VER \in \xrightarrow{K_{CA_1}} CA, VER \triangleleft ZERT_{SIG}}{VER \in \xrightarrow{K_{SIG}} SIG}$	$VER \in \xrightarrow{K_{SIG}} SIG$
	Regel zur Bedeutung von Nachrichten:	
$VER \in \xrightarrow{K_{SIG}} SIG$		
$VER \triangleleft \{H(DATA)\}_{K_{SIG}^{-1}}$	$\dfrac{VER \in \xrightarrow{K_{SIG}} SIG, VER \triangleleft \{H(DATA)\}_{K_{SIG}^{-1}}}{VER \in SIG \vdash H(DATA)}$	$VER \in SIG \vdash H(DATA)$
	Regel für Hash-Funktionen:	
$VER \in SIG \vdash H(DATA)$		
$VER \triangleleft DATA$	$\dfrac{VER \in SIG \vdash H(DATA), VER \triangleleft DATA}{VER \in SIG \vdash DATA}$	***Ergebnis:*** **$VER \in SIG \vdash DATA$**