

## Grundlagen der elektronischen Signatur

Recht Technik Anwendung

Bundesamt für Sicherheit in der Informationstechnik  
Postfach 20 03 63  
53133 Bonn  
E-Mail: [dgsig@bsi.bund.de](mailto:dgsig@bsi.bund.de)  
© Bundesamt für Sicherheit in der Informationstechnik 2006

## Vorwort

Das Thema elektronische Signatur hat sich in den letzten Jahren erheblich fortentwickelt. Nach der Schaffung der rechtlichen Rahmenbedingungen und Sicherheitsinfrastrukturen hält die Signatur zunehmend Einzug in zahlreiche Anwendungs- und Geschäftsbereiche. Die Umsetzung der eCard-Strategie der Bundesregierung, zu deren wesentlichen Stützpfeilern die Authentifizierung und qualifizierte elektronische Signatur mittels Chipkarten unterschiedlicher Ausprägung zählt, wird weitere Fortschritte erzeugen.

Durch die steigende Verbreitung und Nutzung der elektronischen Signatur wird ein signifikanter Beitrag zur Entwicklung des elektronischen Geschäftsverkehrs im E-Government, E-Business und elektronischen Rechtsverkehr geleistet. Hieraus ergibt sich ein neuer Schub für die Erhöhung der Leistungs- und Wettbewerbsfähigkeit sowie für die Modernisierung und Effizienzsteigerung der deutschen Wirtschaft und öffentlichen Verwaltung.

Die erfolgreiche Einführung und Nutzung dieser Schlüsseltechnologie erfordert aber ein interdisziplinäres Verständnis für das komplexe Zusammenwirken juristischer, mathematischer und technischer Methoden und Mechanismen. Deshalb enthält diese Veröffentlichung nicht nur eine für Techniker verständliche Zusammenfassung der rechtlichen Rahmenbedingungen der elektronischen Signatur in Deutschland, sondern auch den für Juristen und Betriebswirte zugänglichen aktuellen Stand der technischen Forschung und Entwicklung. Darüber hinaus werden exemplarisch in der Praxis besonders relevante Anwendungen aufgezeigt.

Ich bin zuversichtlich, dass diese Publikation einen Beitrag zum Verständnis der elektronischen Signatur leistet und eine Grundlage bietet, damit interessierte Fach- und Führungskräfte von den Vorteilen dieser Technologie profitieren können.

Bonn, im März 2006

A handwritten signature in black ink, appearing to read "U. Helmbrecht".

Dr. Udo Helmbrecht, Präsident des BSI

## Autoren

Die vorliegende Publikation wurde im Auftrag und in Zusammenarbeit mit dem Bundesamt für Sicherheit in der Informationstechnik erstellt von:

- Dr. Detlef Hühnlein  
[detlef.huehnlein@secunet.com](mailto:detlef.huehnlein@secunet.com)  
secunet Security Networks AG  
und
- Dr. Ulrike Korte  
[ulrike.korte@bsi.bund.de](mailto:ulrike.korte@bsi.bund.de)  
Bundesamt für Sicherheit  
in der Informationstechnik

## Danksagung

Wir bedanken uns herzlich bei allen Kolleginnen und Kollegen, die die Entstehung dieses Dokumentes unterstützt und uns wertvolle Hinweise und Korrekturen geliefert haben:

- Dr. Astrid Albrecht
- Manuel Bach
- Dr. Rainer Baumgart
- Dr. Steffen Frischat
- Utz Gnaida
- Ingo Hahlen
- Torsten Henn
- Klaus Keus
- Kai Lehmann
- Dr. Ernst-Otto Liebetrau
- Dr. Johannes Merkle
- Tom Monhaupt
- Dr. Christian Mrugalla
- Thomas Prauß
- Dr. Kay Rathke
- Frank Rustemeyer
- Dr. Ernst Schulte-Geers und
- Dr. Thomas Schöller.

**Vielen Dank.**

## **Tabellenverzeichnis**

Tabelle 1: Erweiterter Euklidischer Algorithmus (EEA) .....	27
Tabelle 2: Inversion von $2 \pmod{7}$ mit dem EEA.....	28
Tabelle 3: Inversion von $3 \pmod{160}$ mit dem EEA.....	33
Tabelle 4: Populäre Hashfunktionen.....	46
Tabelle 5: Massensignatur-Benchmark.....	84

# Abbildungsverzeichnis

Abbildung 1: Entwicklung der rechtlichen Rahmenbedingungen im Überblick .....	3
Abbildung 2: Ausprägungen der elektronischen Signatur .....	8
Abbildung 3: Prinzip der digitalen Signatur .....	22
Abbildung 4: Gruppenoperation auf einer elliptischen Kurve .....	29
Abbildung 5: Die Elliptische Kurve $y^2 = x^3 + 1$ .....	30
Abbildung 6: Vergleich der Schlüssellängen zwischen RSA/DSA und ECDSA .....	31
Abbildung 7: Das RSA-Signaturverfahren .....	33
Abbildung 8: PKCS #1 Version 1.5 Codierung .....	35
Abbildung 9: PSS-Codierung (PKCS #1 Version 2.1) .....	36
Abbildung 10: f()-Funktion zur Maskenerzeugung aus PKCS #1 v2.1 .....	37
Abbildung 11: Das ElGamal-Signaturverfahren .....	38
Abbildung 12: Das Schnorr-Signaturverfahren .....	40
Abbildung 13: Der Digital Signature Algorithm .....	42
Abbildung 14: Der Elliptic Curve Digital Signature Algorithm .....	44
Abbildung 15: Prinzipieller Aufbau iterativer Hashfunktionen .....	45
Abbildung 16: Komponenten einer PKI .....	47
Abbildung 17: Anwenderkomponenten .....	48
Abbildung 18: Chipkarte als Signaturerstellungseinheit .....	49
Abbildung 19: X.509v3 - Zertifikatsformat .....	50
Abbildung 20: X.509v2 - CRL-Format .....	53
Abbildung 21: OCSP im Überblick .....	56
Abbildung 22: OCSP-Request .....	57
Abbildung 23: OCSP-Response .....	58
Abbildung 24: TSP im Überblick .....	59
Abbildung 25: TSP-Request .....	60
Abbildung 26: TSP-Response .....	61
Abbildung 27: Zertifikatspfad .....	64
Abbildung 28: Gültigkeitsmodelle im Überblick .....	67
Abbildung 29: Struktur des SignedData-Containers aus CMS / PKCS #7 .....	70
Abbildung 30: Enveloping und Detached Signatures mit CMS .....	74
Abbildung 31: XML-Signatur-Typen .....	76
Abbildung 32: Struktur einer XML-Signatur .....	77
Abbildung 33: Beispielhafte EDIFACT-Übertragungsdatei .....	78
Abbildung 34: Einsatz der AUTACK-Nachricht .....	80
Abbildung 35: Prüfung der Signatur im PDF-Reader .....	82
Abbildung 36: Massensignatur-System .....	83
Abbildung 37: Hashbaum für Stapelsignatur .....	84
Abbildung 38: Relative zeitliche Ordnung durch Hashfunktion .....	86
Abbildung 39: Konstruktion Intervall-Qualifizierter Zeitstempel .....	87
Abbildung 40: IQ-Zeitstempel-System .....	88
Abbildung 41: Formale Anforderungen für Rechnungen .....	94
Abbildung 42: Rahmenbedingungen für die Aufbewahrung von Rechnungen .....	97
Abbildung 43: Papiergebundene Rechnung .....	104
Abbildung 44: EDI-Rechnung mit Signatur .....	105
Abbildung 45: Elektronische Rechnung im Bilddatenformat .....	106
Abbildung 46: PDF-Rechnung mit Signatur und Prüfbericht .....	107
Abbildung 47: EDI mit papiergebundener Sammelrechnung .....	108

## Abbildungsverzeichnis

Abbildung 48: EDI mit papiergebundener Sammelgutschrift.....	108
Abbildung 49: Abrechnung über Konsolidator.....	109
Abbildung 50: Provisionsabrechnungen bei der Bausparkasse Schwäbisch-Hall .....	111
Abbildung 51: Buchungsbestätigung bei der dba Luftfahrtgesellschaft .....	112

## Inhaltsverzeichnis

<b>VORWORT .....</b>	<b>III</b>
<b>TABELLENVERZEICHNIS .....</b>	<b>V</b>
<b>ABBILDUNGSVERZEICHNIS .....</b>	<b>VI</b>
<b>1 EINFÜHRUNG .....</b>	<b>1</b>
<b>2 RECHTLICHE RAHMENBEDINGUNGEN .....</b>	<b>2</b>
2.1 SIGNATURGESETZGEBUNG.....	2
2.1.1 <i>Historische Entwicklung</i> .....	2
2.1.2 <i>Grundlegende Bestimmungen</i> .....	7
2.1.3 <i>Regelungen für Zertifizierungsdiensteanbieter</i> .....	10
2.1.4 <i>Anwendungsbezogene Aspekte</i> .....	12
2.2 RELEVANZ DER SIGNATUR FÜR DEN ELEKTRONISCHEN RECHTSVERKEHR.....	14
2.2.1 <i>Formvorschriften</i> .....	14
2.2.2 <i>Beweiskraft</i> .....	17
2.3 ZUSAMMENFASSUNG DER GESETZLICHEN ANFORDERUNGEN.....	18
2.3.1 ... <i>an Zertifizierungsdiensteanbieter</i> .....	19
2.3.2 ... <i>an Hersteller von Komponenten</i> .....	19
2.3.3 ... <i>an Signaturanwender</i> .....	19
<b>3 TECHNISCHE REALISIERUNG .....</b>	<b>21</b>
3.1 KRYPTOGRAPHISCHE GRUNDLAGEN .....	21
3.1.1 <i>Das Prinzip der digitalen Signatur</i> .....	21
3.1.2 <i>Einwegfunktionen und zahlentheoretische Probleme</i> .....	22
3.1.3 <i>Signaturalgorithmen</i> .....	32
3.1.4 <i>Hashfunktionen</i> .....	44
3.2 PUBLIC-KEY-INFRASTRUKTUREN.....	46
3.2.1 <i>Anwenderkomponenten</i> .....	48
3.2.2 <i>Registrierungsinstanz</i> .....	49
3.2.3 <i>Attributquelle</i> .....	49
3.2.4 <i>Schlüsselgenerator</i> .....	49
3.2.5 <i>Zertifizierungsinstanz</i> .....	50
3.2.6 <i>Verzeichnisdienst</i> .....	55
3.2.7 <i>Zeitstempeldienst</i> .....	59
<b>4 SIGNATURANWENDUNG .....</b>	<b>63</b>
4.1 SIGNATURERZEUGUNG .....	63
4.2 SIGNATURPRÜFUNG .....	64
4.2.1 <i>Mathematische Gültigkeit der Signaturen</i> .....	65
4.2.2 <i>Gültigkeit der Zertifikate gemäß Gültigkeitsmodell</i> .....	65
4.2.3 <i>Korrektheit des Verwendungszwecks der Zertifikate</i> .....	68
4.3 SIGNATURFORMATE .....	68
4.3.1 <i>Cryptographic Message Syntax / PKCS #7</i>	69
4.3.2 <i>S/MIME</i> .....	74
4.3.3 <i>XML Digital Signature</i> .....	75
4.3.4 <i>EDIFACT</i> .....	78
4.3.5 <i>PGP und PGP/MIME</i> .....	81
4.3.6 <i>Einbettung von Signaturen in PDF-Dokumente</i> .....	81
4.4 MASSENSIGNATUR .....	82
4.5 ZEITSTEMPEL.....	85
4.6 ARCHIVIERUNG VON SIGNIERTEN DATEN .....	88
4.7 CODE-SIGNING .....	89

<b>5 ANWENDUNGSBEREICHE .....</b>	<b>90</b>
5.1 SIGNATURANWENDUNGEN IM BEHÖRDLICHEN UMFELD.....	90
5.1.1 <i>Elektronische Steuererklärung</i> .....	90
5.1.2 <i>Vergabe öffentlicher Aufträge</i> .....	91
5.1.3 <i>Genehmigung der Ein- und Ausfuhr geschützter Tiere und Pflanzen</i> .....	91
5.1.4 <i>Rechnungswesen in der Sozialversicherung</i> .....	91
5.1.5 <i>Patentantrag</i> .....	92
5.1.6 <i>Justizkommunikation</i> .....	92
5.1.7 <i>Einwohnermeldewesen</i> .....	93
5.2 ELEKTRONISCHE RECHNUNGSSTELLUNG.....	93
5.2.1 <i>Rechtliche Rahmenbedingungen</i> .....	94
5.2.2 <i>Abrechnungsszenarien</i> .....	103
5.2.3 <i>Wirtschaftlichkeitsbetrachtung</i> .....	109
5.2.4 <i>Praxisbeispiele</i> .....	110
5.3 eCARD-STRATEGIE DER BUNDESREGIERUNG .....	113
<b>GLOSSAR .....</b>	<b>114</b>
<b>LITERATUR.....</b>	<b>140</b>



# 1 Einführung

Durch die elektronische Abwicklung von Geschäftsprozessen lassen sich Kosten senken sowie Fehlerquoten und Prozesslaufzeiten reduzieren. Dazu werden im Wesentlichen papiergebundene Schriftstücke zunehmend digitalisiert oder durch elektronische Dokumente ersetzt und Prozesse automatisiert. Da bei vielen konventionellen Abläufen – insbesondere auf Grund gesetzlicher Schriftformerfordernisse oder zur beweiskräftigen Dokumentation von Willenserklärungen – eigenhändige Unterschriften eingesetzt werden, muss diese auch bei elektronischen Prozessen adäquat abgebildet werden können. Daher kommt der elektronischen Signatur bei der digitalen Abwicklung von Geschäftsprozessen eine besondere Bedeutung zu.

Eine grundlegende Funktion einer *elektronischen Signatur* ist, dass sie der *Authentifizierung* des Unterzeichners dienen soll. In den meisten Fällen ist es jedoch nicht ausreichend, dass nur der vermeintliche Urheber einer Nachricht ermittelt werden kann. Neben der *Authentizität* muss oft auch die *Integrität* und *Nichtabstrebbarkeit* einer Willenserklärung sicher gestellt werden, damit durch die elektronische Abwicklung von Geschäftsprozessen keine zusätzlichen Risiken entstehen.

Die vorliegende Arbeit erläutert die wichtigsten rechtlichen und technischen Grundlagen der elektronischen Signatur und geht auf Aspekte der Anwendung dieser Schlüsseltechnologie ein.

Die wesentlichen rechtlichen Rahmenbedingungen für die elektronische Signatur sind in *Kapitel 2* zusammen getragen. Nach einem Überblick über die Signaturgesetzgebung in *Abschnitt 2.1* wird in *Abschnitt 2.2* auf die Relevanz der Signatur für den elektronischen Rechtsverkehr eingegangen. Eine Zusammenfassung der gesetzlichen Anforderungen findet sich *Abschnitt 2.3*.

*Kapitel 3* befasst sich mit der technischen Realisierung der elektronischen Signatur. Neben den kryptographischen Grundlagen der elektronischen Signatur in *Abschnitt 3.1* werden in *Abschnitt 3.2* auch die wesentlichen Komponenten und Prozesse einer *Public-Key-Infrastruktur* besprochen.

In *Kapitel 4* werden schließlich verschiedene Aspekte der Anwendung der elektronischen Signatur erörtert. Nach einer Darstellung der Abläufe bei der Erzeugung (vgl. *Abschnitt 4.1*) und Prüfung (vgl. *Abschnitt 4.2*) von elektronischen Signaturen werden in *Abschnitt 4.3* die gebräuchlichsten Signaturformate näher beleuchtet. Außerdem werden in den Abschnitten 4.4 - 4.7 einige für den praktischen Einsatz der Signatur besonders wichtige Aspekte, wie z.B. die Massensignatur oder die Archivierung von signierten Daten, näher betrachtet.

In *Kapitel 5* werden schließlich einige Anwendungen der elektronischen Signatur, wie z.B. die elektronische Rechnungsstellung, näher beleuchtet.

## 2 Rechtliche Rahmenbedingungen

Der rechtliche Rahmen der elektronischen Signatur in Deutschland ist durch das Signaturgesetz [SigG] und die Signaturverordnung [SigV] definiert. Diese im Folgenden näher betrachtete Signaturgesetzgebung ist vor dem Hintergrund der EU-Richtlinie [1999/93/EG] für elektronische Signaturen (vgl. *Abschnitt 2.1.1.2*) und im Kontext weiterer nationaler Rechtsvorschriften zu sehen, die den Einsatz elektronischer Signaturen im Rechtsverkehr regeln.

### 2.1 Signaturgesetzgebung

Nachdem in *Abschnitt 2.1.1* die wichtigsten Meilensteine auf dem Weg zum aktuellen Signaturgesetz [SigG] beleuchtet werden, finden sich in den folgenden Abschnitten grundlegende Definitionen (*Abschnitt 2.1.2*) und die wichtigsten Aspekte der Signaturgesetzgebung für *Zertifizierungsdiensteanbieter* (ZDA) (*Abschnitt 2.1.3*) und Anwender (*Abschnitt 2.1.4*).

#### 2.1.1 Historische Entwicklung

Die Entwicklung des deutschen Signaturgesetzes reicht bis in die frühen neunziger Jahre zurück und mündete – wie in *Abbildung 1* dargestellt – im Jahr 1997 in das erste Signaturgesetz [SigG97], das durch das *Informations- und Kommunikationsdienste-Gesetz* (IuKDG) eingeführt und durch die Signaturverordnung [SigV] präzisiert wurde (vgl. *Abschnitt 2.1.1.1*).

Anfang 2000 wurde mit der in *Abschnitt 2.1.1.2* näher besprochenen EU-Richtlinie [1999/93/EG] ein europaweiter Rahmen für elektronische Signaturen und ihren Einsatz im Rechtsverkehr geschaffen. Die Anforderungen an technische Komponenten und Zertifizierungsdiensteanbieter wurden in den Anhängen der Richtlinie [1999/93/EG] lediglich skizziert – im Jahr 2003 erfolgte die geplante Präzisierung durch die Veröffentlichung von Referenznummern [2003/511/EG] für anerkannte Normen [CWA14167-1, CWA14167-2, CWA14169].

Vor dem Hintergrund dieser Europäischen Rahmenbedingungen wurde die deutsche Signaturgesetzgebung im Jahr 2001 umfassend überarbeitet (vgl. *Abschnitt 2.1.1.3*). Hierbei löste das novellierte Signaturgesetz [SigG] das alte Gesetz ab und die Signaturverordnung [SigV] wurde an das neue Gesetz angepasst. Inzwischen wurden Signaturgesetz und Signaturverordnung durch das erste Gesetz zur Änderung des Signaturgesetzes [SigGAendG] abermals geändert.

Erst nach der Novellierung des Signaturgesetzes wurde eine Anpassung der Formvorschriften im Privatrecht [FormAnpG] und im öffentlichen Recht [VerWVfAendG] vorgenommen (vgl. *Abschnitt 2.1.1.4*). Im Privatrecht waren insbesondere das Bürgerliche Gesetzbuch [BGB] und die Zivilprozessordnung [ZPO] betroffen. Im öffentlichen Recht wurden insbesondere das Verwaltungsverfahrensgesetz [VwVfG], das Sozialgesetzbuch [SGBI, SGBIV, SGBX] und die Abgabenordnung [AO] geändert. Im Jahr 2005 wurden schließlich durch das Justizkommunikationsgesetz [JKomG] die Rahmenbedingungen für die umfassende Nutzung elektronischer Kommunikationsformen in der Justiz geschaffen und unter anderem die Regelungen zur Beweiskraft elektronischer Dokumente in der Zivilprozessordnung [ZPO] angepasst.

## 2 Rechtliche Rahmenbedingungen

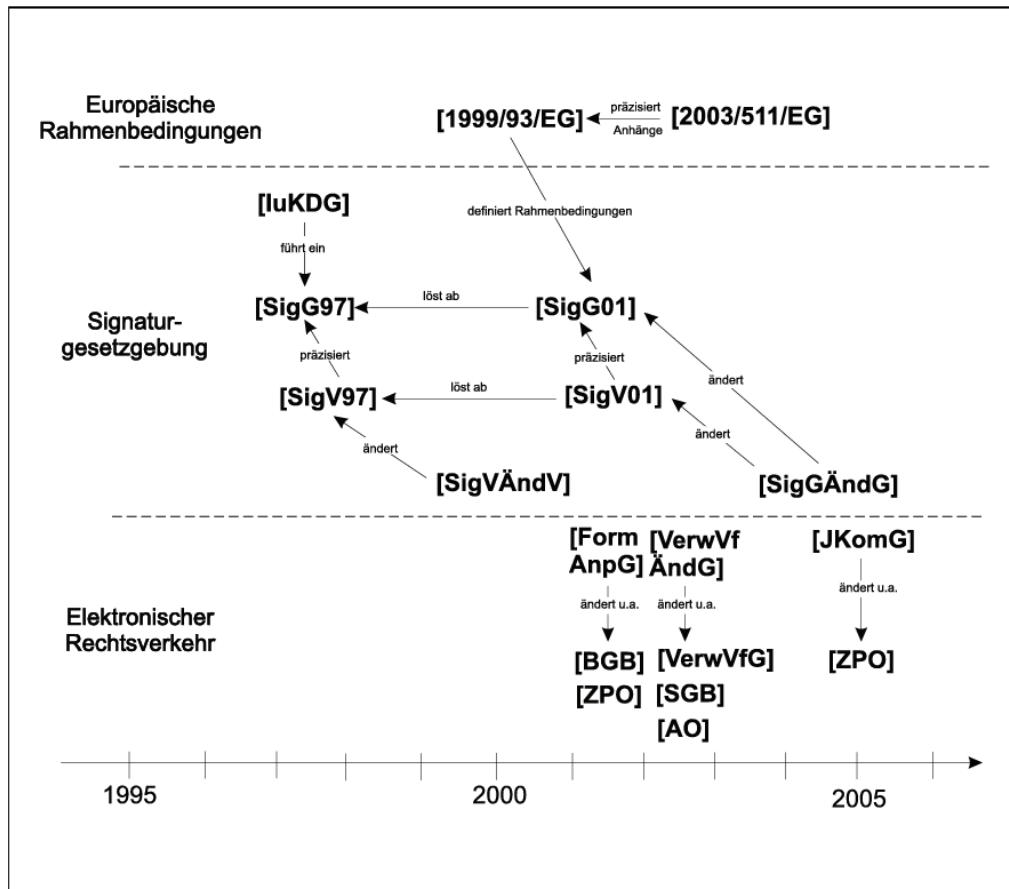


Abbildung 1: Entwicklung der rechtlichen Rahmenbedingungen im Überblick

### 2.1.1.1 Das Signaturgesetz von 1997

Die ersten Schritte auf dem Weg zur deutschen Signaturgesetzgebung wurden von der Bundesnotarkammer und dem TeleTrusT e.V. geprägt. Beispielsweise gab es bereits im November 1993 eine erste Veranstaltung zur Diskussion der rechtlichen Rahmenbedingungen der *digitalen Signatur* (vgl. [BNK95a]). Diese Ansätze mündeten in eine Projektgruppe der Bundesnotarkammer, in der erste Vorschläge für die Signaturgesetzgebung erarbeitet wurden [EbFa96]. In diesem Zusammenhang wurde bereits ein Vorschlag [BNK95b] für einen neuen § 126a [BGB] geschaffen, wonach die elektronische Unterschrift für die „elektronische Form (...) in einem als sicher anerkannten Verfahren (...) hergestellt werden“ muss. Welche Anforderungen damit genau verbunden sind, sollte eine vom Bundesministerium des Innern (BMI) entworfene „Verordnung über die Anerkennung von Verfahren zur elektronischen Unterschrift“ (VEU) [BMI95] regeln. Die vom BMI entworfene Verordnung trat aber nie in Kraft. Vielmehr wurde vom BMI auf Grundlage des Verordnungsentwurfs und dem parallel dazu entstandenen „Utah Digital Signature Act“ [Utah-DSA] vom 1. Mai 1995 ein Vorentwurf des Signaturgesetzes [BMI96] vorgelegt. Aus diesem entstanden schließlich das Signaturgesetz [SigG97] (verkündet als Artikel 3 des *Informations- und Kommunikationsdienste-Gesetzes* [IuKDG]) und die Signaturverordnung [SigV97].

In diesem Signaturgesetz wurden lediglich die Sicherheitsanforderungen an die digitale Signatur spezifiziert – eine rechtliche Gleichstellung derselben mit der handschriftlichen

## 2 Rechtliche Rahmenbedingungen

---

Unterschrift erfolgte nicht. Dieser wichtige Schritt erfolgte erst mit dem Formanpassungsgesetz [FormAnpG] – also insbesondere erst nach der umfassenden Überarbeitung des Signaturgesetzes im Jahr 2001, die insbesondere auf Grund der im nächsten Abschnitt näher beleuchteten EU-Richtlinie [1999/93/EG] nötig wurde.

### 2.1.1.2 Die Europäische Signaturrichtlinie

Parallel zum deutschen Signaturgesetz entwickelten sich auch in anderen Mitgliedstaaten der Europäischen Union, beispielsweise in Italien (vgl. [ISTEV97]), Regelungen für elektronische Signaturen. Deshalb war eine EU-weite Abstimmung der rechtlichen Rahmenbedingungen für elektronische Signaturen geboten.

Vor diesem Hintergrund wurde im Juni 1998 von der Europäischen Kommission ein Vorschlag für eine Signaturrichtlinie [98/0191(COD)] vorgelegt. Dieser Vorschlag verfolgte einen weitgehend konträr zum [SigG97] stehenden Ansatz (vgl. [Ross00, Abschnitt 1.2]). Anstatt die Sicherheit der Signaturinfrastrukturen durch Zulassungsverfahren für Zertifizierungsstellen und technische Komponenten zu regeln, sollte dies durch die Einführung einer umfassenden<sup>1</sup>, vom Verschulden unabhängigen Haftungsregelung für Zertifizierungsdiensteanbieter erfolgen. Dieser Vorschlag, bei dem beispielsweise die Gleichstellung der elektronischen Signatur mit der handschriftlichen Unterschrift (Artikel 5) ohne Einsatz von sicheren Signaturerstellungseinheiten vorgesehen war, wurde im Europäischen Rat am 27. November 1998 mit neun zu sechs Stimmen zurückgewiesen. Der Richtlinienvorschlag wurde mehrfach geändert [COM(1999)195, 1999/C-243/02], bevor die endgültige Richtlinie [1999/93/EG] schließlich nach Verkündung im Amtsblatt der Europäischen Union am 19. Januar 2000 in Kraft trat.

Der Anwendungsbereich der Richtlinie [1999/93/EG], die für Anbieter und Nutzer von Zertifizierungsdiensten und Produkten für elektronische Signaturen nicht unmittelbar gilt, sondern erst von den Mitgliedstaaten in nationales Recht umgesetzt werden musste, ist in Artikel 1 bestimmt:

„Diese Richtlinie soll die Verwendung elektronischer Signaturen erleichtern und zu ihrer rechtlichen Anerkennung beitragen. Sie legt rechtliche Rahmenbedingungen für elektronische Signaturen und für bestimmte Zertifizierungsdienste fest, damit das reibungslose Funktionieren des Binnenmarktes gewährleistet ist.“

Ein wesentlicher Punkt der Richtlinie ist die Klärung der Rechtswirkung elektronischer Signaturen in Artikel 5:

- „(1) Die Mitgliedstaaten tragen dafür Sorge, dass fortgeschrittene elektronische Signaturen, die auf einem qualifizierten Zertifikat beruhen und die von einer sicheren Signaturerstellungseinheit erstellt werden,
- die rechtlichen Anforderungen an eine Unterschrift in bezug auf in elektronischer Form vorliegende Daten in gleicher Weise erfüllen wie handschriftliche Unterschriften in bezug auf Daten, die auf Papier vorliegen, und
  - in Gerichtsverfahren als Beweismittel zugelassen sind.“

Außerdem sorgen die Artikel 4 (Binnenmarktgrundsätze) und Artikel 7 (Internationale Aspekte) dafür, dass qualifizierte Zertifikate aus anderen Mitgliedstaaten sowie Drittländern

---

<sup>1</sup>Nach Artikel 6, Abs. 1 b) [98/0191(COD)] sollte der Zertifizierungsdiensteanbieter dafür haften, dass alle Anforderungen der Richtlinie bei der Ausstellung des qualifizierten Zertifikats eingehalten wurden.

## 2 Rechtliche Rahmenbedingungen

---

unter bestimmten Voraussetzungen EU-weit anerkannt werden. Die Haftungsregelungen gemäß Artikel 6 sehen nun vor, dass ein Zertifizierungsdiensteanbieter nicht haftet, wenn er nachweist, dass er nicht fahrlässig gehandelt hat.

Die „Anforderungen an *qualifizierte Zertifikate*“ sind in Anhang I der Richtlinie definiert. Außerdem spezifizieren die Anhänge II und III „Anforderungen an *Zertifizierungsdiensteanbieter*, die *qualifizierte Zertifikate ausstellen*“ bzw. „Anforderungen an *sichere Signaturerstellungseinheiten*“.

Die Anforderungen der Anhänge II und III wurden durch die Veröffentlichung von Referenznummern für allgemein anerkannte Normen für Produkte für elektronische Signaturen durch den Ausschuss gemäß Artikel 9 weiter präzisiert [2003/511/EG]. Für die Anforderungen an vertrauenswürdige Systeme und Produkte, die von Zertifizierungsdiensteanbietern bei der Ausstellung von qualifizierten Zertifikaten eingesetzt werden (Anhang II f), wird auf die beiden CEN Workshop Agreements [CWA14167-1, CWA14167-2] verwiesen. Die Präzisierung von Anhang III der Richtlinie findet sich in [CWA14169]. Anhang IV enthält schließlich „Empfehlungen für die sichere Signaturprüfung“.

Informationen zur Umsetzung der Richtlinie in den einzelnen EU-Mitgliedstaaten finden sich in [DuRi01, DKN+03, EU-SigNot].

### 2.1.1.3 Die Novellierung des Signaturgesetzes im Jahr 2001

Da der Gesetzgeber mit der Entwicklung des *IuKDG* und des Signaturgesetzes im Jahr 1997 regulatorisches Neuland betrat, hat der Deutsche Bundestag bei dessen Verabschiedung gefordert, dass das Gesetz auf seine Praktikabilität hin überprüft werden solle (vgl. [Tett00]). Der dem Bundestag dazu im Juni 1999 vorgelegte Evaluierungsbericht [BReg97] stellte trotz der insgesamt positiven Erfahrungen mit dem Gesetz auch punktuellen Anpassungsbedarf fest.

Hier wurde die ausschließliche Anwendbarkeit der europäischen Sicherheitskriterien [ITSEC] zur Prüfung von Produkten im Hinblick auf eine internationalen Anerkennung der digitalen Signatur gemäß [SigG97] als Behinderung empfunden. Dies wurde kurzfristig durch eine Änderung der Signaturverordnung [SigV] mittels [SigVAendV] behoben. Damit können nun neben den europäischen Information Technology Security Evaluation Criteria [ITSEC] auch die international anerkannten Common Criteria [CC] zur Prüfung der Produkte genutzt werden. Weiterreichende Anpassungen des Signaturgesetzes erfolgten erst zusammen mit den durch die EU-Richtlinie [1999/93/EG] notwendig gewordenen Änderungen. Das überarbeitete Signaturgesetz [SigG] trat am Tag nach der Verkündung im Bundesgesetzblatt am 21. Mai 2001 in Kraft und löste das alte Signaturgesetz [SigG97] ab, wodurch sich insbesondere die folgenden Änderungen ergaben (vgl. [BrTe01, Seite 341 ff.]):

- *Verschiedene Stufen der elektronischen Signatur*

Das [SigG97] definierte mit der „digitalen Signatur“ lediglich eine Stufe der elektronischen Signatur. Das novellierte [SigG] definiert hingegen analog zu Artikel 2 Nr. 1-2 der Richtlinie [1999/93/EG] die (einfache) *elektronische Signatur* und die *fortgeschrittene elektronische Signatur*, sowie die *qualifizierte elektronische Signatur* im Sinne von Artikel 5 Abs. 1 der Richtlinie [1999/93/EG] und schließlich die qualifizierte elektronische Signatur mit *Anbieterakkreditierung* gemäß § 15 [SigG].

- *Betrieb eines Zertifizierungsdienstes wird genehmigungsfrei*

Während der Betrieb einer *Zertifizierungsstelle* gemäß § 4 Abs. 1 [SigG97] genehmigungspflichtig war, ist dies nach § 4 Abs. 1 [SigG] nicht mehr der Fall (vgl. Erwägungsgrund 10 der Richtlinie [1999/93/EG]). Allerdings muss ein

## 2 Rechtliche Rahmenbedingungen

---

Zertifizierungsdiensteanbieter (ZDA) die Aufnahme des Betriebes bei der Bundesnetzagentur (BNetzA)<sup>2</sup> anzeigen, und die BNetzA ist gemäß § 19 [SigG] zur Aufsicht über die Einhaltung des Signaturgesetzes verpflichtet. Die dem Genehmigungsverfahren ähnliche Akkreditierung kann auf freiwilliger Basis erfolgen.

- *Zertifizierungsdiensteanbieter kann Aufgaben an Dritte übertragen*  
In § 4 Abs. 5 [SigG] wurde klargestellt, dass ein ZDA Aufgaben an Dritte übertragen kann, wenn er diese in sein Sicherheitskonzept einbezieht. Erst dadurch wurde das in [Zeun00, Kap. 5.1] skizzierte und heute weit verbreitete<sup>3</sup> Konstrukt der *virtuellen Zertifizierungsstelle* ermöglicht.
- *Zeitstempeldienst wird optional*  
Die Definition des *Zeitstempels* in § 2 Nr. 14 [SigG] wurde technikneutraler gefasst. Außerdem ist ein ZDA nicht gesetzlich dazu verpflichtet *qualifizierte Zeitstempel* auszustellen. Umgekehrt kann er auch auf die komplexen Infrastrukturen zur Ausgabe von *qualifizierten Zertifikaten* verzichten und nur Zeitstempel ausstellen.
- *Haftung und Deckungsvorsorge*  
Die in Artikel 6 der Richtlinie [1999/93/EG] geforderten Haftungsregelungen fehlten in [SigG97] gänzlich und wurden in § 11 [SigG] neu eingeführt. In diesem Zusammenhang fordert § 12 [SigG] nunmehr auch eine Deckungsvorsorge.
- *Datenschutzregelungen auch für nicht-qualifizierte Zertifikate*  
Die Datenschutzregelungen in § 14 [SigG] gelten – anders als nach § 12 [SigG97] – für alle Aussteller von Zertifikaten.
- *Herstellererklärung ist für bestimmte Komponenten ausreichend*  
Gemäß § 14 Abs. 3 [SigG97] mussten alle technischen Komponenten im Umfeld der digitalen Signatur gemäß Signaturgesetz geprüft und bestätigt sein. Dies gilt nunmehr nur noch für akkreditierte Anbieter (§ 15 Absatz 7 Satz 1 [SigG]) sowie allgemein für die sicheren Signaturerstellungseinheiten (§ 17 Absatz 1 [SigG]) und die durch den ZDA für die Erzeugung und Übertragung von Schlüsseln genutzten Komponenten (§ 17 Absatz 4 [SigG]). Umgekehrt genügt also für manche Komponenten, wie z. B. für *Signaturanwendungskomponenten* im *angezeigten Betrieb*, eine *Herstellererklärung*.
- *Maßnahmenkataloge haben nur noch empfehlenden Charakter*  
Die aus [BSI97] abgeleiteten Maßnahmenkataloge für Zertifizierungsstellen gemäß § 12 Abs. 2 [SigV97] [MKatZS] und technische Komponenten § 16 Abs. 6 [SigV97] [MKatTK] müssen nicht mehr zwingend beachtet werden, sondern besitzen seither nur empfehlenden Charakter.
- *Bußgeldvorschriften*  
Mit § 21 [SigG] werden Bußgeldvorschriften für Verstöße gegen bestimmte Regularien des Signaturgesetzes eingeführt.
- *Internationale Aspekte*  
Schließlich wurden die in Artikel 7 der Richtlinie [1999/93/EG] geforderten Regelungen für die internationale Anerkennung ausländischer Signaturen und Produkte in § 23 [SigG]

---

<sup>2</sup>Mit dem in Kraft treten des [EnWGAendG] am 13. Juli 2005 wurde der Aufgabenbereich der *Regulierungsbehörde für Telekommunikation und Post (RegTP)* erweitert und eine Umbenennung in „*Bundesnetzagentur*“ (BNetzA) durchgeführt.

<sup>3</sup>Während heute mehr als fünfundzwanzig akkreditierte Zertifizierungsdiensteanbieter qualifizierte Zertifikate ausstellen (vgl. [BNetzA-ZDA]), erfolgt der operative Betrieb der dafür notwendigen Systeme in lediglich fünf *Trust-Centern*

## 2 Rechtliche Rahmenbedingungen

neu eingeführt.

### **2.1.1.4 Weitere Entwicklungen**

Kurz nach dem Inkrafttreten des novellierten Signaturgesetzes wurde mit der Anpassung der Formvorschriften im Privatrecht [FormAnpG] die in Artikel 5 Abs. 1 der Richtlinie [1999/93/EG] geforderte Gleichstellung der *qualifizierten elektronischen Signatur* mit der handschriftlichen Unterschrift umgesetzt (vgl. *Abschnitt 2.2.1*). In diesem Zusammenhang wurde in § 292a [ZPO] (Zivilprozessordnung) der Anscheinsbeweis für qualifizierte elektronische Signaturen eingeführt (vgl. *Abschnitt 2.2.2*).

Im Jahr 2002 erfolgte mit dem Verwaltungsverfahrensänderungsgesetz [VerwVfAendG] dann die entsprechende Anpassung der Formvorschriften im öffentlichen Bereich.

Durch das erste Signaturgesetz-Änderungsgesetz [SigGAendG] vom 4. Januar 2005 wurden einige Präzisierungen und vor allem Verfahrenserleichterungen bei der Ausgabe von *qualifizierten Zertifikaten* in das Signaturgesetz [SigG] eingearbeitet.

So kann ein ZDA mit Einwilligung des Antragstellers zur Registrierung bereits zu einem früheren Zeitpunkt erfasste Daten zur Registrierung verwenden (vgl. § 5 Abs. 1 [SigG]), was die Ausgabe kombinierter Signatur- und Bankkarten durch Banken erleichtert. Darüber hinaus kann die Belehrung des Antragstellers nunmehr auch in Textform statt schriftlich erfolgen (vgl. § 6 Abs. 3 [SigG]), wodurch der Einsatz elektronischer Kommunikationsformen vereinfacht wird.

Die jüngsten Gesetzesänderungen mit besonderer Relevanz für den Einsatz der elektronischen Signatur enthält das Justizkommunikationsgesetz [JKomG] vom 22. März 2005. Die hierin enthaltenen Änderungen der Prozessordnungen sowie des Beurkundungsgesetzes schaffen die Voraussetzungen für einen umfassenden Einsatz elektronischer Kommunikationsformen auf Basis der elektronischen Form im gesamten Justizbereich sowie bei Beurkundungen. Darüber hinaus wurden die Regelungen zur Beweisführung bei Einsatz der elektronischen Form modifiziert (vgl. *Abschnitt 2.2.2*). Eine Übersicht über die durch das [JKomG] eingeführten Änderungen findet sich in [Vief05].

### **2.1.2 Grundlegende Bestimmungen**

Wie in *Abbildung 2* visualisiert, definiert das Signaturgesetz in § 2 [SigG] die „elektronische Signatur“, die „fortgeschrittene elektronische Signatur“ und die „qualifizierte elektronische Signatur“.

## 2 Rechtliche Rahmenbedingungen

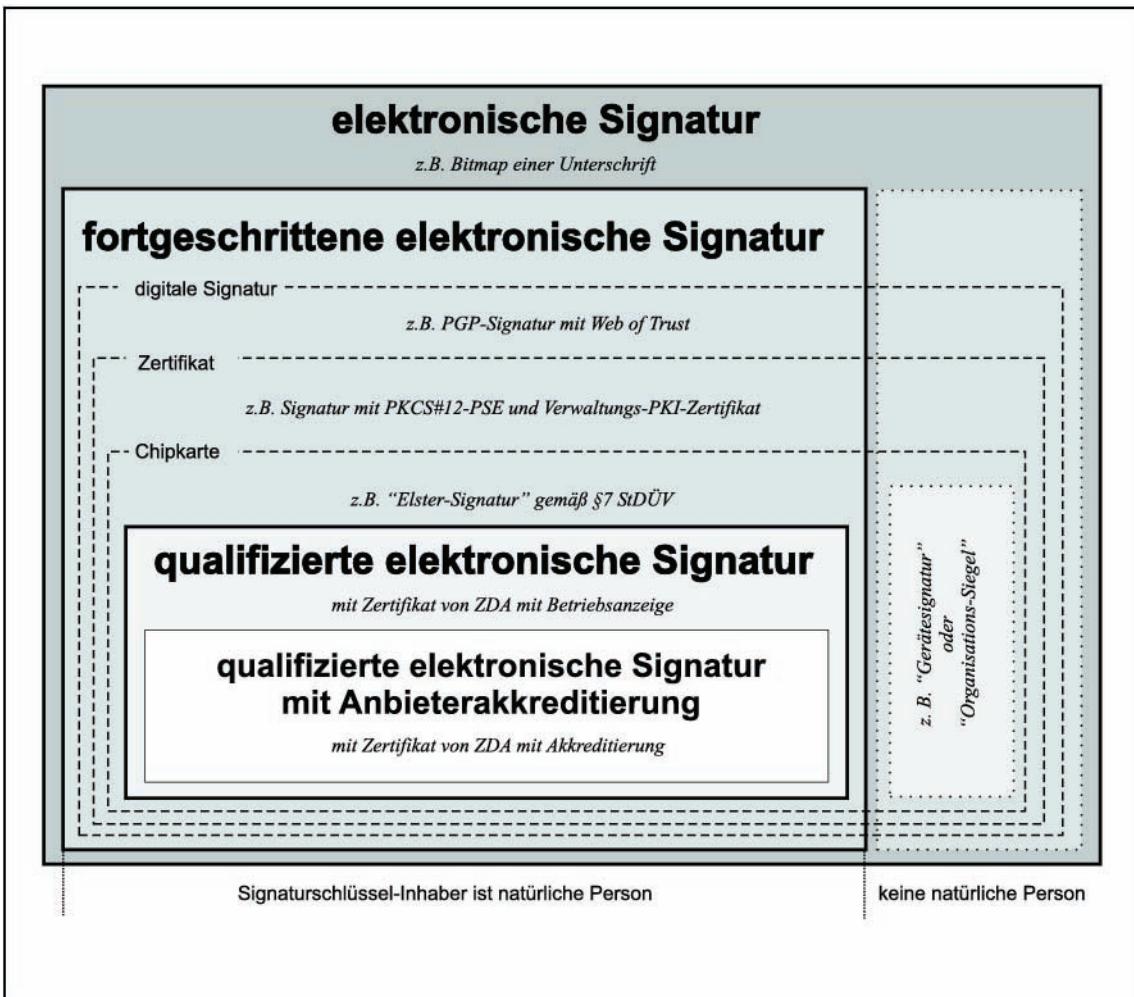


Abbildung 2: Ausprägungen der elektronischen Signatur

Gemäß § 2 [SigG] sind

1. „*elektronische Signaturen*“ Daten in elektronischer Form, die anderen elektronischen Daten beigefügt oder logisch mit ihnen verknüpft sind und die zur *Authentifizierung* dienen,
2. „*fortgeschrittene elektronische Signaturen*“ elektronische Signaturen nach Nummer 1, die
  - ausschließlich dem Signaturschlüssel-Inhaber zugeordnet sind,
  - die Identifizierung des Signaturschlüssel-Inhabers ermöglichen,
  - mit Mitteln erzeugt werden, die der Signaturschlüssel-Inhaber unter seiner alleinigen Kontrolle halten kann, und
  - mit den Daten, auf die sie sich beziehen, so verknüpft sind, dass eine nachträgliche Veränderung der Daten erkannt werden kann,
3. „*qualifizierte elektronische Signaturen*“ elektronische Signaturen nach Nummer 2, die
  - auf einem zum Zeitpunkt ihrer Erzeugung gültigen qualifizierten

## 2 Rechtliche Rahmenbedingungen

---

Zertifikat beruhen und

- b) mit einer sicheren Signaturerstellungseinheit erzeugt werden.

Demnach ist bereits ein eingescanntes Bitmap einer eigenhändigen Unterschrift eine (einfache) *elektronische Signatur*.

Bei der *fortgeschrittenen elektronischen Signatur* kann grob danach unterschieden werden, ob sie unter Verwendung *asymmetrischer Kryptoalgorithmen* erzeugt wird oder nicht. Ist dies der Fall, so spricht man von einer *digitalen Signatur* (vgl. Abschnitt 3.1.1)<sup>4</sup>.

Bei der digitalen Signatur kann die *Authentizität* des *öffentlichen Schlüssels* durch X.509-Zertifikate oder wie beim PGP-Verfahren durch ein so genanntes „*Web of Trust*“ sicher gestellt werden. Außerdem kann entweder ein hardware-basiertes *Personal Security Environment*, beispielsweise in Form einer Chipkarte, oder ein software-basiertes PSE, beispielsweise im PKCS #12-Format, eingesetzt werden. Die Sicherheit der Signaturerzeugung hängt dabei im Wesentlichen vom Nutzerverhalten ab, da dieser die hierzu verwendeten Mittel unter seiner Kontrolle halten kann.

Die *qualifizierte elektronische Signatur* ist eine besondere Form der fortgeschrittenen elektronischen Signatur – bei ihr werden *digitale Signaturen* unter Verwendung von *sicheren Signaturerstellungseinheiten (SSEE)* und *qualifizierten Zertifikaten* erzeugt. Durch die erhöhte technische Sicherheit der SSEE ist die Sicherheit der Signaturerstellung dabei nur noch in geringerem Maße vom Nutzerverhalten abhängig. Bei qualifizierten elektronischen Signaturen unterscheidet das Gesetz außerdem danach, ob die Signaturerstellungseinheiten und Zertifikate von *Zertifizierungsdiensteanbietern* ausgegeben werden, die ihren Betrieb bei der zuständigen Behörde lediglich *angezeigt* haben oder von ihr *akkreditiert* wurden.

Bei der fortgeschrittenen und qualifizierten elektronischen Signatur muss der Signaturschlüssel-Inhaber gemäß § 2 Nr. 9 [SigG] eine *natürliche Person* sein.

Daneben gibt es Anwendungsfälle, bei denen zwar unter Umständen ein ähnliches Sicherheitsniveau wie bei der qualifizierten elektronischen Signatur benötigt wird, aber der Signaturschlüssel-Inhaber eine juristische Person [SigB05] oder ein technisches Gerät [DHLR05] ist. Hierbei handelt es sich aber *nicht* um fortgeschrittene oder qualifizierte elektronische Signaturen im Sinne des Signaturgesetzes.

„*Zertifizierungsdiensteanbieter*“ (§ 2 Nr. 8 [SigG]) sind natürliche oder juristische Personen, die *qualifizierte Zertifikate* oder *qualifizierte Zeitstempel* ausstellen.

Ein „*qualifiziertes Zertifikat*“ (§ 2 Nr. 7 [SigG]) ist eine elektronische Bescheinigung, die von einem ZDA ausgestellt wird. Durch ein qualifiziertes Zertifikat werden insbesondere Signaturprüfschlüssel einer natürlichen Person zugeordnet und die Identität dieser Person, oder weitere Attribute – etwa eine bestehende Vertretungsmacht für einen Dritten – bestätigt. Die detaillierten Inhalte eines qualifizierten Zertifikates sind in § 7 [SigG] näher spezifiziert.

„*Qualifizierte Zeitstempel*“ (§ 2 Nr. 14 [SigG]) sind elektronische Bescheinigungen eines *Zertifizierungsdiensteanbieters*, der die gesetzlichen Anforderungen (vgl. Abschnitt 2.1.3) erfüllt, dass ihm bestimmte Daten zu einem bestimmten Zeitpunkt vorgelegen haben.

Bei einer „*sicheren Signaturerstellungseinheit*“ (SSEE) (§ 2 Nr. 10 [SigG]) handelt es sich um ein Software<sup>5</sup> - oder Hardware-Produkt zur Speicherung und Anwendung der

---

<sup>4</sup>In diesem Dokument wird der Begriff „digitale Signatur“ als Synonym für eine elektronische Signatur, die mittels *asymmetrischer Kryptoalgorithmen* erzeugt wird, verwendet. Es ist also in der Regel *nicht* die „digitale Signatur“ im Sinne des Signaturgesetzes aus dem Jahr 1997 [SigG97] gemeint.

<sup>5</sup>Bei allen heute existierenden SSEEn handelt es sich um Chipkarten (vgl. [BNetzA-Prod])

## 2 Rechtliche Rahmenbedingungen

---

Signaturschlüssel, das bestimmte Eigenschaften (vgl. § 17 Abs. 1 [SigG] und § 15 Abs. 1 [SigV]) erfüllt, die im Rahmen einer Prüfung nach [ITSEC] E3 (hoch) bzw. [CC] EAL4+ und einer Bestätigung durch eine Stelle gemäß § 18 [SigG] nachgewiesen werden müssen. Insbesondere darf es nicht möglich sein, die Signaturschlüssel zu duplizieren (§ 15 Abs. 1 Satz 4 [SigV]). Deshalb darf auch der legitime Benutzer der SSEE nicht in der Lage sein, seinen eigenen Schlüssel aus dieser auslesen zu können.

### 2.1.3 Regelungen für Zertifizierungsdiensteanbieter

#### 2.1.3.1 Grundlegende Anforderungen

Signaturgesetz [SigG] und Signaturverordnung [SigV] formulieren eine Reihe von Anforderungen an *Zertifizierungsdiensteanbieter*.

Abgesehen von den Datenschutzbestimmungen<sup>6</sup>, die gemäß § 14 Abs. 3 [SigG] auch durch Aussteller von nicht-qualifizierten *Zertifikaten* beachtet werden müssen, beziehen sich alle sonstigen Anforderungen des Signaturgesetzes nur auf Aussteller von *qualifizierten* Zertifikaten und Zeitstempeln.

Dies umfasst insbesondere folgende Aspekte:

- Sicherheitskonzept (§ 4 Abs. 2 [SigG] und § 2 [SigV]),
- Nachweisliche Zuverlässigkeit und Fachkunde (§ 4 Abs. 2 [SigG] und § 5 Abs. 3 [SigV]),
- Zuverlässige Identifikation des Antragstellers (§ 5 Abs. 1 Satz 1 [SigG] und § 3 Abs. 1 [SigV]),
- Betrieb eines hochverfüglichen Verzeichnisdienstes (§ 5 Abs. 1 Satz 2 [SigG]),
- Behandlung von Attributen des Signaturschlüsselinhabers:
  - Bestätigung der Attribute (§ 5 Abs. 2 [SigG] und § 3 Abs. 2 [SigV]),
  - Sperrung der Attribute (§ 8 Abs. 2 [SigG]),
- Behandlung von Pseudonymen (§ 5 Abs. 3 [SigG] und § 14 Abs. 2 [SigG]),
- Unterrichtung (§ 6 [SigG]),
- Betrieb eines hochverfüglichen Sperrdienstes (§ 8 [SigG] und § 7 [SigV]),
- Dokumentation (§ 10 [SigG] und § 8 [SigV]),
- Haftung (§ 11 [SigG]),
- Deckungsvorsorge (§ 12 [SigG] und § 9 [SigV]),
- Einsatz von geprüften „Produkten für qualifizierte elektronische Signaturen“<sup>7</sup>, die eine Herstellererklärung oder Bestätigung aufweisen (§ 17 [SigG], § 15 [SigV] und Anlage 1 [SigV]).

Die Summe der Anforderungen soll sicherstellen, dass die herkömmlichen Funktionen der eigenhändigen Unterschrift (vgl. *Abschnitt 2.2.1*) möglichst gleichwertig in die elektronische Welt übertragen werden können. Diese in technischer und organisatorischer Hinsicht teilweise

---

<sup>6</sup>Beispielsweise wird in § 14 Abs. 1 [SigG] klar gestellt, dass ein ZDA personenbezogene Daten nur beim unmittelbar Betroffenen selbst erheben darf oder eine explizite Einwilligung für die Erhebung von Daten bei einem Dritten vorliegen muss. Außerdem wird in § 14 Abs. 2 [SigG] festgelegt, in welchen Fällen ein Pseudonym aufgedeckt werden darf und dass der Betroffene darüber in Kenntnis zu setzen ist.

<sup>7</sup>Gemäß § 2 Nr. 13 [SigG] umfasst dies sichere Signaturerstellungseinheiten, Signaturanwendungskomponenten und technische Komponenten für Zertifizierungsdienste, wie z.B. Komponenten für Verzeichnis- und Zeitstempeldienst sowie ggf. Schlüsselgeneratoren.

## 2 Rechtliche Rahmenbedingungen

anspruchsvollen Anforderungen müssen vom ZDA nicht zwangsläufig selbst realisiert werden. Gemäß § 4 Abs. 5 [SigG] kann ein ZDA – unter Einbeziehung in sein Sicherheitskonzept – auch Aufgaben an Dritte übertragen. Dadurch ist der Aufbau von „*virtuellen Zertifizierungsstellen*“ möglich, bei denen eine Organisation (z.B. Berufskammer oder Sozialversicherungsträger) als Zertifizierungsdiensteanbieter auftritt, aber nahezu alle Pflichten des Signaturgesetztes an einen Dritten, beispielsweise den Betreiber eines *Trust-Centers*, delegiert.

Die Erfüllung der Anforderungen wird durch eine Aufsichtsbehörde überwacht. Gemäß § 3 [SigG] kommt diese Aufgabe der *Bundesnetzagentur* (BNetzA) zu. Nach § 21 Abs. 3 [SigG] ist die BNetzA auch die Verwaltungsbehörde im Sinne des § 36 Abs. 1 Nr. 1 des Gesetzes über Ordnungswidrigkeiten.

### **2.1.3.2 Anzeige oder Akkreditierung**

Gemäß § 4 Abs. 3 [SigG] muss der Betrieb eines Zertifizierungsdienstes zumindest der zuständigen Behörde *angezeigt* werden, wobei die Anforderungen aus § 1 [SigV] zu berücksichtigen sind. Alternativ dazu kann bei der BNetzA ein Antrag auf freiwillige *Akkreditierung* gemäß § 15 [SigG] gestellt werden, der nach § 11 Abs. 1 Satz 2 [SigV] die *Anzeige* ersetzt. Zusätzlich zu den oben aufgeführten Anforderungen, die jeder ZDA erfüllen muss, ist die Akkreditierung mit folgenden zusätzlichen Pflichten verbunden:

- Prüfung und Bestätigung des Sicherheitskonzeptes (§ 15 Abs. 2 [SigG]), die gemäß § 11 Abs. 2 [SigV] mindestens alle drei Jahre zu wiederholen ist.
- Einsatz von geprüften und bestätigten Produkten (§ 15 Abs. 7 Satz 2 Nr. 1 [SigG]). Die Erleichterungen des § 17 Absatz 4 [SigG], wonach insbesondere für Signaturanwendungskomponenten eine Herstellererklärung ausreicht, gelten insofern nicht.
- Ausstellung von qualifizierten Zertifikaten nur für Personen, die nachweislich geprüfte und bestätigte sichere Signaturerstellungseinheiten besitzen (§ 15 Abs. 7 Satz 2 Nr. 2 [SigG]).
- Unterrichtung des Signaturschlüssel-Inhabers über geprüfte und bestätigte Signaturanwendungskomponenten (§ 15 Abs. 7 Satz 2 Nr. 3 [SigG]).
- Nachprüfbarkeit der Zertifikate und Aufbewahrung der Dokumentation über 30 Jahre nach Ablauf der Gültigkeit des Zertifikates (§ 4 Abs. 2 [SigV]); beim lediglich angezeigten Betrieb muss der ZDA dies nur über fünf Jahre nach Ablauf der Gültigkeit des Zertifikates gewährleisten.

Auf der anderen Seite kommt der ZDA durch die freiwillige Akkreditierung in den Genuss einiger Vorteile:

## 2 Rechtliche Rahmenbedingungen

---

- Der ZDA erhält ein Gütesiegel der BNetzA als Ausdruck der nachgewiesenen Sicherheit (§ 15 Abs. 1 [SigG]).
- Die BNetzA stellt dem akkreditierten ZDA die für seine Tätigkeit notwendigen qualifizierten Zertifikate aus (§ 16 Abs. 1 [SigG]).
- Durch § 15 Abs. 6 [SigG] ist die langfristige Aufbewahrung der Dokumentation des ZDA und dadurch insbesondere auch der qualifizierten Zertifikate sowie etwaiger Sperrungen (§ 8 Abs. 2 Nr. 6 und Nr. 7 [SigG]) staatlich garantiert. Da die BNetzA bei Vorliegen eines berechtigten Interesses gemäß § 13 Abs. 2 [SigG] darüber entsprechende Auskünfte erteilt, sofern dies technisch ohne unverhältnismäßig großen Aufwand möglich ist, können Signaturen bis zu dreißig Jahre nach Ablauf des Jahres, in dem die Gültigkeit des Zertifikates endet, noch geprüft werden.

### 2.1.4 Anwendungsbezogene Aspekte

Während das Signaturgesetz im Wesentlichen die Rechte und Pflichten der *Zertifizierungsdiensteanbieter* reguliert, werden Aspekte der Nutzung der elektronischen Signatur durch den Anwender im Signaturgesetz lediglich gestreift.

Beim Endanwender sind zum Einsatz der qualifizierten elektronischen Signatur zwei Komponenten nötig:

- Eine sichere Signaturerstellungseinheit (SSEE) (§ 2 Nr. 10 [SigG]) auf der, möglicherweise neben weiteren kryptographischen Schlüsseln und Zertifikaten zur Authentifizierung und Verschlüsselung, ein Signaturschlüssel und ein zugehöriges qualifiziertes Zertifikat gespeichert sind.
- Eine Signaturanwendungskomponente (§ 2 Nr. 11 [SigG]), die das Bindeglied zwischen existierenden Anwendungssystemen, der SSEE und den zentralen Diensten des ZDA darstellt und
  - Daten dem Prozess der Erzeugung oder Prüfung *qualifizierter elektronischer Signaturen* zuführt oder
  - qualifizierte elektronische Signaturen oder qualifizierte Zertifikate *prüft* und die Ergebnisse anzeigt.

Das Signaturgesetz fordert in § 17 Abs. 1 [SigG], dass die sichere Signaturerstellungseinheit vor unberechtigter Nutzung zu schützen ist. § 15 Abs. 1 [SigV] fordert hierfür eine Identifikation „durch Besitz und Wissen oder durch Besitz und ein oder mehrere biometrische Merkmale“. Da bislang keine Implementierungen biometrischer Verfahren bekannt sind, die die Anforderungen des Signaturgesetzes (vgl. Anlage 1 zu [SigV]) nachweislich erfüllen, werden für qualifizierte elektronische Signaturen in der Praxis immer *Personal Identification Numbers* (PIN) als *Identifikationsdaten* eingesetzt.

Die SSEE mit den darauf gespeicherten Schlüsseln und Zertifikaten sowie die zugehörige PIN für die Freischaltung der SSEE erhält der Endanwender nach entsprechender Identifikation (§ 5 Abs. 1 [SigG] und § 3 Abs. 1 [SigV]) und Unterrichtung (§ 6 [SigG] und § 6 [SigV]) vom ZDA. Im Rahmen dieser Unterrichtung muss der ZDA den Anwender über notwendige Sicherheitsvorkehrungen beim Einsatz der elektronischen Signatur und die regelmäßige rechtliche Gleichstellung der *qualifizierten elektronischen Signatur* mit der eigenhändigen Unterschrift unterrichten (vgl. Abschnitt 2.2.1).

Bei der Realisierung der notwendigen Sicherheitsmaßnahmen spielt die *Signaturanwendungskomponente* eine zentrale Rolle. Die Anforderungen an diese sind in § 17 Abs. 2 [SigG] festgehalten:

„Für die Darstellung zu signierender Daten sind Signaturanwendungskomponenten

## 2 Rechtliche Rahmenbedingungen

---

erforderlich, die die Erzeugung einer *qualifizierten elektronischen Signatur* vorher eindeutig anzeigen und feststellen lassen, auf welche Daten sich die Signatur bezieht. Für die Überprüfung signierter Daten sind Signaturanwendungskomponenten erforderlich, die feststellen lassen,

1. auf welche Daten sich die Signatur bezieht,
2. ob die signierten Daten unverändert sind,
3. welchem Signaturschlüssel-Inhaber die Signatur zuzuordnen ist,
4. welche Inhalte das *qualifizierte Zertifikat*, auf dem die Signatur beruht, und zugehörige qualifizierte *Attribut-Zertifikate* aufweisen und
5. zu welchem Ergebnis die Nachprüfung von *Zertifikaten* nach § 5 Abs. 1 Satz 2 geführt hat.

Signaturanwendungskomponenten müssen nach Bedarf auch den Inhalt der zu signierenden oder signierten Daten hinreichend erkennen lassen. Die Signaturschlüssel-Inhaber sollen solche Signaturanwendungskomponenten einsetzen oder andere geeignete Maßnahmen zur Sicherheit qualifizierter elektronischer Signaturen treffen.“

§ 15 Abs. 2 und Abs. 4 [SigV] machen weitere Angaben zu den Anforderungen an solche Signaturanwendungskomponenten:

- „(2) Signaturanwendungskomponenten nach § 17 Abs. 2 des Signaturgesetzes müssen gewährleisten, dass
1. bei der Erzeugung einer qualifizierten elektronischen Signatur
    - a) die Identifikationsdaten nicht preisgegeben und diese nur auf der jeweiligen *sicheren Signaturerstellungseinheit* gespeichert werden,
    - b) eine Signatur nur durch die berechtigt signierende Person erfolgt,
    - c) die Erzeugung einer Signatur vorher eindeutig angezeigt wird und
  2. bei der Prüfung einer qualifizierten elektronischen Signatur
    - a) die Korrektheit der Signatur zuverlässig geprüft und zutreffend angezeigt wird und
    - b) eindeutig erkennbar wird, ob die nachgeprüften *qualifizierten Zertifikate* im jeweiligen *Zertifikat-Verzeichnis* zum angegebenen Zeitpunkt vorhanden und nicht gesperrt waren.“

Nach § 15 Abs. 4 [SigV] müssen „sicherheitstechnische Veränderungen an technischen Komponenten (...) für den Nutzer erkennbar werden.“ Während Zertifizierungsdiensteanbieter zur Verwendung von geprüften und mit einer *Herstellererklärung* bzw. *Bestätigung* versehenen Signaturanwendungskomponenten verpflichtet sind, wird dem Anwender in § 17 Abs. 2 [SigG] die Verwendung geeigneter Signaturanwendungskomponenten lediglich *empfohlen*. Die Begründung des Regierungsentwurfs [SigGBeg] macht deutlich, dass die Nutzung „geeigneter Signaturanwendungskomponenten in das Ermessen der Signaturschlüssel-Inhaber gestellt bleibt“ und sagt zu der Formulierung „soll“, dass damit klargestellt wird, dass „die Verwendung von geeigneten Signaturanwendungskomponenten nicht Voraussetzung für die Erzeugung einer qualifizierten elektronischen Signatur ist“. Einer Signatur (auch einer qualifizierten) ist es nach deren Erzeugung ohnehin nicht mehr anzusehen, ob sie mit einer „solchen“ Anwendung erstellt wurde, oder mit einer anderen.

Es liegt im Ermessen eines Signaturschlüssel-Inhabers, eine nach dem Signaturgesetz *geprüfte und bestätigte* Anwendung zur Erzeugung einer qualifizierten elektronischen Signatur zu verwenden.

## 2 Rechtliche Rahmenbedingungen

---

In Erfüllung ihrer Verpflichtung aus § 15 Abs. 7 Nr. 3 [SigG], „die Signaturschlüssel-Inhaber im Rahmen des § 6 Abs. 1 über nach Satz 1 geprüfte und bestätigte Signaturanwendungskomponenten zu unterrichten“, liefern die *akkreditierten* Zertifizierungsdiensteanbieter dem Anwender entweder eine Liste der derzeit verfügbaren geprüften und bestätigten Signaturanwendungskomponenten oder verweisen auf die durch die BNetzA geführte Liste [BNetzA-Prod]. Hierdurch entsteht allerdings keine Verpflichtung der Anwender, diese geprüften und bestätigten Anwendungen tatsächlich einzusetzen.

Während die Signaturgesetzgebung dem Anwender bzgl. der einzusetzenden Signaturanwendungskomponenten also einen gewissen Freiraum zubilligt, so sollte der Anwender dennoch die Empfehlungen des Signaturgesetzes bei der Auswahl seiner Signaturanwendungskomponenten entsprechend berücksichtigen. Wegen des erhöhten Missbrauchspotenzials gilt dies insbesondere bei der automatisierten Erzeugung qualifizierter elektronischer Signaturen – der so genannten „Massensignatur“ (vgl. *Abschnitt 4.4*).

Während nach der Novellierung des Signaturgesetzes (vgl. *Abschnitt 2.1.1.3*) zur Erstellung *qualifizierter Zeitstempel* nicht zwingend *qualifizierte elektronische Signaturen* eingesetzt werden müssen, so ist dies in der Praxis heute aber immer der Fall. Da Komponenten zum Anfordern und Prüfen solcher qualifizierten Zeitstempel *Signaturanwendungskomponenten* im Sinne des Signaturgesetzes sind, sollten die Empfehlungen für *Signaturanwendungskomponenten* in § 17 [SigG] und § 15 [SigV] auch bei der Anforderung und der Prüfung von *qualifizierten Zeitstempeln* berücksichtigt werden.

## 2.2 Relevanz der Signatur für den elektronischen Rechtsverkehr

Gemäß Artikel 5 der Richtlinie [1999/93/EG] mussten die EU-Mitgliedstaaten dafür Sorge tragen, dass qualifizierte elektronische Signaturen der eigenhändigen Unterschrift rechtlich gleich gestellt und als Beweismittel vor Gericht zugelassen sind. Während in der deutschen Zivilprozessordnung elektronische Signaturen als Objekte des Augenscheins ohnehin als Beweismittel vor Gericht zugelassen waren, bedurfte es für die vorgeschriebene Gleichstellung der qualifizierten elektronischen Signatur mit der eigenhändigen Unterschrift zusätzlicher Anpassungen.

Anders als bei der österreichischen Umsetzung der Signaturrichtlinie (vgl. § 4 Abs. 1 [OeSigG]) hat der Deutsche Gesetzgeber die erforderliche Gleichstellung aber nicht pauschal im Signaturgesetz, sondern einzeln in den jeweiligen Fachgesetzen geregelt. Auf das Gesetz zur Anpassung der Formvorschriften des Privatrechts an den modernen Geschäftsverkehr [FormAnpG] und das Gesetz zur Änderung der verwaltungsverfahrensrechtlichen Vorschriften [VerwVfAendG] soll in *Abschnitt 2.2.1* näher eingegangen werden.

Daneben wurde durch das [FormAnpG] der gesetzliche Anscheinsbeweis des § 292a [ZPO] eingeführt, wodurch einer mit einer qualifizierten elektronischen Signatur versehenen Willenserklärung eine erhöhte Beweiskraft vor Gericht verliehen wird. Auf die Aspekte der Beweiskraft von elektronischen Signaturen soll in *Abschnitt 2.2.2* näher eingegangen werden.

Bei der Anwendung der elektronischen Signatur sind stets auch die weiteren durch das jeweilige Fachgesetz bestimmten Rahmenbedingungen zu beachten. Beispielsweise sind bei der elektronischen Übermittlung von Rechnungen die in *Abschnitt 5.2.1* skizzierten besonderen Regelungen des Umsatzsteuergesetzes und der dazu ergangenen Verwaltungsvorschriften zu beachten.

### 2.2.1 Formvorschriften

Willenserklärungen können nach dem deutschen Rechtssystem grundsätzlich formfrei

## 2 Rechtliche Rahmenbedingungen

---

abgegeben werden. Dieser Grundsatz trägt den Gegebenheiten des modernen Wirtschaftsverkehrs Rechnung. Sieht der Gesetzgeber aber für bestimmte *Rechtsgeschäfte* eine besondere Form vor, so legt § 125 [BGB] für den privatrechtlichen Bereich fest, dass ein Rechtsgeschäft, das der gesetzlich geforderten – oder durch Rechtsgeschäft bestimmten – Form nicht genügt, nichtig ist.

Sinn einer gesetzlichen Formvorschrift kann es beispielsweise sein (vgl. [FormAnpGBeg]), die Identität des Erklärenden (Identitätsfunktion) oder den Abschluss einer Willenserklärung zu dokumentieren (Abschlussfunktion), die Erklärenden auf mögliche negative Folgen ihrer Erklärungen hinzuweisen (Warnfunktion), Beweise zu sichern (Beweisfunktion), Dritten die Überprüfung zu ermöglichen (Kontrollfunktion) oder die Beratung der Erklärenden durch Rechtskundige, z.B. durch einen Notar, sicherzustellen (Beratungsfunktion).

Vor den im Zuge der Signaturgesetzgebung eingeführten Änderungen der Formvorschriften kannte das Bürgerliche Gesetzbuch [BGB]

- die Schriftform (§ 126 [BGB]), bei der „die Urkunde von dem Aussteller eigenhändig (...) unterzeichnet werden muss.“
- die notarielle Beurkundung (§ 128 [BGB]) eines Vertrages, bei der der „Antrag und sodann die Annahme des Antrags von einem Notar beurkundet wird“ und
- die öffentliche Beglaubigung (§ 129 [BGB]), bei der „die Erklärung schriftlich abfasst und die Unterschrift des Erklärenden von einem Notar beglaubigt werden“ muss.

Im Zuge des Gesetzes zur Anpassung der Formvorschriften an den modernen Rechtsgeschäftsverkehr [FormAnpG] wurden zusätzlich folgende Formen eingeführt:

- Textform (§ 126b [BGB])

Hier muss „die Erklärung in einer *Urkunde* oder auf andere zur dauerhaften Wiedergabe in Schriftzeichen geeignete Weise abgegeben, die Person des Erklärenden genannt und der Abschluss der Erklärung durch Nachbildung der Namensunterschrift oder anders erkennbar gemacht werden.“

Anders als bei der Schriftform sind bei der Textform also die Verkörperung der Erklärung in einer Urkunde sowie die eigenhändige Unterschrift entbehrlich, was zu erheblichen Erleichterungen im Rechtsverkehr führt. Dies betrifft insbesondere die verbraucherschutzrechtlichen Unterrichtungspflichten (§ 312c Abs. 2 [BGB]) etwa im Bereich des Internet-Versandhandels, die nunmehr auch per E-Mail oder durch Anzeige von Internet-Seiten erfüllt werden können.

- Elektronische Form (§ 126a [BGB])

Hier muss „der Aussteller der Erklärung dieser seinen Namen hinzufügen und das elektronische Dokument mit einer *qualifizierten elektronischen Signatur* nach dem Signaturgesetz versehen“.

Nach § 126 Abs. 3 [BGB] kann die schriftliche Form „durch die elektronische Form ersetzt werden, wenn sich nicht aus dem Gesetz ein anderes ergibt.“

Diese in Artikel 5 Abs. 1 a) der Richtlinie [1999/93/EG] geforderte Gleichstellung der *qualifizierten elektronischen Signatur* mit der handschriftlichen Unterschrift wurde in [FormAnpGBeg] mit der „Funktionsäquivalenz“ der *elektronischen Form* und der *Schriftform* begründet.

Die Funktionen der Schriftform sind, wie in [FormAnpGBeg] und [BrTe01] erläutert, im Einzelnen wie folgt:

- „Abschlussfunktion

Die eigenhändige Unterschrift ist der räumliche Abschluss eines Textes und bringt zum Ausdruck, dass die *Willenserklärung* abgeschlossen ist. Dadurch wird das Stadium der

## 2 Rechtliche Rahmenbedingungen

---

Vorverhandlungen und des bloßen Entwurfs von dem der rechtlichen Bindung abgegrenzt.

- Perpetuierungsfunktion

Das Schriftformerfordernis führt dazu, dass die Unterschrift und vor allem der Text fort dauernd und lesbar in einer *Urkunde* wiedergegeben werden und einer dauerhaften Überprüfung zugänglich sind. Hierdurch wird gewährleistet, dass eine Information über die Erklärung nicht nur flüchtig möglich ist und die Erklärung dokumentiert werden kann.

- Identitätsfunktion

Durch die eigenhändige Namensunterschrift wird zum einen der Aussteller der *Urkunde* erkennbar. Dariüber hinaus soll der Erklärende identifiziert werden können, weil die unverwechselbare Unterschrift eine unzweideutige Verbindung zur Person des Unterzeichners herstellt.

- Echtheitsfunktion

Die räumliche Verbindung der Unterschrift mit der *Urkunde*, die den Erklärungstext enthält, stellt einen Zusammenhang zwischen Dokument und Unterschrift her. Hierdurch soll gewährleistet werden, dass die Erklärung inhaltlich vom Unterzeichner herrührt.

- Verifikationsfunktion

Die Verifikationsfunktion steht im engen Zusammenhang mit der Echtheits- und der Identitätsfunktion. Sie wird dadurch erreicht, dass der Empfänger eines Dokuments die Möglichkeit hat zu überprüfen, ob die unverwechselbare Unterschrift echt ist, z. B. durch einen Unterschriftenvergleich.

- Beweisfunktion

Die eigenhändige Unterschrift unter einem fixierten Text dient dem Interesse an der Beweisführung und Offenlegung des Geschäftsinhalts und führt zu dauerhafter Klarheit. Die Schriftform erleichtert dem Beweispflichtigen seine Beweisführung, sofern der Beweisgegner die Echtheit der Unterschrift nicht bestreitet (§ 439 Abs. 1, 2, § 440 Abs. 1 [ZPO]).

- Warnfunktion

Durch den bewussten Akt des Unterzeichnens wird der Erklärende hingewiesen auf die erhöhte rechtliche Verbindlichkeit und die persönliche Zurechnung der unterzeichneten Erklärung. Hierdurch soll er vor übereilten Rechtsgeschäften geschützt werden.“

Wie in der Begründung zum Formanpassungsgesetz [FormAnpGBeg] näher erläutert, geht der Gesetzgeber von einer (weitgehenden) Funktionsäquivalenz durch den Einsatz der *qualifizierten elektronischen Signatur* aus. Beispielsweise sind die zuverlässige Identifizierung des Antragstellers gemäß § 5 Abs. 1 Satz 1 [SigG] und § 3 Abs. 1 [SigV] sowie die anspruchsvollen Auflagen für die Signaturerstellungseinheit und den Umgang mit den *Identifikationsdaten*<sup>8</sup> notwendig, um die *Identitätsfunktion* möglichst gut abzubilden. In ähnlicher Weise müssen Aspekte der Nachsignatur gemäß § 17 [SigV] (vgl. *Abschnitt 4.6*) beachtet werden, um die *Perpetuierungsfunktion* zu erfüllen. Die Unterrichtung gemäß § 6 [SigG] zielt insbesondere auf die Unterstützung der *Abschlussfunktion* und der *Warnfunktion* ab.

Im öffentlichen Bereich existieren ähnliche Formvorschriften, die durch das Verwaltungsverfahrensänderungsgesetz [VerwVfAendG] maßgeblich angepasst wurden, so

---

<sup>8</sup>Da zusammen mit der Signaturerstellungseinheit auch die PIN an andere Personen weitergegeben oder missbraucht werden könnte, und damit rechtswirksame Signaturen erstellt werden können, erscheint für eine verlässliche Verifikation der Einsatz hinreichend sicherer biometrischer Verfahren vorteilhaft (vgl. [Albr03]).

## 2 Rechtliche Rahmenbedingungen

---

dass die qualifizierte elektronische Signatur auch im öffentlichen Recht die eigenhändige Unterschrift ersetzen kann.

Beispielsweise ermöglicht der durch Artikel 1 [VerwVfAendG] eingeführte § 3a Abs. 1 Verwaltungsverfahrensgesetz ([VwVfG]) die Übermittlung elektronischer Dokumente, soweit der Empfänger hierfür einen Zugang eröffnet. Außerdem kann – analog zu § 126 Abs. 3 [BGB] in Verbindung mit § 126a [BGB] – gemäß § 3a Abs. 2 Satz 1 und 2 [VwVfG] die *Schriftform*, „soweit nicht durch Rechtsvorschrift etwas anderes bestimmt ist, durch die elektronische Form ersetzt werden. In diesem Fall ist das elektronische Dokument mit einer qualifizierten elektronischen Signatur nach dem Signaturgesetz zu versehen.“

Analoge Regelungen für den allgemeinen Teil des Sozialgesetzbuches [SGBI] und der Abgabenordnung [AO] wurden durch Artikel 2 und 4 [VerwVfAendG] eingeführt. Gemäß § 36a [SGBI] und § 87a [AO] kann auch in diesen Bereichen die eigenhändige Unterschrift durch die qualifizierte elektronische Signatur ersetzt werden.

Erwähnenswert ist, dass durch § 87a Abs. 6 [AO] bis Ende 2005 auch die „qualifizierte elektronische Signatur mit Einschränkungen“ gemäß § 7 [StDUeV] (Steuerdaten-Übermittlungsverordnung) als Ersatz der eigenhändigen Unterschrift – insbesondere für Zwecke der Elektronischen Steuererklärung (ELSTER) (vgl. *Abschnitt 5.1.1*) – eingesetzt werden konnte (siehe auch [Ross03b]).

Ein Überblick zum [VerwVfAendG] und der dadurch ermöglichten Modernisierung der Verwaltungsverfahren finden sich auch in [Ross03a].

### 2.2.2 Beweiskraft

Die Europäische Richtlinie [1999/93/EG] fordert in Artikel 5 Abs. 1 b) und Abs. 2, dass (qualifizierte) elektronische Signaturen als Beweismittel vor Gericht zugelassen sein müssen.

Da elektronische Dokumente seit jeher dem Beweis durch Augenschein<sup>9</sup> zugänglich waren, bedurfte es insofern keiner Änderungen, um die Anforderungen der Richtlinie zu erfüllen. Dies gilt sowohl für unsignierte, als auch für elektronisch signierte Dokumente unabhängig von der Stufe der eingesetzten Signatur.

Allerdings wurde mit dem [FormAnpG] der Paragraph 292a in die Zivilprozessordnung eingefügt, durch den sich eine Beweiserleichterung für *Willenserklärungen* ergibt, die der *elektronischen Form* gemäß § 126a [BGB] genügen. Nach § 292a [ZPO] konnte „der Anschein der Echtheit einer in elektronischer Form vorliegenden Willenserklärung, der sich auf Grund der Prüfung nach dem SigG ergibt, nur durch Tatsachen erschüttert werden, die ernstliche Zweifel daran begründen, dass die Erklärung mit dem Willen des Signaturschlüsselhabers abgegeben worden ist.“ Diese Regelung war an die im ursprünglichen Signaturgesetz [SigG97] formulierte „Sicherheitsvermutung“<sup>10</sup> der digitalen Signatur gemäß Signaturgesetz angelehnt.

Von einer beweisrechtlichen Gleichstellung eines elektronischen Dokumentes mit einer Urkunde wurde bei der Anpassung der Formvorschriften im Privatrecht im Jahre 2001 dagegen bewusst abgesehen (vgl. [FormAnpGBeg]), da einem elektronischen Dokument im Vergleich zur Urkunde die Verkörperung fehle, so dass es nicht unmittelbar ohne technische Hilfsmittel lesbar sei. Zudem bestehe die Gefahr, dass die Echtheit der (elektronischen)

---

<sup>9</sup>Daneben sieht die Zivilprozessordnung den *Zeugenbeweis* (§§ 373ff [ZPO]), den *Beweis durch Sachverständige* (§§ 402ff [ZPO]), den Beweis durch Urkunden (§§ 415ff [ZPO]) und schließlich den *Beweis durch Parteivernehmung* (§§ 445ff [ZPO]) vor.

<sup>10</sup>vgl. § 1 Abs. 1 [SigG97] und [Ross98]

## 2 Rechtliche Rahmenbedingungen

---

Unterschrift bei der Anwendung des Urkundsbeweises gemäß §§415 f. [ZPO] von einer Partei bestritten werde, so dass sie von der beweisbelasteten Partei zur vollen Überzeugung des Gerichts bewiesen werden müsse (§ 440 Abs. 1 [ZPO]). Bei diesem Beweis seien aber keine Erleichterungen vorgesehen, weshalb die Beweisregeln für private Urkunden dem hohen Beweiswert, der mit dem Einsatz qualifizierter elektronischer Signaturen verbunden ist, nicht gerecht würden.

Weitere Aspekte der Nutzung elektronischer Dokumente als Beweismittel unter Geltung des § 292a [ZPO] erörtern [FGPS02, Jung02].

Inzwischen wurde der Anscheinsbeweis des § 292a [ZPO] durch das Justizkommunikationsgesetz [JKomG] in einen neuen § 371a [ZPO] übernommen, welcher für private und öffentliche elektronische Dokumente nunmehr auch die Regeln für den Urkundsbeweis durch private bzw. öffentliche Urkunden anwendbar macht.

§ 371a [ZPO] (Beweiskraft elektronischer Dokumente) lautet wie folgt:

„(1) Auf private elektronische Dokumente, die mit einer qualifizierten elektronischen Signatur versehen sind, finden die Vorschriften über die *Beweiskraft privater Urkunden* entsprechende Anwendung. Der Anschein der Echtheit einer in elektronischer Form vorliegenden Erklärung, der sich auf Grund der Prüfung nach dem Signaturgesetz ergibt, kann nur durch Tatsachen erschüttert werden, die ernstliche Zweifel daran begründen, dass die Erklärung vom Signaturschlüssel-Inhaber abgegeben worden ist.

(2) Auf elektronische Dokumente, die von einer öffentlichen Behörde innerhalb der Grenzen ihrer Amtsbefugnisse oder von einer mit öffentlichem Glauben versehenen Person innerhalb des ihr zugewiesenen Geschäftskreises in der vorgeschriebenen Form erstellt worden sind (öffentliche elektronische Dokumente), finden die Vorschriften über die *Beweiskraft öffentlicher Urkunden* entsprechende Anwendung. Ist das Dokument mit einer qualifizierten elektronischen Signatur versehen, gilt § 437 entsprechend.“

Es sei angemerkt, dass somit die neu gefasste Beweiserleichterung nicht nur für *Willenserklärungen*, sondern auch für „Wissenserklärungen“, wie z.B. Quittungen, gilt.

Für die Beweiskraft von (*qualifizierten*) *Zeitstempeln* existieren im deutschen Recht keine gesonderten Regelungen, so dass die beweisrechtlichen Vorschriften für elektronische Dokumente anzuwenden sind. Werden Zeitstempel wie beim *Time Stamp Protocol* durch die Signatur von um eine Zeitangabe ergänzte Daten erstellt, so hängt die Beweiskraft insbesondere von der Qualität der Signatur und der Vertrauenswürdigkeit der zur Produktion der Zeitstempel verwendeten Zeitangaben ab. Durch den Einsatz *qualifizierter elektronischer Signaturen* kommt man in den Genuss der beweisrechtlichen Vorteile gemäß § 371a [ZPO], so dass im Streitfall lediglich die Vertrauenswürdigkeit der Zeitinformation in Frage gestellt werden kann. Hier bieten die von *Zertifizierungsdienstanbietern* ausgestellten *qualifizierten Zeitstempel* eine besonders hohe Beweiskraft, da sie durch geprüfte Produkte in einer besonders gesicherten Umgebung erstellt werden. Weitere Betrachtungen zum Beweiswert von Zeitstempeln finden sich in [Hueh04c, Abschnitt 3.3].

Damit der Beweiswert signierter Daten langfristig erhalten werden kann, müssen die in *Abschnitt 4.6* näher erläuterten Mechanismen zur Nachsignatur gemäß § 17 [SigV] eingesetzt werden.

## 2.3 Zusammenfassung der gesetzlichen Anforderungen

Schließlich sollen die wichtigsten Aspekte des Signaturgesetzes hier in kompakter Art und

## 2 Rechtliche Rahmenbedingungen

Weise zusammengefasst werden. Hierbei wird getrennt auf die Anforderungen für Zertifizierungsdiensteanbieter (vgl. *Abschnitt 2.3.1*), Hersteller von Komponenten (vgl. *Abschnitt 2.3.2*) und schließlich Signaturanwender (vgl. *Abschnitt 2.3.3*) eingegangen.

### **2.3.1 ... an Zertifizierungsdiensteanbieter**

Zertifizierungsdiensteanbieter müssen alle Anforderungen der §§ 4 bis 14 sowie § 17 [SigG] oder § 23 [SigG] und der Signaturverordnung [SigV] erfüllen. Was dies im einzelnen umfasst, ist in *Abschnitt 2.1.3* erläutert. Soweit durch die Nichteinhaltung der gesetzlichen Anforderungen Schäden entstehen, hat diese der *ZDA* zu ersetzen. Die in § 21 [SigG] genannten Anforderungen sind darüber hinaus bußgeldbewehrt, wobei schwerwiegende bzw. dauernde Verstöße ggf. auch zu gewerberechtlichen Maßnahmen bis hin zur Betriebsuntersagung führen können.

### **2.3.2 ... an Hersteller von Komponenten**

Hersteller von sicheren Signaturerstellungseinheiten (§ 2 Nr. 10 [SigG]), Signaturanwendungskomponenten (§ 2 Nr. 11 [SigG]) oder technischen Komponenten für Zertifizierungsdienste (§ 2 Nr. 12 [SigG]) müssen bei der Entwicklung und Auslieferung ihrer Produkte die in § 17 [SigG] und § 15 [SigV] definierten Anforderungen erfüllen und dies im Regelfall durch eine *Prüfung und Bestätigung* nachweisen.

Abweichend davon genügt für Signaturanwendungskomponenten sowie Komponenten für den *Verzeichnisdienst* und *Zeitstempeldienst*, die nicht von *akkreditierten Zertifizierungsdiensteanbietern* eingesetzt werden sollen (vgl. § 15 Abs. 7 [SigG]) gemäß § 17 Abs. 4 Satz 2 [SigG] eine *Herstellererklärung*.

Durch Artikel 1 Nr. 8 [SigGAendG] wurden dem § 17 Abs. 4 [SigG] die folgenden beiden Sätze angefügt:

„Der Hersteller hat spätestens zum Zeitpunkt des Inverkehrbringens des Produkts eine Ausfertigung seiner Erklärung in schriftlicher Form bei der Regulierungsbehörde für Telekommunikation und Post zu hinterlegen. Herstellererklärungen, die den Anforderungen des Gesetzes und der Rechtsverordnung nach § 24 entsprechen, werden im Amtsblatt der Regulierungsbehörde für Telekommunikation und Post veröffentlicht.“

Einem Hersteller von Komponenten für die elektronische Signatur, der die Anforderungen des Signaturgesetzes missachtet, drohen – anders als dem *ZDA* (vgl. *Abschnitt 2.3.1*) – keine Bußgelder gemäß § 21 [SigG].

Allerdings ist auch der Hersteller nach den allgemeinen Vorschriften zum Ersatz entstehender Schäden verpflichtet. Insbesondere kann sich ein Hersteller mit Schadensersatzansprüchen gemäß dem Bürgerlichen Gesetzbuch [BGB] oder gemäß dem Produkthaftungsgesetz [ProdHG] konfrontiert sehen.

Darüber hinaus sind auch hier bei besonders schwerwiegenden bzw. dauernden Verstößen auch gewerberechtliche Maßnahmen denkbar.

### **2.3.3 ... an Signaturanwender**

Der Signaturanwender benötigt für den Umgang mit Signaturen zunächst eine *Signaturanwendungskomponente*, welche die Anforderungen des § 17 Abs. 2 [SigG] erfüllen soll. Für die Erzeugung *qualifizierter elektronischer Signaturen* nach [SigG] bedarf es darüber hinaus einer *sicheren Signaturerstellungseinheit*.

## 2 Rechtliche Rahmenbedingungen

Über die weiteren erforderlichen Sicherheitsvorkehrungen wird der Anwender durch den ZDA informiert (§ 6 [SigG]). Deren Nichteinhaltung wird zwar durch das [SigG] selbst nicht sanktioniert; aus den jeweils einschlägigen Fachgesetzen können sich allerdings teils erhebliche Nachteile ergeben.

Beispielsweise ist gemäß § 125 [BGB] ein *Rechtsgeschäft*, das der gesetzlich vorgeschriebenen oder vereinbarten Form ermangelt, nichtig. Wird bei elektronisch übermittelten Rechnungen auf den Einsatz der *qualifizierten elektronischen Signatur* verzichtet, so darf sie nicht zum Vorsteuerabzug gemäß § 15 [UStG] herangezogen werden, wodurch der Rechnungsempfänger erhebliche Verluste erleidet (vgl. *Abschnitt 5.2*). Missachtet ein Sozialversicherungsträger die Vorschriften der Sozialversicherungsrechnungsverordnung [SVRV] und der zugehörigen Verwaltungsvorschrift [SRVwV], so drohen Sanktionen durch das Bundesversicherungsamt als zuständige Aufsichtsbehörde.

Weitere Informationen zu den rechtlichen Rahmenbedingungen für den Einsatz der elektronischen Signatur in verschiedenen Anwendungsgebieten finden sich in *Abschnitt 5*.

## 3 Technische Realisierung

Während die Definition der (fortgeschrittenen) elektronischen Signatur in der Europäischen Signaturrichtlinie [1999/93/EG] und der Deutschen Umsetzung im Signaturgesetz [SigG] technologienutral gestaltet ist und somit Raum für alternative Realisierungen lassen würde, werden sie in der Praxis zumeist<sup>11</sup> als so genannte *digitale Signaturen* unter Verwendung von *asymmetrischen Kryptoalgorithmen* und *Zertifikaten* verwirklicht. Insbesondere bei der *qualifizierten elektronischen Signatur* müssen diese Technologien, in besonders sicherer Art und Weise, zum Einsatz kommen.

Gegenstand dieses Kapitels ist die Erläuterung der kryptographischen Grundlagen der digitalen Signatur in *Abschnitt 3.1* und darauf basierender *Public-Key-Infrastrukturen* in *Abschnitt 3.2*.

### 3.1 Kryptographische Grundlagen

Im Folgenden sollen die wichtigsten kryptographischen Grundlagen der digitalen Signatur erläutert werden. Eine systematische Einführung in die Kryptographie findet sich in [Buch99]. [MOV97] liefert eine umfassende Behandlung der Materie.

#### 3.1.1 Das Prinzip der digitalen Signatur

Zu den wesentlichen Funktionen der Schriftform (vgl. *Abschnitt 2.2.1*), bei der eine Urkunde eigenhändig unterschrieben werden muss, zählt der Nachweis, dass die durch die Urkunde verkörperte Erklärung vom Unterzeichner herrührt (Echtheitsfunktion) und dies vom Empfänger des Dokumentes geprüft werden kann (Verifikationsfunktion). Hierbei soll die Signatur ausschließlich vom Unterzeichner erzeugt, aber durch jedermann verifiziert werden können.

Zur Konstruktion der digitalen Signatur als elektronische Variante der eigenhändigen Unterschrift bietet sich – wie in Abbildung 3 dargestellt – der Einsatz asymmetrischer Kryptoalgorithmen an, bei denen ein Schlüsselpaar, bestehend aus einem privaten Schlüssel (private key,  $K_{priv}$ ) und einem öffentlichen Schlüssel (public key,  $K_{pub}$ ), verwendet wird.

Hierbei ist der private Schlüssel  $K_{priv}$ , der zur Erzeugung einer Signatur  $\sigma = \text{Sign}(m, K_{priv})$  verwendet wird, nur für den Signaturschlüsselhaber zugänglich. Der öffentliche Schlüssel  $K_{pub}$ , der zur Prüfung von Signaturen mit der Funktion  $\text{Verify}(m, \sigma, K_{pub})$  verwendet wird, steht hingegen jedermann zur Verfügung. Die Schlüssel stehen insofern in einer Beziehung, als man mit ihrer Hilfe die zueinander inversen Operationen des Erstellens und Prüfens von Signaturen ausführen kann. Die Sicherheit der Methode beruht darauf, dass es praktisch unmöglich ist, den privaten Schlüssel aus dem öffentlichen Schlüssel herzuleiten, selbst unter Zuhilfenahme leistungsfähigster Rechnersysteme und Millionen von Jahren an Rechenzeit. Hierbei berechnet sich der öffentliche Schlüssel  $K_{pub} = f(K_{priv})$  durch Anwendung einer so genannten Einwegfunktion  $f$  aus dem privaten Schlüssel  $K_{priv}$ . Daher kann der öffentliche Schlüssel in einem öffentlich zugänglichen Verzeichnis hinterlegt werden, ohne damit den

---

<sup>11</sup>Neben den hier behandelten digitalen Signaturverfahren auf Basis von asymmetrischen Kryptoalgorithmen ist es auch möglich, elektronische Signaturen unter Verwendung von Hashfunktionen [Merk80], symmetrischen Kryptoalgorithmen [Merk87] oder biometrischen Verfahren [LeSc04] zu konstruieren.

privaten Schlüssel preiszugeben.

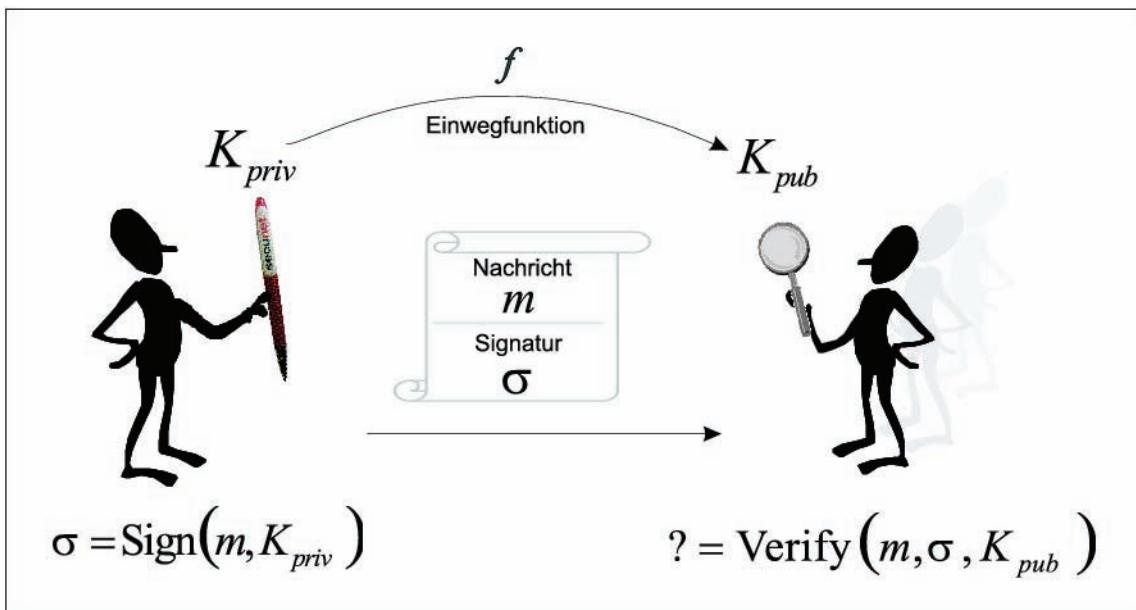


Abbildung 3: Prinzip der digitalen Signatur

### 3.1.2 Einwegfunktionen und zahlentheoretische Probleme

Wie oben erläutert, wird zur Konstruktion eines digitalen Signaturverfahrens eine *Einwegfunktion*  $f$  benötigt<sup>12</sup>. Eine Einwegfunktion  $f : x \rightarrow y$  ist intuitiv eine Funktion, die sich „leicht“ berechnen, aber nur sehr „schwer“ invertieren lässt. Während es leicht ist, aus  $x$  den Wert  $y = f(x)$  zu berechnen, ist es in der Praxis nicht möglich, aus dem Wert  $y$  den ursprünglichen Wert  $x = f^{-1}(y)$  zu ermitteln.

Nach den Komplexitätstheoretischen Betrachtungen in Abschnitt 3.1.2.1, die vom eiligen Leser auch übersprungen werden können, soll in den folgenden Unterabschnitten auf die Konstruktion von Einwegfunktionen auf Basis schwieriger zahlentheoretischer Probleme eingegangen werden.

#### 3.1.2.1 Komplexität von Algorithmen

Um die oben verwendeten Begriffe „leicht“ und „schwer“ präziser zu fassen, betrachtet man die Laufzeit der für die Lösung der jeweiligen Probleme existierenden Algorithmen.

Hierbei verwendet man die Bitoperation, die den Aufwand für die Verknüpfung von zwei Bits bezeichnet, als elementare Einheit und die  $\mathcal{O}$ -Notation, die die asymptotische Abschätzung der Laufzeit erleichtert. Für zwei reellwertige Funktionen  $f, g$  schreiben wir  $f = \mathcal{O}(g)$ , wenn für hinreichend große Werte von  $x$  die Ungleichung  $f(x) \leq c \cdot g(x)$  für einen konstanten Wert  $c > 0$  gilt. Diese Betrachtungsweise konzentriert sich also auf die Schwierigkeit eines Problems bei sehr großen Eingabewerten und abstrahiert von implementierungsabhängigen Details, die die Laufzeit eines Algorithmus nur um einen konstanten Faktor beeinflussen würden.

<sup>12</sup>In [Romp90] wurde gezeigt, dass Einwegfunktionen nicht nur notwendig, sondern auch hinreichend für sichere Signaturverfahren sind.

Erhält ein Algorithmus eine Zahl  $n$ , die größer als die Euler'sche Zahl  $e \approx 2.71828$  ist, als Eingabe und benötigt zur Berechnung des Ergebnisses  $O(\log(n)^c)$  Operationen für eine Konstante  $c > 1$ , so besitzt der Algorithmus *polynomielle Laufzeit*, bezogen auf die Eingabelänge (Anzahl der zur Darstellung der Zahl  $n$  benötigten binären Zeichen) der Größe  $O(\log(n))$ . Es sei angemerkt, dass bei der *O-Notation* die Basis des Logarithmus unerheblich ist. Denn es gilt  $\log_a(n) = c \cdot \log_b(n)$  mit  $c = \log_a(b)$  und die *O-Notation* abstrahiert gerade von konstanten Faktoren.

Bei der Betrachtung von Algorithmen unterscheidet man außerdem danach, ob die Folge der einzelnen Schritte bei der Ausführung des Algorithmus allein von den Eingabedaten bestimmt ist und deshalb bei gleichen Eingaben identisch abläuft, oder ob der Ablauf zusätzlich von Zufallswerten bestimmt ist. Ist der Ablauf eines Algorithmus nicht vom Zufall abhängig, so spricht man von einem *deterministischen Algorithmus*. Ist die Folge der Schritte bei der Ausführung des Algorithmus aber vom Zufall abhängig, so spricht man von einem nicht-deterministischen oder *probabilistischen Algorithmus*.

Ein Problem, das mit einem *deterministischen Algorithmus* in *Polynomialzeit* gelöst werden kann, nennt man „leicht“. Die Menge aller solcher Probleme bezeichnet man mit  $\mathcal{P}$ . Beispielsweise ist die Multiplikation von ganzen Zahlen in  $\mathcal{P}$ .

Die Menge aller *probabilistischen Algorithmen* mit *polynomieller Laufzeit* bezeichnet man mit  $\mathcal{NP}$ . Eine gegebene Lösung zu einem Problem in der Klasse  $\mathcal{NP}$  kann mit einem Algorithmus aus  $\mathcal{P}$  geprüft werden. Beispielsweise ist das in *Abschnitt 3.1.2.2* betrachtete Faktorisierungsproblem in  $\mathcal{NP}$ .

Ein Problem nennt man „schwer“, wenn die Erfolgswahrscheinlichkeit für *jeden* Algorithmus aus der Klasse  $\mathcal{NP}$  „vernachlässigbar“<sup>13</sup> ist.

Wenn eine Funktion  $f$  „leicht“ und die Umkehrfunktion  $f^{-1}$  „schwer“ ist, so nennt man  $f$  eine *Einwegfunktion* (vgl. [Gold01], Definition 2.1).

Da jedes Problem, das mit einem *deterministischen Algorithmus* in *Polynomialzeit* gelöst werden kann, auch mit einem *probabilistischen Algorithmus* in *Polynomialzeit* gelöst werden kann, gilt  $\mathcal{P} \subseteq \mathcal{NP}$ . Umgekehrt ist nicht bekannt, ob die Menge  $\mathcal{NP}$  echt größer ist als  $\mathcal{P}$ . Es ist das vielleicht wichtigste offene Problem der theoretischen Informatik, ob  $\mathcal{P} = \mathcal{NP}$  ist. Könnte man die Existenz einer *Einwegfunktion* beweisen, so würde das  $\mathcal{P} \neq \mathcal{NP}$  implizieren.

Da man bis heute keine einzige Funktion  $f$  aus  $\mathcal{P}$  finden konnte, für die  $f^{-1}$  bewiesenermaßen „schwer“ ist, muss man sich für die Konstruktion von digitalen Signaturverfahren mit Funktionen begnügen, die nach heutigem Kenntnisstand die Einweg-Eigenschaft zu besitzen *scheinen*. Statt der Forderung, dass *jeder mögliche Algorithmus* aus  $\mathcal{NP}$  vernachlässigbare Erfolgswahrscheinlichkeit hat, begnügt man sich in der Praxis damit, dass man bis heute *keinen Algorithmus kennt*, der eine nicht-vernachlässigbare Erfolgswahrscheinlichkeit hat.

Zur Konstruktion von digitalen Signatursystemen verwendet man Probleme, wie das in *Abschnitt 3.1.2.2* betrachtete *Faktorisierungsproblem* oder das in *Abschnitt 3.1.2.3* diskutierte Problem des *Diskreten Logarithmus*, für deren Lösung nur Algorithmen mit *subexponentieller* oder *exponentieller Laufzeit* bekannt sind. Insbesondere ist für geeignet gewählte Instanzen dieser Probleme kein *Polynomialzeit*-Algorithmus bekannt, durch den man den *privaten Schlüssel*  $K_{priv}$  effizient aus dem *öffentlichen Schlüssel*  $K_{pub}$  berechnen könnte.

---

<sup>13</sup>„Vernachlässigbar“ ist ein Wert, wenn er kleiner als  $\frac{1}{p(l)}$  für jedes Polynom  $p$  mit hinreichend großer Eingabelänge  $l$  ist. Beispielsweise ist eine Erfolgswahrscheinlichkeit von  $\frac{1}{2^l}$  vernachlässigbar.

Erhält ein Algorithmus eine Zahl  $n$  als Eingabe und benötigt zur Berechnung des Ergebnisses  $\mathcal{O}(c^{\log(n)})$  Operationen für eine Konstante  $c > 1$ , so besitzt dieser *exponentielle Laufzeit*, bezogen auf die Eingabelänge der Zahl  $n$  mit der Größe  $\mathcal{O}(\log(n))$ .

Sei  $e$  die Euler'sche Zahl,  $n > e$ ,  $\log(n)$  der natürliche Logarithmus der Zahl  $n$ ,  $u$  eine reelle Zahl mit  $0 \leq u \leq 1$  und  $v$  eine positive reelle Zahl. Dann stützt man sich für die Beschreibung der Laufzeit von Algorithmen mit *subexponentieller Laufzeit* meist auf die in [BLP93] definierte Funktion

$$(3.1.1) \quad L_n[u, v] = e^{v(\log(n))^u (\log(\log(n)))^{1-u}}.$$

Da  $L_n[0, v] = e^{v(\log(n))^0 \log(\log(n))} = e^{v \log(\log(n))} = \log(n)^v = \mathcal{O}(\log(n)^c)$  für eine Konstante  $c$ , entspricht  $L_n[0, v]$  der *Polynomialzeit*. In ähnlicher Weise entspricht  $L_n[1, v] = e^{v \log(n)} = \mathcal{O}(n^v)$  der *Exponentialzeit*, und für  $0 < u < 1$  liegt der Aufwand zwischen diesen beiden Extremen, und der Algorithmus hat in diesem Fall subexponentielle Laufzeit.

Beispielsweise besitzt der beste bekannte Algorithmus für das *Faktorisierungsproblem*, das *Zahlkörpersieb* [LLMP93], eine erwartete Laufzeit von  $L_n[1/3, (64/9)^{1/3} + o(1)]$ , wobei  $o(1)$  eine Funktion ist, die im Unendlichen gegen Null strebt und als eine Verallgemeinerung der Konstante bei der *O-Notation* angesehen werden kann. Für geeignet gewählte *DL-Probleme* in Punktegruppen elliptischer Kurven benötigt man nach heutigem Kenntnisstand sogar *exponentielle Laufzeit*. Dies erklärt, wieso Kryptosysteme auf Basis elliptischer Kurven das gleiche Maß an Sicherheit bei deutlich kürzeren Schlüsseln erreichen können (vgl. Abbildung 6).

Allerdings muss betont werden, dass alle heutigen Aussagen über die Sicherheit von *asymmetrischen Kryptoalgorithmen* auf unbewiesenen Vermutungen basieren. Die Einweg-Eigenschaft der ihnen zu Grunde liegenden „*Einwegfunktionen*“ ist leider selbst eine Funktion der Zeit. Durch den glücklichen Einfall einer einzigen Mathematikerin oder eines einzigen Mathematikers kann sich ein vermeintlich schwieriges Problem als leicht lösbar erweisen und darauf basierende Signaturen wären plötzlich leicht fälschbar. Deshalb müssen bei der langfristigen Aufbewahrung von *qualifizierten elektronischen Signaturen* auf jeden Fall die in Abschnitt 4.6 diskutierten Aspekte der *Nachsignatur* berücksichtigt werden. Außerdem sollte bei kritischen Applikationen der gleichzeitige Einsatz von mehreren, unabhängigen Kryptoalgorithmen erwogen werden.

#### 3.1.2.2 Das Faktorisierungsproblem

Das Faktorisierungsproblem, das beispielsweise dem in Abschnitt 3.1.3.1 vorgestellten RSA-Verfahren zu Grunde liegt, besteht darin, eine große zusammengesetzte Zahl in ihre Primfaktoren zu zerlegen. Da es leicht ist, *Primzahlen* miteinander zu multiplizieren, aber ungleich schwieriger ist, eine zusammengesetzte Zahl in ihre Primfaktoren zu zerlegen, scheint das Faktorisierungsproblem für die Konstruktion von *Einwegfunktionen* geeignet zu sein.

Wie das Zitat des berühmten Mathematikers Carl Friedrich Gauß aus dem Jahr 1801 [Gaus01, Artikel 329] belegt, handelt es sich beim Faktorisierungsproblem keineswegs um eine neuartige Problemstellung:

„Dass die Aufgabe, die Primzahlen von den zusammengesetzten zu unterscheiden und letztere in ihre Primfactoren zu zerlegen, zu den wichtigsten und nützlichsten der gesamten Arithmetik gehört und die Bemühungen und den Scharfsinn sowohl der alten wie auch der neueren Geometer in Anspruch genommen hat, ist so bekannt, dass es überflüssig wäre, hierüber viele Worte zu verlieren. (...), so bietet sich doch dem erfahrenen Rechner nicht selten die Gelegenheit dar, aus der Zerlegung grosser Zahlen in Factoren grosse Vorteile zu ziehen,

welche den mässigen Aufwand an Zeit reichlich wieder ausgleichen; außerdem aber dürfte es die Würde der Wissenschaft erheischen, alle Hilfsmittel zur Lösung jenes so eleganten und berühmten Problems fleissig zu vervollkommen.“

Wie kürzlich in [AKS04] gezeigt (siehe auch [Born02]), existiert für das erste von Gauß erwähnte Problem, bei dem entschieden werden muss, ob es sich bei einer gegebenen Zahl um eine *Primzahl* handelt oder nicht, ein Algorithmus aus  $\mathcal{P}$ .

Für das zweite angesprochene Problem, das *Faktorisierungsproblem*, ist hingegen bisher kein Algorithmus mit *polynomieller Laufzeit* bekannt. Ein Überblick über die populärsten Faktorisierungsalgorithmen findet sich in [Nebe00]. Aus dem Blickwinkel der Kryptographie können diese Algorithmen in drei wesentliche Kategorien eingeteilt werden:

- Klassische Faktorisierungsalgorithmen

Neben der naheliegenden Versuchsdision existieren bereits seit langer Zeit verschiedene Faktorisierungsalgorithmen, die auf berühmte Mathematiker wie Fermat, Legendre, Euler und Gauß zurück gehen (vgl. [Ries94, Chapter 5]). Diese Algorithmen – wie auch der vor einigen Jahren vorgeschlagene  $\beta$ -Algorithmus von Pollard [Poll75] oder der SQUFOF<sup>14</sup> - Algorithmus von Shanks [Ries94, S. 186 ff.] – haben keine praktische Bedeutung, da sie lediglich *exponentielle Laufzeit* besitzen und deshalb nur bei sehr kleinen Schlüssellängen erfolgreich eingesetzt werden könnten.

- Faktorisierung durch glatte Gruppenordnungen

Daneben existieren eine Reihe von Faktorisierungsalgorithmen [Poll74, Will82, ScLe84, Lens87], die Exponentiationen in einer endlichen, *abelschen Gruppe*  $G_n$ , die mit der zu faktorisierenden Zahl  $n$  zusammenhängt, durchführen und genau dann erfolgreich sind, wenn die Gruppenordnung  $|G_n|$  „glatt“ ist – wenn diese also ausschließlich aus kleinen *Primzahlen* zusammengesetzt ist. Solche Algorithmen können effizient kleine Faktoren in einer großen zusammengesetzten Zahl  $n$  ermitteln. Damit digitale Signaturverfahren nicht durch diese Algorithmen gebrochen werden können, verwendet man in der Kryptographie meist<sup>15</sup> Zahlen der Form  $n = pq$ , wobei  $p$  und  $q$  etwa gleich groß sind.

- Siebbasierte Faktorisierungsalgorithmen

Die leistungsfähigsten Faktorisierungsalgorithmen für große Zahlen der Form  $n = pq$  verwenden eine so genannte Faktorbasis für die durch Anwendung von Siebverfahren Relationen gesammelt werden. Sind ausreichend viele von diesen Relationen vorhanden, so wird ein lineares Gleichungssystem gelöst, aus dem schließlich zwei ganze Zahlen  $x$  und  $y$  konstruiert werden, für die  $x^2 \equiv y^2 \pmod{n}$ , aber  $x \not\equiv \pm y \pmod{n}$  gilt. Sind solche Zahlen ermittelt, so kann ein Faktor von  $n$  leicht durch die Berechnung des größten gemeinsamen Teilers  $\text{ggT}(x - y, n)$  gefunden werden. Hierbei unterscheiden sich das quadratische Sieb [Pome85, Silv87] und das so genannte *Zahlkörpersieb* [LLMP93] in der Art und Weise wie sie die Relationen ermitteln. Hat die zu faktorisierende Zahl mehr als etwa 130 Dezimalstellen, so ist das *Zahlkörpersieb* effizienter. Der aktuelle Rekord, der von Wissenschaftlern der Universität Bonn in Zusammenarbeit mit dem Bundesamt für Sicherheit in der Informationstechnik aufgestellt wurde, liegt bei 200 Dezimalstellen [RSA200]. Kürzlich wurde von dieser Arbeitsgruppe auch die etwas kleinere Zahl [RSA640] faktorisiert.

---

<sup>14</sup>SQUare FOrms Factorization

<sup>15</sup>[PKCS1(v2.1)] sieht, entgegen den Vorgängerversionen dieses Standards, auch die Verwendung von mehreren unabhängigen Primzahlen für das RSA-Verfahren vor. Außerdem basieren einige Kryptosysteme auf der Schwierigkeit der Faktorisierung von Zahlen der Form  $n = pq^2$  [OkUc98, Pail99, Hueh04b] oder  $n = p^k q$  [Taka98].

### 3.1.2.3 Das Problem des diskreten Logarithmus

Das *Diskrete Logarithmus-Problem (DLP)* in einer endlichen, *abelschen Gruppe*  $G$ , das beispielsweise den in Abschnitt 3.1.3.2 vorgestellten (EC)DSA-Verfahren zu Grunde liegt, besteht darin, aus den gegebenen Gruppenelementen  $g, g^n \in G$  den Exponenten  $n$  zu ermitteln.

Zur Berechnung von  $g^n$  verwendet man in der Praxis das „Square & Multiply“-Verfahren, oder eine seiner Varianten (vgl. [Gord98]).

**Beispiel 3.1.1** Es soll die Potenz  $g^5$  berechnet werden. Hierfür betrachtet man die Binärdarstellung des Exponenten  $5_{10} = 101_2$ . Nun startet man mit dem Wert  $g$  und durchwandert die Binärdarstellung des Exponenten von links nach rechts, wobei bei jedem Schritt zum nächsten Bit erst quadriert und dann, falls das Bit, zu dem man springt, gleich „1“ ist, der aktuelle Wert mit  $g$  multipliziert wird. Man berechnet also  $g^5 = (g^2)^2 \cdot g$ .

Es ist leicht zu sehen, dass man mit einer solchen Strategie nur  $O(\log(n))$  Gruppenoperationen für die Berechnung von  $g^n$  benötigt.

In den für kryptographische Zwecke eingesetzten *Gruppen* kann  $g^n$  in *Polynomialzeit* berechnet werden, während für die Lösung des Diskreten Logarithmus-Problems nur *subexponentielle* oder *exponentielle Algorithmen* existieren, so dass auf dieser Basis *digitale Signatursysteme* konstruiert werden können.

Für die Lösung des *DLP* stehen zwei grundsätzliche Algorithmentypen zur Verfügung.

Auf der einen Seite existieren so genannte *generische Algorithmen*, die in beliebigen endlichen, *abelschen Gruppen* eingesetzt werden können. Die bekanntesten generischen Algorithmen für die Berechnung Diskreter Logarithmen sind der „Baby-Step-Giant-Step-Algorithmus“ von Shanks [Shan72] und das „Rho-Verfahren“ von Pollard [Poll78]. Beide haben eine (erwartete) Laufzeit von  $O(\sqrt{|G|})$  Gruppenoperationen, wobei  $|G|$  die Ordnung der Gruppe  $G$  ist. Somit besitzen diese Algorithmen – bezogen auf die Eingabelänge der Größe  $O(\log(|G|))$  – *exponentielle Laufzeit*. Außerdem kann durch den Algorithmus von Pohlig und Hellman die Berechnung des *Diskreten Logarithmen* in einer endlichen, *abelschen Gruppe*  $G$ , bei der die Faktorisierung der Gruppenordnung bekannt ist, auf die Berechnung von Diskreten Logarithmen in den *Untergruppen*  $U \subset G$  zurück geführt werden. Deshalb dürfen in der Kryptographie keine Gruppen eingesetzt werden, deren *Gruppenordnung* ausschließlich aus kleinen *Primzahlen* besteht.

Daneben existieren für bestimmte Gruppen, wie beispielsweise für multiplikative Gruppen endlicher Körper, *siebbasierte Algorithmen*, die die Berechnung von Diskreten Logarithmen mit *subexponentiellem Aufwand* erlauben. Wie schwierig die Lösung des *DLP* ist, hängt also maßgeblich von der verwendeten *Gruppe*  $G$  ab.

In der kryptographischen Praxis werden insbesondere multiplikative *Gruppen endlicher Körper* und Punktegruppen *elliptischer Kurven* eingesetzt. Daneben wurden beispielsweise auch so genannte Divisor-Klassengruppen auf hyperelliptischen Kurven [Kobl89] oder Klassengruppen imaginärquadratischer Körper [BuWi88] zur Konstruktion schwieriger *DL-Probleme* vorgeschlagen.

#### Multiplikative Gruppen endlicher Körper

Die Verwendung multiplikativer Gruppen endlicher Körper zur Konstruktion von schwierigen *DL-Problemen* ist so alt wie das *DL-Problem* selbst, das von Diffie und Hellman vorgeschlagen wurde [DiHe76].

Hierzu wählt man einfach eine *Primzahl*  $p$  und betrachtet bei allen Zahlen, mit denen

gerechnet wird, lediglich den kleinsten Rest, der übrig bleibt, wenn man ganzzahlige Vielfache von  $p$  abzieht. Man rechnet also „modulo“  $p$  und schreibt  $a \equiv b \pmod{p}$ , wenn sich  $a$  und  $b$  nur um ein Vielfaches von  $p$  unterscheiden.

Die *Restklassen modulo  $p$*  ( $0, 1, 2, \dots, p-1$ ) bilden einen *endlichen Körper* – den „Primkörper“<sup>16</sup>  $\mathbb{F}_p$ .

Die grundlegende Arithmetik in  $\mathbb{F}_p$  ist denkbar einfach: Man verwendet die gewöhnliche Addition und Multiplikation und reduziert das Ergebnis modulo  $p$ .

**Beispiel 3.1.2** Wir wählen  $p = 7$  und berechnen  $2^5$  in  $\mathbb{F}_7$ . Wie in Beispiel 3.1.1 gezeigt, kann eine Exponentiation mit 5 durch zwei Quadrierungsschritte und eine darauffolgende Multiplikation realisiert werden. Deshalb ist  $2^5 = (2^2)^2 \cdot 2 = 4^2 \cdot 2 = 16 \cdot 2 \equiv 2 \cdot 2 \equiv 4 \pmod{7}$ .

Für die Inversion eines Elementes  $x \in \mathbb{F}_p$ , also die Berechnung eines  $x^{-1}$ , so dass  $x^{-1} \cdot x \equiv 1 \pmod{p}$ , existieren zwei unterschiedliche Ansätze:

1. Exponentiation mit  $p - 2$

Durch den „kleinen Satz von Fermat“ gilt  $x^{p-1} \equiv 1 \pmod{p}$  für alle zu  $p$  teilerfremden Elemente  $x$  und deshalb erhält man durch die Exponentiation mit  $p - 2$  das inverse Element  $x^{-1} \equiv x^{p-2} \pmod{p}$ . Deshalb ist im obigen Beispiel  $2^5 \equiv 4 \pmod{7}$  das inverse Element von 2 modulo 7.

2. Erweiterter Euklidischer Algorithmus

Mit dem erweiterten Euklidischen Algorithmus (EEA) (vgl. [BaSh96, Chapter 4.3]) berechnet man für zwei ganze Zahlen  $a$  und  $b$  eine Linearkombination des größten gemeinsamen Teilers  $\text{ggT}(a, b) = s \cdot a + t \cdot b$  der eingegebenen Zahlen.

Zeile	$r_i$	=	$s_i \cdot a$	+	$t_i \cdot b$	$q_i$
1	$a$	=	$1 \cdot a$	+	$0 \cdot b$	
2	$b$	=	$0 \cdot a$	+	$1 \cdot b$	$q_2 = \lfloor a/b \rfloor$
			$\vdots$			
$i$	$r_{i-2} - q_{i-1} \cdot r_{i-1}$	=	$(s_{i-2} - q_{i-1} \cdot s_{i-1}) \cdot a$	+	$(t_{i-2} - q_{i-1} \cdot t_{i-1}) \cdot b$	$q_i = \lfloor r_{i-1}/r_i \rfloor$
			$\vdots$			
$n-1$	$\text{ggT}(a, b)$	=	$s \cdot a$	+	$t \cdot b$	
$n$	0					

Tabelle 1: Erweiterter Euklidischer Algorithmus (EEA)

Hierfür initialisiert man die oben abgebildete Tabelle mit den ersten beiden Zeilen und berechnet den Wert  $q_2 = \lfloor a/b \rfloor$  durch eine ganzzahlige Division. Eine neue Zeile ( $i$ ) erhält man jeweils dadurch, dass man von der vorletzten Zeile ( $i-2$ ) das  $q_{i-1}$ -fache der letzten Zeile ( $i-1$ ) abzieht und den Faktor  $q_i$  für die Ermittlung der nächsten Zeile ( $i+1$ ) berechnet. Dies wird so lange durchgeführt, bis man schließlich den Rest  $r_n = 0$  erhält. Die gesuchte Linearkombination findet sich dann in Zeile  $n-1$ .

Ruft man den Algorithmus für ein zum Modul  $p$  teilerfremdes  $x$  auf und erhält die Linearkombination  $1 = \text{ggT}(p, x) = s \cdot p + t \cdot x$ , so ist  $t \equiv x^{-1} \pmod{p}$  das gesuchte inverse Element.

<sup>16</sup>Allgemeiner kann man zu jeder Primzahlpotenz  $p^n$  endliche Körper mit  $p^n$  Elementen konstruieren. Die Grundlagen endlicher Körper sind in [Kobl94, Chapter II.1] erläutert. Eine umfassende Behandlung der Materie findet sich in [LiNi86].

**Beispiel 3.1.3** Entsprechend dem oben genannten Beispiel soll wieder das inverse Element von  $2 \pmod{7}$ , diesmal aber mit dem erweiterten Euklidischen Algorithmus, berechnet werden.

Zeile	$r_i = s_i \cdot a + t_i \cdot b$	$q_i$
1	$7 = 1 \cdot 7 + 0 \cdot 2$	
2	$2 = 0 \cdot 7 + 1 \cdot 2$	$q_2 = \lfloor 7/2 \rfloor = 3$
3	$(7 - 2 \cdot 3) = (1 - 3 \cdot 0) \cdot 7 + (0 - 3 \cdot 1) \cdot 2$	
4	$1 = 1 \cdot 7 + (-3) \cdot 2$	$q_3 = \lfloor 2/1 \rfloor = 2$
	$0$	

**Tabelle 2: Inversion von  $2 \pmod{7}$  mit dem EEA**

Hierzu initialisiert man in Tabelle 2 die ersten beiden Zeilen und berechnet  $q_2 = \lfloor 7/2 \rfloor = 3$ . Die Zeile 3 erhält man nun dadurch, dass man das 3-fache der Zeile 2 von Zeile 1 abzieht. Da man in Zeile 4 bereits den Rest  $r_4 = 0$  erhält, ist in Zeile 3 abzulesen, dass  $\text{ggT}(7, 2) = r_3 = 1$  und das Inverse von 2 durch  $-3 \equiv 4 \pmod{7}$  gegeben ist.

Während die Inversion in  $\mathbb{F}_p^*$  mittels Exponentiation  $O(\log(p)^3)$  Bitoperationen benötigt, braucht die Inversion unter Verwendung des erweiterten Euklidischen Algorithmus gemäß [BaSh96, Corollary 4.3.3] nur  $O(\log(p)^2)$  Bitoperationen. Deshalb verwendet man in der Praxis meist den erweiterten Euklidischen Algorithmus für die modulare Inversion.

Zur Lösung des *DLP* in der multiplikativen Gruppe eines endlichen Körpers stehen neben den generischen Algorithmen, die in jeder endlichen abelschen Gruppe funktionieren, ähnlich effiziente Algorithmen wie für das Faktorisierungsproblem zur Verfügung. Insbesondere kann auch das Zahlkörpersieb [Gord93, Webe96, JoLe03] zur Lösung des *DLP* in diesen Gruppen eingesetzt werden. Die asymptotische Laufzeit dieses Algorithmus ist identisch mit der des Zahlkörpersiebes für die Faktorisierung.

### Punktegruppen elliptischer Kurven über endlichen Körpern

Eine elliptische Kurve (über einem Körper  $K$  mit Charakteristik  $\text{char}(K) \neq 2, 3$ <sup>17</sup>) ist die Menge aller Punkte  $P = (x, y)$  auf der „glatten“<sup>18</sup> Kurve

$$(3.1.2) \quad y^2 = x^3 + ax + b$$

zusammen mit dem Punkt  $\mathcal{O}$ , „im Unendlichen“.

Für die Menge der Punkte auf dieser elliptischen Kurve kann man eine Verknüpfung „+“ definieren, so dass diese zu einer abelschen Gruppe mit  $\mathcal{O}$  als neutralem Element werden. Ist die Kurve über dem endlichen Körper  $\mathbb{F}_p$  definiert, so erhält man eine endliche, abelsche Gruppe. Solche Gruppen werden bei Lenstra’s „Elliptic Curve Method“ [Lens87] zum Faktorisieren eingesetzt oder, wie von Miller [Mill85] und Koblitz [Kobl87] vorgeschlagen, als Basis für schwierige *DL-Probleme* verwendet.

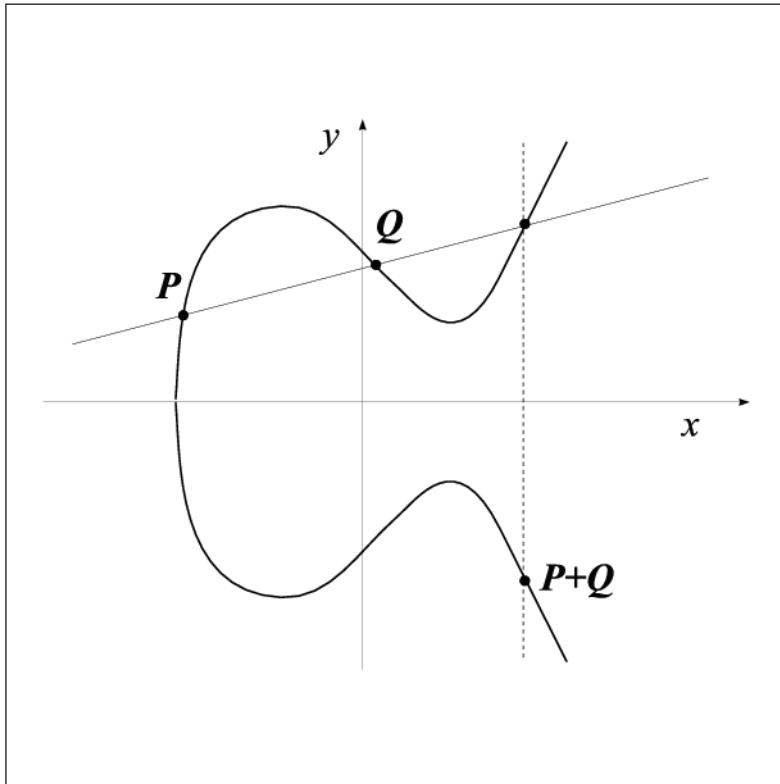
Wie in Abbildung 4 veranschaulicht, erhält man bei einer elliptischen Kurve über den reellen Zahlen, falls  $P \neq \pm Q \neq \mathcal{O}$ , den Punkt  $P + Q$ , indem man eine Gerade durch die beiden

<sup>17</sup>Darunter fallen beispielsweise die unendlichen Körper der rationalen  $\mathbb{Q}$ , reellen  $\mathbb{R}$  oder komplexen Zahlen  $\mathbb{C}$  und die endlichen Körper  $\mathbb{F}_p$ , für eine Primzahl  $p > 5$ . Entsprechende Betrachtungen für Körper mit  $\text{char}(K) = 2, 3$  finden sich beispielsweise in [Mene93].

<sup>18</sup>Man nennt eine Kurve „glatt“, wenn man in jedem Punkt eine Tangente anlegen kann. Außerdem lässt sich nachprüfen, dass dies genau dann der Fall ist, wenn  $4a^3 + 27b^2 \neq 0$  gilt.

Punkte  $P$  und  $Q$  legt und den dritten Schnittpunkt dieser Gerade mit der elliptischen Kurve an der  $x$ -Achse spiegelt. Ist  $P = Q$ , so erhält man das Doppelte  $2P$  des Punktes  $P$ , indem man die Tangente an den Punkt  $P$  legt und den zweiten Schnittpunkt der Tangente mit der Kurve an der  $x$ -Achse spiegelt. Ist  $Q = -P$ , so führt die Punktaddition  $P + Q = P - P = \mathcal{O}$  zum unendlich fernen Punkt – dem neutralen Element der Punktegruppe.

Für die Inversion eines Punktes  $P = (x, y)$  spiegelt man diesen einfach an der  $x$ -Achse ( $-P = (x, -y)$ ).



**Abbildung 4: Gruppenoperation auf einer elliptischen Kurve**

Seien  $P = (x_1, y_1)$  und  $Q = (x_2, y_2)$  zwei Punkte auf der elliptischen Kurve. Dann kann man die Summe  $R = P + Q = (x_3, y_3)$  dieser beiden Punkte leicht berechnen. Es gilt

$$(3.1.3) \quad \begin{aligned} x_3 &= \lambda^2 - x_1 - x_2, \\ y_3 &= -y_1 + \lambda(x_1 - x_3) \end{aligned}$$

mit

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1}, & \text{falls } P \neq Q \text{ und } x_1 \neq x_2 \\ \frac{3x_1^2 + a}{2y_1}, & \text{falls } P = Q \text{ und } y_1 \neq 0. \end{cases}$$

Ist  $P \neq Q$  und  $x_1 = x_2$ , so ist die Gerade durch die beiden Punkte  $P$  und  $Q$  vertikal und schneidet die elliptische Kurve im Unendlichen. In diesem Fall ist  $P + Q = \mathcal{O}$ .

Falls  $P = Q$  und  $y_1 = 0$ , so ist auch  $P + Q = 2P = \mathcal{O}$  der unendlich ferne Punkt.

**Beispiel 3.1.4** Wir betrachten die elliptische Kurve mit der Gleichung  $E : y^2 = x^3 + 1$  über den reellen Zahlen.

Nimmt man zu den Lösungen der Gleichung  $E$  den Punkt  $\mathcal{O}$  im Unendlichen hinzu, so erhält man die Punktegruppe der elliptischen Kurve  $E$  über  $\mathbb{R}$ . Außerdem sieht man, dass es fünf Punkte auf der Kurve mit ganzzahligen Koordinaten gibt, die zusammen mit dem Punkt  $\mathcal{O}$  im Unendlichen eine Untergruppe mit Gruppenordnung 6 bilden. Wie in Abbildung 5 ersichtlich, wird diese Untergruppe vom Punkt  $P_1$  „erzeugt“. Die einzelnen Punkte sind folgendermaßen gegeben:

$$P_1 = (2, -3)$$

$$P_2 = 2 \cdot P_1 = P_1 + P_1 = (0, -1)$$

$$P_3 = 3 \cdot P_1 = P_2 + P_1 = (-1, 0)$$

$$P_4 = 4 \cdot P_1 = P_3 + P_1 = (0, 1)$$

$$P_5 = 5 \cdot P_1 = P_4 + P_1 = (2, 3)$$

$$P_6 = 6 \cdot P_1 = P_5 + P_1 = \mathcal{O}$$

Für die Addition von Punkten verwendet man die Formel für die Gruppenverknüpfung (3.1.3). Ist die elliptische Kurve über einem endlichen Primkörper  $\mathbb{F}_p$  definiert, so reduziert man in jedem Rechenschritt zusätzlich modulo  $p$ . Die „Multiplikation“ eines Punktes mit einer ganzen Zahl  $n$ , also die  $n$ -malige Verknüpfung des Punktes mit sich selbst, ist das Analog zur Exponentiation in einer multiplikativ geschriebenen Gruppe – das DL-Problem ist die Ermittlung des „Faktors“  $n$  bei gegebenen Punkten  $P$  und  $[n] \cdot P$ . Weitere beispielhafte Kurven lassen sich im Online-Tutorial [Laub99] leicht erzeugen.

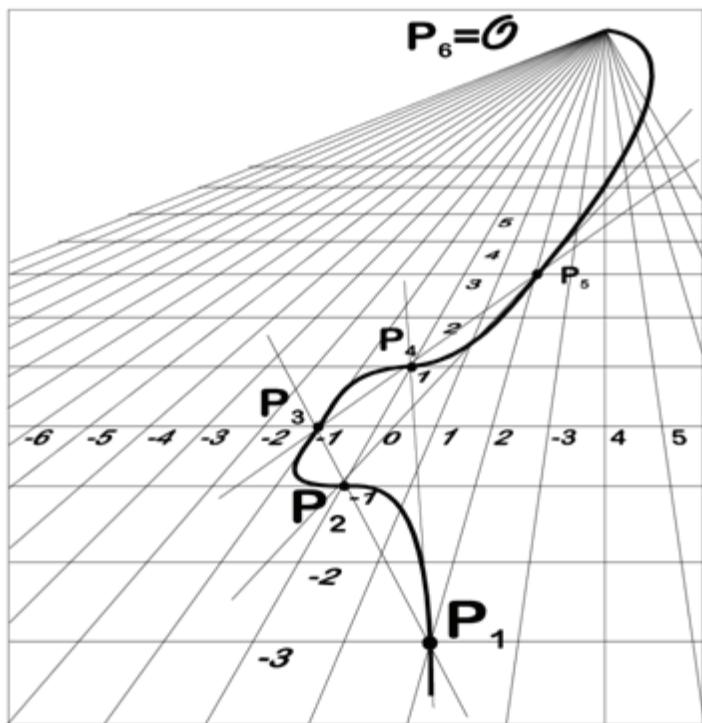


Abbildung 5: Die Elliptische Kurve  $y^2 = x^3 + 1$

Für das *DL-Problem* in einer elliptischen Kurve über einem *endlichen Körper* existieren im Allgemeinen<sup>19</sup> *keine subexponentiellen Algorithmen* im Stile des *Zahlkörpersiebs*, wie er für das Faktorisieren [LLMP93] oder die Berechnung *Diskreter Logarithmen* in *endlichen Körpern* [Gord93] eingesetzt wird. Außerdem erscheint die Anwendung solcher Verfahren für elliptische Kurven grundsätzlich wenig erfolgsversprechend [SiSu98]. Für die Lösung des *DL-Problems (DLP)* in der Punktgruppe einer elliptischen Kurve stehen nur generische Algorithmen [Tesk98], wie beispielsweise das so genannte  $\rho$ -Verfahren von Pollard [Poll78], zur Verfügung. Deshalb bieten Kryptosysteme auf Basis elliptischer Kurven bei gleicher Schlüssellänge ein viel höheres Maß an Sicherheit.

### 3.1.2.4 Vergleich der notwendigen Schlüssellängen

Wie oben erläutert, stehen zur Lösung des *Faktorisierungsproblems* und für das *DLP* in der multiplikativen *Gruppe endlicher Körper* *subexponentielle Algorithmen* zur Verfügung. Für das *DLP* in der Punktgruppe *elliptischer Kurven* existieren aber nur Algorithmen mit *exponentieller Komplexität*. Deshalb kann bei Kryptosystemen auf Basis elliptischer Kurven das gleiche Maß an Sicherheit mit viel kürzeren Schlüsseln erreicht werden.

Abbildung 6 stützt sich auf [LeVe01, Tabelle 1] und veranschaulicht diese Zusammenhänge.

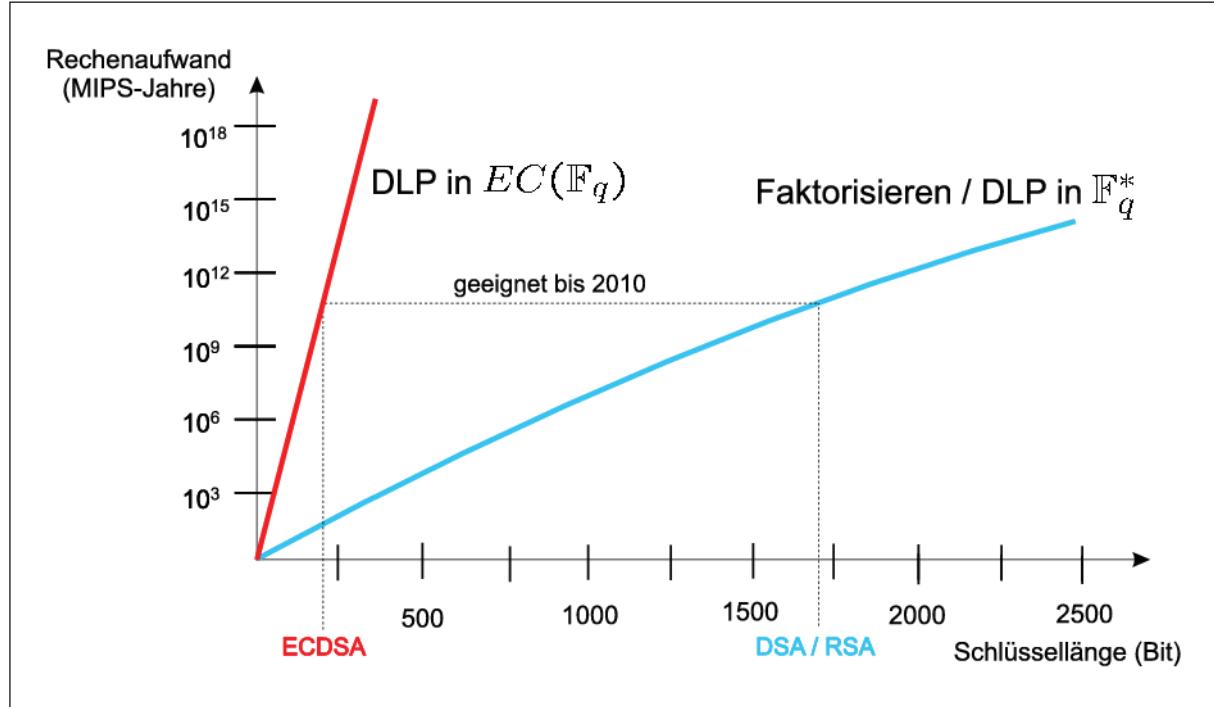


Abbildung 6: Vergleich der Schlüssellängen zwischen RSA/DSA und ECDSA

Gemäß [BNetzA-Alg05] bietet eine *RSA-Signatur* mit 1728 Bit langem Modul ausreichende Sicherheit bis zum Jahr 2010. Danach müssen mindestens 2048 Bit eingesetzt werden,

<sup>19</sup>Ausnahmen bilden so genannte „supersinguläre“ und „anomale“ elliptische Kurven, da für die Lösung des *DL-Problems* in diesen Kurven *subexponentielle* [MOV91, FMH99] bzw. *polynomiale* [SaAr98, Smar99] Algorithmen zur Verfügung stehen. Deshalb sollten solche Kurven nicht für die Konstruktion digitaler Signatursysteme verwendet werden. Außerdem sollten gewisse *endliche Körper* vermieden werden [MTW04].

während beim Einsatz *elliptischer Kurven* 224 Bit ausreichend sind.

### 3.1.3 Signaturalgorithmen

In diesem Abschnitt nutzen wir das Faktorisierungsproblem und das Diskrete Logarithmus-Problem zur Konstruktion digitaler Signatursysteme.

#### 3.1.3.1 RSA

Das nach seinen Erfindern Rivest, Shamir und Adleman benannte RSA-Verfahren [RSA78] war das erste digitale Signaturverfahren – heute ist es das in der Praxis am häufigsten eingesetzte. Es basiert<sup>20</sup> auf dem in *Abschnitt 3.1.2.2* näher diskutierten *Faktorisierungsproblem* und kann in ähnlicher Form auch zur *Verschlüsselung* von Nachrichten eingesetzt werden. Hierbei entsprechen die Abläufe zur *Verschlüsselung* (*Encryption*) mit dem öffentlichen Exponenten  $e$  etwa dem Ablauf bei der *Signaturprüfung* und die *Entschlüsselung* (*Decryption*) etwa der *Signaturerstellung* mit dem privaten Schlüssel  $d$ .

#### Das RSA-Verfahren

Wie in *Abbildung 7* dargestellt, erfolgt der Systemaufbau durch die Wahl von zwei zufälligen, etwa gleich großen *Primzahlen*  $p, q$  und einem *privaten Schlüssel*  $d$ . Hierbei werden die beiden *Primzahlen*  $p$  und  $q$  so gewählt, dass  $n = pq$  nach heutigem Kenntnisstand nicht faktorisiert werden kann. Um Angriffe durch spezialisierte Faktorisierungsverfahren auszuschließen, sollten  $p$  und  $q$  zwar etwa gleich groß sein, aber nicht zu nah zusammen liegen. Außerdem soll der größte gemeinsame Teiler der beiden Zahlen  $p - 1$  und  $q - 1$  klein sein. Diese Eigenschaften sind bei zufälliger Wahl von großen Primzahlen mit sehr großer Wahrscheinlichkeit gegeben. Wie in *Abschnitt 3.1.2.4* angedeutet, bietet die Verwendung eines RSA-Moduls  $n = pq$  mit 1728 Bit voraussichtlich Sicherheit bis zum Jahr 2010. „Für die Gewährleistung eines langfristigen Sicherheitsniveaus wird grundsätzlich die Erhöhung auf 2048 Bit empfohlen“ (vgl. [BNetzA-Alg05]).

Der Exponent  $d$  wird zufällig<sup>21</sup> gewählt, so dass er teilerfremd zur Gruppenordnung von  $(\mathbb{Z}/n\mathbb{Z})^*$ , der „primen Restklassengruppe“ modulo  $n$ , ist.

Die *Gruppe*  $(\mathbb{Z}/n\mathbb{Z})^*$  besteht aus allen Elementen, die teilerfremd zu  $n$  sind. Die Gruppenverknüpfung ist durch die Multiplikation und Reduktion modulo  $n$ , die *Gruppenordnung* durch die so genannte „Eulersche  $\varphi$ -Funktion“ gegeben. Da in unserem Fall  $n = pq$  genau aus zwei Primzahlen besteht, ist  $\varphi(n) = (p - 1)(q - 1)$ . Durch den „Satz von Euler“, der besagt, dass man in jeder endlichen *abelschen Gruppe* durch die Exponentiation mit der *Gruppenordnung* das neutrale Element erhält, gilt  $a^{\varphi(n)} \equiv 1 \pmod{n}$  für jedes  $a \in (\mathbb{Z}/n\mathbb{Z})^*$ .

Der öffentliche Schlüssel besteht aus dem Modul  $n = pq$  und dem Exponenten  $e$ , der  $ed \equiv 1 \pmod{\varphi(n)}$  erfüllt. Die Erzeugung einer Signatur  $s \equiv m^d \pmod{n}$  geschieht durch die

<sup>20</sup>Während die Berechnung des privaten Exponenten  $d$  aus dem öffentlichen Exponenten  $e$  unter deterministischen Polynomialeitreduktionen äquivalent zum Faktorisieren von  $n = pq$  ist [May05], ist bisher nicht klar, ob man für die Lösung des so genannten „RSA-Problemes“ (Berechnung  $e$ -ter Wurzeln modulo  $n$ ) zwingend faktorisieren muss. Es ist bisher theoretisch nicht ausgeschlossen, dass man  $e$ -te Wurzeln auch ohne die vorherige Lösung des Faktorisierungsproblems berechnen kann.

<sup>21</sup>Wird der private Schlüssel  $d$  zufällig gewählt oder als inverses Element eines festgelegten öffentlichen Schlüssels  $e$  berechnet, ist die Wahrscheinlichkeit, einen kleinen und dadurch unsicheren (vgl. [BoDu99]) privaten Schlüssel  $d$  zu erhalten, sehr gering.

modulare Exponentiation der Nachricht  $m$  mit dem *privaten Schlüssel*  $d$ . Für die Prüfung einer solchen Signatur wird eine Exponentiation der Signatur  $s$  mit dem *öffentlichen Schlüssel*  $e$  durchgeführt.

Die Signatur ist genau dann gültig, wenn

$$s^e \equiv (m^d)^e \equiv m^{ed} \equiv m^1 \pmod{\varphi(n)} \equiv m \pmod{n}.$$

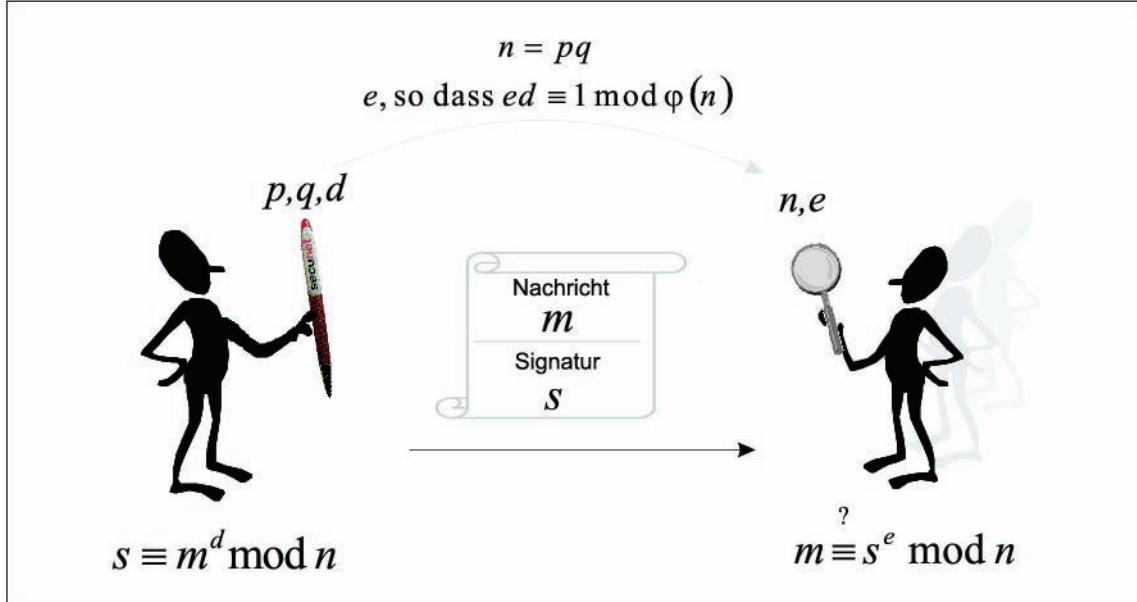


Abbildung 7: Das RSA-Signaturverfahren

**Beispiel 3.1.5** Das RSA-Signaturverfahren soll anhand eines Beispieles mit kleinen Parametern erläutert werden: Sei  $p = 11$  und  $q = 17$ . Dann ist  $n = 11 \cdot 17 = 187$  und  $\varphi(n) = (p-1)(q-1) = 10 \cdot 16 = 160$ . Wir verwenden den öffentlichen Schlüssel  $e = 3$  und ermitteln den privaten Schlüssel  $d \equiv e^{-1} \pmod{\varphi(n)}$  mit dem in Abschnitt 3.1.2.3 näher beschriebenen erweiterten Euklidischen Algorithmus.

Zeile	$r_i$	=	$s_i \cdot a$	+	$t_i \cdot b$	$q_i$
1	160	=	$1 \cdot 160$	+	$0 \cdot 3$	
2	3	=	$0 \cdot 160$	+	$1 \cdot 3$	$q_2 = \lfloor 160/3 \rfloor = 53$
3	$(160 - 53 \cdot 3)$	=	$(1 - 53 \cdot 0) \cdot 160$	+	$(0 - 53 \cdot 1) \cdot 3$	
4	1	=	$1 \cdot 160$	+	$(-53) \cdot 3$	$q_3 = \lfloor 3/1 \rfloor = 3$
	0					

Tabelle 3: Inversion von 3(mod 160) mit dem EEA

Wie in Tabelle 3 ersichtlich, erhalten wir  $d \equiv -53 \equiv 107 \pmod{160}$  und könnten leicht prüfen, dass  $e \cdot d \equiv 3 \cdot 107 \equiv 321 \equiv 1 \pmod{160}$ . Um eine Nachricht  $m = 10$  zu signieren, berechnet man  $s \equiv m^d \equiv 10^{107} \equiv 54 \pmod{187}$ . Zur Prüfung der Nachricht berechnet man  $s^e \equiv 54^3 \equiv 10 \pmod{187}$  und stellt fest, dass die Signatur gültig ist.

Die naive Verwendung des RSA-Verfahrens für digitale Signaturen, wie sie hier vorgestellt wurde, birgt allerdings eine Reihe von Gefahren.

Zum einen kann ein Angreifer eine willkürliche Signatur  $s$  wählen und durch

Exponentiation mit dem öffentlichen Exponenten  $e$  die zu dieser Signatur gehörige Nachricht  $m \equiv s^e \pmod{n}$  bestimmen. Während die auf diese Weise erzeugte „Nachricht“  $m$  wahrscheinlich kein sinnvoller Text ist, sollte man solche „existentiellen Fälschungen“ für die Wahrung der *Nichtabstreitbarkeit* dennoch vermeiden.

Zu dem ist das RSA-Verfahren „multiplikativ“. Aus zwei Nachrichten  $m_1, m_2$  und den zugehörigen Signaturen  $s_1 \equiv m_1^d \pmod{n}$  und  $s_2 \equiv m_2^d \pmod{n}$  erhält man für die Nachricht  $m \equiv m_1 \cdot m_2 \pmod{n}$  die Signatur  $s \equiv s_1 \cdot s_2 \equiv m_1^d \cdot m_2^d \equiv (m_1 \cdot m_2)^d \equiv m^d \pmod{n}$ .

Um diese beiden Angriffe auszuschließen, verwendet man in der Praxis meist<sup>22</sup> *Hashfunktionen* (vgl. *Abschnitt 3.1.4*), um existentielle Fälschungen zu verhindern und die multiplikative Struktur beim RSA-Verfahren zu zerstören. Insbesondere verwendet das nachfolgend diskutierte und in der Praxis sehr weit verbreitete *PKCS #1*-Format eine *Hashfunktion* und mehr oder weniger ausgefeilte *Paddingmechanismen*.

Außerdem kann ein Angreifer versuchen, dem Prüfer der Signatur seinen eigenen öffentlichen Schlüssel unter einem falschen Namen unterzuschieben, so dass dieser vom Angreifer gefälschte Signaturen positiv verifiziert. Damit solche Angriffe nicht erfolgreich sind, setzt man Zertifikate ein, die den öffentlichen Schlüssel und den Namen des Schlüsselinhabers tragen und von einer vertrauenswürdigen Stelle signiert werden (vgl. *Abschnitt 3.2*).

Weitere Angriffe auf das RSA-Kryptosystem, von denen die meisten aber nur für die Verschlüsselungsvariante anwendbar sind, sind auch in [Bone99] diskutiert.

#### **Signaturformate für RSA**

Für das RSA-Verfahren sind verschiedene Signaturformate standardisiert, die sich im Wesentlichen dadurch unterscheiden, wie der Hashwert vor der Anwendung des RSA-Verfahrens mit Füllzeichen (*Padding*) aufgefüllt wird.

#### **PKCS #1**

Der *PKCS #1* - Standard spezifiziert ein RSA-basiertes und in der Praxis sehr weit verbreitetes Signaturformat für elektronische Signaturen, bei dem die Signatur von der Nachricht separiert ist.

Hierbei wird neben der oben beschriebenen Anwendung des RSA-Algorithmus insbesondere spezifiziert, wie die Nachricht  $m$  auf den tatsächlich durch Exponentiation mit dem privaten Schlüssel  $d$  signierten Wert  $\bar{m} \in (\mathbb{Z}/n\mathbb{Z})^*$  abgebildet wird. Wegen der oben genannten Sicherheitsaspekte und um idealerweise beliebig lange Nachrichten signieren zu können, wird nicht die Nachricht selbst, sondern vielmehr deren kryptographischer Hashwert für diese Codierung herangezogen.

PKCS #1 liegt derzeit in Version 2.1 [PKCS1(v2.1), RFC3447] vor und enthält zwei Varianten für die Codierung des Hashwertes der Nachricht:

---

<sup>22</sup>Das in [ISO9796-1] definierte Signaturformat bildete hier eine Ausnahme. Hier wurde keine Hashfunktion eingesetzt, sondern die Nachricht vor der Signaturerzeugung um von der Nachricht unabhängige Füllzeichen ergänzt. Wegen der Angriffe aus [CNS99, CHJ99] sollte dieses Signaturformat heute nicht mehr verwendet werden. Außerdem zeigen die Betrachtungen in [BCCN01], dass ein deterministisches *Padding* grundsätzlich problematisch ist, sofern nicht mindestens zwei Drittel der Nachricht aus Füllzeichen bestehen. Weitere Angriffe gegen *Padding*-Mechanismen für RSA-basierte Signaturen finden sich auch in [Misa98].

- PKCS #1 Version 1.5  
ist eine einfache und in der Praxis häufiger eingesetzte Codierungsvariante, die bereits Gegenstand von [PKCS1(v1.5), RFC2313] war.
- Probabilistic Signature Scheme  
ist eine auf [BeRo96, BeRo98] zurückgehende, erstmals in PKCS #1 Version 2.1 enthaltene Codierungsvariante, für die ein Sicherheitsbeweis (im so genannten *Random Oracle Model*) existiert, wonach das Fälschen einer Signatur in einem adaptiven aktiven Angriff äquivalent zur Lösung des RSA-Problemes (Berechnung  $e$ -ter Wurzeln modulo  $n$ ) ist.

### PKCS #1 Version 1.5

Bei der bereits in [PKCS1(v1.5), RFC2313] spezifizierten Codierungsvariante erhält man den Wert  $\overline{m} \in \mathbb{Z}$  aus dem folgendermaßen aufgebauten Octet-String  $\overline{M}$  der Länge  $l_{\overline{M}} = \lceil (l_n - 1)/8 \rceil$ , wobei  $l_n$  die Bitlänge des RSA-Moduls  $n$  ist:

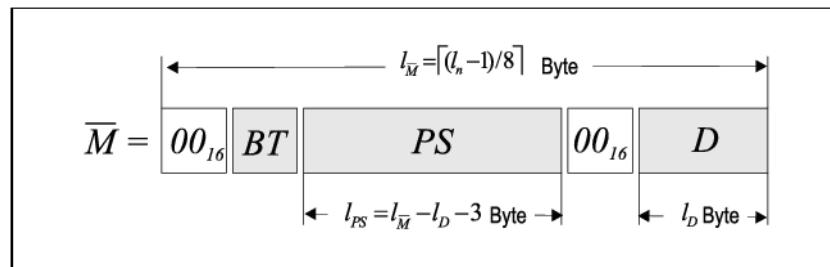


Abbildung 8: PKCS #1 Version 1.5 Codierung

Hierbei sind die Teile von  $\overline{M}$  folgendermaßen gegeben:

- $BT$  – Blocktyp  
ist bei der elektronischen Signatur mit dem Wert  $BT = 01_{16}$  belegt. Daneben definiert PKCS #1 noch die beiden Blocktypen  $BT = 00_{16}$  und  $BT = 02_{16}$ . Blocktyp  $02_{16}$  wird bei der Verschlüsselung eingesetzt. Blocktyp  $00_{16}$  sieht überhaupt kein Padding vor, so dass die Angriffe aus [DeOd85, Misa98, CNS99, CHJ99, BCCN01] möglich wären. Deshalb sollte dieser Blocktyp nicht eingesetzt werden.
- $PS$  – Paddingstring  
füllt die zur Verfügung stehenden  $l_{PS} = l_{\overline{M}} - l_D - 3$  Bytes mit vom Blocktyp abhängigen Füllzeichen. Beim Blocktyp  $01_{16}$  ist der Wert der Füllzeichen gleich  $FF_{16}$ .
- $D$  – Digest  
ist die DER<sup>23</sup> -Codierung des DigestInfo – einer Folge aus *Hashalgorithmus* und *Hashwert*.  
Die Länge  $l_D$  des DER-codierten DigestInfo hängt von der eingesetzten *Hashfunktion* ab (vgl. [RFC3447] und Tabelle 4).

### Probabilistic Signature Scheme

Bei der erstmals in [PKCS1(v2.1), RFC3447] spezifizierten Codierungsvariante werden nicht

<sup>23</sup>Während in PKCS #1 Version 1.5 noch die BER-Codierung gefordert wurde, muss ab Version 2.0 die DER-Codierung angewandt werden, so dass es bei bestimmten Hashfunktionen – zumindest theoretisch – zu Interoperabilitätsproblemen zwischen den verschiedenen Versionen kommen könnte.

wie bei [PKCS1(v1.5)] fest definierten Füllzeichen verwendet, sondern das *Padding* in Abhängigkeit von der Nachricht und einem Zufallswert berechnet.

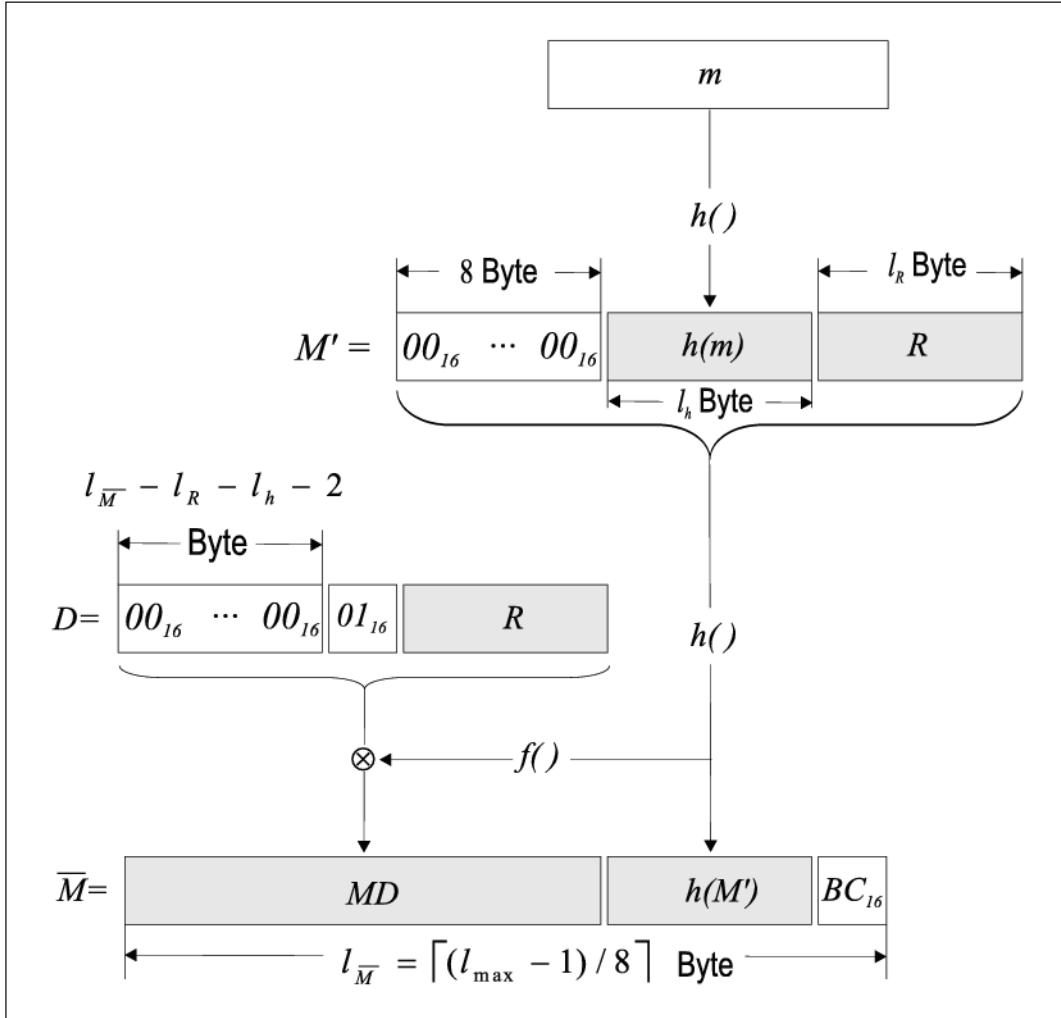
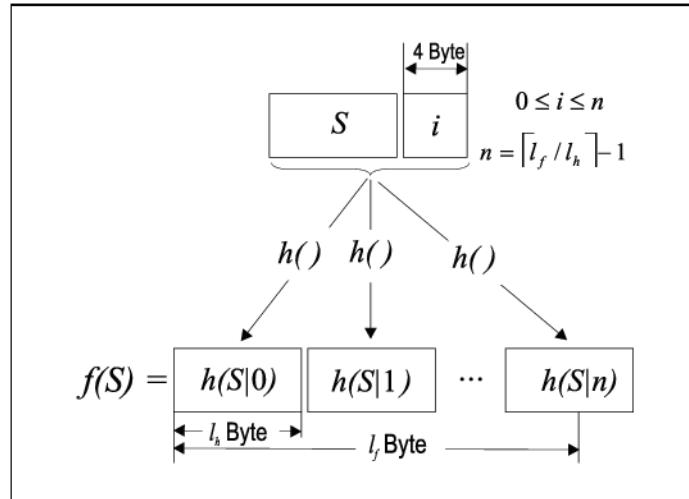


Abbildung 9: PSS-Codierung (PKCS #1 Version 2.1)

Wie in Abbildung 9 ersichtlich, werden die Füllzeichen  $MD$  durch die exklusive Oder (XOR) - Verknüpfung des im Wesentlichen aus dem Zufallswert  $R$  bestehenden Wertes  $D$  mit einer aus der Nachricht und dem Zufallswert  $R$  abgeleiteten Maske  $f(h(M'))$  berechnet.

Die Funktion  $f()$  zur Berechnung dieser Maske kann, wie in [PKCS1(v2.1), Anhang B 2.1] erläutert und in Abbildung 10 dargestellt, leicht aus der *Hashfunktion*  $h()$  konstruiert werden. Hierbei wird an den eingehenden Wert  $S$  ein fortlaufender Index  $i$  angehängt und dieser String  $S|i$  jeweils der *Hashfunktion*  $h()$  übergeben.



**Abbildung 10: f()-Funktion zur Maskenerzeugung aus PKCS #1 v2.1**

Der Octet-String  $\overline{M}$ , der schließlich der eigentlichen Signaturerzeugung (Exponentiation mit dem privaten Schlüssel  $d$ ) zugeführt wird, besteht aus folgenden drei Teilen:

- $MD$   
ist der mit  $f(h(M'))$  durch XOR maskierte Wert von  $D$ . Hierbei setzt sich  $D$ , wie in Abbildung 9 ersichtlich, zusammen aus einer Folge von  $l_{\overline{M}} - l_R - l_h - 2$  Null-Bytes, einem  $01_{16}$ -Byte und dem Zufallswert  $R$  der Länge  $l_R$ .
- $h(M')$   
ist der *Hashwert* von  $M'$ , wobei sich  $M'$  aus 8 Null-Bytes, dem *Hashwert*  $h(m)$  der Nachricht  $m$  und den  $l_R$  Zufallsbytes  $R$  zusammensetzt.
- $BC_{16}$   
ist ein konstanter Wert, der die Signatur abschließt und aus Gründen der Kompatibilität zu den Rabin-Williams-Varianten in [IEEE-P1363] und [ISO9796-2] eingeführt wurde.

Diese Vorgehensweise hat den Vorteil, dass auch ein besonders mächtiger<sup>24</sup> Angreifer in der Praxis nur dann Signaturen fälschen kann, wenn er auch das RSA-Problem lösen kann [BeRo96, BeRo98].

### Weitere Signaturformate für RSA

Neben den im PKCS #1-Standard festgelegten Signaturformaten existieren weitere Standards für RSA, die aber in der Praxis weniger weit verbreitet sind.

Der Standard [ISO9796-2] erlaubt es, einen Teil der Nachricht, oder bei entsprechend kurzen Nachrichten die gesamte Nachricht, aus der Signatur zu rekonstruieren. RSA-Signaturen gemäß [ANSI-X9.31] sind ähnlich aufgebaut wie [PKCS1(v1.5)]-Signaturen, wobei andere Füllzeichen verwendet werden und die bei der Signaturerzeugung verwendete *Hashfunktion*

<sup>24</sup>Bei der Mächtigkeit eines Angreifers gegen ein Signatursystem unterscheidet man grundsätzlich zwischen passiven und aktiven Angreifern. Ein aktiver Angreifer besitzt die Möglichkeit, Signaturen für von ihm gewählte Nachrichten berechnet zu bekommen. Hierbei unterscheidet man wiederum danach, ob er alle Nachrichten, die er signiert haben möchte, zu Beginn seines Angriffes wählen muss, oder ob er die Nachrichten, die er signiert haben möchte, im Laufe des Angriffs anpassen darf. Hier spricht man von einem „adaptiven aktiven Angriff“ (adaptively chosen message attack). Idealerweise ist ein Signaturverfahren auch gegen solche Angriffe immun.

anders codiert wird (vgl. [Dono01]).

### 3.1.3.2 Signaturalgorithmen auf Basis des Diskreten Logarithmus

Das erste Signaturverfahren, das auf dem *Diskreten Logarithmus-Problem (DLP)* basiert, wurde im Jahr 1984 von Taher ElGamal beschrieben [ElGa85]. Daraufhin schlug Claus-Peter Schnorr [Schn89, Schn91] vor, maßgebliche Teile der Signaturberechnung nicht in der kompletten Gruppe  $\mathbb{F}_p^*$ , sondern lediglich in einer Untergruppe der Ordnung  $q$  mit  $q|(p-1)$  durchzuführen, was zu deutlich schnelleren Berechnungen und kleineren Signaturen führt. Auf Basis dieser Ansätze wurde der Digital Signature Algorithm (DSA) [FIPS186] im Jahr 1994 vom *NIST* standardisiert. Der Elliptic Curve Digital Signature Algorithm (ECDSA) [ANSI-X9.62] ist die von *ANSI* standardisierte Variante in der Punktegruppe einer elliptischen Kurve.

#### ElGamal-Signatur

Der Signaturalgorithmus von ElGamal [ElGa85] hat den gleichen Systemaufbau wie das Diffie-Hellman-Verfahren zum Schlüsselaustausch [DiHe76]. Hierzu wählt man eine große *Primzahl*  $p$  und einen Erzeuger  $g$  der multiplikativen *Gruppe*  $\mathbb{F}_p^*$  des *endlichen Körpers*  $\mathbb{F}_p$ , so dass die Berechnung *Diskreter Logarithmen* in  $\mathbb{F}_p^*$  in der Praxis nicht möglich ist.

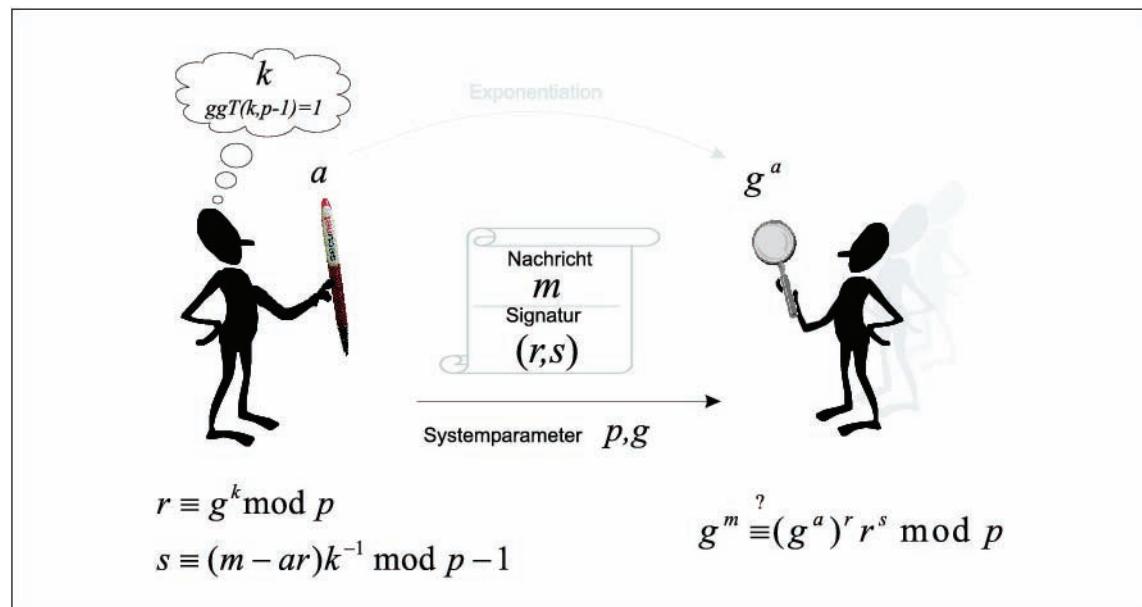


Abbildung 11: Das ElGamal-Signaturverfahren

Wie in Abbildung 11 ersichtlich, besteht der *private Schlüssel* aus einer Zufallszahl  $a$  mit  $1 \leq a \leq p-2$ . Der zugehörige *öffentliche Schlüssel*  $g^a$  wird durch eine Exponentiation berechnet.

Für die Erzeugung einer Signatur wählt man eine Zufallszahl  $k$ , die modulo  $p-1$  invertierbar ist, und erzeugt die Signatur  $(r, s)$  für eine Nachricht  $m$  durch eine Exponentiation modulo  $p$  und eine Berechnung modulo  $p-1$ . Hierbei ist  $r \equiv g^k \pmod{p}$  und  $s \equiv (m - ar)k^{-1} \pmod{p-1}$ . Zur Prüfung der Signatur berechnet man  $(g^a)^r r^s \pmod{p}$  und vergleicht diesen Wert mit  $g^m \pmod{p}$ . Die Korrektheit des Signaturverfahrens ist gegeben, da

$$(3.1.4) \quad (g^a)^r r^s \equiv g^{ar} g^{k(m-ar)k^{-1}} \equiv g^m \pmod{p}.$$

Um die modulare Inversion zur Berechnung von  $k^{-1} \pmod{p-1}$  bei der Signaturerzeugung zu vermeiden, kann auch eine Variante dieses Verfahrens [AMV90] verwendet werden, bei der nicht die Zufallszahl  $k$ , sondern der *private Schlüssel*  $a$  modulo  $p-1$  invertierbar ist. Ähnliche Variationen des Verfahrens sind in [HMP94, HMP95] beschrieben.

**Beispiel 3.1.6** Wir demonstrieren die ElGamal-Signatur anhand eines kleinen Beispiels: Wir wählen  $p = 11$  und den Erzeuger  $g = 2$  der multiplikativen Gruppe  $\mathbb{F}_{11}^*$ . Als privaten Schlüssel wählen wir  $a = 8$  und erhalten  $g^a \equiv 2^8 \equiv 3 \pmod{11}$  als öffentlichen Schlüssel. Um die Nachricht  $m = 5$  zu signieren, wählen wir ein zufälliges  $k = 9$ , das  $\text{ggT}(k, p-1) = \text{ggT}(9, 10) = 1$  erfüllt. Nun kann die Berechnung der Signatur  $(r, s)$  durchgeführt werden. Wir erhalten  $r \equiv g^k \equiv 2^9 \equiv 6 \pmod{11}$  und müssen  $s \equiv (m - ar)k^{-1} \equiv (5 - 8 \cdot 6)9^{-1} \pmod{p-1}$  berechnen. Das Element  $9 \pmod{10}$  ist zu sich selbst invers; es gilt  $9 \cdot 9 \equiv 1 \pmod{10}$  und wir erhalten  $s \equiv (5 - 8 \cdot 6)9 \equiv 3 \pmod{10}$ . Die Signatur für die Nachricht  $m = 5$  ist also das Paar  $(6, 3)$ . Für die Prüfung der Signatur muss schließlich  $(g^a)^r r^s \equiv 3^6 \cdot 6^3 \equiv 10 \pmod{11}$  berechnet und mit dem Wert  $2^5 \equiv 10 \pmod{11}$  verglichen werden.

Da das Verfahren vollständig gebrochen wäre, wenn ein Angreifer den *privaten Schlüssel* aus dem *öffentlichen* errechnen könnte, müssen die Parameter so gewählt werden, dass die Lösung des *Diskreten Logarithmus-Problems* in der Praxis nicht möglich ist. Gemäß der Bekanntmachung der Bundesnetzagentur [BNetzA-Alg05] bezüglich des unten näher erläuterten DSA-Verfahrens, bietet die Wahl einer *Primzahl*  $p$  mit 1024 Bit ausreichend Sicherheit bis Ende 2007 – ab 2010 müssen 2048-Bit Moduli eingesetzt werden.

Die Berechnung von  $r \equiv g^k \pmod{p}$  bei der ElGamal-Signatur ist unabhängig von der Nachricht und kann bereits in einer Vorausberechnungsphase gemacht werden. Allerdings ist darauf zu achten, dass die Zufallszahl  $k$  nicht öffentlich bekannt wird. Denn sonst könnte der *private Schlüssel*  $a \equiv (m - sk)r^{-1} \pmod{p-1}$  leicht aus einer einzigen Signatur  $(r, s)$  berechnet werden. Hier sei angemerkt, dass auch eine teilweise Kenntnis der verwendeten Zufallszahlen problematisch ist [NgSh02]. Besonders effektiv ist ein solcher Angriff bei kurzen *privaten Schlüsseln* und Zufallszahlen, die bei naiven Implementierungen auch in der Praxis eingesetzt werden (vgl. [Nguy04]).

Würde man die Zufallszahl mehrfach verwenden, wäre das auch problematisch, denn aus den beiden Nachrichten  $m_1$  und  $m_2$  und den zugehörigen Signaturen  $(r, s_1)$  und  $(r, s_2)$  kann man mit großer Wahrscheinlichkeit die Zufallszahl  $k$  bestimmen.

Hierzu betrachtet man das Gleichungssystem

$$\begin{aligned} s_1 &\equiv (m_1 - ar)k^{-1} \pmod{p-1} \\ s_2 &\equiv (m_2 - ar)k^{-1} \pmod{p-1} \end{aligned}$$

und erhält

$$s_1 - s_2 \equiv (m_1 - ar)k^{-1} - (m_2 - ar)k^{-1} \equiv (m_1 - m_2)k^{-1} \pmod{p-1}.$$

Ist nun der Wert  $s_1 - s_2$  modulo  $p-1$  invertierbar, so kann

$$k \equiv (m_1 - m_2)(s_1 - s_2)^{-1} \pmod{p-1}$$

berechnet und der *private Schlüssel*  $a$  wie oben beschrieben ermittelt werden.

Auch beim ElGamal-Signaturverfahren, wie es hier vorgestellt wurde, ist – ähnlich wie beim in Abschnitt 3.1.3.1 diskutierten RSA-Verfahren – die existentielle Fälschung einer Signatur möglich (vgl. [ElGa85] und [Buch99]). Deshalb sollte man auch bei diesem Verfahren nicht die Nachrichten selbst, sondern nur *Hashwerte* von Nachrichten signieren.

Außerdem ist es wichtig, dass bei der Signaturprüfung überprüft wird, ob  $1 \leq r < p$  gilt. Fehlt eine solche Prüfung, könnte ein Angreifer, wie in [Blei96] gezeigt, aus einer gültigen Signatur

$(r, s)$  für die Nachricht  $m$  eine „gültige“ Signatur  $(r', s')$  für eine beliebige Nachricht  $m'$  erzeugen. In [Blei96] wurde auch gezeigt, dass man Signaturen für beliebige Nachrichten fälschen kann, wenn der Erzeuger  $g$  nur aus kleinen *Primzahlen* besteht und ein Teiler von  $p - 1$  ist. Deshalb darf insbesondere  $g = 2$  niemals eingesetzt werden.

### Schnorr-Signatur

Das Schnorr-Verfahren zur Identifikation und Signatur wurde erstmals auf der „Rump-Session“ der EUROCRYPT 1989 vorgestellt. Das ausführliche Papier erschien wenige Wochen später im Tagungsbuch der CRYPTO [Schn89]. Es verbindet die Ideen von [ElGa85], [FiSh86] und [CEG87], um ein effizientes Signatursystem auf Basis des *Diskreten Logarithmus* zu konstruieren.

Wesentlich ist, dass nicht nur die Struktur der multiplikativen Gruppe  $\mathbb{F}_p^*$  verwendet wird, sondern auch ausgenutzt wird, dass in dieser *Gruppe Untergruppen* existieren. Welche Untergruppen in der Gruppe  $\mathbb{F}_p^*$  existieren, hängt von der Faktorisierung von  $p - 1$  ab. Da  $p$  eine große *Primzahl* ist, hat  $p - 1 = 2 \cdot r$  mindestens zwei Faktoren und deshalb mindestens zwei Untergruppen:

- eine Untergruppe der Ordnung 2 und
- eine Untergruppe der Ordnung  $r$ .

Beim Schnorr-Signaturverfahren wählt man nun die Primzahl  $p$  so, dass  $p - 1$  von einer Primzahl  $q$  mit beispielsweise 160 Bit geteilt wird und  $g$  eine Untergruppe von  $\mathbb{F}_p^*$  mit Ordnung  $q$  erzeugt. Während ein Teil der Berechnungen weiterhin modulo  $p$  durchgeführt wird, bewegt man sich dennoch in der von  $g$  erzeugten Untergruppe, so dass alle Exponenten modulo  $q$  reduziert werden dürfen. Die Länge der Signatur  $(s, t)$  ist nunmehr nur noch von der Bitlänge des Hashwertes und der Größe der Untergruppe  $q$  abhängig. Dadurch erhält man vergleichsweise kompakte Signaturen.

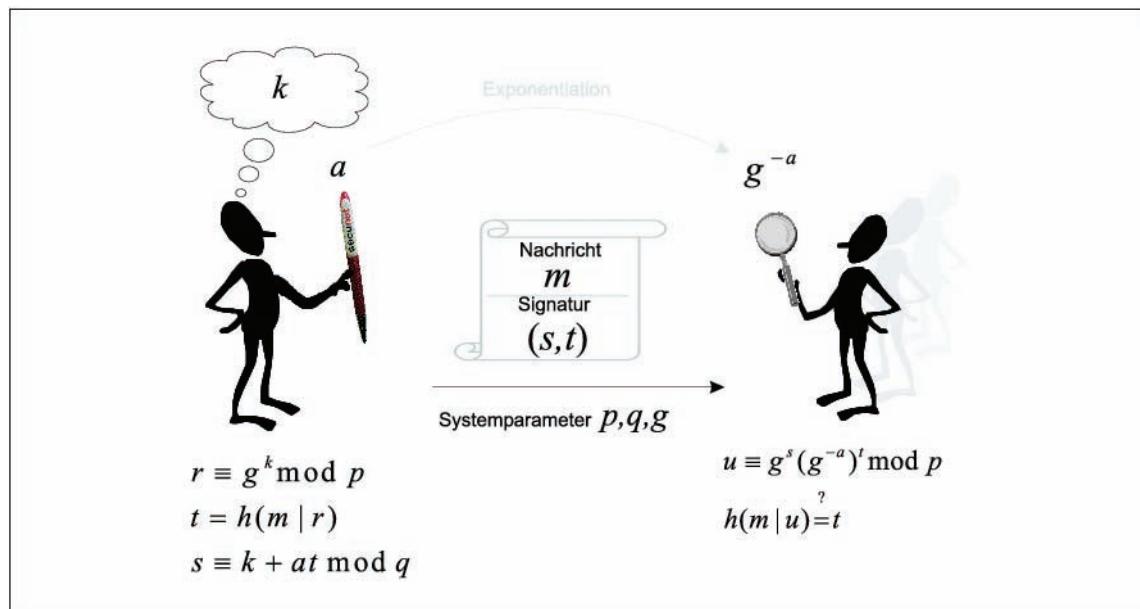


Abbildung 12: Das Schnorr-Signaturverfahren

Abbildung 12 zeigt, dass der öffentliche Schlüssel, wie bei der ElGamal-Variante aus [AMV90], durch  $g^{-a} \pmod p$  gegeben ist.

Für die Erzeugung einer Signatur wählt man – genau wie beim oben diskutierten ElGamal-

Verfahren – eine Zufallszahl  $k$  und berechnet  $r \equiv g^k \pmod{p}$ . Während der Wert  $r$  beim ElGamal-Verfahren ein Teil der Signatur ist, wird er beim Schnorr-Verfahren in einen Octet-String gewandelt, der an die Nachricht  $m$  angehängt und mittels einer *Hashfunktion*  $h$  (vgl. Abschnitt 3.1.4) auf den Wert  $t = h(m|r)$  abgebildet wird. Aus dem Wert  $t$ , der Zufallszahl  $k$  und dem privaten Schlüssel  $a$  wird schließlich  $s \equiv k + at \pmod{q}$  berechnet. Die Signatur besteht aus den beiden Werten  $(s, t)$ .

Zur Prüfung der Signatur berechnet man den Wert  $u \equiv g^s (g^{-a})^t \pmod{p}$  und prüft, ob  $h(m|u) = h(m|t)$  gilt. Diese Prüfvorschrift für die Signaturen beruht darauf, dass

$$[u_1]G + [u_2]A = [h(m)s^{-1}]G + [ars^{-1}]G = [(h(m) + ar)s^{-1}]G = [k]G.$$

**Beispiel 3.1.7** Um das Schnorr-Signatur-Verfahren zu veranschaulichen, wählen wir die Primzahlen  $q = 11$  und  $p = 23$ ,  $p - 1 = 2 \cdot q$  sowie einen Erzeuger  $g = 2$  für die Untergruppe mit Ordnung  $Q$ . Als privaten Schlüssel verwenden wir  $a = 5$  und erhalten dadurch den öffentlichen Schlüssel  $g^{-a} \equiv 2^{-5} \equiv 2^6 \equiv 18 \pmod{23}$ . Um eine Signatur für eine Nachricht  $m$  zu erzeugen, wählen wir die Zufallszahl  $k = 7$  und berechnen  $r \equiv g^k \equiv 2^7 \equiv 13 \pmod{p}$ . Nun berechnet man den Hashwert  $h(m|r)$  und erhält beispielsweise den Wert  $t = h(m|r) = h(m|13) = 4$  und dadurch  $s \equiv k + at \equiv 7 + 5 \cdot 4 \equiv 5 \pmod{q}$ . Die Signatur besteht aus den beiden Werten  $(s, t) = (5, 4)$ . Um die Signatur zu überprüfen berechnet man  $u \equiv g^s (g^{-a})^t \equiv 2^5 \cdot 18^4 \equiv 9 \cdot 4 \equiv 13 \pmod{p}$  und sieht schließlich, dass  $t = h(m|13) = 4$ .

Die Sicherheit des Verfahrens beruht auf zwei unterschiedlichen, aber eng miteinander verbundenen *DL-Problemen*: Zum einen dem *DL-Problem* modulo  $p$ , das mit dem *subexponentiellen Zahlkörpersieb* [Gord93, Webe96, JoLe03] gelöst werden kann und zum anderen auf dem *DL-Problem* in der Untergruppe der Ordnung  $q$ , das mit dem generischen  $\rho$ -Algorithmus [Poll78] gelöst werden kann. Damit diese beiden Algorithmen nicht anwendbar sind, wählt man bis Ende 2007 eine Primzahl  $p$  mit 1024 Bit, bis Ende 2009  $q$  mit 160 Bit und ab dem Jahr 2010 2048 Bit für  $p$  und 224 Bit für  $q$  [BNetzA-Alg05]. Bislang ist nicht bekannt, wie man die Untergruppen-Struktur bei der Berechnung *Diskreter Logarithmen* modulo  $p$  mit dem *Zahlkörpersieb* nutzen könnte.

Das Schnorr-Signatur-Verfahren hat übrigens die interessante Eigenschaft, dass – anders als beim Digital Signature Algorithm (DSA) – die Ordnung der *Untergruppe*  $Q$  nicht zur Verifikation benötigt wird und deshalb Teil des *privaten Schlüssels* sein kann. Dadurch lassen sich Varianten des Verfahrens konstruieren, die zusätzlich auf dem *Faktorisierungsproblem* beruhen [HuMe00, Hueh01] und eine besonders effiziente Signaturerzeugung erlauben.

### Digital Signature Algorithm (DSA)

Der Digital Signature Algorithm [FIPS186] ist eine Variante des Schnorr-Verfahrens, die im Jahr 1994 vom *NIST* standardisiert wurde. Neben der präzisen Spezifikation, wie  $p, q$  und  $g$  zu wählen sind und dem Einsatz des „Secure Hash Algorithm“ [FIPS180-1] als Hashfunktion, gibt es weitere geringfügige Unterschiede beim Systemaufbau sowie bei der Erzeugung und Prüfung der Signatur.

Für die Ordnung der Untergruppe wählt man eine Primzahl  $q$  mit  $2^{159} < q < 2^{160}$  und eine Primzahl  $p$  mit (bis zu) 1024 Bit, so dass  $q|(p-1)$ . Um einen Erzeuger  $g$  der Untergruppe der Ordnung  $q$  zu erhalten, wählt man einen zufälligen Hilfswert  $h$  mit  $1 < h < p-1$  und berechnet  $g = h^{(p-1)/q} \pmod{p}$  bis man ein  $g \not\equiv 1 \pmod{p}$  erhält. Dadurch ist sichergestellt, dass  $g$  die Untergruppe der Ordnung  $q$  erzeugt. Der private Schlüssel ist ein zufälliger Wert  $a$  mit  $0 < a < q$ . Wie beim ursprünglichen ElGamal-Verfahren ist der öffentliche Schlüssel  $g^a \pmod{p}$ . Man invertiert den privaten Schlüssel  $a$  nicht bei der Berechnung des öffentlichen Schlüssels während des Systemaufbaus, sondern muss die Zufallszahl  $k$  bei

jeder Signaturerzeugung invertieren.

Wie in Abbildung 13 angedeutet, wählt man für die Signaturerzeugung eine Zufallszahl  $k$  mit  $0 < k < q$  und berechnet den Wert  $r \equiv (g^k \pmod p) \pmod q$ . Für die Berechnung des zweiten Wertes  $s$  invertiert man die Zufallszahl  $k$  modulo  $q$  und erhält

$$(3.1.5) \quad s \equiv k^{-1}(h(m) + ar) \pmod q,$$

wobei  $h(m)$  der Hashwert der Nachricht  $m$  ist. Zur Prüfung der Signatur berechnet man  $w \equiv s^{-1} \pmod q$ ,  $u_1 \equiv h(m) \cdot w \pmod q$ ,  $u_2 \equiv r \cdot w \pmod q$  und schließlich  $r' \equiv (g^{u_1} (g^a)^{u_2} \pmod p) \pmod q$ . Die Signatur wird genau dann als gültig akzeptiert, wenn  $r' \equiv r \pmod q$ .

Diese Prüfvorschrift beruht darauf, dass

$$(3.1.6) \quad g^{u_1} (g^a)^{u_2} = g^{h(m)s^{-1}} g^{ars^{-1}} = g^{(h(m)-ar)s^{-1}} = g^k = r \pmod p.$$

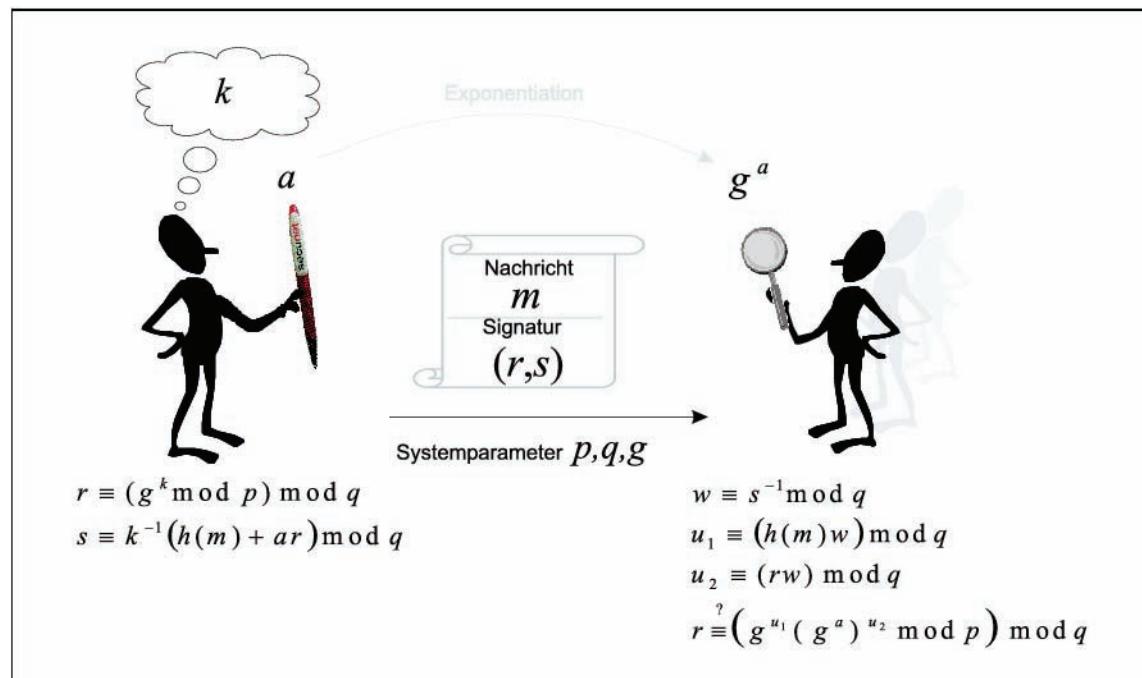


Abbildung 13: Der Digital Signature Algorithm

**Beispiel 3.1.8** Wie in Beispiel 3.1.7 wählen wir  $p = 23$ ,  $q = 11$ ,  $g = 2$  und  $a = 5$ . Den öffentlichen Schlüssel erhalten wir hier als  $g^a \equiv 2^5 \equiv 9 \pmod{23}$ . Für die Erzeugung einer Signatur wählen wir die Zufallszahl  $k = 7$  und berechnen  $r \equiv (g^k \equiv 13 \pmod{23}) \equiv 2 \pmod{11}$ . Das Inverse der Zufallszahl  $k$  modulo  $q$  ist  $k^{-1} \equiv 8 \pmod{11}$  und wir erhalten für den beispielhaften Hashwert  $h(m) = 4$  den Wert  $s \equiv k^{-1}(h(m) + ar) \equiv 8 \cdot (4 + 5 \cdot 2) \equiv 2 \pmod{11}$ . Die Signatur besteht aus den Werten  $(r, s) = (2, 2)$ . Für die Prüfung der Signatur berechnet man  $w \equiv s^{-1} \equiv 2^{-1} \equiv 6 \pmod{11}$ ,  $u_1 \equiv h(m)w \equiv 4 \cdot 6 \equiv 2 \pmod{11}$ ,  $u_2 \equiv rw \equiv 2 \cdot 6 \equiv 1 \pmod{11}$  und  $r' \equiv g^{u_1} (g^a)^{u_2} \equiv 2^2 \cdot 9^1 \pmod{23} \equiv 36 \pmod{23} \equiv 13 \pmod{23} \equiv 2 \pmod{11}$ . Da  $r' \equiv r \pmod{11}$  ist die Signatur gültig.

Die oben angeführten Sicherheitsbetrachtungen für ElGamal- und Schnorr-Signaturen sind auch beim Digital Signature Algorithm zu beachten. Insbesondere darf kein Teil der Zufallszahl  $k$  öffentlich werden [NgSh02]. Außerdem wurde in [Vaud96] gezeigt, dass ein Angreifer beim Systemaufbau bewusst solche Parameter  $p, q$  und  $g$  wählen kann, um

Signaturen zu fälschen. Deshalb ist es wichtig, dass man prüft, ob die in [FIPS186, FIPS186-2] vorgeschriebenen Prozeduren bei der Erzeugung der Systemparameter eingehalten wurden. Neben dem durch *NIST* standardisierten Digital Signature Algorithm existieren einige weitere Varianten des Verfahrens:

- Nyberg und Rueppel [NyRu94] schlugen eine DSA-Variante vor, bei der ein Teil der Nachricht aus der Signatur wiedergewonnen werden kann. Dieses Verfahren ist Bestandteil des [IEEE-P1363]-Standards.
- Der „Korean Certificate-based Digital Signature Algorithm (KCDSA)“ [KCDSA98, LiLe98] ist eine DSA-Variante, bei der die Verknüpfung des Hashwertes der Nachricht mit dem privaten Schlüssel  $a$  und der Zufallszahl  $k$  in (3.1.5) nicht durch eine Multiplikation modulo  $q$  sondern durch eine Exklusive-Oder-Operation (XOR) erfolgt. Außerdem fließt der Hashwert des Zertifikates für den öffentlichen Schlüssel  $g^a$  in die Signatur ein.
- Beim russischen DSA-Analog GOST 34.10 [MNP96] wird statt (3.1.5) die Signaturgleichung  $s \equiv ra + kh(m) \pmod{q}$  verwendet und der Einsatz von Untergruppen mit einer Größe von 256 Bit verlangt.

#### Elliptic Curve Digital Signature Algorithm (ECDSA)

Der „Elliptic Curve Digital Signature Algorithm“ (ECDSA) [JoMe99] ist ein DSA-Analog, bei dem die Gruppe  $\mathbb{F}_p^*$  durch die Punktegruppe einer *Elliptischen Kurve* ersetzt wird. ECDSA wurde 1998 von *ISO* [ISO14888-3] und *ANSI* [ANSI-X9.62] standardisiert, im Jahr 2000 in den Standard [IEEE-P1363] aufgenommen und inzwischen vom *NIST* [FIPS186-2] akzeptiert.

Wie beim Vergleich von *Abbildung 13* und *Abbildung 14* zu erkennen ist, ist das ECDSA-Verfahren ein DSA-Analog, bei dem lediglich die modulare Multiplikation in  $\mathbb{F}_p^*$  durch die Punktaddition auf einer elliptischen Kurve ersetzt wurde.

Durch die Systemparameter  $a, b, p$  wird die elliptische Kurve  $EC : y^2 \equiv x^3 + ax + b \pmod{p}$  definiert, in der der Punkt  $G$  eine zyklische *Untergruppe* mit Ordnung  $q$  erzeugt. Der *private Schlüssel* ist  $a$  mit  $1 \leq a \leq q$ , der *öffentliche Schlüssel* ist der Punkt  $A = [a]G$ .

Für die Erzeugung einer Signatur wählt man eine Zufallszahl  $k$  und berechnet den Punkt  $(x, y) = [k]G$ . Die  $x$ -Koordinate dieses Punktes wird modulo  $q$  reduziert, um den ersten Teil der Signatur  $r \equiv x \pmod{q}$  zu erhalten. Die Berechnung des zweiten Signaturwertes  $s$  erfolgt genau wie beim DSA-Verfahren in (3.1.5). Auch bei der Signaturprüfung ist die Berechnung von  $w, u_1$  und  $u_2$  identisch. Nur die Berechnung des Punktes  $(x, y) = [u_1]G + [u_2]A$  findet wieder in der Punktegruppe der elliptischen Kurve statt.

Die Prüfung der Signatur ist korrekt, da analog zu (3.1.6) Folgendes gilt:

$$(3.1.7) \quad [u_1]G + [u_2]A = [h(m)s^{-1}]G + [ars^{-1}]G = [(h(m) + ar)s^{-1}]G = [k]G.$$

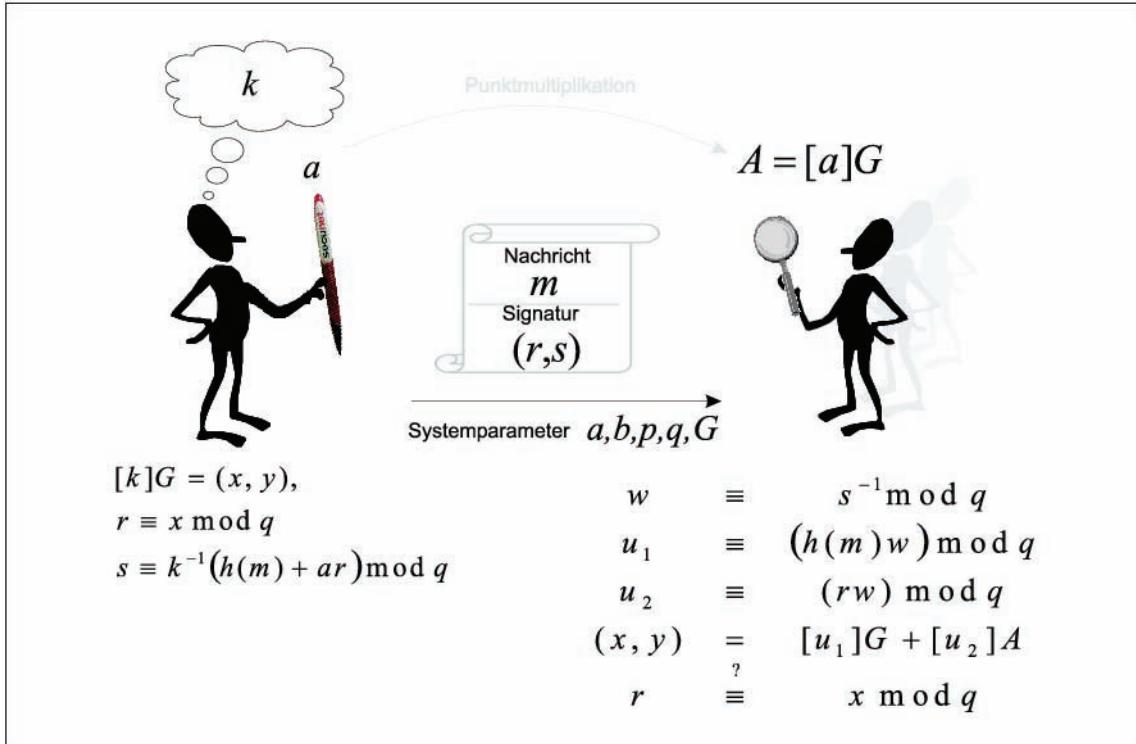


Abbildung 14: Der Elliptic Curve Digital Signature Algorithm

Für die Sicherheit des ECDSA-Verfahrens sind die gleichen Angriffe wie beim DSA zu berücksichtigen. Insbesondere muss auch hier die teilweise Veröffentlichung der Zufallszahl  $k$  vermieden werden [NgSh03].

Da für die Sicherheit des Verfahrens nun nicht mehr die Schwierigkeit des *DLP* in  $\mathbb{F}_p^*$  sondern die Schwierigkeit des *DLP* in der Punktegruppe der elliptischen Kurve maßgeblich ist, können signifikant kürzere Parameter verwendet werden (vgl. Abschnitt 3.1.2.4). Beim Siganturverfahren aus [BSL04] nutzt man die so genannte Weil-Paarung auf einer *elliptischen Kurve*, um besonders kurze Signaturen zu ermöglichen.

### 3.1.4 Hashfunktionen

Eine *Hashfunktion*  $h : \{0,1\}^* \rightarrow \{0,1\}^{l_h}$  bildet eine beliebig lange Nachricht  $m = \{0,1\}^*$  auf einen *Hashwert*  $h(m) = \{0,1\}^{l_h}$  mit der festen Bitlänge  $l_h$  ab.

Bei einer kryptographisch geeigneten Hashfunktion  $h$  fordert man, dass es praktisch nicht möglich ist, aus einem Hashwert  $h(m)$  eine zugehörige Nachricht  $m$  zu berechnen (*Einweg-Eigenschaft*, vgl. Abschnitt 3.1.2) und dass es nicht möglich ist, zwei Nachrichten  $m_1$  und  $m_2$  zu finden, die auf den gleichen Hashwert  $h(m_1) = h(m_2)$  abgebildet werden (*Kollisionsresistenz*). Deshalb bezeichnet man den Hashwert  $h(m)$  einer Nachricht  $m$  auch als „digitalen Fingerabdruck“ der Nachricht.

In der Praxis verwendet man zur Konstruktion von Hashfunktionen meist das Design-Prinzip von Merkle und Damgård [Damg89, Merk89]. Sie wird hierbei durch iterative Anwendung einer so genannten *Kompressionsfunktion*  $f : \{0,1\}^{l_b} \times \{0,1\}^{l_h} \rightarrow \{0,1\}^{l_h}$  gebildet, der man die Nachricht  $m$  blockweise zuführt.

Wie in Abbildung 15 angedeutet, wird die Nachricht  $m$  der Bitlänge  $l_m$  während der

*Initialisierungsphase* in Blöcke der Länge  $l_b$  geteilt und der letzte Block  $m_k$  durch einen *Paddingmechanismus*, bei dem auch die Länge der Nachricht berücksichtigt wird, aufgefüllt. Außerdem wird bei der Initialisierung der Wert  $h_0$  festgelegt. Während der *Iterationsphase* wird der Wert  $h_i = f(m_i, h_{i-1})$  berechnet. Der Hashwert  $h(m)$  der Nachricht  $m$  ist schließlich gleich dem Wert  $h_k$ , den man erhält, wenn auch der letzte Nachrichtenblock  $m_k$  verarbeitet ist.

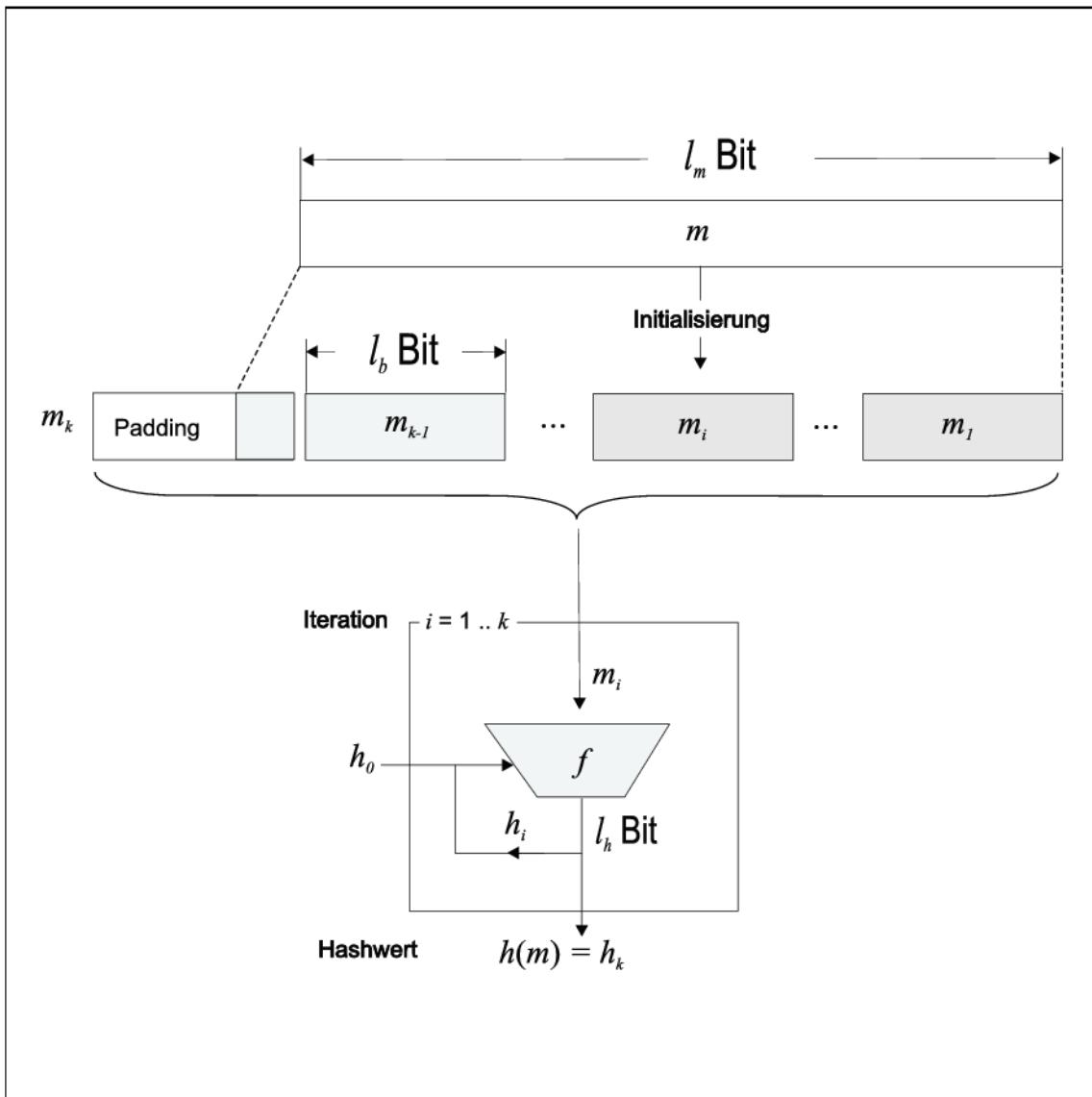


Abbildung 15: Prinzipieller Aufbau iterativer Hashfunktionen

Diesem generellen Aufbau folgen alle Hashfunktionen in *Tabelle 4*.

Hashfunktion	Länge des Hashwertes ( $l_h$ Bit)	Blocklänge ( $l_b$ Bit)	Länge des DER-codierten DigestInfo ( $l_D$ Byte)	Referenz
MD5	128	512	34	[RFC1321]
RIPEMD-160	160	512	29	[DoBP96]
SHA-1	160	512	35	[FIPS180-2]
SHA-256	256	512	51	[FIPS180-2]
SHA-384	384	1024	67	[FIPS180-2]
SHA-512	512	1024	83	[FIPS180-2]

**Tabelle 4: Populäre Hashfunktionen**

Ein naheliegender Ansatz zur Erzeugung von Kollisionen besteht darin, zufällige Nachrichten  $m$  zu wählen und zusammen mit dem zugehörigen Hashwert  $h(m)$  zu speichern, bis auf diesem Weg eine Kollision gefunden wird. Auf Grund des so genannten „Geburtstagsparadox“ (vgl. [Buch99, Kapitel 4.3]) ist die Wahrscheinlichkeit, auf diesem Weg eine Kollision zu finden, größer als  $\frac{1}{2}$ , wenn man etwas mehr als  $2^{l_h/2}$  zufällige Nachrichten wählt.

Damit man bei diesem Angriff  $2^{80}$  Hashwerte berechnen und speichern müsste, verwendet man aus Sicherheitsgründen heute nur kryptographische Hashfunktionen, die mindestens 160 Bit-Hashwerte produzieren. Schon aus diesem Grund ist die Verwendung von MD5 [RFC1321] im Kontext digitaler Signaturen problematisch, da dieser nur Hashwerte mit 128 Bit erzeugt. Spätestens seit den jüngsten Angriffen [WaYu05, Klim05, LeWe05] gilt MD5 als endgültig gebrochen.

Auch gegen die heute möglicherweise in der Praxis am häufigsten eingesetzte Hashfunktion SHA-1 gab es bereits einen Angriff, der die Erzeugung zufälliger Kollisionen mit  $2^{69}$  [WYY05a] Operationen ermöglicht. Im Rahmen der Rump-Session der CRYPTO 2005 wurde gar ein Angriff skizziert, der nur  $2^{63}$  Operationen benötigen soll<sup>25</sup>.

Wie in [DaLu05, GIS05] gezeigt, kann man bei Hashfunktionen, die gemäß den Design-Prinzipien von Merkle und Damgård konstruiert sind (vgl. Abbildung 15), aus einer einzigen, zufälligen Kollision verschiedene sinnvolle Nachrichten in populären Dokumentenformaten (z.B. Postscript, TIFF, PDF, Word 97) konstruieren. Deshalb sollten für Zwecke der elektronischen Signatur grundsätzlich keine gebrochenen Hashfunktionen, wie beispielsweise MD5, eingesetzt werden. Außerdem sollte verstärkt über die Verwendung der Nachfolger SHA-256, SHA-384 und SHA-512 nachgedacht werden. Ob die kürzlich präsentierten Angriffe auch ähnlich effektiv gegen diese Nachfolge-Hashfunktionen und gegen RIPEMD-160 eingesetzt werden können, ist bislang unklar.

## 3.2 Public-Key-Infrastrukturen

Wie in Abschnitt 3.1.3 erläutert, könnte ein Angreifer versuchen, dem Prüfer der Signatur seinen eigenen öffentlichen Schlüssel unter einem falschen Namen unterzuschieben, so dass dieser vom Angreifer gefälschte Signaturen positiv verifiziert und die vom Signierenden erstellten korrekten Signaturen für falsch hält. Damit solche Angriffe nicht erfolgreich sind, setzt man *Zertifikate* ein, die insbesondere den *öffentlichen Schlüssel* und den Namen des

<sup>25</sup>Siehe [http://www.schneier.com/blog/archives/2005/08/new\\_cryptanalyt.html](http://www.schneier.com/blog/archives/2005/08/new_cryptanalyt.html).

Schlüsselhabers tragen (vgl. Abbildung 19) und von einer vertrauenswürdigen Zertifizierungsinstanz signiert werden. Durch die Signatur wird eine Bindung zwischen dem öffentlichen Schlüssel und dem Schlüsselhaber erzeugt. Neben dieser Zertifizierungsinstanz benötigt man weitere Komponenten zur Ausgabe und Verwaltung von Zertifikaten. Die Gesamtheit dieser Komponenten und die zugehörigen Prozesse bezeichnet man als *Public-Key-Infrastruktur (PKI)*.

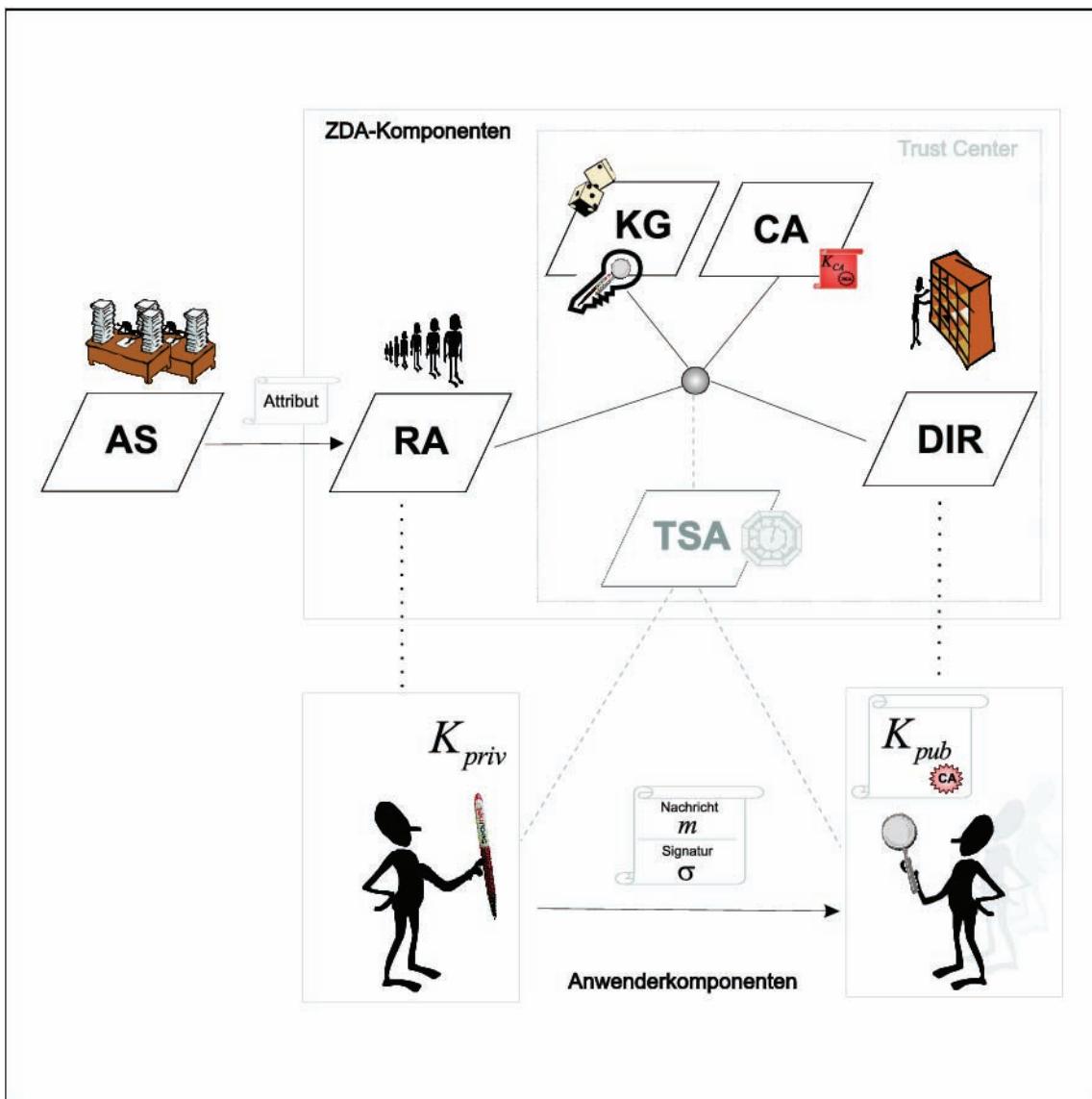


Abbildung 16: Komponenten einer PKI

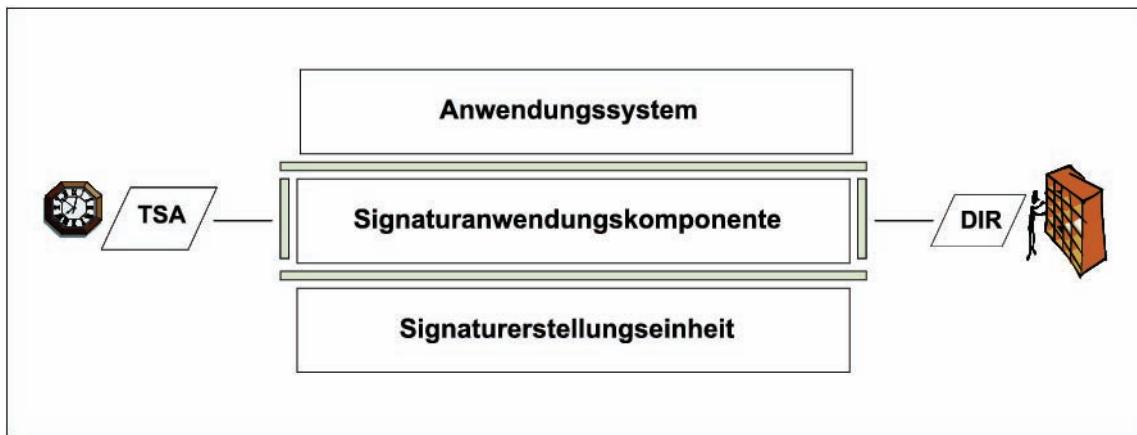
Wie in Abbildung 16 skizziert, umfasst eine solche PKI neben den Anwenderkomponenten (vgl. Abschnitt 3.2.1) auch eine Reihe von Infrastrukturkomponenten, die abgesehen von der Attributquelle (engl. Attribute Source (AS), vgl. Abschnitt 3.2.3) im Verantwortungsbereich des Zertifizierungsdiensteanbieters (ZDA) liegen. Zu diesen ZDA-Komponenten zählen

- die Registrierungsinstanz (engl. Registration Authority (RA)),
- der Schlüsselgenerator (engl. Key Generator (KG)),
- die Zertifizierungsinstanz (engl. Certification Authority (CA)),
- der Verzeichnisdienst (engl. Directory (DIR)) und möglicherweise
- der Zeitstempeldienst (engl. Time Stamping Authority (TSA)).

Abgesehen von der Registrierungsinstanz werden diese Komponenten typischerweise in einer besonders geschützten *Trust-Center*-Umgebung betrieben.

#### 3.2.1 Anwenderkomponenten

Wie in *Abbildung 17* skizziert, benötigt der Anwender zur Erstellung und Prüfung von elektronischen Signaturen im Rahmen elektronischer Geschäftsprozesse verschiedene Komponenten:



**Abbildung 17: Anwenderkomponenten**

- **Anwendungssystem**

Das Anwendungssystem steuert die generellen Abläufe und stößt die Erzeugung oder Prüfung von Signaturen bei der Signaturanwendungskomponente an. Die Grenze zwischen diesen beiden Funktionseinheiten ergibt sich aus den gesetzlichen Anforderungen an die Signaturanwendungskomponente (vgl. *Abschnitt 2.1.4*) und der daraus resultierenden Definition des *Evaluationsgegenstandes* bei einer Prüfung gemäß *ITSEC* oder *Common Criteria*.

- **Signaturanwendungskomponente**

Die *Signaturanwendungskomponente* besitzt Schnittstellen zum Anwendungssystem und zur *Signaturerstellungseinheit*, sowie Client-Server-Schnittstellen zu Zeistempel- (vgl. *Abschnitt 3.2.7*) und Verzeichnisdiensten (vgl. *Abschnitt 3.2.6*). Die Schnittstellen zum Anwendungssystem und zur Signaturerstellungseinheit werden häufig durch Programmierschnittstellen realisiert. Ein Überblick über die wichtigsten Standard-*APIs* findet sich in [Hueh05]. Sofern der private Signaturschlüssel auf einer *Chipkarte* abgelegt oder angewandt wird, umfasst die *Signaturanwendungskomponente* auch ein *Chipkartenterminal*.

- **Signaturerstellungseinheit**

In der *Signaturerstellungseinheit* wird der private Signaturschlüssel aufbewahrt und angewandt. Die Signaturerstellungseinheit kann in Form eines Hardware-Tokens, beispielsweise als *Chipkarte* (vgl. *Abbildung 18*) oder *Hardware Security Module*, oder

durch das Rechnersystem, auf dem auch die *Signaturanwendungskomponente* läuft, realisiert sein. In diesem Fall werden die privaten Schlüssel und vertrauenswürdigen Zertifikate meist in Form von mittels Passwort verschlüsselten Dateien, beispielsweise im *PKCS #12*-Format, abgelegt. Die Signaturerstellungseinheit erhält der Anwender typischerweise von der *Registrierungsinstanz* (vgl. Abschnitt 3.2.2).



**Abbildung 18: Chipkarte als Signaturerstellungseinheit**

#### 3.2.2 Registrierungsinstanz

Die Registrierungsinstanz bildet die Schnittstelle zwischen dem Anwender und den zentralen Diensten der Public-Key-Infrastruktur, die meist in einer besonders vertrauenswürdigen *Trust-Center*-Umgebung betrieben werden. Der Anwender beantragt bei der Registrierungsinstanz *Zertifikate* und veranlasst dort bei Bedarf auch wieder die Sperrung derselben.

Im Zuge des erstmaligen Registrierungsprozesses werden die Identität des Antragstellers und möglicherweise zusätzliche *Attribute* überprüft, so dass später die Korrektheit der Angaben im *Zertifikat* gewährleistet ist. Für die Feststellung der *Attribute* kommuniziert die Registrierungsinstanz mit der Attributquelle (vgl. Abschnitt 3.2.3). Sobald alle Informationen verifiziert sind, werden die für die Produktion der Zertifikate notwendigen Daten zur Zertifizierungsinstanz (vgl. Abschnitt 3.2.5) übermittelt. Hierfür kann das Certificate Management Protocol (CMP) [RFC2510] zusammen mit dem Certificate Request Message Format [RFC2511] oder alternativ das CMS-basierte Nachrichtenformat aus [RFC2797] verwendet werden.

#### 3.2.3 Attributquelle

Die Attributquelle (engl. Attribute Source, AS) bescheinigt, dass der Antragsteller für ein *Zertifikat* eine bestimmte Eigenschaft besitzt, so dass diese als *Attribut* in das beantragte Zertifikat aufgenommen werden kann.

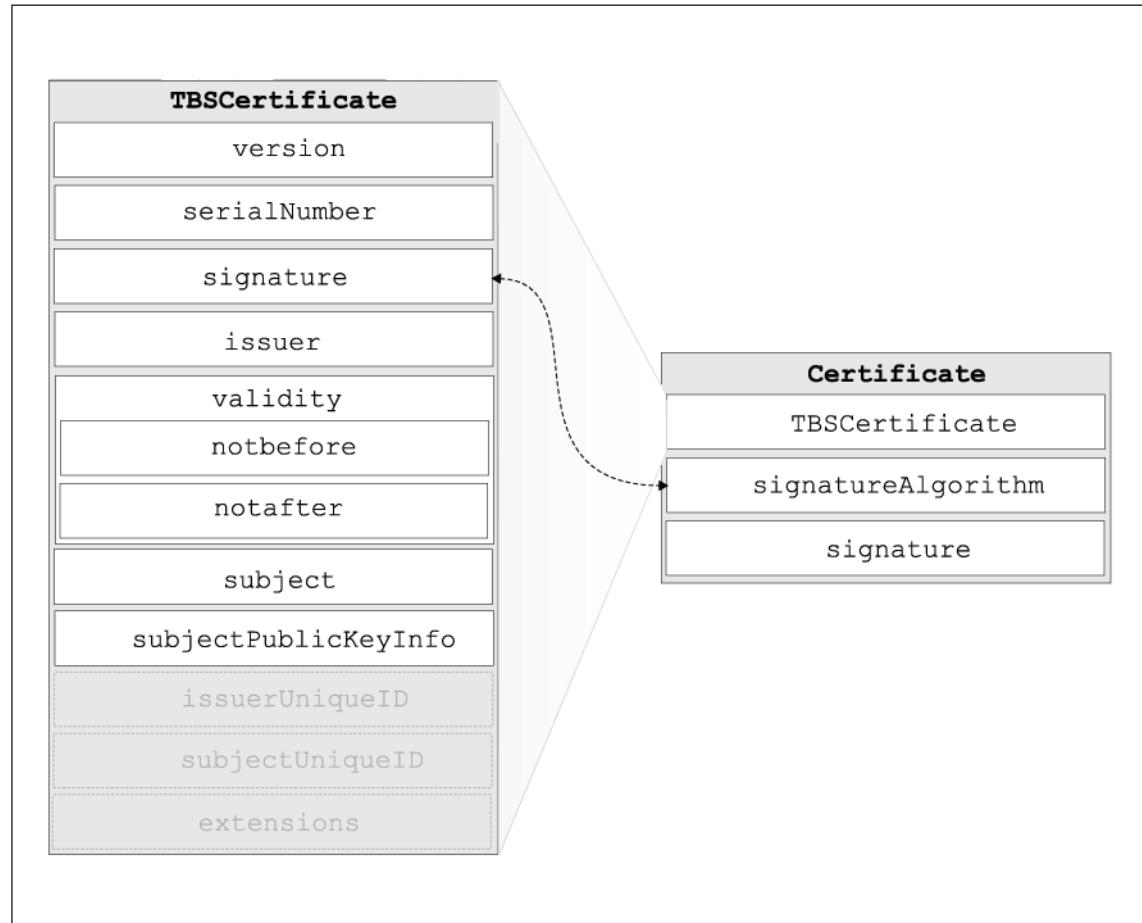
#### 3.2.4 Schlüsselgenerator

Durch den Schlüsselgenerator (engl. Key Generator, KG) werden die Schlüsselpaare entsprechend dem eingesetzten *asymmetrischen Kryptoalgorithmus* erzeugt. Beispielsweise werden für den *RSA*-Algorithmus zwei zufällige Primzahlen  $p$  und  $q$  gewählt und der *private* Schlüssel  $d$  aus dem *öffentlichen* Schlüssel  $e$  berechnet (vgl. Abschnitt 3.1.3.1). Dies kann entweder durch einen externen Schlüsselgenerator oder bei Einsatz bestimmter

Chipkarten direkt auf der Karte geschehen, so dass der private Schlüssel die Signaturerstellungseinheit niemals verlässt.

### 3.2.5 Zertifizierungsinstanz

Die Zertifizierungsinstanz (engl. Certification Authority, CA) erhält die notwendigen Daten zur Produktion der *Zertifikate* von der Registrierungsinstanz (vgl. Abschnitt 3.2.2) und erzeugt daraus die Zertifikate. Hierfür wird meist<sup>26</sup> das X.509 (Version 3)-Format [X.509:00] verwendet.



**Abbildung 19: X.509v3 - Zertifikatsformat**

Wie in Abbildung 19 angedeutet, besteht ein Certificate aus einem Zertifikatsprototyp TBSCertificate, der von der Zertifizierungsinstanz signiert wird. Dieser Zertifikatsprototyp besteht wiederum aus folgenden Daten:

- **version**  
gibt die Version des Zertifikatsformats an. Sind die optionalen **Extensions** vorhanden, so handelt es sich um ein X.509 (Version 3) - Zertifikat.
- **serialNumber**  
enthält die Seriennummer des Zertifikates. Jeder Aussteller muss dafür Sorge tragen, dass

<sup>26</sup>Alternative, aber derzeit in der Praxis nur sehr selten eingesetzte, Zertifikatsformate finden sich beispielsweise in [PKCS6, SPKC-ID, ISO9735-5, ANSI-X9.68, WAP-Cert].

er Zertifikate mit einer eindeutigen Seriennummer versieht.

- **signature**  
identifiziert den Signaturalgorithmus, der von der Zertifizierungsinstanz zur Erstellung des Zertifikates durch Signatur des Zertifikatsprototypen verwendet wird. Dieses Feld vom Typ `AlgorithmIdentifier` muss identisch sein mit dem Feld `signatureAlgorithm` in der `Certificate`-Struktur.
- **issuer**  
enthält den Namen der Zertifizierungsinstanz, die das betreffende Zertifikat erstellt und signiert.
- **validity**  
spezifiziert den Gültigkeitszeitraum des Zertifikates durch Angabe der ausgeschlossenen Grenzzeitpunkte `notbefore` und `notafter`.
- **subject**  
enthält den Namen des Zertifikatsinhabers.
- **subjectPublicKeyInfo**  
ist der *öffentliche Schlüssel* des Zertifikatsinhabers.
- **issuerUniqueID** und **subjectUniqueID**  
sind optionale Felder, um den Aussteller und Inhaber des Zertifikates eindeutig zu bestimmen, sofern `issuer` und `subject` nicht eindeutig sind. Das weithin akzeptierte Zertifikatsprofil [RFC3280] empfiehlt, diese Felder nicht zu verwenden.
- **extensions**  
wurden erst mit Version 3 der X.509-Zertifikatsstruktur eingeführt und ermöglichen die Erweiterung von Zertifikaten in definierter Art und Weise. Einer solchen Erweiterung ist eine Kritikalität, nämlich „kritisch“ oder „nicht-kritisch“, zugeordnet. Anwendungen müssen Erweiterungen, die durch die Angabe „kritisch“ gekennzeichnet werden, immer auswerten. Kennt eine Anwendung eine als kritisch markierte Zertifikaterweiterung nicht, so muss sie das Zertifikat als ungültig ansehen. Nicht-kritische Erweiterungen haben nur einen Informationscharakter und könnten bei der Gültigkeitsprüfung eines Zertifikats ignoriert werden. Zu den wichtigsten Zertifikaterweiterungen zählen die folgenden beiden, die gemäß [RFC3280, ISIS-MTT] als kritisch markiert sein müssen:

- **KeyUsage**  
spezifiziert den Nutzungszweck des privaten Schlüssels. Diese Erweiterung hat folgende ASN.1-Spezifikation:

```
KeyUsage ::= BIT  
STRING {  
    digitalSignature    (0),  
    nonRepudiation     (1),  
    keyEncipherment   (2),  
    dataEncipherment  (3),  
    keyAgreement       (4),  
    keyCertSign        (5),  
    cRLSign            (6),  
    encipherOnly       (7),  
    decipherOnly       (8) }
```

Die erlaubte Schlüsselnutzung ergibt sich daraus, welche Bits in der `KeyUsage`-Struktur gesetzt sind. Beispielsweise sind bei einem Signatur-Zertifikat die Bits `digitalSignature` und `nonRepudiation` gesetzt. Weitere Details zur Schlüsselnutzung können auch in einer zusätzlichen `ExtendedKeyUsage`-

Erweiterung spezifiziert sein. Wie in *Abschnitt 4.2.3* erläutert, wird die (Extended) KeyUsage-Extension bei der Prüfung von Signaturen ausgewertet.

- BasicConstraints

gibt an, ob es sich um ein Zertifikat einer Zertifizierungsinstanz oder eines Benutzers handelt. Ein wesentlicher Kritikpunkt an der ersten Version des X.509-Zertifikatsformats, bei der noch keinerlei Erweiterungen vorgesehen war, war die fehlende Möglichkeit der Unterscheidung zwischen CA- und Benutzerzertifikaten, so dass auch Endnutzer uneingeschränkt als Zertifizierungsinstanz hätten auftreten können.

Daneben sind insbesondere die folgenden beiden Zertifikaterweiterungen von Bedeutung, da sie nähere Auskunft über den generellen Charakter des Zertifikates geben:

- QCStatements

Die QCStatements-Erweiterung aus [RFC3039, ETSI-101862] signalisiert, dass es sich um ein *qualifiziertes Zertifikat* im Sinne des Signaturgesetzes handelt.

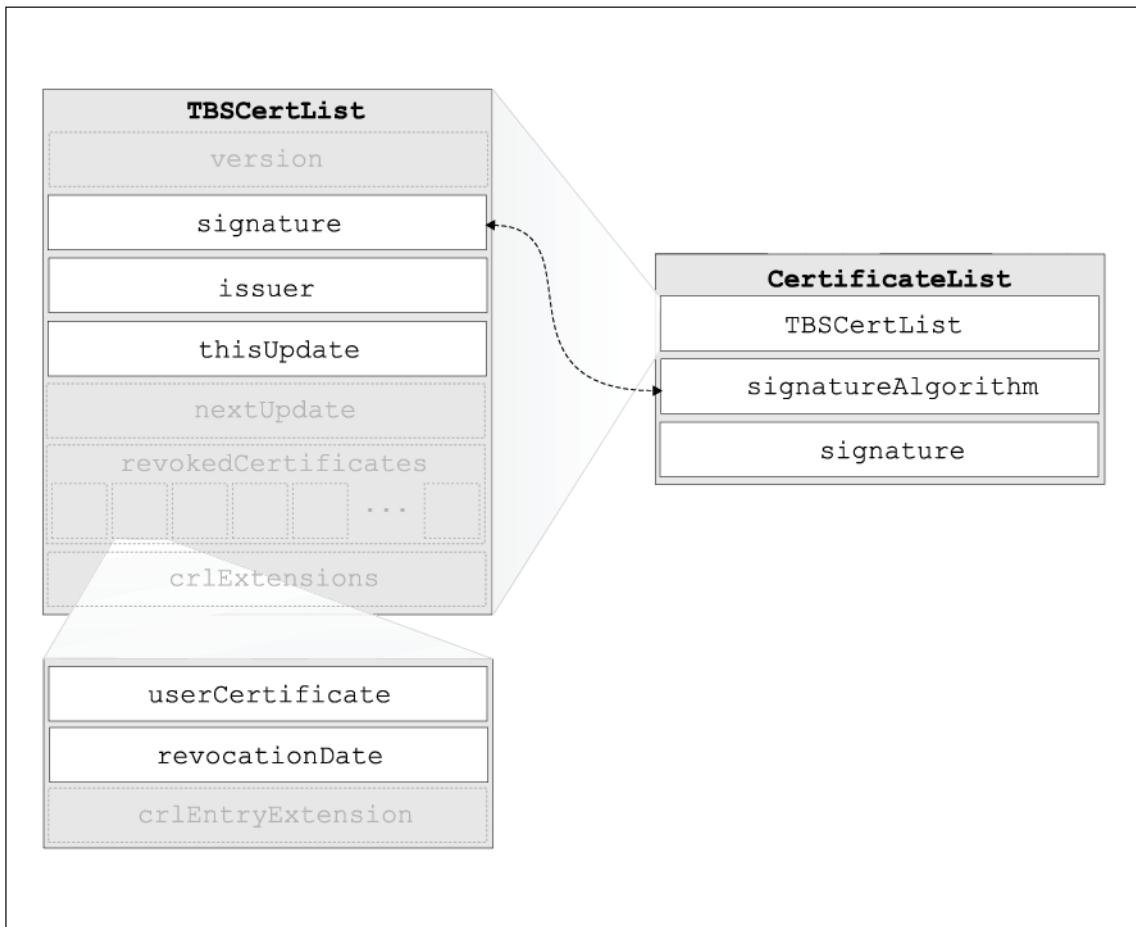
- CertificatePolicies

Durch die CertificatePolicies-Erweiterung aus [RFC3280] wird angegeben, welche *Certificate Policies* für das vorliegende Zertifikat anwendbar sind.

Weitere Erweiterungen für *Public-Key-Zertifikate* sind in [ISIS-MTT, Tabelle 10] erläutert.

Außerdem kann ein Public-Key-Zertifikat weitere *Attribute* in Form von Erweiterungen beinhalten. Alternativ dazu können solche Attribute auch in separaten *Attributzertifikaten* [RFC3281] enthalten sein, die selbst keinen *öffentlichen Schlüssel* beinhalten, sondern auf ein *Public-Key-Zertifikat* verweisen.

Neben den Zertifikaten stellt eine Zertifizierungsinstanz meist auch *Sperrlisten* aus.



**Abbildung 20: X.509v2 - CRL-Format**

Wie in *Abbildung 20* angedeutet, besteht eine CertificateList aus einem Sperrlistenprototyp TBSCertList, der von der Zertifizierungsinstanz signiert wird. Dieser Sperrlistenprototyp besteht wiederum aus folgenden Daten:

- **version**  
gibt optional die Version (1 oder 2) des Sperrlistenformates an. Fehlt sie, so dürfen keine Extensions vorhanden sein und es handelt sich automatisch um Version 1.
- **signature**  
identifiziert den Signaturalgorithmus, der von der Zertifizierungsinstanz zur Erstellung der Sperrliste durch Signatur des Sperrlistenprototypen verwendet wird. Dieses Feld vom Typ AlgorithmIdentifier muss identisch sein mit dem Feld **signatureAlgorithm** in der CertificateList-Struktur.
- **issuer**  
enthält den Namen der Zertifizierungsinstanz, die das betreffende Zertifikat erstellt und signiert.
- **thisUpdate**  
gibt den Zeitpunkt der Erstellung der Sperrliste an.
- **nextUpdate**  
ist ein optionales Feld, das den Zeitpunkt angibt, zu dem die nächste Sperrliste erstellt wird.

- revokedCertificates

enthält eine Liste der gesperrten Zertifikate, wobei das userCertificate durch seine Seriennummer identifiziert wird und der Sperrzeitpunkt (revocationDate) sowie möglicherweise weitere auf den Eintrag bezogene Erweiterungen (crlEntryExtension), wie z.B. der Grund der Sperrung (CRLReason), gespeichert werden. Der Sperrgrund hat folgende ASN.1-Spezifikation:

```
CRLReason ::= ENUMERATED {
    unspecified                      (0),
    keyCompromise                     (1),
    cACompromise                      (2),
    affiliationChanged                (3),
    superseded                        (4),
    cessationOfOperation              (5),
    certificateHold                   (6),
    removeFromCRL                    (8),
    privilegeWithdrawn                (9),
    aACompromise                      (10)}
```

Die Bedeutung der einzelnen Sperrgründe ist folgendermaßen gegeben (vgl. [X.509:00], Abschnitt 8.5.2.2):

- keyCompromise  
wird bei Anwender-Zertifikaten verwendet und zeigt an, dass mindestens der Verdacht auf Kompromittierung des privaten Schlüssels besteht. Dieser Sperrgrund wird beispielsweise verwendet, wenn die *Signaturerstellungseinheit* verloren und die *Identifikationsdaten* möglicherweise ausgeforscht wurden oder der *Signaturalgorithmus* nicht mehr sicher ist.
- cACompromise  
wird bei CA-Zertifikaten analog zu keyCompromise verwendet.
- affiliationChanged  
wird verwendet, wenn sich der Name oder andere im Zertifikat angegebenen Informationen geändert haben, aber kein Verdacht auf Kompromittierung des privaten Schlüssels existiert.
- superseded  
zeigt an, dass ein neues Zertifikat ausgestellt wurde, beispielsweise weil der private Schlüssel auf Grund einer defekten Signaturerstellungseinheit nicht mehr genutzt werden konnte.
- cessationOfOperation  
wird verwendet, wenn das Zertifikat nicht mehr für den vorgesehenen Zweck gebraucht wird, aber kein Verdacht auf Kompromittierung des privaten Schlüssels existiert. Beispielsweise kommt dieser Sperrgrund zum Einsatz, wenn der Betrieb einer Zertifizierungsinstanz eingestellt wird.
- certificateHold  
kann (bei nicht-qualifizierten Zertifikaten<sup>27</sup>) für eine temporäre Suspendierung

---

<sup>27</sup>Gemäß § 15 Abs. 3 [SigV] müssen die technischen Komponenten sicher stellen, „dass die Sperrung eines qualifizierten Zertifikates nicht unbemerkt rückgängig gemacht werden kann“.

verwendet werden.

- removeFromCRL  
wird ausschließlich bei *Delta-CRLs* eingesetzt und gibt an, dass der Eintrag in der Sperrliste aufgehoben wird, beispielsweise weil die temporäre Suspendierung des Zertifikates rückgängig gemacht wurde.
- privilegeWithdrawn  
wird verwendet, wenn ein Attribut in einem *Public-Key-* oder *Attributzertifikat* nicht mehr gültig ist.
- aACompromise  
wird eingesetzt, wenn vermutet wird, dass bei einer *Attributbestätigungsinstanz* oder bei der Ausstellung von *Attributzertifikaten* durch eine *Zertifizierungsinstanz* eine Kompromittierung vorliegt.

[RFC3280] und [ISIS-MTT] empfehlen nachdrücklich die differenzierte Verwendung dieser Sperrgründe. Anstatt den wenig aussagekräftigen Sperrgrund unspecified zu verwenden, sollen Aussteller von Sperrlisten besser komplett auf die CRLReason-Erweiterung verzichten.

- crlExtension  
enthält möglicherweise zusätzliche Erweiterungen, die sich auf die gesamte Sperrliste beziehen. Hier kann beispielsweise durch den DeltaCRLIndicator angegeben werden, dass es sich nicht um eine Sperrliste handelt, die alle gesperrten Zertifikate enthält, sondern nur die zusätzlich zu den in einer Basis-Sperrliste mit der Nummer BaseCRLNumber enthaltenen.

#### 3.2.6 Verzeichnisdienst

Der Verzeichnisdienst dient dazu, *Zertifikate* nachprüfbar (vgl. Abschnitt 4.2) und möglicherweise abrufbar zu halten. Auch wenn ein X.500-kompatibles Verzeichnis verwendet wird, so erfolgt der Abruf von Zertifikaten und *Sperrlisten* in der Regel durch das *Lightweight Directory Access Protocol (LDAP)* [RFC2251]. Durch Inspektion einer aus dem Verzeichnis heruntergeladenen Sperrliste kann lokal überprüft werden, ob ein Zertifikat gesperrt worden ist oder nicht. Da aber eine Sperrliste nur in bestimmten Zeitintervallen, beispielsweise einmal täglich, erstellt wird und das Fehlen eines Eintrags in einer Sperrliste kein Beleg dafür ist, dass ein bestimmtes Zertifikat jemals von der Zertifizierungsinstanz ausgestellt wurde, – sind die verwendeten Algorithmen nicht mehr sicher, so könnte ein Angreifer im Nachhinein selbst ein Zertifikat fälschen – verwendet man bei kritischen Applikationen und insbesondere für die Prüfung von *qualifizierten Zertifikaten* in der Regel das in [RFC2560] spezifizierte *Online Certificate Status Protocol (OCSP)*.

Hierbei handelt es sich, wie in Abbildung 21 dargestellt, um ein Client-Server-Protokoll, mit dem der aktuelle Status eines Zertifikates bei einem so genannten „OCSP-Responder“, der vom *Zertifizierungsdiensteanbieter* betrieben wird, ermittelt werden kann. Der „OCSP-Client“ baut über *TCP* eine Verbindung zum Server auf und schickt dann ggf. über *HTTP* einen OCSPRequest, der insbesondere die CertID-Struktur zur Identifikation des fraglichen Zertifikates enthält, an den „OCSP-Responder“, der daraufhin eine OCSPResponse mit dem zugehörigen Zertifikatstatus und möglicherweise weiteren Erweiterungen zurück schickt.

---

Deshalb scheint es fraglich, ob eine temporäre Suspendierung von qualifizierten Zertifikaten mit der Signaturgesetzgebung in Einklang zu bringen ist.



**Abbildung 21: OCSP im Überblick**

Wie in *Abbildung 22* dargestellt, besteht der OCSPRequest aus möglicherweise signierten und um für die Prüfung dieser Signatur notwendige Zertifikatsinformationen ergänzte Anfragedaten (TBSRequest). Diese bestehen wiederum aus folgenden Daten:

- **version**  
gibt optional die Version der TBSRequest-Syntax an.
- **requestorName**  
ist ein optionales Feld, das den Namen des Anfragenden enthält.
- **requestList**  
ist eine Liste von einzelnen Anfragen vom Typ Request. Eine solche Anfrage enthält die CertID und möglicherweise zusätzlichen Erweiterungen (singleRequestExtensions), die sich auf eine einzelne Anfrage beziehen. Beispielsweise kann der OCSP-Responder durch die in [ISIS-MTT-SigG] definierte RetrieveIfAllowed-Erweiterung angewiesen werden, auch das fragliche Zertifikat zurückzuliefern, sofern dies erlaubt ist (vgl. § 5 Abs. 1 Satz 3 [SigG]).

Die CertID setzt sich aus folgenden Informationen zusammen:

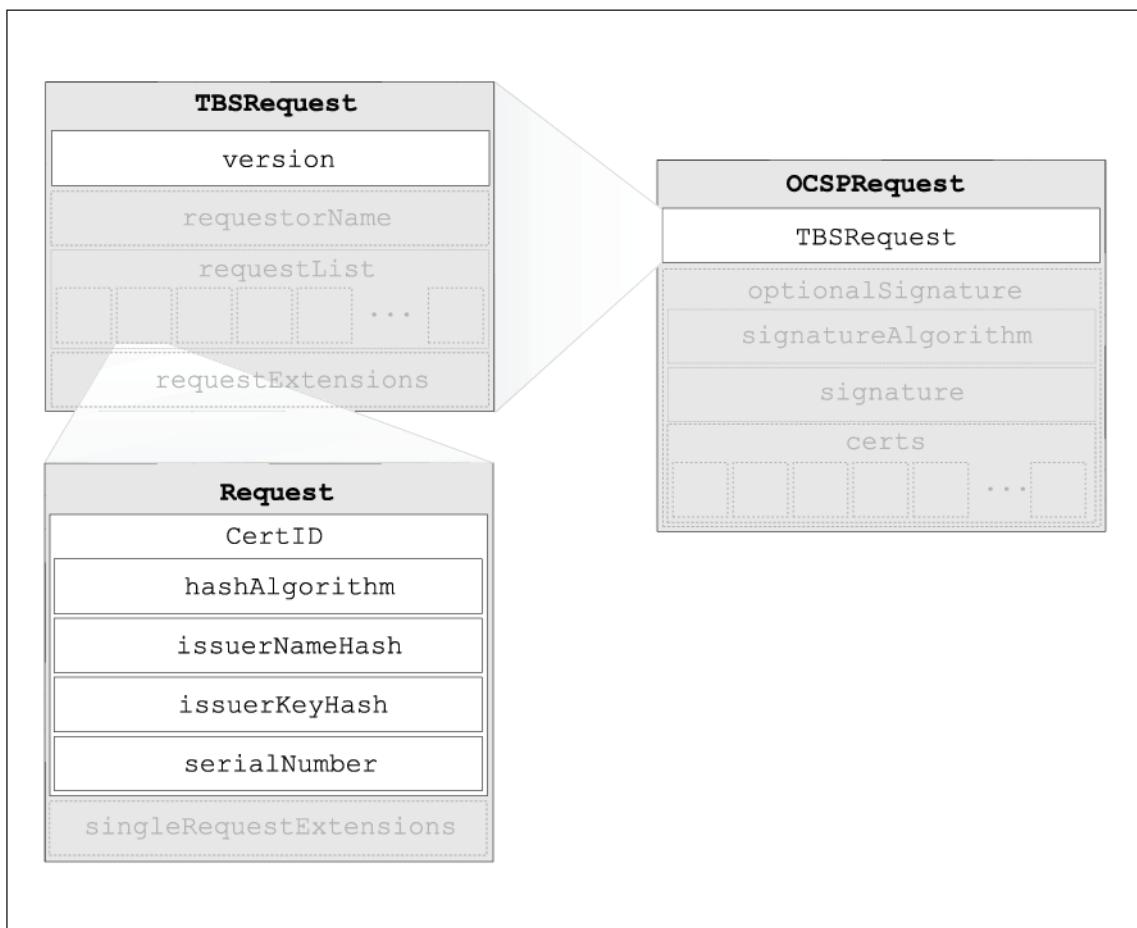
- **hashAlgorithm**  
spezifiziert den Hash-Algorithmus, der für die Berechnung der beiden folgenden Felder (issuerNameHash und issuerKeyHash) verwendet wird.
- **issuerNameHash**  
ist der *Hashwert* des *Distinguished Name* des Ausstellers des Zertifikates.
- **issuerKeyHash**  
ist der *Hashwert* des *öffentlichen Schlüssels* des betreffenden Zertifikates.
- **serialNumber**  
ist die Seriennummer des in Frage stehenden Zertifikates.
- **requestExtensions**  
enthält Erweiterungen, durch die – unter Verwendung von Zufallszahlen (Nonce) – Replay-Attacken abgewehrt werden oder die zulässigen Antworttypen (AcceptableResponses) spezifiziert werden können.

Der OCSP-Responder schickt daraufhin eine OCSPResponse, die – wie in *Abbildung 23*

ersichtlich – mindestens aus dem `responseStatus`-Feld besteht, an den Client zurück. Im Erfolgsfall wird die Antwort des OCSP-Responders auch eine `response` eines bestimmten `responseTypes` enthalten. [RFC2560] spezifiziert hierfür die `BasicOCSPResponse`, die im Wesentlichen aus den signierten `responseData` besteht.

Die `responseData` bestehen wiederum aus folgenden Informationen:

- `version`  
gibt die Version der `responseData`-Syntax an.
- `responderID`  
enthält den Namen des OCSP-Responders oder den *SHA-1*-Wert des *öffentlichen Schlüssels* des OCSP-Responders.



**Abbildung 22: OCSP-Request**

- `producedAt`  
beinhaltet den Zeitpunkt, als die OCSP-Antwort vom OCSP-Responder signiert wurde.
- `responses`  
ist eine Folge von einzelnen Antworten (`singleResponse`), die jeweils aus folgenden Informationen bestehen:
  - `CertID`  
besteht aus den oben beschriebenen Feldern (`hashAlgorithm`, `issuerNameHash`, `issuerKeyHash` und `serialNumber`) und identifiziert,

das Zertifikat eindeutig.

- certStatus  
enthält den Status des fraglichen Zertifikates, wobei die Werte good, revoked und unknown möglich sind. Ist das Zertifikat gesperrt, wird zumindest der Sperrzeitpunkt (revocationTime) und möglicherweise der Sperrgrund in Form des oben beschriebenen CRLReason-Feldes zurückgeliefert.
- thisUpdate und ggf. nextUpdate  
enthält die Zeitpunkte der letzten bzw. nächsten Aktualisierung der Datenbasis des OCSP-Responders. Wird der OCSP-Responder durch Sperrlisten gespeist, so werden diese beiden Felder aus den jüngsten *Sperrlisten* (vgl. Abschnitt 3.2.5) entnommen. Hat der OCSP-Responder, wie dies im Umfeld *qualifizierter elektronischer Signaturen* üblich ist, eine Datenbank, in der alle ausgestellten Zertifikate und der jeweils aktuelle Zertifikatstatus eingetragen ist, so wird im Feld thisUpdate der aktuelle Zeitpunkt zurückgeliefert.

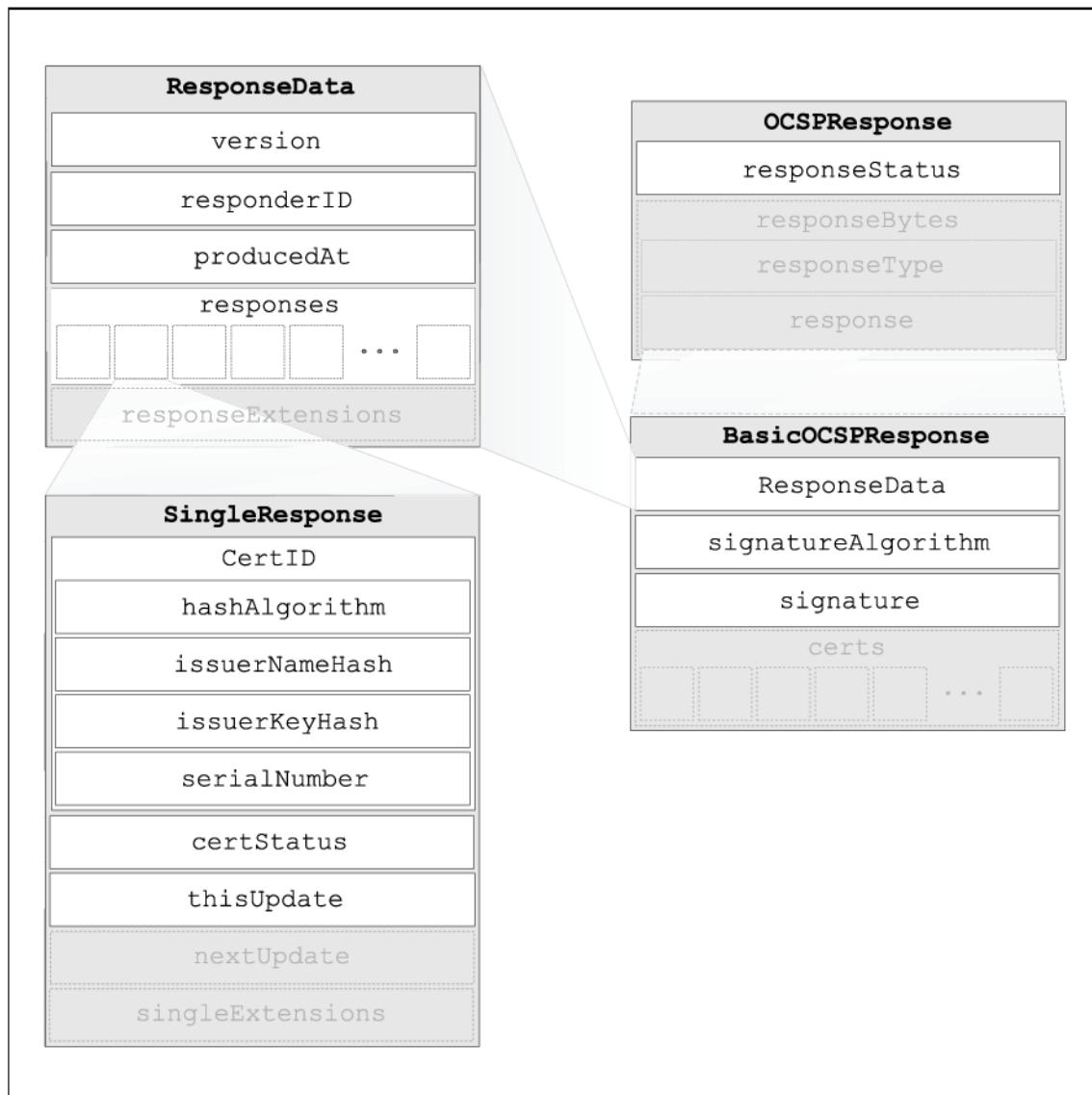


Abbildung 23: OCSP-Response

- singleExtensions  
enthält Erweiterungen, die sich auf die einzelne Antwort beziehen. Beispielsweise definiert [ISIS-MTT-SigG] die folgenden Erweiterungen:
- CertHash  
enthält den Hashwert des fraglichen Zertifikates und dient somit als Nachweis, dass der OCSP-Responder dieses Zertifikat tatsächlich kennt.
- CertInDirSince  
liefert den Zeitpunkt zurück, an dem das Zertifikat in das Verzeichnis eingestellt wurde.
- RequestedCertificate  
liefert das mittels RetrieveIfAllowed angeforderte Zertifikat zurück, sofern es abrufbar ist.
- responseExtensions  
enthält Erweiterungen, die sich auf die komplette Antwort beziehen. Beispielsweise würde hier die möglicherweise in der Anfrage enthaltene Zufallszahl (Nonce) zurückgeliefert werden.

#### 3.2.7 Zeitstempeldienst

Ein Zeitstempeldienst stellt *Zeitstempel* aus. Diese dienen gemäß [ISO18014-1] zum Nachweis, dass bestimmte Daten bereits vor einem bestimmten Zeitpunkt existiert haben. In der Praxis verwendet man hierzu meist das in [RFC3161] spezifizierte *Time Stamp Protocol (TSP)*. Beispielsweise verwenden alle heute existierenden *Zertifizierungsdiensteanbieter* dieses Protokoll zur Ausstellung von *qualifizierten Zeitstempeln*.



**Abbildung 24: TSP im Überblick**

Wie in Abbildung 24 dargestellt, handelt es sich beim *TSP* um ein Client-Server-Protocol, bei dem der Client mit dem Server eine *TCP*-Verbindung unterhält. Direkt über diese Socket-Verbindung oder über darüberliegende Protokolle, wie z.B. *HTTP*, schickt der TSP-Client eine Zeitstempelanfrage (*TimeStampReq*), die insbesondere den Hashwert der zeitzustempelnden Daten enthält, an den TSP-Responder. Dieser ergänzt im Wesentlichen den empfangenen Hashwert um die aktuelle Uhrzeit, signiert diese Daten und schickt diesen Zeitstempel an den Client zurück (*TimeStampResp*). Bei der Ausstellung von *qualifizierten Zeitstempeln* muss der *Zertifizierungsdiensteanbieter* gemäß § 15 Abs. 3 Satz 4 [SigV] die

„gültige gesetzliche Zeit“ gemäß [ZeitG] zur Produktion der Zeitstempel verwenden. Da die gesetzlich gültige Zeit über das DCF77-Funksignal von der Physikalisch-Technischen Bundesanstalt ausgestrahlt wird, verwenden alle *Zertifizierungsdiensteanbieter* TSP-Responder, die mit einem DCF77-Empfänger ausgestattet sind.

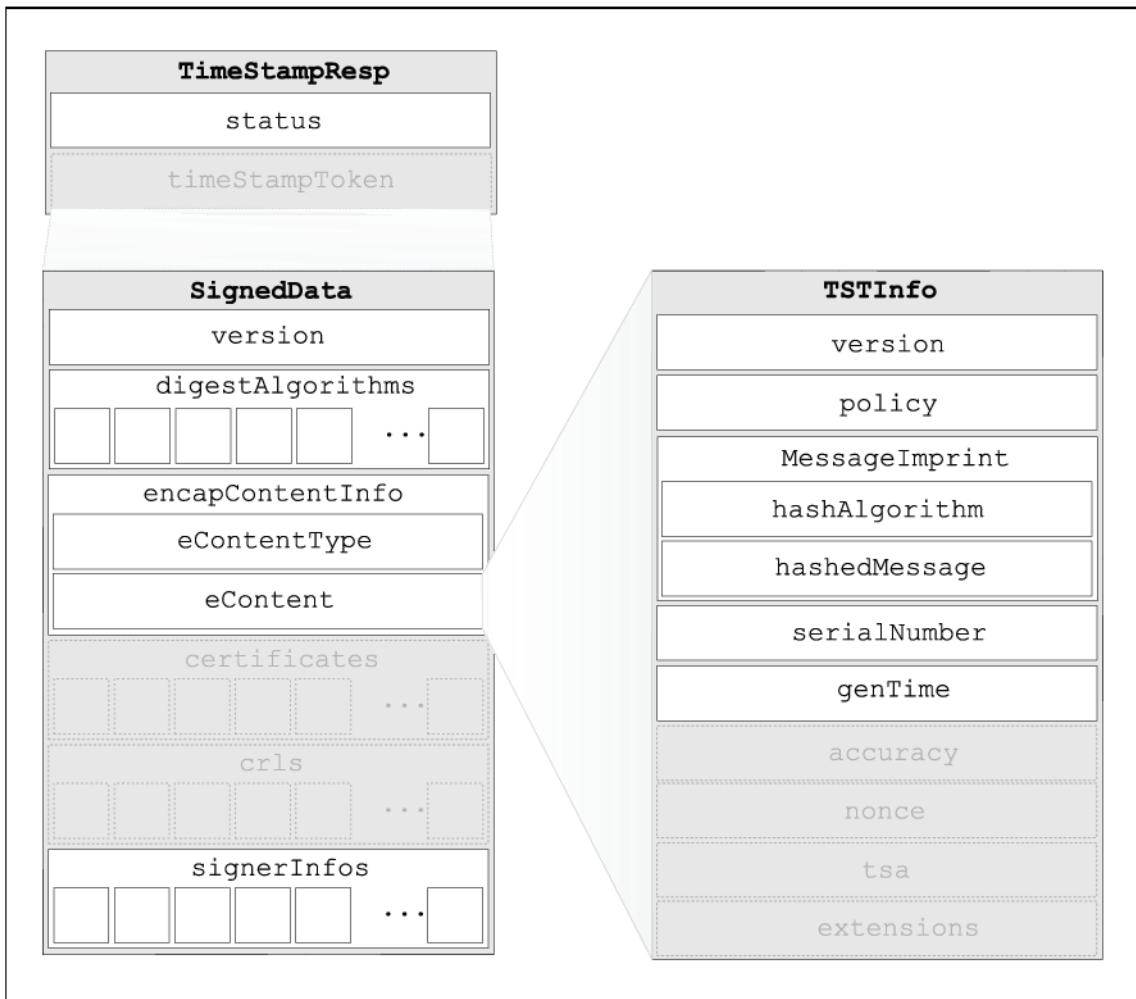


**Abbildung 25: TSP-Request**

Wie in *Abbildung 25* angedeutet, besteht eine Zeitstempelanfrage beim TSP-Protokoll (TimeStampReq) aus folgenden Daten:

- **version**  
gibt die Version der TimeStampReq-Syntax an.
- **MessageImprint**  
enthält den Hashwert der zeitzustempelnden Daten (hashedMessage) und die Information, welche Hashfunktion hierfür verwendet wurde (hashAlgorithm).
- **reqPolicy**  
gibt an, unter welcher Policy der Zeitstempel durch den Zeitstempeldienst erstellt werden soll.
- **nonce**  
ermöglicht es, in der Anfrage einen zufälligen Wert an den Server zu schicken, so dass die Aktualität der Antwort vom Client auch ohne eine zuverlässige Zeitquelle geprüft werden kann.
- **certReq**  
gibt an, ob der Zeitstempeldienst einen Verweis auf das Signaturzertifikat, auf Basis dessen der Zeitstempel erstellt wurde, als SigningCertificate-Attribut gemäß [RFC2634] in den zurückgelieferten Zeitstempel einfügen soll oder nicht.
- **extensions**  
bietet eine generische Möglichkeit für Erweiterungen. Allerdings werden in [RFC3161] keine spezifischen Erweiterungen angegeben, sondern nur auf die allgemeine Definition

von Erweiterungen in [RFC2459]<sup>28</sup> verwiesen.



**Abbildung 26: TSP-Response**

Wie in *Abbildung 26* angedeutet, besteht die Antwort des TSP-Responders zumindest aus einer Statusangabe. Im Erfolgsfall wird dem Client außerdem der gewünschte Zeitstempel (`timeStampToken`) zurückgeliefert. Bei diesem Zeitstempel handelt es sich um eine CMS-Struktur gemäß [RFC2630] vom Typ `SignedData` (vgl. *Abschnitt 4.3.1.2*), bei der der signierte Inhalt (`eContent`) aus den eigentlichen Zeitstempeldaten besteht. Die `TSTInfo`-Struktur umfasst folgende Informationen:

- `version`  
gibt die Version der `TSTInfo`-Syntax an.
- `policy`  
gibt die Policy an, unter der der Zeitstempel erzeugt wurde.
- `MessageImprint`  
enthält den Hashwert der zeitzustempelnden Daten (`hashedMessage`) und die Information, welche Hashfunktion hierfür verwendet (`hashAlgorithm`) und bei der Zeitstempelanfrage übergeben wurde.

---

<sup>28</sup>Dieser Standard wurde inzwischen durch [RFC3280] abgelöst.

- serialNumber  
ist eine ganze Zahl mit einer Größe von bis zu 160 Bit. Wie diese Zahl gebildet wird, ist abhängig von der Policy des Zeitstempeldienstes. Meist wird hier eine fortlaufende Seriennummer für die ausgestellten Zeitstempel gewählt.
- genTime  
ist der Zeitpunkt der Zeitstempelung.
- accuracy  
gibt optional die Genauigkeit der zur Zeitstempelung verwendeten Zeitquelle in Sekunden, Millisekunden und Microsekunden an.
- nonce  
ist die möglicherweise bei der Zeitstempelanfrage übergebene Zufallszahl.
- tsa  
ist der Name des Zeitstempeldienstes, wie im `subject`-Feld des zugehörigen Zertifikats angegeben. Die präzise Identifikation des Zeitstempelzertifikats erfolgt unter Verwendung des `SigningCertificate`-Attribut gemäß [RFC2634], sofern dies bei der Anfrage gewünscht wurde (vgl. `certReq` bei dem TSP-Request).
- extensions  
bietet eine generische Möglichkeit für Erweiterungen. Allerdings werden auch hier in [RFC3161] keine spezifischen Erweiterungen angegeben.

## 4 Signaturanwendung

### 4.1 Signaturerzeugung

Die Erzeugung einer digitalen Signatur umfasst drei Berechnungs-Schritte:

1. Hashing

Das zu signierende Dokument wird durch eine kryptographische *Hashfunktion* auf einen *Hashwert* fester Länge gebracht.

2. Padding

Der Bitstring mit dem Hashwert wird in geeigneter Weise auf die für das Signaturverfahren und den Signaturschlüssel notwendige Länge aufgefüllt.

3. Signatur

Der aufgefüllte Bitstring wird je nach *Signaturalgorithmus* (vgl. *Abschnitt 3.1.3*) mit dem *privaten Signaturschlüssel* zu einer Signatur verknüpft.

Nur im dritten Schritt werden geheime Informationen verarbeitet:

- Der private Schlüssel und ggf.
- auch geheime Zufallszahlen, die in die Signatur mit eingehen.

Diese Berechnungen sollten daher in einer Umgebung erfolgen, die gegen Abhören durch Dritte gesichert ist. Idealerweise wird der *private Signaturschlüssel* ausschließlich in einer speziellen Hardware, der *Signaturerstellungseinheit*, gespeichert und angewendet, die ein Auslesen wirksam verhindert. In der Praxis werden dafür meist *Chipkarten* mit integriertem Mikroprozessor (*Smart Cards*) oder USB-Token eingesetzt. Bei Anwendungen, die eine hohe Performance erfordern, kommen auch spezialisierte *Hardware Security Module* zum Einsatz. Moderne *Chipkarten* und USB-Token können auch zufällige Signaturschlüssel-Paare generieren, so dass der *private Schlüssel* niemals das Gerät verlässt.

In den ersten beiden Schritten müssen dagegen keine geheimen Informationen geschützt werden. In der Praxis erfolgt die Berechnung des *Hashwertes* daher auch meist außerhalb der *Signaturerstellungseinheit*, so dass dieser nur der kurze Hashwert und nicht eine große Nachricht übergeben werden muss. Damit auch wirklich die korrekten Daten signiert werden, muss der gesamte Prozess der Signaturerstellung vor Manipulationen (z.B. durch Viren oder Trojaner) sicher sein. Dies betrifft nicht nur die Berechnungen zur Erzeugung der digitalen Signatur, sondern auch die Übergabe der zu signierenden Daten und Zwischenergebnisse (z.B. dem Hashwert) zwischen den beteiligten Komponenten. In Fällen, in denen eine elektronische Signatur als Willenserklärung einer Person aufgefasst werden soll, sollte diese die zu signierenden Daten zuvor angezeigt bekommen. Insbesondere bei *qualifizierten elektronischen Signaturen*, die vom Gesetzgeber der eigenhändigen Unterschrift in den meisten Fällen gleichgestellt worden sind, muss der Ersteller der Signatur sicher sein können, dass er nur das signiert, was er sieht. Dateiformate, die versteckte Informationen (z.B. Kommentare, Meta-Daten, Text mit weißer Schriftfarbe, etc.) enthalten können, eröffnen Betrügern Tür und Tor und sind daher eher ungeeignet. Wichtig ist aber auch, dass die *Signaturanwendungskomponente* – die zur Signierung verwendete Software oder Hardware – zuverlässig (d. h. ohne schwerwiegende Fehler) und vertrauenswürdig (d. h. ohne böswillige, versteckte Funktionen) ist. Die gesetzlichen Anforderungen an Signaturanwendungskomponenten sind in *Abschnitt 2.1.4* skizziert.

Damit dem Signaturschlüssel-Inhaber keine Nachteile entstehen, sollte er dafür Sorge tragen, dass er

- seine Signaturerstellungseinheit und die dazugehörige PIN sicher verwahrt,
- Dokumente nur nach Kenntnisnahme und Prüfung signiert,
- seine Signaturen nur mit vertrauenswürdigen Signaturanwendungskomponenten erstellt und
- bei Kompromittierung seiner Signaturerstellungseinheit sein Zertifikat umgehend sperren lässt.

In Anwendungen, in denen *qualifizierte elektronische Signaturen* in automatisierter Art und Weise erstellt werden (vgl. Abschnitt 4.4), existieren besonders hohe Sicherheitsanforderungen. Insbesondere muss sichergestellt sein, dass dem Signaturserver nicht unberechtigt Dokumente zur Signierung untergeschoben werden können.

## 4.2 Signaturprüfung

Die Verifikation einer digitalen Signatur umfasst in der Praxis nicht nur die Prüfung ihrer mathematischen Korrektheit, sondern auch die Prüfung der Gültigkeit und Anwendbarkeit des zugeordneten Zertifikates. Dazu muss ein Zertifizierungspfad zu einer – aus Sicht des Verifizierenden – vertrauenswürdigen Zertifizierungsinstanz gebildet und geprüft werden. Bei den meisten Signaturanwendungskomponenten und Web-Browsern sind bereits Zertifikate vertrauenswürdiger Zertifizierungsinstanzen (Trusted CAs) integriert und voreingestellt. Der Anwender kann jedoch in der Regel weitere Zertifikate von Zertifizierungsinstanzen, die er für vertrauenswürdig hält, importieren. Wichtig ist dabei, dass er die *Authentizität* der Zertifikate in einer geeigneten Weise prüft. Die Bundesnetzagentur veröffentlicht beispielsweise zu diesem Zweck die *Hashwerte* ihrer aktuellen Zertifikate im Bundesanzeiger.

Wie in Abbildung 27 angedeutet, besteht der *Zertifizierungspfad* aus einer Kette von Zertifikaten  $Z_1 - Z_2 - \dots - Z_n$ , wobei

- $Z_1$  das Zertifikat des Signaturschlüsselhabers (zu dem öffentlichen Signaturschlüssel, mit dem sich die zu verifizierende Signatur prüfen lässt) ist,
- $Z_n$  das Zertifikat der vertrauenswürdigen Zertifizierungsinstanz, die oft als „Wurzel-Zertifizierungsinstanz“ oder „Root-CA“ bezeichnet wird, und
- für alle  $i$  von 1 bis  $n - 1$  der Eigentümer von  $Z_{i+1}$  das Zertifikat  $Z_i$  ausgestellt hat, so dass sich die Signatur von  $Z_i$  mit dem in  $Z_{i+1}$  zertifizierten öffentlichen Schlüssel prüfen lässt.

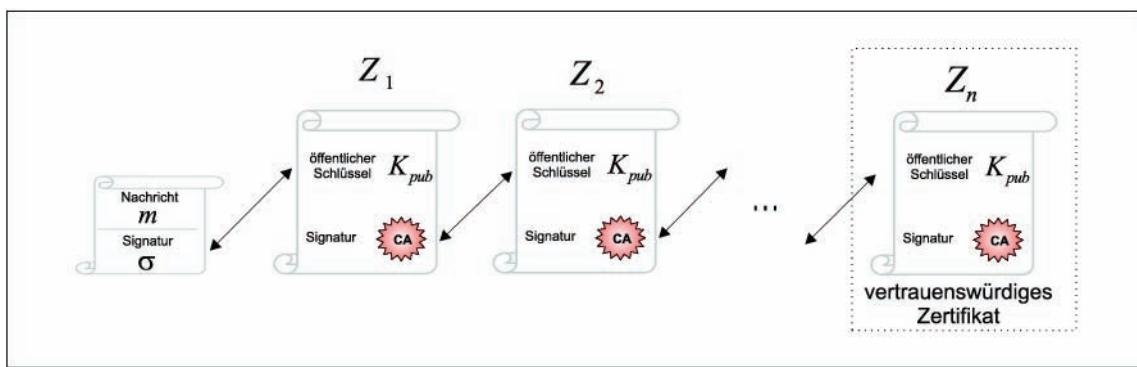


Abbildung 27: Zertifikatspfad

Zu einer Signatur und einer Liste von Zertifikaten vertrauenswürdiger Zertifizierungsinstanzen kann es mehrere – und sogar unterschiedlich lange – *Zertifizierungspfade* geben. In der Praxis bestehen die Zertifizierungspfade jedoch meist aus nur drei Zertifikaten und sind

offensichtlich. Komplizierter wird es erst, wenn *Cross-Zertifikate* hinzukommen. Die Konstruktion eines *Zertifizierungspfades* erfolgt in der Regel automatisch durch die *Signaturanwendungskomponente*.

Die Prüfung erfolgt nun entlang des *Zertifizierungspfades*. Dabei werden mindestens die folgenden Punkte geprüft:

- Mathematische Gültigkeit der Signaturen,
- Gültigkeit der Zertifikate gemäß Gültigkeitsmodell,
- Korrektheit des Verwendungszwecks der Zertifikate.

Je nach Anwendungsfall kann die Verifikation der Signatur noch weitere Aspekte umfassen, z.B. ob das Zertifikat  $Z_1$  des Erstellers der Signatur ein qualifiziertes ist, oder ob die Zertifikate unter einer bestimmten Zertifizierungspolitik (*Certificate Policy*) ausgestellt wurden. Für diese beiden Beispiele sind die QCStatements-Erweiterung und die Erweiterung CertificatePolicies standardisiert (vgl. Abschnitt 3.2.5). Eine umfassende Profilierung der relevantesten Standards findet sich in [ISIS-MTT, ISIS-MTT-SigG].

### 4.2.1 Mathematische Gültigkeit der Signaturen

Die Signatur über  $Z_i$  muss sich mit dem in  $Z_{i+1}$  enthaltenen *öffentlichen Schlüssel* prüfen lassen. Dieser Schritt entfällt natürlich für das vertrauenswürdige Zertifikat  $Z_n$ . Sofern es sich um das Zertifikat einer *Wurzel-Zertifizierungsinstantz* (Root-CA) handelt, ist  $Z_n$  selbstsigniert, d.h. es könnte mit dem in ihm enthaltenen öffentlichen Schlüssel geprüft werden. Dies ist jedoch überflüssig, da das Zertifikat als vertrauenswürdig angesehen wird. Meist wird der Fingerprint des Root-CA-Zertifikates, d.h. der Hashwert des Zertifikates, das insbesondere den öffentlichen Schlüssel enthält, veröffentlicht. Dadurch kann die *Integrität* des Schlüssels geprüft werden.

### 4.2.2 Gültigkeit der Zertifikate gemäß Gültigkeitsmodell

Die Zertifikate müssen zum entscheidenden Zeitpunkt gültig (d.h. nicht gesperrt oder abgelaufen) gewesen sein. Welcher Zeitpunkt hierbei der entscheidende ist, hängt vom angewandten *Gültigkeitsmodell* (*Schalenmodell*, *Hybridmodell* und *Kettenmodell*) ab (vgl. [BNetzA-GMod]). Die Prüfung der Gültigkeit der Zertifikate umfasst die Prüfung des Gültigkeitszeitraums und die Auswertung der Statusinformationen – jeweils bezüglich der durch das *Gültigkeitsmodell* vorgegebenen Zeitpunkte.

Die Statusinformationen können aus *Sperrlisten* (vgl. Abbildung 20) oder von *OCSP-Respondern* (vgl. Abschnitt 3.2.6) bezogen werden. Beim *Schalenmodell* und bei der Nutzung von *Sperrlisten* ist es wichtig, dass der Verifizierende eine zuverlässige Systemzeit besitzt, damit ihm keine abgelaufenen Zertifikate oder Sperrlisten unbemerkt untergeschoben werden können.

Da *Sperrlisten* und *OCSP*-Antworten signiert sind, beinhaltet ihre Prüfung wiederum die Verifikation einer Signatur – hier beginnt also eine Rekursion. Um den Aufwand dafür zu minimieren, können *Signaturanwendungskomponenten* die Signatur der Sperrliste nur nach dem Herunterladen prüfen oder die Statusinformationen für alle Zertifikate des *Zertifizierungspfades* in einem einzigen OCSP-Request abfragen.

#### 4.2.2.1 Schalenmodell

Beim Schalenmodell fordert man, dass alle Zertifikate des Zertifizierungspfades zum

Zeitpunkt der Prüfung gültig sind, d.h. der entscheidende Zeitpunkt ist „jetzt“. Das Schalenmodell hat zur Folge, dass Signaturen, die zum Zeitpunkt ihrer Erstellung gültig waren, durch die Sperrung oder den Ablauf eines der Zertifikate im Zertifizierungspfad ungültig werden können. Das Schalenmodell ist daher sinnvoll, wenn man nicht prüfen kann, wann die Signatur erstellt wurde. Es hat allerdings auch die Konsequenz, dass die Gültigkeit von Zertifikaten nicht über die der Zertifizierungsinstanz hinausgehen darf. Aus diesem Grund müssen in einer *PKI*, die das Schalenmodell nutzt, die Zertifikate der CA erheblich länger gültig sein als die der Endbenutzer, und diese verlieren bei einer Sperrung des entsprechenden CA-Zertifikates ebenfalls ihre Gültigkeit. Das Schalenmodell wird beim X.509-Standard, der insbesondere für Zwecke der *Authentifizierung* geschaffen wurde, eingesetzt. Da es bei der Prüfung von *qualifizierten elektronischen Signaturen* aber nicht auf den Zeitpunkt der Prüfung, sondern auf den Zeitpunkt der Erstellung der Signatur ankommt (vgl. § 2 Nr. 3 [SigG]), ist das Schalenmodell für die Prüfung von qualifizierten elektronischen Signaturen nicht ohne weiteres einsetzbar.

### 4.2.2.2 Hybridmodell

Beim Hybridmodell handelt es sich um eine Mischform aus Ketten- und Schalenmodell. Hier fordert man, dass alle Zertifikate des Zertifizierungspfades bei der Erstellung der zu verifizierenden Signatur gültig waren. Dieses Gültigkeitsmodell setzt also voraus, dass dieser Zeitpunkt nachprüfbar ist, z.B. anhand von Zeitstempeln. Gültige Signaturen können in diesem Modell also nicht mehr ungültig werden. Wie im Schalenmodell nützt es nichts, wenn die Zertifikate der Endbenutzer länger gültig sind als die der CA, da zur Gültigkeit der Signatur alle Zertifikate im Zertifizierungspfad gültig sein müssen. Außerdem kann ein Anwender keine gültigen Signaturen mehr erzeugen, nachdem das Zertifikat der Zertifizierungsinstanz gesperrt wurde. Diese Eigenschaft ist sinnvoll, wenn kein *OCSP*-Responder zur Verfügung steht, so dass sich nach der Kompromittierung eines CA-Signaturschlüssels gefälschte Zertifikate nicht mehr von echten unterscheiden lassen.

### 4.2.2.3 Kettenmodell

Beim Kettenmodell fordert man, dass alle Zertifikate zum Zeitpunkt der Anwendung des (dem zertifizierten öffentlichen Schlüssel) zugeordneten privaten Signaturschlüssels gültig waren. Dies bedeutet konkret, dass für  $i \geq 2$  das Zertifikat  $Z_i$  bei der Ausstellung von  $Z_{i-1}$  und  $Z_1$  bei der Erstellung der zu verifizierenden Signatur gültig gewesen sein muss. Dies ist eine wesentlich schwächere Forderung als beim Schalenmodell oder dem Hybridmodell, denn wenn bei der Erstellung der zu verifizierenden Signatur beispielsweise  $Z_2$  abgelaufen oder gesperrt war, wäre die Signatur deswegen nicht ungültig. Um sicherzustellen, dass  $Z_1$  nicht ein gefälschtes Zertifikat ist, das ein Betrüger mit einem außer Betrieb genommenen oder kompromittierten Schlüssel einer CA erstellt hat, benötigt man einen Dienst wie *OCSP*, der einem zu gefälschten Zertifikaten die Antwort unknown (d.h. „nicht von dieser CA“) gibt. Außerdem muss man den Zeitpunkt der Erstellung der zu verifizierenden Signatur – z.B. anhand von *Zeitstempeln* – bestimmen können. Das Signaturgesetz geht vom Kettenmodell aus, indem es in § 19 Abs. 5 [SigG] vorschreibt, dass die von einem Zertifizierungsdienstanbieter ausgestellten Zertifikate von der Einstellung seines Betriebes (und der damit verbundenen Sperrung der CA-Zertifikate) unberührt bleiben.

### 4.2.2.4 Zusammenfassung

Wie in Abbildung 28 dargestellt, führen die verschiedenen *Gültigkeitsmodelle* unter Umständen zu unterschiedlichen Prüfungsergebnissen und verschieden großen „effektiven

Nutzungszeiträumen<sup>29</sup> für Zertifikate.

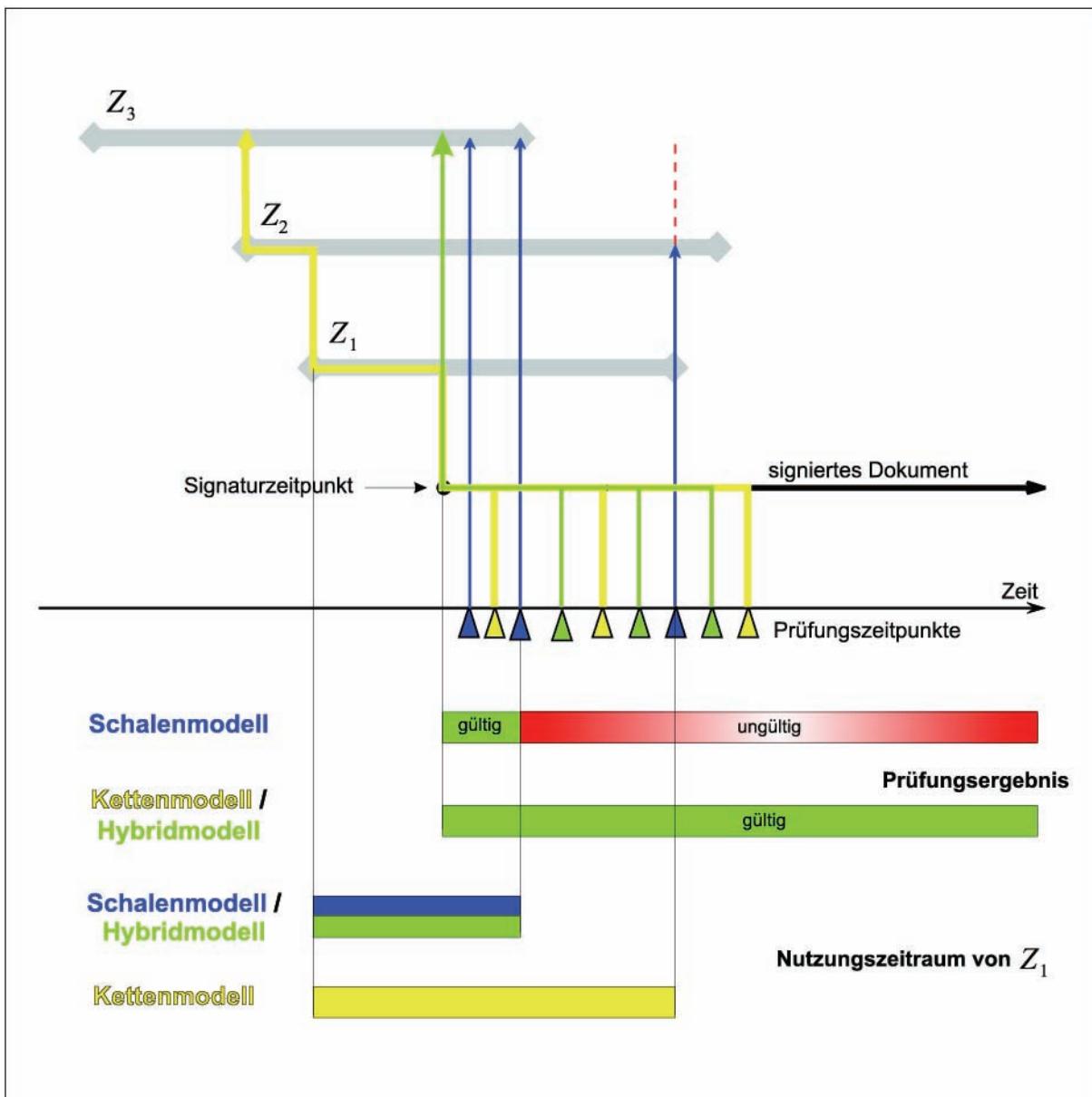


Abbildung 28: Gültigkeitsmodelle im Überblick

Da beim *Kettenmodell* und beim *Hybridmodell* auf den Signaturzeitpunkt abgestellt wird, bleibt die Signatur auch dann noch gültig, wenn das Zertifikat  $Z_1$  bereits abgelaufen ist. Beim *Schalenmodell* hingegen wird geprüft, ob alle Zertifikate zum Prüfungszeitpunkt gültig sind. Deshalb ist die Signatur nach dem Ablauf des Zertifikates  $Z_1$  nicht mehr gültig.

Der effektive Nutzungszeitraum des Zertifikates beim *Schalenmodell* und beim *Hybridmodell* ergibt sich aus dem Zeitraum, in dem alle Zertifikate gültig sind. Beim *Kettenmodell* hingegen ist der effektive Nutzungszeitraum der Gültigkeitszeitraum des Zertifikates selbst.

<sup>29</sup>Der effektive Nutzungszeitraum eines Zertifikates ist der Zeitraum, in dem mit dem Zertifikat gültige Signaturen erstellt werden können.

## 4.2.3 Korrektheit des Verwendungszwecks der Zertifikate

Die Zertifikate müssen für den entsprechenden Verwendungszweck zugelassen sein. Dafür werden in X.509-Zertifikaten die Erweiterungen KeyUsage und ExtendedKeyUsage (vgl. Abschnitt 3.2.5) verwendet, in denen einzelne oder mehrere Verwendungszwecke eingetragen sein können.

### 4.2.3.1 Zertifikat des Anwenders ( $Z_1$ )

Die im Zertifikat  $Z_1$  eingetragenen Verwendungszwecke müssen dem Zweck der zu verifizierenden digitalen Signatur entsprechen.

- Falls die Signatur eine qualifizierte elektronische Signatur im Sinne des Signaturgesetzes ist, sollte nonRepudiation als KeyUsage eingetragen sein.
- Für Signaturen über Sperrlisten ist cRLSign als KeyUsage vorgesehen.
- Für Signaturen eines OCSP-Responders ist id-kp-OCSPSigning als ExtendedKeyUsage vorgesehen.
- Für die im Rahmen einer Authentisierung über SSL oder IPSec erstellten Signaturen sind die ExtendedKeyUsages id-kp-clientAuth und ServerAuth vorgesehen, zusätzlich wird digitalSignature als KeyUsage gesetzt.
- Für Signaturen in Zeitstempeln (siehe Abschnitt 3.2.7) ist die ExtendedKeyUsage id-kp-timeStamping vorgesehen. Zusätzlich wird digitalSignature als KeyUsage gesetzt.

Weitere Einschränkungen oder Legitimationen für bestimmte Verwendungszwecke können in einem dem Zertifikat des Signaturschlüsselinhabers  $Z_1$  zugeordneten Attributzertifikat festgelegt sein. In Fällen, in denen dies relevant sein könnte (z.B. wenn eine Zeichnungsbefugnis „in Vertretung (i.V.)“ verlangt wird), muss daher auch das Attributzertifikat geprüft werden.

### 4.2.3.2 CA-Zertifikate ( $Z_2 \dots Z_n$ )

Die CA-Zertifikate  $Z_2 \dots Z_n$  müssen KeyCertSign als KeyUsage eingetragen haben. Dieser Wert gibt an, dass der zertifizierte öffentliche Signaturschlüssel zu einer Zertifizierungsinstanz gehört und für die Ausstellung (d.h. Signierung) von Zertifikaten verwendet werden darf. Außerdem müssen die CA-Zertifikate  $Z_2 \dots Z_n$  in der Erweiterung BasicConstraints im Feld ca den Wert TRUE und im Feld pathLenConstraint einen ausreichend großen Wert gesetzt haben. Die Erweiterung BasicConstraints wurde eingeführt, damit eine CA die Tiefe der von ihr aufgespannten Zertifizierungshierarchie festlegen kann und pathLenConstraint gibt an, wieviele CA-Zertifikate im Zertifizierungspfad vor dem aktuellen Zertifikat stehen dürfen. Im Zertifikat  $Z_i$  (mit  $i \geq 2$ ) muss also pathLenConstraint mindestens  $i - 1$  sein.

## 4.3 Signaturformate

Bei Signaturformaten kann in grober Art und Weise zwischen grundlegenden *Low-Level-Signaturformaten* und erweiterten *High-Level-Signaturformaten* unterschieden werden. Beim Low-Level-Signaturformat ist der *Signaturalgorithmus* (z.B. RSA, DSA, ECDSA) sowie ein möglicherweise notwendiges *Padding* spezifiziert. Das Format selbst ist maßgeblich vom eingesetzten Kryptoalgorithmus geprägt und deshalb in Abschnitt 3.1.3 für den RSA-Algorithmus näher beschrieben.

*High-Level-Signaturformate* umfassen die rohen Signaturdaten, wie sie durch das Low-Level-Signaturformat vorgegeben sind, sowie weitere Informationen, die bei der Gültigkeitsprüfung

der Signatur herangezogen werden, wie z.B. den Zeitpunkt der Signaturerstellung und die zur Prüfung der Signatur notwendigen *Zertifikate*.

Während anhand einer Signatur im Low-Level-Format nur die mathematische Gültigkeit der Signatur geprüft werden kann, unterstützen die High-Level-Signaturformate in der Regel die komplette Gültigkeitsprüfung von Signaturen und Zertifikaten, sowie beispielsweise auch die Mehrfachsignatur. Im Folgenden werden ausgewählte *High-Level-Signaturformate* näher beschrieben.

### 4.3.1 Cryptographic Message Syntax / PKCS #7

Mit der auf den PKCS #7 - Standard [PKCS7(v1.5)] zurückgehenden *Cryptographic Message Syntax (CMS)* [RFC2630, RFC3369, RFC3852] wurde das in der Praxis vielleicht gebräuchlichste Format für kryptographisch behandelte Nachrichten spezifiziert.

Die Betrachtung des CMS-Standards ist in folgende Abschnitte gegliedert:

- Containertypen für CMS (vgl. Abschnitt 4.3.1.1),
- Die Struktur von SignedData und SignerInfo (vgl. Abschnitt 4.3.1.2),
- Einige SignerInfo-Attribute (vgl. Abschnitt 4.3.1.3),
- Abläufe bei der Erstellung und Prüfung von Signaturen (vgl. Abschnitt 4.3.1.4),
- Enveloping und Detached Signatures mit CMS (vgl. Abschnitt 4.3.1.5).

#### 4.3.1.1 Containertypen für CMS

Die CMS-Spezifikation sieht die Ablage von Nachrichten in ContentInfo-Containern vor. Diese Container haben jeweils einen bestimmten Typ, wobei in [RFC3852] folgende Containertypen definiert sind:

- Data  
bezeichnet beliebige Daten.
- SignedData  
wird für elektronische Signaturen verwendet und in *Abschnitt 4.3.1.2* näher erläutert.
- EnvelopedData  
enthält mit einem Nachrichtenschlüssel verschlüsselte Daten und die Verschlüsselung des verwendeten Nachrichtenschlüssels für den oder die Empfänger.
- DigestedData  
enthält Daten, die zum Schutz der *Integrität* um einen *Hashwert* der Daten ergänzt wurden.
- EncryptedData  
enthält verschlüsselte Daten. Anders als beim EnvelopedData-Typ wird hier allerdings der zur Verschlüsselung verwendete Nachrichtenschlüssel nicht zusätzlich für die Empfänger verschlüsselt. Das Schlüsselmanagement muss also auf einem anderen Weg geschehen.
- AuthenticatedData  
enthält Daten, die mit einem *Message Authentication Code (MAC)* gesichert sind, sowie die Verschlüsselung des zur MAC-Erzeugung verwendeten symmetrischen Schlüssels für einen oder mehrere Empfänger.
- Weitere Containertypen  
können auch außerhalb der CMS-Spezifikation [RFC3852] definiert werden.  
Insbesondere sind hier folgende zu nennen:
  - SignedAndEnvelopedData  
wurde in [PKCS7(v1.5), RFC2315] definiert und erlaubt es, Inhaltsdaten zu

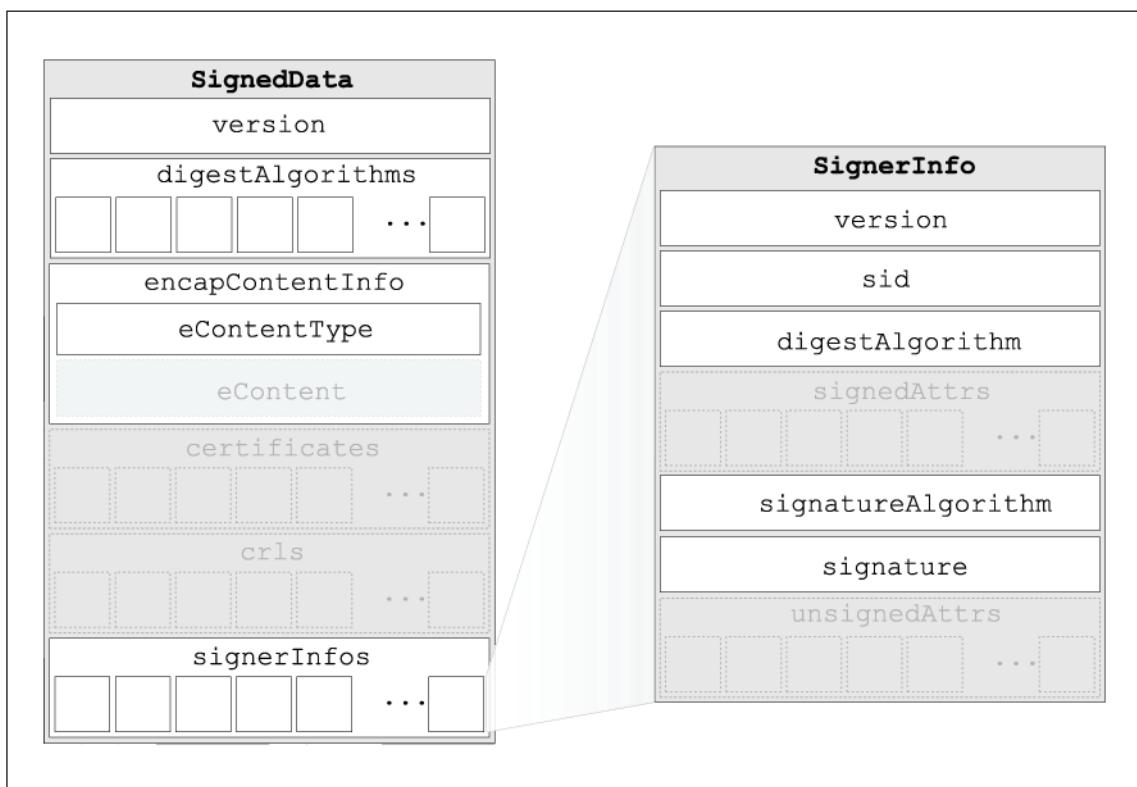
signieren und direkt danach zu verschlüsseln. Da man ein ähnliches Ergebnis auch durch die Schachtelung von SignedData- und EnvelopedData-Containern erreichen kann, wurde dieses Format nicht in die CMS-Spezifikationen der IETF [RFC2630, RFC3369, RFC3852] aufgenommen.

- CompressedData  
wurde in [RFC3274] definiert und erlaubt die Komprimierung von Daten mit dem ZLIB-Verfahren [RFC1950, RFC1951].

### 4.3.1.2 Die Struktur von SignedData und SignerInfo

Das Herzstück des CMS-Signaturformats ist durch die ASN.1-Strukturen SignedData und SignerInfo gegeben.

Der SignedData-Container enthält im Wesentlichen die zu signierenden Daten und eine oder mehrere Signaturen in Form der SignerInfo-Struktur.



**Abbildung 29: Struktur des SignedData-Containers aus CMS / PKCS #7**

Wie in Abbildung 29 angedeutet, besteht der SignedData-Container aus folgenden Teilen:

- version  
ist ein ganzzahliger Wert, durch den die Kompatibilität zwischen den verschiedenen Versionen der CMS-Spezifikationen [PKCS7(v1.5), RFC2630, RFC3369, RFC3852] sichergestellt wird.
- digestAlgorithms  
listet die Hashalgorithmen auf, die später in den signerInfos wieder vorkommen. Durch die Angabe der verwendeten Hashalgorithmen vor dem zu hashenden Inhalt kann eine Verarbeitung in einem Durchlauf erreicht werden.

- `encapContentInfo`  
besteht aus den Feldern `eContentType`, das den Typ der Daten angibt, und dem eigentlichen Inhalt (`eContent`). Als Datentypen kommen beispielsweise die in *Abschnitt 4.3.1.1* definierten CMS-Containertypen in Frage. Da das `eContent`-Feld fehlen darf, können auch so genannte „Detached Signatures“ erzeugt werden, die sich auf extern abgelegte Inhalte beziehen (vgl. *Abschnitt 4.3.1.5*).
- `certificates`  
ist ein optionales Feld, das eine Menge von *Public-Key-* und *Attributzertifikaten* enthalten kann, die zur Signaturprüfung herangezogen werden können.
- `crls`  
ist ein optionales Feld, das Informationen zur Prüfung des Zertifikatsstatus enthalten kann. Gemäß der jüngsten CMS-Spezifikation [RFC3852] können hier neben *Sperrlisten* auch andere Sperrinformationen, beispielsweise vom Typ `OCSPResponse`, enthalten sein.
- `signerinfos`  
enthält eine Menge von Signaturen, die aus folgenden Informationen zusammengesetzt sind:
  - `version`  
ist ein ganzzahliger Wert, durch den die verschiedenen Versionen der `SignerInfo`-Struktur unterschieden werden können. Unterschiede existieren hier lediglich beim `sid` vom Typ `SignerIdentifier`.
  - `sid`  
ermöglicht die Identifikation des (Zertifikates des) Signierenden.
  - `digestAlgorithm`  
gibt an, welcher Hashalgorithmus zur Erzeugung der Signatur verwendet wurde.
  - `signedAttrs`  
kann eine Menge von Attributen enthalten, die in die Signatur einbezogen werden.
  - `signatureAlgorithm`  
enthält den Signaturalgorithmus. Beispielsweise kann hier der in [RFC2313] spezifizierte und in *Abschnitt 3.1.3.1* näher erläuterte Algorithmus `rsaEncryption` eingesetzt werden.
  - `signature`  
enthält die tatsächliche Signatur im angegebenen *Low-Level-Signaturformat*.
  - `unsignedAttrs`  
kann eine Menge von Attributen enthalten, die jedoch – anders als bei `signedAttrs` oben – nicht in die Signatur einbezogen werden.

### Bemerkung 1 (Mehrfachsignaturen)

Durch die Präsenz mehrerer `SignerInfo`-Felder in einem `SignedData`-Container können parallele Mehrfachsignaturen realisiert werden.

Da die zu signierenden Nutzdaten wiederum aus einem `ContentInfo`-Container vom Typ `SignedData` bestehen können, ist es möglich, sequentielle Mehrfachsignaturen zu konstruieren.

Soll nicht die gesamte `SignedData`-Struktur – und damit möglicherweise mehrere parallele Signaturen – erneut signiert werden, sondern lediglich eine einzelne Signatur

„gegenzeichnet“ werden, so kann dies unter Verwendung des *countersignature-Attributes* geschehen.

### 4.3.1.3 Einige SignerInfo-Attribute

Wie oben erläutert, kann die SignerInfo-Struktur im Feld `signedAttrs` signierte und im Feld `unsignedAttrs` unsignierte Attribute enthalten. Im Folgenden sollen einige nützliche Attribute besprochen werden:

- **ContentType**

Das `ContentType`-Attribut aus [RFC3447, PKCS9] wird bei der Signaturerzeugung als signiertes Attribut verwendet, um den Typ der signierten Daten im `eContent` anzugeben. Sofern überhaupt signierte Attribute vorhanden sind, ist die Präsenz der `ContentType`- und `MessageDigest`-Attribute obligatorisch.

- **MessageDigest**

Das `MessageDigest`-Attribut aus [RFC3447, PKCS9] wird bei der Signaturerzeugung als signiertes Attribut verwendet, um den Hashwert des `eContent` anzugeben.

- **SigningTime**

Das `SigningTime`-Attribut aus [RFC3447, PKCS9] wird als signiertes Attribut verwendet, um den Zeitpunkt der Signaturerzeugung zu dokumentieren.

- **Countersignature**

Mit dem `countersignature`-Attribut aus [RFC3447, PKCS9], das als unsigniertes Attribut eingesetzt wird, kann eine existierende Signatur mit einer oder mehreren Signaturen gegenzeichnet werden.

- **SigningCertificate**

Da die im `SignedData`-Container enthaltenen Zertifikate bei der unten näher erläuterten Erzeugung der Signatur nicht mit signiert werden, wäre es möglich, die Zertifikate zu einem späteren Zeitpunkt auszutauschen. Um einen solchen Angriff zu verhindern, kann das in [RFC2634] definierte `SigningCertificate`-Attribut verwendet werden. Dieses Attribut wird mit signiert und enthält eine Liste von Zertifikaten, die über ihre Hashwerte<sup>30</sup> und möglicherweise zusätzlich über ein `IssuerSerial`-Feld identifiziert werden.

- **SignerAttributes**

Mit dem `SignerAttributes`-Attribut aus [ETSI-101733] kann der Unterzeichner der Signatur weitere Attribute zuweisen. Neben lediglich behaupteten Attributen können auch von einer vertrauenswürdigen dritten Partei bestätigte Attribute – in Form von Attributzertifikaten – eingefügt werden.

- **ContentTimestamp**

Mit dem `ContentTimestamp`-Attribut aus [ETSI-101733] kann ein existierender Zeitstempel gemäß [RFC3161] als signiertes Attribut in die Signatur eingefügt werden. Dadurch kann der Nachweis erbracht werden, dass die Nutzdaten bereits vor dem im Zeitstempel angegebenen Zeitpunkt existiert haben.

Weitere Attribute für CMS-Signaturen sind beispielsweise in [ETSI-101733] definiert.

---

<sup>30</sup>Bei dem in [RFC2634] spezifizierten Attribut ist die Verwendung des SHA-1 Hashalgorithmus zwingend vorgeschrieben. Deshalb wurde in [RFC3126, ETSI-101733] mit `OtherSigningCertificate` ein ähnliches Attribut definiert, bei dem auch andere Hashfunktionen eingesetzt werden können.

#### 4.3.1.4 Abläufe bei der Erzeugung und Verifikation von Signaturen

Wie in *Abschnitt 4.1* erläutert, erfolgt die Erzeugung der Signatur durch die Erzeugung des Hashwertes, ggf. das Padding des Hashwertes und schließlich die Berechnung der Signatur. Während die beiden letzten Schritte vom verwendeten kryptographischen Signaturalgorithmus abhängen, werden die zur Erzeugung des Hashwertes zu verwendenden Daten von der CMS-Spezifikation festgelegt.

Sind keine signierten Attribute vorhanden, so wird einfach der Hashwert des Inhalts<sup>31</sup> des eContent-Felds gehasht.

Sind jedoch signierte Attribute vorhanden, so wird das gesamte signedAttributes-Feld – die komplette ASN.1/DER-Codierung – für die Berechnung des Hashwertes herangezogen. Da in diesem Fall die beiden Attribute ContentType und MessageDigest vorhanden sein müssen (vgl. *Abschnitt 4.3.1.3*), fließt der Hashwert des eContent indirekt in die Berechnung des Hashwertes ein.

Bei der Prüfung der Signatur verfährt man analog, wobei – wie in *Abschnitt 4.2* erläutert – außerdem die Prüfung des Zertifikatspfades zu erfolgen hat.

#### 4.3.1.5 Enveloping und Detached Signatures mit CMS

Wie in *Abbildung 30* angedeutet, ermöglicht das CMS-Signaturformat zwei grundsätzliche Arten der Signatur:

- Enveloping Signature

Ist die Nachricht ein Teil der Signatur, so spricht man von einer „Enveloping Signature“. In diesem Fall wird die Nachricht gemäß ASN.1 codiert und in das eContent-Feld der SignedData-Struktur eingefügt.

- Detached Signature

Andernfalls fehlt das eContent-Feld und die Nachricht wird in einer separaten Datei abgelegt. In diesem Fall spricht man von einer „Detached Signature“.

---

<sup>31</sup>Der ASN.1-Overhead (TAG und LENGTH) fließt nicht in die Berechnung des Hashwertes ein.

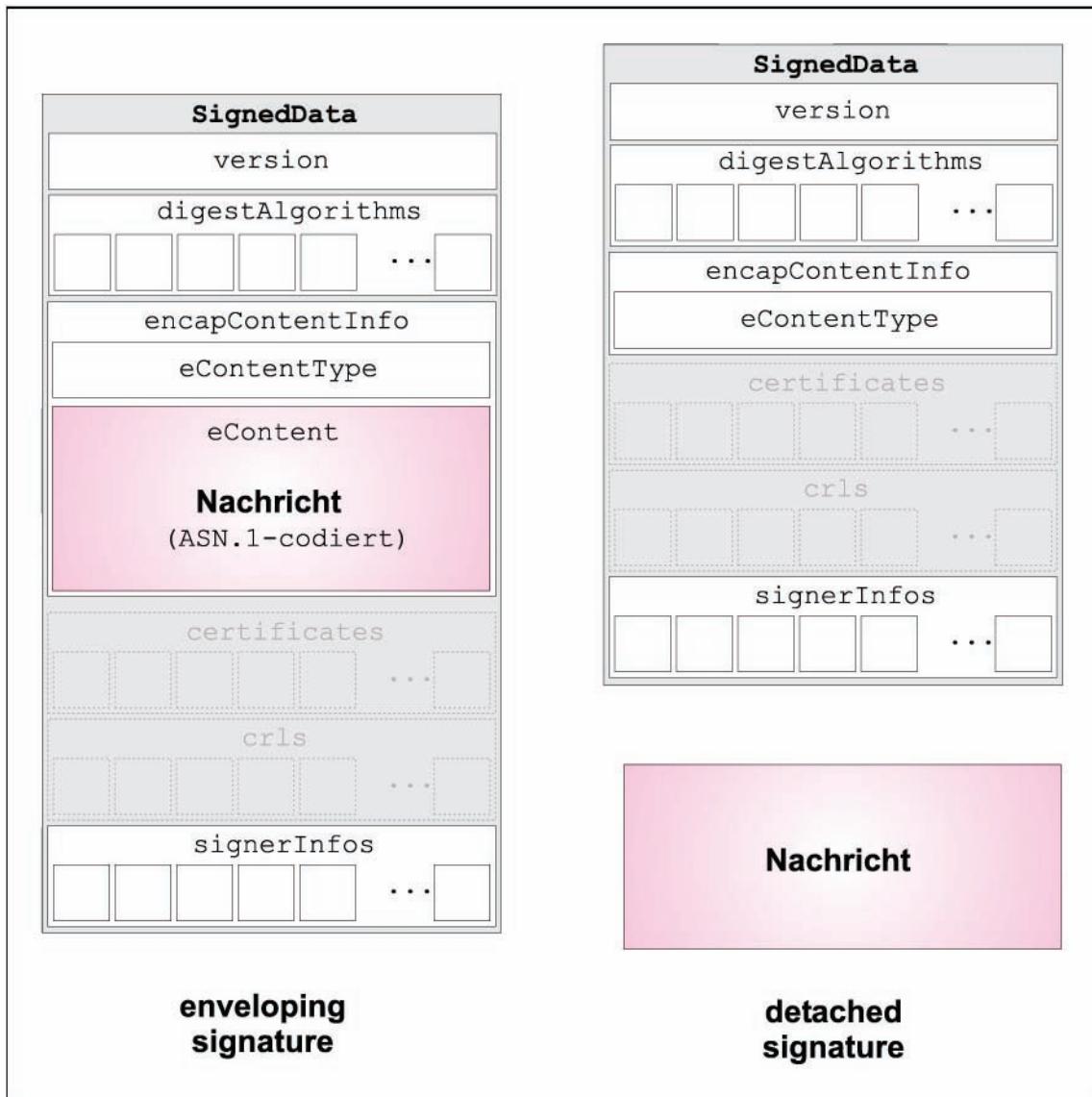


Abbildung 30: Enveloping und Detached Signatures mit CMS

Während im ersten Fall die Nachricht und die Signatur in einer einzigen (ASN.1-codierten) Datei vorliegen und deshalb keine Probleme mit dem Zusammenfügen auftreten können, so ist die Nachricht hier allerdings nur dann lesbar, wenn ein CMS-fähiges Anwendungsprogramm existiert. Da dies in vielen Fällen problematisch ist, verwendet man in der Praxis häufig die zweite Variante, bei der die Nachricht und die Signatur in separaten Dateien vorliegen. Um die automatisierte Zuordnung einer Nachrichten- und Signaturdatei zu erleichtern, wählt man für die Signaturdatei in der Regel den gleichen Dateinamen und hängt lediglich die zusätzliche Dateiendung .p7s an – standardisiert ist diese Vorgehensweise zur Zuordnung zwischen Signatur- und Nachrichtendatei aber leider nicht.

### 4.3.2 S/MIME

Secure Multipurpose Internet Mail Extensions (S/MIME) ist ein ursprünglich von den Laboratorien der RSA Security Inc. entwickeltes, nunmehr in der IETF in [RFC2632, RFC2633] standardisiertes, auf PKCS #7/CMS aufbauendes Format für die *Verschlüsselung*

und *Signatur* von E-Mails und E-Mail-Anhängen im *MIME*-Format [RFC1521].

Für die Signatur von E-Mails sieht der S/MIME-Standard [RFC2633] zwei verschiedene Varianten vor:

- application/pkcs7-mime mit SignedData,
- multipart/signed.

Da die signierte Mail bei der multipart/signed-Variante auch mit E-Mail-Clients gelesen werden kann, die S/MIME nicht unterstützen, sollte sie vom Sender generell bevorzugt werden.

### 4.3.2.1 application/pkcs7-mime mit SignedData

In diesem Fall besteht die übertragene Nachricht aus einer *CMS*-Struktur vom Typ SignedData, die in einer *MIME*-Nachricht vom Typ application/pkcs7-mime transportiert wird.

Hierfür wird in einem ersten Schritt die Nachricht in eine kanonische Form gebracht, indem bei einer Text-Nachricht beispielsweise sichergestellt wird, dass eine Zeile mit den Zeichen (CR)/(LF) (Carriage Return und Line Feed) endet. Diese Daten sollten<sup>32</sup> in eine Transport-Codierung, wie base64 oder quoted-printable, überführt werden, so dass eine fehlerfreie Übertragung der E-Mail auch dann gewährleistet ist, wenn eine Mail-Relay-Station auf dem Weg zwischen Sender und Empfänger nur in der Lage ist, 7-Bit-Daten zu verarbeiten. Aus diesen Daten, die den eContent der *CMS*-Struktur vom Typ Data bilden, wird eine CMS-Signatur gebildet (vgl. *Abschnitt 4.3.1*). Diese CMS-Struktur wird schließlich in einen MIME-Typ vom Typ application/pkcs7-mime mit signed-data im optionalen „smime-type“-Parameter eingebettet.

### 4.3.2.2 multipart/signed

Bei der multipart/signed-Variante, die auf [RFC1847] zurück geht, besteht die S/MIME-Nachricht aus zwei Teilen, die als *MIME*-Nachrichten codiert sind. Im ersten Teil befindet sich die zu signierende Nachricht, im zweiten Teil befindet sich die zugehörige Signatur im *CMS*-Format, in der der eContent leer ist. Auch hier wird die zu signierende *MIME*-Nachricht in eine kanonische Form gebracht und die beiden *MIME*-Nachrichten in eine entsprechende Transport-Codierung überführt. Der Vorteil dieser Variante ist, dass die signierte Nachricht auch mit nicht-S/MIME-fähigen Mail-Programmen gelesen werden können.

## 4.3.3 XML Digital Signature

Für die *digitale Signatur* von Daten im *XML*-Format wurde von einer Arbeitsgruppe des W3C ein spezifisches Signaturformat entwickelt [XML-DSig, RFC3275]. Im Vergleich zum in *Abschnitt 4.3.1* erläuterten *Cryptographic Message Syntax*-Signaturformat bietet die *XML*-Signatur ein höheres Maß an Flexibilität, das notwendig ist, um das volle Potenzial von *XML* auch im Bereich der digitalen Signatur ausnutzen zu können.

---

<sup>32</sup>Auch wenn die Transport-Codierung nur bei der multipart/signed-Nachricht zwingend notwendig ist, wird in [RFC2633, Section 3.1.2] empfohlen, sie bei allen S/MIME-Varianten zu verwenden, da so eine fehlerfreie Übertragung auch dann sichergestellt ist, wenn bei der internen Verarbeitung der E-Mail beim Empfänger Systeme eingesetzt werden, die nur in der Lage sind 7-Bit-Daten zu verarbeiten.

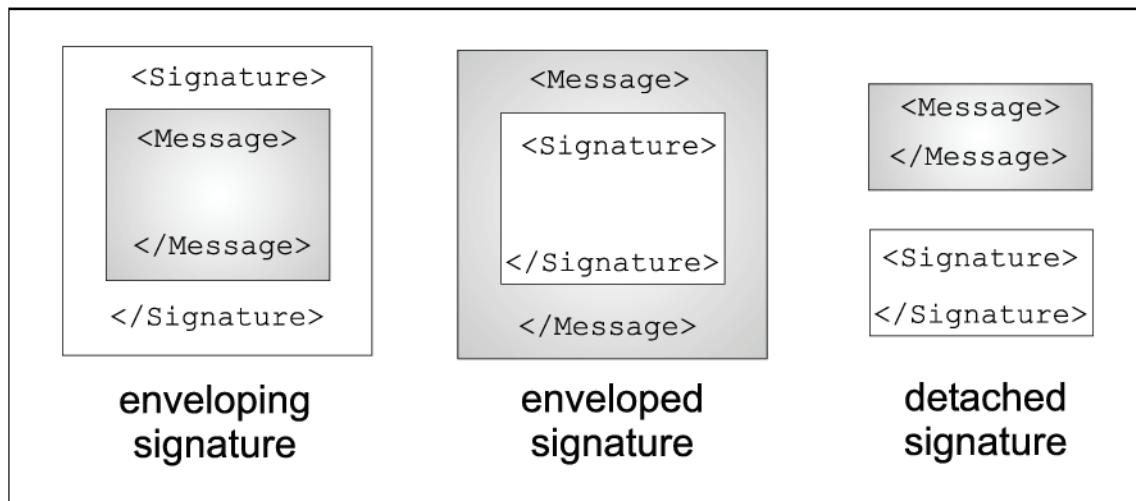


Abbildung 31: XML-Signatur-Typen

Während das *CMS*-Format nur die Erstellung von umschließenden Signaturen (enveloping signature) und von der Nachricht getrennten Signaturen (detached signature) unterstützt, kann bei der XML-Signatur gemäß [XML-DSig, RFC3275] auch die Signatur in die zu signierende Nachricht eingebettet sein (enveloped signature) (vgl. Abbildung 31). Ein ähnliches Konstrukt ist für *CMS*-basierte Signaturen nur in Verbindung mit bestimmten Dokumenten-Typen, wie beispielsweise *PDF* (vgl. Abschnitt 4.3.6), in standardisierter Art und Weise möglich.

Bei der XML-Signatur kann man komplett Dateien eines beliebigen Typs oder ganz gezielt bestimmte Teile eines XML-Dokumentes in die Signatur einbeziehen oder die zu signierenden Daten vor der Signaturerstellung in einer bestimmten Art und Weise transformieren. Hierbei können beispielsweise XPath- oder XSL-Transformationen durchgeführt werden. Mit der XPath-Transformation [XPath] ist es möglich, bestimmte Bestandteile eines XML-Dokumentes bei der Erzeugung der Signatur auszusparen, so dass sich diese Datenfelder später ändern können, ohne dass die Signatur ungültig werden würde. Dies kann<sup>33</sup> beispielsweise für die Erzeugung von eingebetteten Signaturen (enveloped signature) verwendet werden. Mit einer XSL-Transformation [XSLT] können die Daten im XML-Format vor der Erzeugung und Prüfung einer Signatur mit einem bestimmten Layout [XSL] verknüpft werden.

---

<sup>33</sup>Für den Fall der eingebetteten Signatur existiert außerdem eine spezielle Transformation (vgl. [RFC3275]).

```

<Signature ID>
    <SignedInfo>
        <CanonicalizationMethod/>
        <SignatureMethod/>
        <Reference URI>
            <Transforms>
                <DigestMethod>
                <DigestValue>
            </Reference>
    </SignedInfo>
    <SignatureValue>
    <KeyInfo>
    <Object ID>
</Signature>

```

**Abbildung 32: Struktur einer XML-Signatur**

Eine XML-Signatur beginnt, wie in *Abbildung 32* dargestellt, mit dem `<Signature>`-Tag, endet mit dem `</Signature>`-Tag und enthält im Wesentlichen folgende Bestandteile:

- `SignedInfo`
  - Enthält Informationen, wie welche Daten zu signieren sind. Insbesondere enthält es folgende Bestandteile:
    - `CanonicalizationMethod`  
gibt an, welche Kanonisierungsmethode eingesetzt wird.
    - `SignatureMethod`  
spezifiziert den Signaturalgorithmus.
    - `Reference`  
ist ein Feld, das ein- oder mehrmals vorhanden sein kann und einen Verweis auf die zu signierenden Daten, die mittels eines Uniform Resource Identifier (URI) adressiert werden, und Informationen zur Aufbereitung derselben enthält. Es enthält insbesondere folgende Elemente:
      - `Transforms`  
gibt möglicherweise an, wie die Daten vor der Anwendung der *Hashfunktion* aufzubereiten sind.
      - `DigestMethod`  
spezifiziert die zu verwendende *Hashfunktion*.
      - `DigestValue`  
enthält den *Hashwert*, der mit der angegebenen *Hashfunktion* aus den referenzierten und möglicherweise transformierten Daten berechnet wurde.
- `SignatureValue`  
enthält die Signatur
- `KeyInfo`  
ist ein optionales Feld, in dem beispielsweise *Zertifikate* abgelegt sein können.
- `Object`  
kann beliebig oft (auch keinmal) auftreten und ein beliebiges Datenobjekt beinhalten.

Beispielsweise kann hier eine zu signierende Nachricht, zusätzliche Eigenschaften der Signatur (`SignatureProperties`), eine weitere Signatur oder ein so genanntes Manifest in die Signatur einbezogen werden. Ein Manifest enthält eine Liste von Referenzen auf bestimmte Daten und den zugehörigen Hashwert derselben.

Anders als beim CMS-Signaturformat werden die Daten nicht gemäß ASN.1 codiert, sondern liegen, wie in Abbildung 32 angedeutet, in „lesbaren“ XML-Strukturen vor.

#### 4.3.4 EDIFACT

EDIFACT ist ein branchenübergreifender internationaler Standard für den automatisierten Austausch elektronischer Daten im Geschäftsverkehr, dessen Syntax in ISO9735 festgelegt ist.

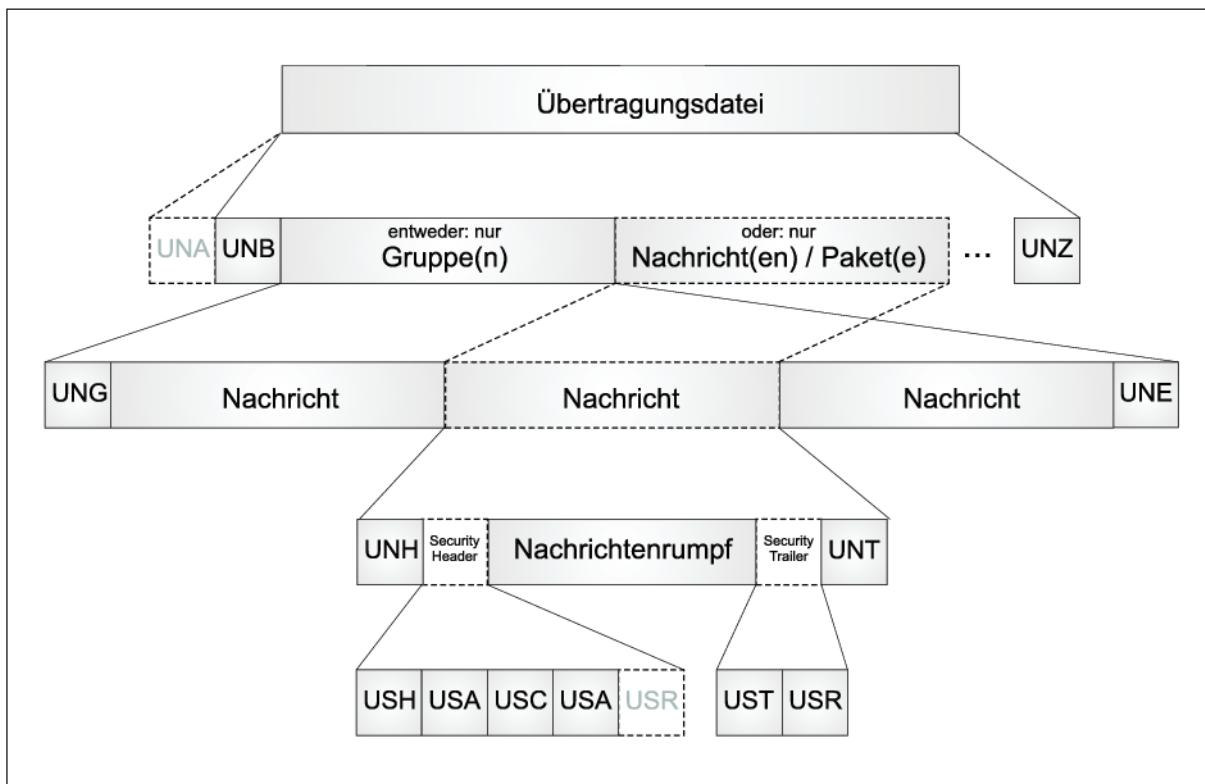


Abbildung 33: Beispielhafte EDIFACT-Übertragungsdatei

Wie in Abbildung 33 angedeutet, besteht eine EDIFACT-Übertragungsdatei (Interchange) aus einem optionalen UNA-Segment, durch das u.a. die verwendeten Trennzeichen festgelegt sind, dem Interchange-Header-Segment (UNB), den zu übertragenden Daten und schließlich dem Interchange-Trailer-Segment (UNZ). Die zu übertragenden Daten bestehen entweder aus Gruppen (von Nachrichten oder Paketen) oder nur aus Nachrichten oder Paketen. Eine Gruppe besteht aus dem Gruppen-Header (UNG), mehreren Nachrichten oder Paketen und endet mit dem Gruppen-Trailer (UNE). Eine Nachricht besteht wiederum aus einem Nachrichtenkopf (Message-Header, UNH), möglicherweise<sup>34</sup> einer Security Header Gruppe, dem Nachrichtenrumpf (Message Body), einer Security Trailer Gruppe und schließlich dem Nachrichten-Ende-Segment (Message-Trailer, UNT). Seit der Version 4 des ISO9735-

<sup>34</sup>Die Verwendung von Sicherheitsmechanismen für EDIFACT ist optional und erst mit der Syntax Version 4 eingeführt worden.

Standards gibt es auch die Möglichkeit, Paket-Objekte einzubinden. Hierfür wird der Objekt-Header (UNO) und der Objekt-Trailer (UNP) eingesetzt.

Wie in [ISO9735-5] spezifiziert, besteht die Security Header Gruppe aus folgenden Segmenten:

- Security Header (USH)  
gibt an, welcher Sicherheitsdienst, in welcher Art und Weise, auf die eingeschlossenen oder referenzierten Daten angewandt wird. Beispielsweise wird man beim Einsatz digitaler Signaturen zur Übermittlung elektronischer Rechnungen in Form von EDIFACT [INVOIC]-Nachrichten (vgl. *Abschnitt 5.2*) hier den Sicherheitsdienst „Non-repudiation of origin“ angeben. Neben dem Sicherheitsdienst können weitere Elemente, wie z.B. eine Sicherheitsfolgenummer oder ein Zeitstempel, angegeben werden.
  - Security Algorithm (USA)  
spezifiziert im allgemeinen den zu verwendenden kryptographischen Algorithmus. Beim Einsatz der digitalen Signatur wird hier der verwendete *Hashalgorithmus* festgelegt.
  - Certificate (USC)  
enthält einen *öffentlichen Schlüssel* oder Informationen zu einem verwendeten *Zertifikat*, wie beispielsweise die Seriennummer und den Aussteller des Zertifikates.
  - Security Algorithm (USA)  
Bei der digitalen Signatur kann dieses USA-Segment in drei Ausprägungen vorkommen:
    - a) als der vom Aussteller eines EDIFACT-Zertifikates verwendete Algorithmus zur Berechnung des Hashwertes des Zertifikats,
    - b) als der vom Zertifikatsaussteller verwendete Signaturalgorithmus zur Erstellung des Zertifikats und
    - c) als der vom Absender verwendeten Signaturalgorithmus zur Signierung einer Nachricht bzw. eines Pakets.
- Sofern X.509-Zertifikate eingesetzt werden, enthält dieses USA-Segment also den zur Signatur der Nutzdaten eingesetzten Algorithmus.
- Security Result (USR)  
enthält möglicherweise die Signatur des im USC-Segment angegebenen öffentlichen Schlüssels. Wird im USC-Segment auf ein X.509-Zertifikat verwiesen, so kann dieses USR-Segment entfallen.

Die Security Trailer Gruppe besteht aus folgenden Segmenten:

- Security Trailer (UST)  
sorgt für die Verbindung zur oben erläuterten Security Header Gruppe und gibt an, aus wie vielen USR-Segmenten die Security Trailer Gruppe besteht.
- Security Result (USR)  
enthält schließlich das Ergebnis der Sicherheitsoperation, beispielsweise eine digitale Signatur.

Neben der Signatur von EDIFACT-Nachrichten mittels Header und Trailer kann auch die in [ISO9735-6] spezifizierte AUTACK-Nachricht zum Einsatz kommen. Wie in *Abbildung 34* dargestellt, dient die AUTACK-Nachricht entweder der AUThentisierung von gesendeten Übertragungsdateien, Gruppen, Nachrichten oder Paketen oder der kryptographisch gesicherten Bestätigung des Empfangs (ACKnowledgement) von Übertragungsdateien,

Gruppen, Nachrichten oder Paketen.

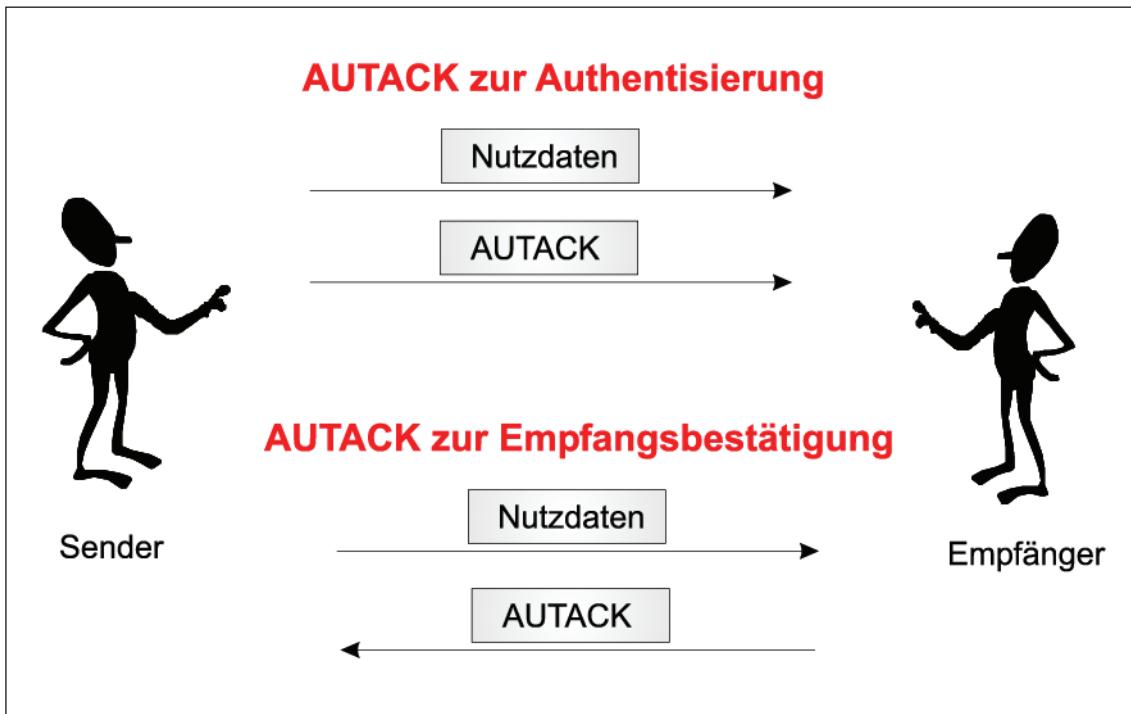


Abbildung 34: Einsatz der AUTACK-Nachricht

Die AUTACK enthält selbst eine oben erläuterte Security Header (USH) Gruppe und ein Security Trailer Segment (UST), die eine Reihe von AUTACK-spezifischen Segmenten umfassen:

- Secured Data Identification (USB)  
enthält Informationen zum Sender und Empfänger der AUTACK-Nachricht.
- Security References (USX)  
enthält den Verweis auf die durch die AUTACK-Nachricht gesicherten Daten in Form von Referenznummern für Interchanges oder Messages.
- Security on References (USY)  
enthält das Ergebnis der Sicherheitsoperation, die auf die referenzierten Daten angewandt wird. Beispielsweise kann hier eine digitale Signatur der referenzierten Daten enthalten sein.

Eine AUTACK-Authentisierungs-Nachricht kann auf zwei Arten angewendet werden. Die erste Methode transportiert die Hashwerte der referenzierten EDIFACT-Strukturen, die durch die AUTACK selbst gesichert werden. Bei der zweiten Methode wird die AUTACK nur dazu verwendet, die digitalen Signaturen der referenzierten EDIFACT-Strukturen zu transportieren.

Da bei der Verwendung der AUTACK-Nachricht die eigentlichen EDIFACT-Nutzdaten, beispielsweise eine Rechnung im [INVOIC]-Format, noch unter Verwendung der EDIFACT Syntax Version 3 vorliegen können, empfiehlt der EDI-Anwenderkreis Handel in [EDI-AK-Handel] den Einsatz der AUTACK zur Übermittlung elektronischer Rechnungen.

Auch bei den EDIFACT-Signaturen wird, ähnlich wie bei der XML-Signatur (vgl. Abschnitt 4.3.3), keine ASN.1-Codierung eingesetzt, sondern auf im EDIFACT-Umfeld übliche Codes für bestimmte Datenstrukturen zurückgegriffen.

### 4.3.5 PGP und PGP/MIME

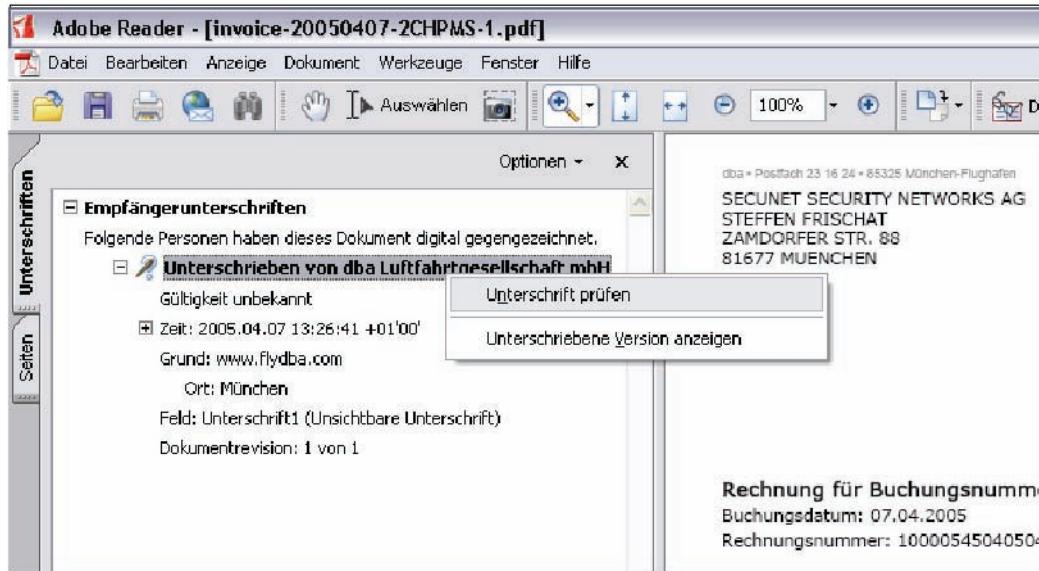
Pretty Good Privacy (PGP) ist ein von Phil Zimmermann entwickeltes Programm zur *Verschlüsselung* sowie Erzeugung und Prüfung von *digitalen Signaturen* von Daten. Die *Authentizität* der *öffentlichen Schlüssel* wird im Regelfall durch ein so genanntes „*Web of Trust*“ sichergestellt – alternativ ist auch die Verwendung von *X.509-Zertifikaten* möglich. Das PGP-Nachrichtenformat ist in [RFC2440] spezifiziert. Es verwendet nicht durchgängig ASN.1-Codierung, sondern eine so genannte Radix-64-Codierung, bei der eine base64-Codierung gemäß [RFC2231] und eine 24-Bit CRC-Prüfsumme zum Einsatz kommen. Außerdem können Nachrichten durch so genannte „ASCII-Armors“ der Form „-----BEGIN PGP (...)-----“ und „-----END PGP (...)-----“ abgegrenzt werden, so dass insbesondere die Verarbeitung von Text-Nachrichten leicht möglich ist. Die OpenPGP-Spezifikation [RFC2440] legt fest, wie sich eine Nachricht aus verschiedenen Paketen (Packets) zusammensetzt. Ein solches Paket besteht aus einem Header, der den Typ über ein „Packet Tag“ und die Länge festlegt und den eigentlichen Inhalt enthält. Tag=2 gibt beispielsweise an, dass es sich um eine Signatur handelt. Die Spezifikation beschreibt verschiedene Signaturtypen, durch die nicht nur Nachrichten signiert, sondern auch Schlüssel verwaltet werden können. Bei der Version 4 des PGP-Nachrichtenformats besteht eine Signatur aus Sub-Packets, in denen weitere Informationen über die Signatur, wie z.B. der Signaturzeitpunkt, enthalten sein können.

Neben der OpenPGP-Spezifikation [RFC2440], die die Signatur von binären Daten oder ASCII-Text spezifiziert, existieren ähnliche Festlegungen für MIME-Daten [RFC3156].

### 4.3.6 Einbettung von Signaturen in PDF-Dokumente

Wie oben erläutert, ist es bei *CMS-Signaturen* (vgl. *Abschnitt 4.3.1*) nicht in jedem Fall möglich, die Signatur in standardisierter Art und Weise in ein signiertes Dokument einzubetten (vgl. „enveloped signature“ in *Abbildung 31*). Eine Ausnahme bildet das *PDF*-Format [PDF(v1.3), PDF(v1.4), PDF(v1.5), PDF(v1.6)], das spezifische Felder vorsieht, in denen Signaturen eingefügt werden können. Eine Signatur wird in Form eines Verzeichnisses (*Signature Directory*) in das PDF-Dokument eingebettet. Dieses Verzeichnis kann beispielsweise aus folgenden Feldern bestehen:

- **Type**  
gibt den Typ des Verzeichnisses an. Der Wert `Sig` gibt an, dass es sich um ein *Signature Directory* handelt.
- **Filter**  
enthält den Namen des zu verwendenden Signatur-Handlers, z.B. `AdobePPKLite`.
- **SubFilter**  
gibt an, um welche Art der Signatur es sich handelt, z.B. `adbe.pkcs7.detached`.
- **ByteRange**  
gibt an, auf welche Daten sich die Signatur bezieht, wobei ein Teil des Dokuments jeweils durch einen Offset und seine Länge referenziert wird. Wie bei der XML-enveloped-signature muss auch hier der Bereich, in den die Signatur eingefügt wird, bei der Signaturerzeugung ausgespart werden.
- **Contents**  
enthält die digitale Signatur.



**Abbildung 35: Prüfung der Signatur im PDF-Reader**

Der Vorteil der Einbettung der Signatur in das *PDF*-Dokument besteht insbesondere darin, dass die Prüfung der Signatur, wie in Abbildung 35 angedeutet, mit einer aktuellen Version des kostenlosen Adobe Reader möglich ist. Da dieser meist zur Standard-Rechnerausstattung zählt, kann für die Prüfung elektronischer Signaturen auf die zusätzliche Installation einer *Signaturanwendungskomponente* verzichtet werden.

### 4.4 Massensignatur

Unter der „Massensignatur“ versteht man die automatisierte Erzeugung *qualifizierter elektronischer Signaturen*. Das Signaturgesetz [SigG] benennt die automatisierte Signaturerstellung nicht explizit, verbietet sie aber auch nicht. Dies liegt hauptsächlich daran, dass das Signaturgesetz insbesondere die Belange von *Zertifizierungsdiensteanbietern* (ZDA) regelt, nicht aber die Art und Weise der Nutzung des Signaturschlüssels durch den Anwender. Die Begründung zur Signaturverordnung [SigVBeg] nimmt jedoch im Zusammenhang mit § 15 Abs. 2 [SigV] Bezug auf die automatisierte Signaturerstellung:

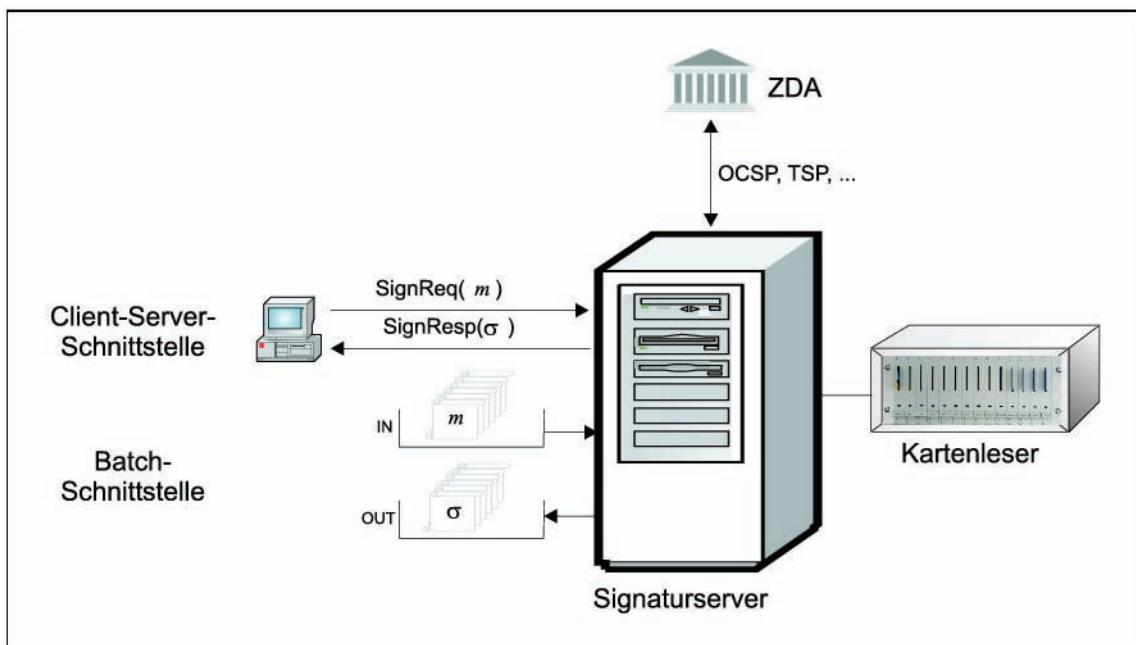
„Insbesondere bei der automatischen Erzeugung von Signaturen („Massensignaturen“) muss sichergestellt sein, dass Signaturen nur zu dem voreingestellten Zweck (z. B. Signaturen zu Zahlungsanweisungen bei Großanwendern) und durch eine zuvor geprüfte und abgenommene Anwendung vorgenommen werden können.“

§ 15 Abs. 2 [SigV] sowie dessen Begründung gibt sicherheitstechnische Hinweise, die dazu dienen, dass eine Signatur nur durch die berechtigte Person erzeugt werden kann und nur über die Daten erzeugt wird, welche diese Person signieren will. Es handelt sich also um Maßnahmen, die den Signaturschlüssel-Inhaber vor dem Missbrauch seines Schlüssels schützen sollen. Beispielsweise wird in [SigVBeg] empfohlen, dass man die Signaturerstellungseinheit nur für einen bestimmten Zeitraum oder für eine bestimmte Anzahl an Signaturen freischaltet. Weitere Sicherheitsaspekte beim Einsatz der Massensignatur sind beispielsweise in [HuKn03, Abschnitt 3.1] beleuchtet.

Außerdem gibt die Antwort zu Frage 18 der FAQ der *BNetzA* Folgendes zu bedenken:

„Trotz Verwendung dieser technischen Hilfsmittel werden die Erklärungen aus den signierten Dokumenten dem Absender persönlich zugerechnet. Daher sollte bei derartigen „automatisch“ erstellten Signaturen immer ein besonderer Schutz gegen Missbrauch implementiert werden. Dieser Schutz sollte sich an dem Aktivierungszeitraum orientieren, was von einem verschlossenen Stahlschrank für Karte und Kartenleser, bis hin zur Trust-Center-Umgebung reichen kann.“

In technischer Hinsicht verwendet man zur Massensignatur – wie in Abbildung 36 angedeutet – spezialisierte und besonders geschützte Signaturserver, denen die zu signierenden Nachrichten stapelweise, über eine Batch-Schnittstelle, oder einzeln über eine Client-Server-Schnittstelle im Stile von OCSP oder TSP zugeführt werden. Die Batch-Schnittstelle kann einfach dadurch realisiert werden, dass die zu signierenden Daten in ein Eingangsverzeichnis (IN) gelegt werden, das vom Signaturserver durch einen Polling-Mechanismus überwacht wird. Der Signaturserver entnimmt die Dateien diesem Verzeichnis, erstellt eine zugehörige Signatur und legt diese Signatur im Ausgangsverzeichnis (OUT) ab. Da bisher nur *sichere Signaturerstellungseinheiten* in Form von Chipkarten existieren, ist ein Chipkartenleser mit mehreren Steckplätzen an den Signaturserver angeschlossen. Soll der Server auch Signaturen mittels OCSP prüfen oder Zeitstempel über TSP anfordern, so ist eine entsprechende Verbindung zum Trust-Center des Zertifizierungsdiensteanbieters nötig.

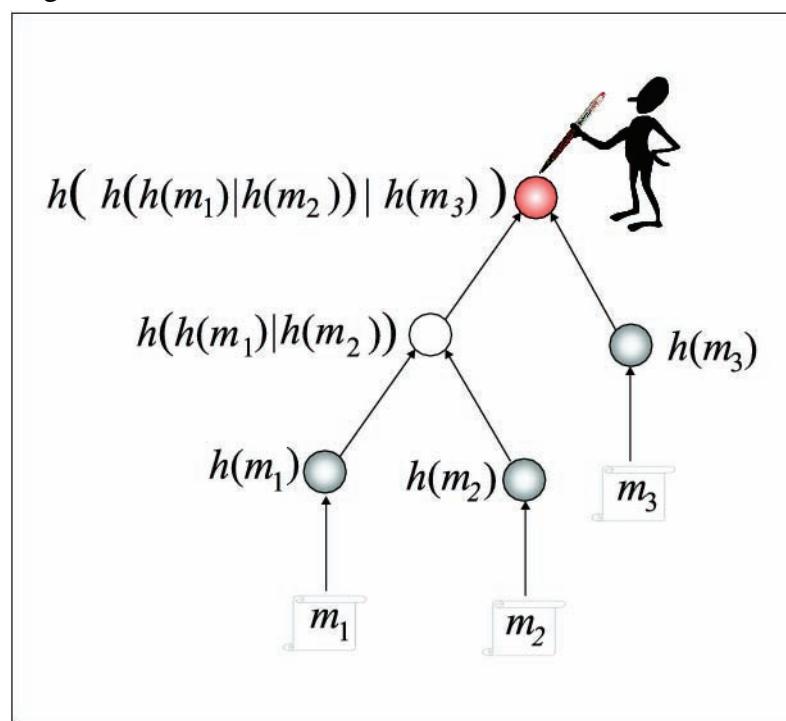


**Abbildung 36:** Massensignatur-System

Karten	Signaturen pro Minute	Signaturen pro Tag	Ideale Anzahl
1	46	66.240	66.240
2	79	113.760	132.480
4	132	190.080	264.960
8	199	286.560	529.920
16	376	540.889	1.059.840
32	717	1.032.943	2.119.680
64	1.401	2.017.052	4.239.360
128	2.768	3.985.269	8.478.720
256	5.501	7.921.703	16.957.440

**Tabelle 5: Massensignatur-Benchmark**

Die Tatsache, dass zur Erstellung *qualifizierter elektronischer Signaturen sichere Signaturerstellungseinheiten* eingesetzt werden müssen, die derzeit nur in Form von Chipkarten zur Verfügung stehen, führt in der Praxis zu keinen unüberwindbaren Performanz-Problemen. Denn wie Benchmarks (vgl. *Tabelle 5*)<sup>35</sup> belegen, kann ein Signaturserver, der mit 32 parallel arbeitenden Chipkarten ausgestattet ist, mehr als eine Million Signaturen pro Tag erstellen.



**Abbildung 37: Hashbaum für Stapelsignatur**

Muss ein signifikant größeres Signaturvolumen bewältigt werden, so kann auch die Stapel-Signatur-Strategie aus [PaBo99] eingesetzt werden. Wie in *Abbildung 37* angedeutet, wird

<sup>35</sup>Die Daten beziehen sich auf einen Solaris-basierten Signaturserver, 200kB Nachrichten, PKCS #7-Signaturen und Chipkarten mit 1024 Bit RSA.

nicht jede Nachricht  $m_i$  einzeln signiert, sondern erst daraus eine Hashbaum im Stile von [Merk80] konstruiert, bei dem lediglich die Wurzel signiert wird. Wie in [PaBo99, Lemma 1] gezeigt wurde, bleibt die Sicherheit des Signaturverfahrens erhalten, sofern man eine kollisionsresistente kryptographische Hashfunktion zur Konstruktion des Hashbaumes verwendet.

Der Nachteil dieser Vorgehensweise ist, dass dieses Stapelsignatur-Verfahren bislang nicht standardisiert ist. Hierfür könnte beispielsweise die `SignedData`-Struktur aus [RFC3369] entsprechend erweitert werden.

## 4.5 Zeitstempel

Da bei der Prüfung von *qualifizierten elektronischen Signaturen* gemäß § 2 Nr. 3 [SigG] der Zeitpunkt der *Erstellung* der Signatur heranzuziehen ist (vgl. *Abschnitt 4.2*), ist die beweiskräftige Dokumentation dieses Zeitpunktes bei diesen Signaturen von besonderer Bedeutung. Da man mit *Zeitstempeln* beweisen kann, dass bestimmte Daten bereits vor einem bestimmten Zeitpunkt existiert haben, bietet sich für die Dokumentation des Signaturerstellungszeitpunktes die Verwendung von Zeitstempeln an.

Ist das *qualifizierte Zertifikat*, auf dem eine Signatur beruht, zum Zeitpunkt der Prüfung nicht gesperrt, so gilt dies auch für den Zeitpunkt der Signaturerstellung (vgl. [DGI05]). Wird das Zertifikat aber vor dem Zeitpunkt der Prüfung gesperrt, so kann ohne Einsatz vertrauenswürdiger Zeitstempel nicht mit Sicherheit entschieden werden, ob diese Signatur gültig ist oder nicht. Eine vor dem Zeitpunkt der Sperrung erstellte Signatur wäre gültig, eine danach erstellte Signatur wäre hingegen ungültig. Deshalb sollte eine qualifizierte elektronische Signatur stets direkt nach der Erstellung mit einem vertrauenswürdigen Zeitstempel versehen werden.

Grundsätzlich können Zeitstempel vom Anwender selbst erstellt, oder – beispielsweise mittels *Time Stamp Protocol* – von einem vertrauenswürdigen Zeitstempeldienst bezogen werden (vgl. *Abschnitt 3.2.7*). Unterschiede ergeben sich hierbei insbesondere im Hinblick auf die Beweiskraft des Zeitstempels (vgl. *Abschnitt 2.2.2*).

Ein durch den Anwender selbst erstellter Zeitstempel entfaltet unter Umständen nur eine sehr geringe Beweiskraft. Wird der Zeitstempeldienst aber von einem *Zertifizierungsdiensteanbieter* gemäß Signaturgesetz betrieben, so handelt es sich um *qualifizierte Zeitstempel*, die eine sehr hohe Beweiskraft vor Gericht versprechen. Leider ist der Bezug von qualifizierten Zeitstempeln in der Regel mit Transaktionsgebühren verbunden, die sich bei einer großen Anzahl an Zeitstempeln als signifikanter Kostenfaktor erweisen.

Deshalb wurde in [Hueh04c] ein Verfahren vorgeschlagen, wie eine beliebige Anzahl an selbst erzeugten, und dadurch kostengünstigen, Zeitstempeln mit der hohen Beweiskraft qualifizierter Zeitstempel versehen werden kann. Hierfür bedient man sich kryptographisch geeigneter Hashfunktionen (vgl. *Abschnitt 3.1.4*), deren Einwegeigenschaft eine relative zeitliche Ordnung induziert. Fließt der Hashwert eines zum Zeitpunkt  $T_1$  ausgestellten qualifizierten Zeitstempels  $QS_1$  in den zum Zeitpunkt  $t$  vom Anwender selbst erstellten Zeitstempel  $S$  ein, so gilt

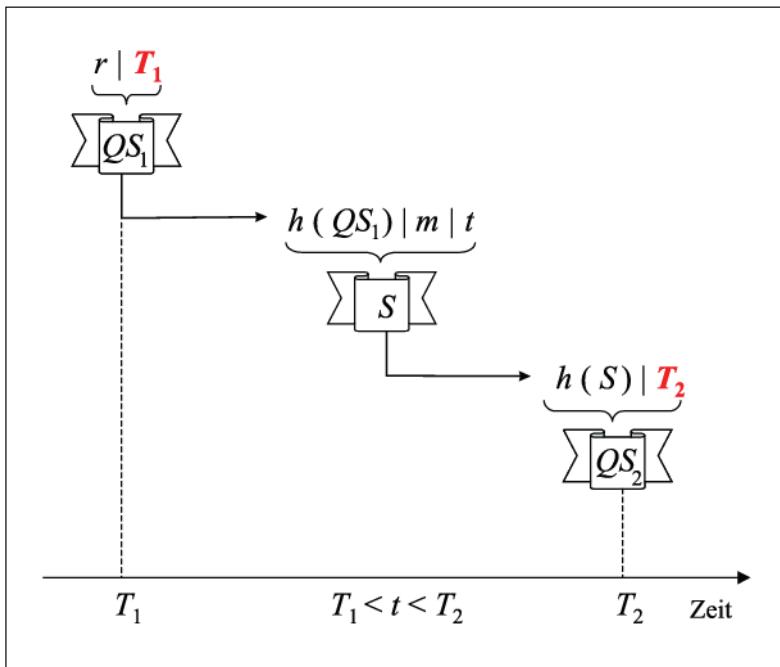
$$t > T_1.$$

Wird über den Hashwert  $h(S)$  zum Zeitpunkt  $T_2$  ein zweiter qualifizierter Zeitstempel  $QS_2$  erstellt, so gilt

$$T_2 > t.$$

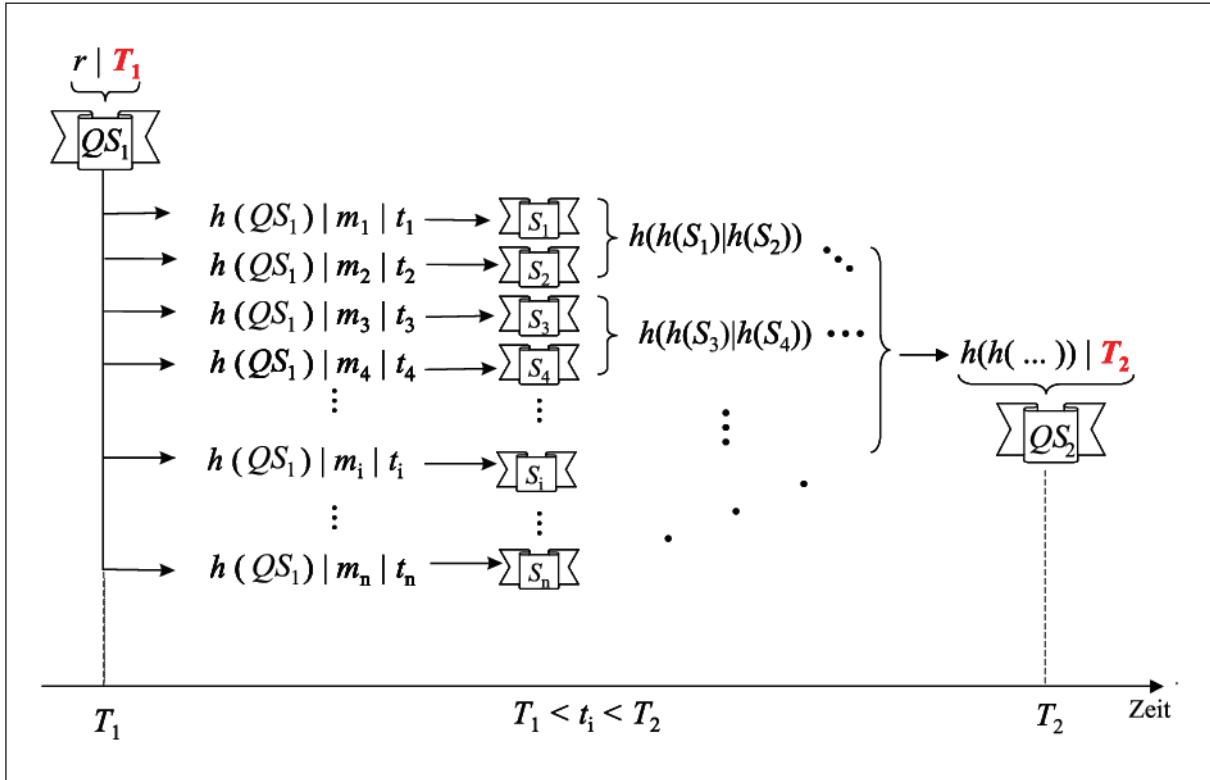
Aus (4.5.1) und (4.5.2) folgt schließlich  $T_1 < t < T_2$ . Der Anwender muss also den Zeitstempel  $S$ , wie in *Abbildung 38* veranschaulicht, im offenen Zeitintervall zwischen  $T_1$

und  $T_2$  erstellt haben.



**Abbildung 38: Relative zeitliche Ordnung durch Hashfunktion**

Sollen nun mehrere Zeitstempel  $S_i$ ,  $1 \leq i \leq n$ , zwischen den beiden qualifizierten Zeitstempeln  $Q_1$  und  $Q_2$  erzeugt werden, so wird der Zeitstempel  $QS_1$  in jeden Zeitstempel  $S_i$  einbezogen und die verschiedenen Zeitstempel  $S_i$  wiederum in geeigneter Weise in den Zeitstempel  $QS_2$  integriert. Hierfür bietet sich die Verwendung der in Abbildung 37 dargestellten Stapelsignatur-Strategie aus [PaBo99] an. Wie in Abbildung 39 abgebildet, wird hierzu aus den vom Anwender selbst erstellten Zeitstempeln  $S_i$  ein Hashbaum aufgebaut, dessen Wurzel für die Erzeugung des Zeitstempels  $QS_2$  verwendet wird.



**Abbildung 39: Konstruktion Intervall-Qualifizierter Zeitstempel**

Dadurch kann bewiesen werden (vgl. [Hueh04a, Theorem 1]), dass alle Zeitstempel  $S_i$  im Intervall zwischen  $T_1$  und  $T_2$  erzeugt wurden. Durch diese Konstruktion ist ein Anwender zu den Zeitpunkten  $t_i$  selbst in der Lage, „Intervall-qualifizierte (IQ) Zeitstempel“ mit interessanten Eigenschaften zu erzeugen. Denn diese Zeitstempel besitzen bezüglich der Intervallzugehörigkeit  $T_1 < t_i < T_2$  den hohen Beweiswert der qualifizierten Zeitstempel, wohingegen der genaue Zeitpunkt  $t_i$  der Erstellung von  $S_i$  nur mit dem geringeren Beweiswert des selbsterzeugten Zeitstempels dokumentiert ist. Wird beispielsweise täglich ein qualifizierter Zeitstempel angefordert, so ist zwar das Datum der Intervall-qualifizierten Zeitstempel nachweislich authentisch, aber möglicherweise nicht die Uhrzeit. Somit kann ein „IQ-Zeitstempeldienst“, wie in Abbildung 40 angedeutet, auf kostengünstige Art und Weise selbst beliebig viele Zeitstempel  $S_i$  erzeugen und muss nur einmal täglich einen kostenpflichtigen qualifizierten Zeitstempel bei einem Zertifizierungsdienstanbieter beziehen.

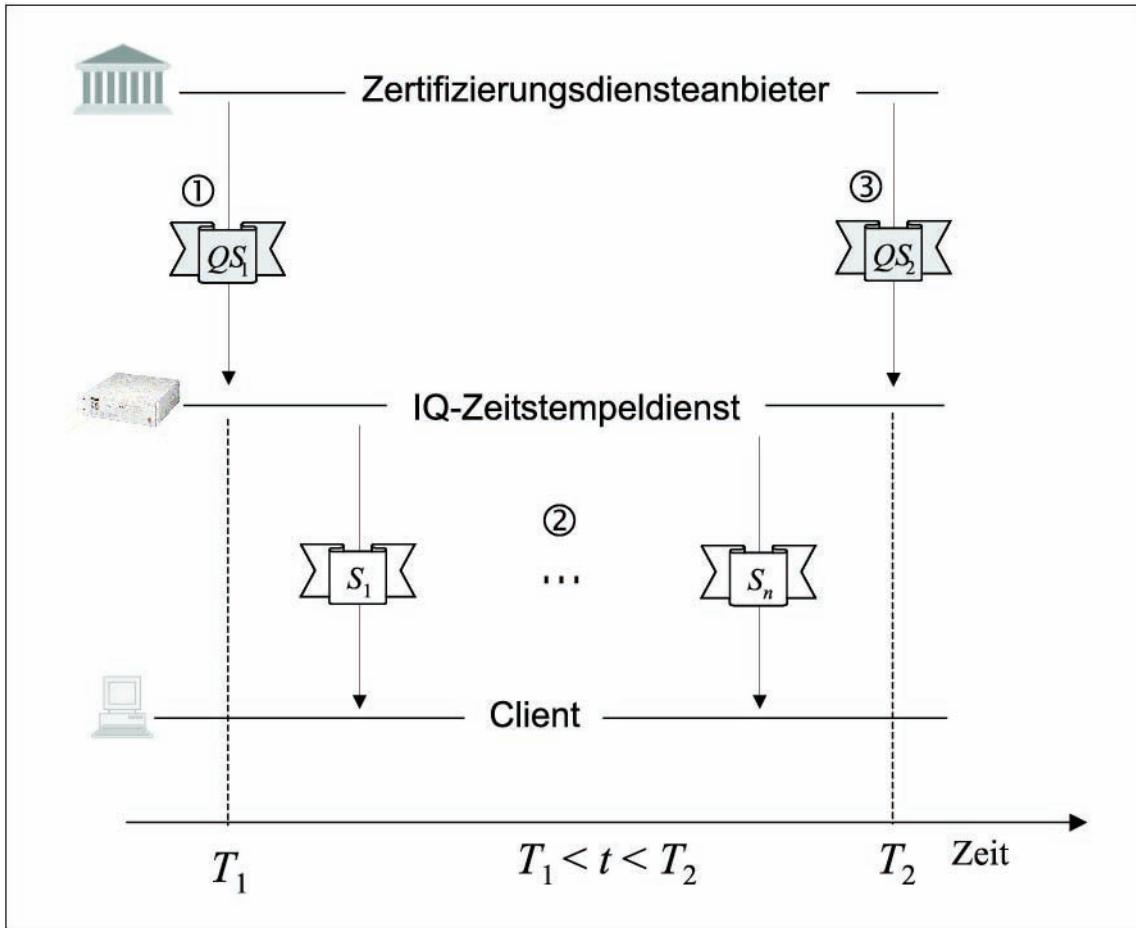


Abbildung 40: IQ-Zeitstempel-System

## 4.6 Archivierung von signierten Daten

Bei der Archivierung von signierten Daten muss neben den sonstigen Aspekten bei der elektronischen Archivierung (vgl. [KaRo97, VOI05]) insbesondere berücksichtigt werden, dass kryptographische „Einwegfunktionen“ (vgl. Abschnitt 3.1.2) nach heutigem Kenntnisstand selbst Funktionen der Zeit sind und kryptographische Mechanismen deshalb im Laufe der Zeit ihre Sicherheitseignung verlieren können. Eine Signatur, die heute ausreichende Fälschungssicherheit bietet, kann durch einen glücklichen Einfall eines einzigen Mathematikers vielleicht schon morgen leicht fälschbar sein.

Um die Beweiskraft elektronischer Signaturen langfristig erhalten zu können, muss deshalb insbesondere auch der Zeitpunkt der Signaturerstellung beweiskräftig dokumentiert werden, so dass im Streitfall nachgewiesen werden kann, dass ein Fälschen der Signatur zu diesem Zeitpunkt nicht möglich war. Deshalb sieht § 17 [SigV] folgendes Verfahren zur langfristigen Datensicherung vor:

„Daten mit einer qualifizierten elektronischen Signatur sind nach § 6 Abs. 1 Satz 2 des Signaturgesetzes neu zu signieren, wenn diese für längere Zeit in signierter Form benötigt werden, als die für ihre Erzeugung und Prüfung eingesetzten Algorithmen und zugehörigen Parameter als geeignet beurteilt sind. In diesem Falle sind die Daten vor dem Zeitpunkt des Ablaufs der Eignung der Algorithmen oder der zugehörigen Parameter mit einer neuen qualifizierten elektronischen Signatur zu

versehen. Diese muss mit geeigneten neuen Algorithmen oder zugehörigen Parametern erfolgen, frühere Signaturen einschließen und einen qualifizierten Zeitstempel tragen.“

Nach dem Sinn und Zweck dieser Regel wird man zu dem Schluss gelangen, dass es ausreichend ist, wenn die Nachsignatur durch einen qualifizierten Zeitstempel erfolgt, der durch eine qualifizierte elektronische Signatur erstellt wird.

Bei einer juristischen Auslegung dieser Regel nach ihrem Sinn und Zweck wird man zu dem Schluß gelangen, dass es ausreichend ist, wenn die Nachsignatur durch einen *qualifizierten Zeitstempel* erfolgt, der durch eine *qualifizierte elektronische Signatur* realisiert wird. Hierfür kann für jede Signatur, beispielsweise über *TSP* (vgl. *Abschnitt 3.2.7*), ein einzelner *qualifizierter Zeitstempel* angefordert werden, oder – wie im Rahmen des ArchiSig-Projektes [RoSc05] vorgeschlagen und nun auch beispielsweise im ArchiSafe-Projekt der Physikalisch-Technischen Bundesanstalt realisiert [PTB05a, PTB05b] – aus einer großen Anzahl an Signaturen, ähnlich wie in *Abbildung 37*, ein Hashbaum konstruiert werden, bei dem lediglich die Wurzel mit einem *qualifizierten Zeitstempel* versehen wird. Hierfür kann die im Rahmen der „Long Term Archiving and Notary Services (LTANS)“-Arbeitsgruppe der *IETF* entwickelte „Evidence Record Syntax“ (ERS) [BrPo05] verwendet werden. Alternativ dazu können die in *Abschnitt 4.5* erläuterten „Intervall-qualifizierten Zeitstempel“ zur Nachsignatur eingesetzt werden.

### 4.7 Code-Signing

Ein Anwendungsfall der elektronischen Signatur, der einen wichtigen Beitrag zur Sicherheit in modernen IT-Systemen liefert, ist die Sicherstellung der *Authentizität* und *Integrität* von ausführbarem Programmcode zum Schutz vor Viren, Würmern und Trojanern (siehe auch <http://www.bsi.de/av/>).

Hierbei wird der ausführbare Programmcode von einer vertrauenswürdigen Stelle mit einer digitalen Signatur versehen und auf der Zielplattform nur dann aktiviert, wenn die Echtheit der Herkunft und die Unversehrtheit des Codes durch Prüfen der Signatur nachgewiesen werden kann.

Beispielsweise sieht das Microsoft-Code-Signing-System „Authenticode“ [MSAC] vor, dass der Programmautor ein so genanntes *Software Publishing Certificate* (SPC) von einer vertrauenswürdigen *Zertifizierungsinstanz* erhält, mit dem er seinen Code signiert. Diese Signatur wird bei der Installation von Systemprogrammen, Gerätetreibern oder vor der Ausführung von Active-X-Controls im Internet Explorer geprüft.

Ähnliche, aber untereinander nicht kompatible, Code-Signing-Systeme existieren auch bei anderen Herstellern, wie Sun oder Netscape. Beispielsweise kann ein Java-Applet [SunJCS] dadurch signiert werden, dass es zusammen mit einem so genannten „Manifest“ und der zugehörigen Signatur in eine Java-ARchive (JAR) Datei gepackt wird. Im Manifest finden sich Meta-Informationen über den Inhalt der JAR-Datei, die bei der Prüfung der Signatur ausgewertet werden können.

## 5 Anwendungsbereiche

In diesem Kapitel sollen einige Anwendungen der elektronischen Signatur näher betrachtet werden. Neben verschiedenen Anwendungsfällen der elektronischen Signatur im behördlichen Umfeld wird in *Abschnitt 5.2* insbesondere auf die elektronische Rechnungsstellung eingegangen.

Hierbei ist zu beachten, dass im Folgenden lediglich die Einsatzmöglichkeiten der elektronischen Signatur veranschaulicht werden sollen. Mit einer Nennung von tatsächlichen Einsatzgebieten und Projekten ist keine Bewertung durch das Bundesamt für Sicherheit in der Informationstechnik verbunden. Die folgenden rechtlichen Betrachtungen erheben keinen Anspruch auf Vollständigkeit. Somit kann die Lektüre des vorliegenden Werkes die eingehende Prüfung der relevanten rechtlichen Rahmenbedingungen im Einzelfall keinesfalls ersetzen.

### 5.1 Signaturanwendungen im behördlichen Umfeld

Weltweit sind immer mehr nationale und regionale Behörden rund um die Uhr online erreichbar – ohne Warteschlangen und Schalterschlusszeiten. So können Bürger und Unternehmen amtliche Vorgänge zeit- und ortsunabhängig über das Medium Internet abwickeln. Auf Grund der regelmäßig erforderlichen Schriftform spielt der Einsatz der *qualifizierten elektronischen Signatur* hierbei eine wesentliche Rolle.

In diesem Abschnitt sollen einige Anwendungsbereiche der elektronischen Signatur im behördlichen Umfeld aufgezeigt werden.

#### 5.1.1 Elektronische Steuererklärung

Seit 01.01.2005 müssen Lohnsteuer- und Umsatzsteuervoranmeldungen gemäß § 41a Abs. 1 [EStG] und § 18 Abs. 1 [UStG] grundsätzlich elektronisch erfolgen. Nähere Erläuterungen hierzu enthält das Schreiben des BMF vom 29.11.2004 [BMF04b].

Hierfür wurde im Auftrag der Finanzverwaltung eine neue Sicherheitsplattform für das ElsterOnline-Portal [ElsterOnline] realisiert, die die *Authentifizierung*, *Verschlüsselung* und elektronische Signatur für Web-Anwendungen über drei zertifikatsbasierte Verfahren unterstützt:

- **ELSTER-Plus**

Hierbei werden *Smart Cards* und *Zertifikate* von verschiedenen kommerziellen Anbietern unterstützt, so dass bei der Kommunikation mit der Finanzverwaltung *qualifizierte elektronische Signaturen* und *fortgeschrittene elektronische Signaturen* gemäß § 7 [StDUeV] zum Einsatz kommen können.

- **ELSTER-Spezial**

In diesem Fall kann der Anwender einen so genannten „ElsterStick“ – einen USB-Stick, in dem *Smart Card* und *Chipkartenterminal* vereinigt sind – erwerben. Das zugehörige Zertifikat erhält der Anwender von der Finanzverwaltung kostenlos.

- **ELSTER-Basis**

Hierbei wird für den Anwender kostenlos ein softwarebasiertes Schlüsselpaar und ein Zertifikat erzeugt. In diesem Fall wird der *private Schlüssel* in Form einer mit einem Passwort verschlüsselten *PKCS # 12* - Datei gespeichert.

Der Einsatz dieser Verfahren öffnet neue komfortable Online-Wege für nahezu alle steuerrelevanten Bereiche (z.B. Steueranmeldungen, Lohnsteuerkarte, Einkommensteuer und

## 5 Anwendungsbereiche

---

Steuerkontoabfrage).

Durch die vom Portal gebotenen personalisierten Dienste ergeben sich erweiterte Einsatzmöglichkeiten für Besitzer von Signaturkarten (ELSTER-Plus), welche den Nutzen dieser Signaturkarten erhöhen:

- Einfachere Registrierung zur Nutzung der personalisierten Dienste,
- Authentisierung zum Nachweis der Identität eines Nutzers für den persönlichen Zugang,
- Virtuelles E-Mail Postfach für verschlüsselte Kommunikation mit der Finanzbehörde,
- Elektronische Möglichkeit für einen Antrag auf Steuerkontoeinsicht,
- Funktion der Steuerkontoabfrage.

Besitzt der Anwender keine Signaturkarte, wird für ihn im Rahmen seiner Registrierung im Portal ein Softwareschlüssel (ELSTER-Basis) oder ein Schlüssel für den ElsterStick (ELSTER-Spezial) erzeugt.

Weitere Informationen zum ELSTER-Projekt finden sich unter [Elster].

### 5.1.2 Vergabe öffentlicher Aufträge

Gemäß § 15 [VgV] können Angebote für öffentliche Aufträge elektronisch abgegeben werden. „Digitale Angebote sind mit einer *qualifizierten elektronischen Signatur* nach dem Signaturgesetz zu versehen und zu *verschlüsseln*.“

Hierfür wird vom Bundesbeschaffungsamt eine Beschaffungsplattform [eVergabe] bereitgestellt, die von den Bundesbehörden genutzt werden kann. Ähnliche Plattformen existieren auch auf Landesebene.

### 5.1.3 Genehmigung der Ein- und Ausfuhr geschützter Tiere und Pflanzen

Das Bundesamt für Naturschutz (BfN) ist für die Genehmigung der Ein- und Ausfuhr geschützter Tiere und Pflanzen nach dem Washingtoner Artenschutzübereinkommen (Convention on International Trade in Endangered Species of Wild Fauna and Flora – CITES) zuständig.

Um die Antragstellung und Genehmigung zu erleichtern, wurden die relevanten Prozesse vollständig elektronisch abgebildet [CITES]. Die eigenhändigen Unterschriften an den Antragsformularen werden durch *qualifizierte elektronische Signaturen* ersetzt. Weitere Informationen zu diesem Projekt finden sich in [Ley04].

### 5.1.4 Rechnungswesen in der Sozialversicherung

Im Umfeld des Rechnungswesens in der Sozialversicherung existieren eine Reihe von detaillierten Vorschriften, die entweder eigenhändige Unterschriften, und damit Papiergebundene Prozesse, oder den Einsatz von *qualifizierten elektronischen Signaturen* vorsehen. Hierbei ist insbesondere die Sozialversicherungsrechnungsverordnung [SVRV] und die zugehörige allgemeine Verwaltungsvorschrift [SRVwV] zu berücksichtigen. Für die Aufbewahrung von Unterlagen sind die §§ 110a-d [SGBIV] maßgebend, wobei durch den Einsatz *qualifizierter elektronischer Signaturen* gemäß § 110d [SGBIV] eine hohe Beweiskraft erreicht werden kann. Ein Überblick über die Einsatzgebiete der elektronischen Signatur in der Sozialversicherung findet sich in [HeHu02].

### 5.1.5 Patentantrag

Das Deutsche Patent- und Markenamt (DPMA) nimmt Anträge zur Patent- und Markenanmeldung elektronisch entgegen. Dies umfasst Anträge, die für das DPMA bestimmt sind, und solche, die an das Europäische Patentamt (EPA) gerichtet sind. Vorgesehen ist, dass die Antragsteller die Antragsunterlagen mit der vom DPMA unentgeltlich bereit gestellten Software PaTrAS erstellen und anschließend per E-Mail oder Datenträger an die Annahmestelle übermitteln. Die übermittelten Daten sind in einer ZIP-Datei enthalten, die qualifiziert zu signieren und zu verschlüsseln ist. Weitere Informationen zu diesem Projekt finden sich unter [BOS-DPMA].

### 5.1.6 Justizkommunikation

Bei der elektronischen Kommunikation innerhalb und mit Justizbehörden spielt der Einsatz der *qualifizierten elektronischen Signatur* aus zwei Gründen eine wichtige Rolle: Einerseits gelten auch hier Schriftformerfordernisse, beispielsweise für die so genannten bestimmenden Schriftsätze wie z.B. Klageschriften. Darüber hinaus besteht hier ein besonderes Bedürfnis an der Sicherung der *Integrität* und *Authentizität* elektronischer Dokumente – insbesondere zur Sicherung ihres Beweiswerts.

Die grundsätzliche Öffnung der Justiz für den elektronischen Rechtsverkehr wurde in zwei Schritten vollzogen: Zunächst wurde durch das [FormAnpG] vom 13.07.2001 die elektronische Form für Schriftsätze zugelassen. Durch das [JKomG] vom 22.03.2005 ist nunmehr auch die Zustellung gerichtlicher Entscheidungen in elektronischer Form sowie die elektronische Aktenführung in der Justiz möglich.

Von diesen Möglichkeiten haben Bund (für die Bundesgerichte) und Länder (für die Instanzgerichte) in unterschiedlichem Maße Gebrauch gemacht. Die folgenden diesbezüglichen Projekte sind dabei besonders erwähnenswert:

- Elektronischer Antrag zur Eröffnung von Mahnverfahren

Zur Eröffnung eines Mahnverfahrens kann bereits mit vielen Mahngerichten elektronisch kommuniziert werden. Dabei werden Anträge auf Erlass eines Mahn- oder eines Vollstreckungsbescheids mit einer geeigneten Software in Form von Datensätzen erzeugt, die direkt in das Fachverfahren beim Mahngericht eingespielt werden können. In vielen Fällen kann das Fachverfahren automatisch eine Entscheidung treffen und die Bescheide erstellen. Entsprechende Softwareangebote finden sich unter [OptiMahn] und [ProfiMahn].

- Elektronischer Rechtsverkehr am Finanzgericht Hamburg

Seit Mai 1999 wurde beim Finanzgericht Hamburg der elektronische Rechtsverkehr in einem Feldversuch erprobt. Seit Mai 2002 können beim Finanzgericht Hamburg nun Klagen, vorläufige Rechtsschutzgesuche und Schriftsätze per E-Mail eingereicht werden. Weitere Informationen zu diesem Projekt finden sich unter [FGHH-ERV].

- Elektronischer Rechtsverkehr am Bundesgerichtshof

Der Bundesgerichtshof erprobte im Rahmen eines Modellversuchs den Umgang mit elektronischen Schriftsätzen im Zivilprozess. Seit November 2001 haben Rechtsanwälte die Möglichkeit, in Zivilprozessen Schriftsätze per E-Mail oder über den elektronischen Gerichtsbriefkasten einzureichen. In einem der Zivilsenate werden die elektronisch eingegangenen Dokumente auch intern elektronisch weitergeleitet und bearbeitet. Weitere Informationen zu diesem Projekt finden sich unter [BGH-ERV]. Außerdem war der Bundesgerichtshof an der Entwicklung der ersten Version von XJustiz maßgeblich beteiligt.

## 5 Anwendungsbereiche

---

- XJustiz  
Die Bund-Länder-Kommission für Datenverarbeitung und Rationalisierung in der Justiz hat die XML-basierte Datensatzbeschreibung [XJustiz] entwickelt, die den medienbruchfreien Austausch möglichst vieler verfahrensrelevanter Daten ermöglichen soll.
- Elektronisches Verwaltungs- und Gerichtspostfach am Bundesverwaltungsgericht und Bundesfinanzhof  
Das elektronische Verwaltungs- und Gerichtspostfach soll dem Bundesverwaltungsgericht und dem Bundesfinanzhof ermöglichen, unstrukturierte Nachrichten zwischen den Gerichten und den Parteien rechtsverbindlich auszutauschen. Der sichere Nachrichtentransport wird dabei unter Einsatz der Virtuellen Poststelle des Bundes [BSI-VPS] realisiert, wobei die Rechtsverbindlichkeit der Erklärungen durch Einsatz qualifizierter elektronischer Signaturen erreicht wird.
- Justizkommunikation in Rheinland-Pfalz  
Die elektronische Kommunikation zum Oberverwaltungsgericht und zu den vier erstinstanzlichen Verwaltungsgerichten in Rheinland-Pfalz (Koblenz, Mainz, Neustadt an der Weinstraße und Trier) ist flächendeckend eröffnet. Damit ist die Verwaltungsgerichtsbarkeit Rheinland-Pfalz bundesweit die erste Gerichtsbarkeit, in der eine medienbruchfreie, instanzübergreifende Kommunikation möglich ist. Die Gerichte stellen neben einer Mailadresse für den Posteingang – dem „elektronischen Gerichtsbriefkasten“ – auch die elektronische Akteneinsicht und Verfahrensstandabfrage per Internet bereit [RLP-JP].

### 5.1.7 Einwohnermeldewesen

Durch das Melderecht ist eine Vielzahl von staatlichen Stellen über verwaltungsrelevante Daten der Einwohner informiert. Durch die Novellierung des Melderechtsrahmengesetzes [MRRG] im Jahre 2002 hat der Gesetzgeber die Rahmenbedingungen für ein effizientes Einwohnermeldewesen geschaffen:

- Melderegisterauskünfte dürfen über das Internet erteilt werden.
- Bürger können sich über das Internet anmelden.
- Bei Umzügen im Inland entfällt die Abmeldepflicht.
- Die Datenübermittlung zwischen Meldebehörden soll automatisiert stattfinden.

Gemäß § 11 Abs. 6 [MRRG] muss die Urheberschaft der Anmeldung durch eine *qualifizierte elektronische Signatur* nachgewiesen werden. In ähnlicher Art und Weise erhält der Betroffene gemäß § 8 Abs. 2 [MRRG] Auskunft über seine gespeicherten Daten, nachdem er einen mit einer qualifizierten elektronischen Signatur versehenen Antrag stellt. Für den Datenaustausch zwischen den einzelnen Stellen wird das OSCI-Nachrichtenformat [XMeld] verwendet.

Das elektronische Meldeverfahren ist beispielsweise in Hamburg bereits weitestgehend umgesetzt. Die rechtlichen Rahmenbedingungen sind in [HmbMG] und [HmbMDUeV] geregelt. Seit Oktober 2003 können elektronische Melderegisterauskünfte über das Internet abgerufen werden (vgl. [HH03]).

## 5.2 Elektronische Rechnungsstellung

Ein in der Praxis besonders wichtiger Anwendungsfall für die elektronische Signatur ist die elektronische Übermittlung von Rechnungen zwischen Unternehmen. Damit der Rechnungsempfänger eine erhaltene Rechnung zum Vorsteuerabzug heranziehen darf, muss

## 5 Anwendungsbereiche

diese den Anforderungen des § 14 [UStG] genügen und somit entweder auf Papier ausgestellt oder mit einer *qualifizierten elektronischen Signatur* versehen sein. Da durch die elektronische Übermittlung von Rechnungen zum Teil erhebliche Prozessoptimierungen und Kostensenkungen möglich werden, erfreut sich die elektronische Rechnungsstellung zunehmender Beliebtheit.

Nach der Vorstellung der rechtlichen Rahmenbedingungen der elektronischen Rechnung in *Abschnitt 5.2.1* soll in den folgenden Abschnitten auf technische, organisatorische und wirtschaftliche Aspekte bei der Realisierung von Systemen zur elektronischen Rechnungsstellung eingegangen werden.

### 5.2.1 Rechtliche Rahmenbedingungen

Im Folgenden werden die rechtlichen Rahmenbedingungen der (elektronischen) Rechnungsstellung näher beleuchtet. Wie in *Abbildung 41* angedeutet, sind hier die Europäischen Rahmenbedingungen und vor allem die deutsche Umsetzung dieser Vorgaben zu berücksichtigen.

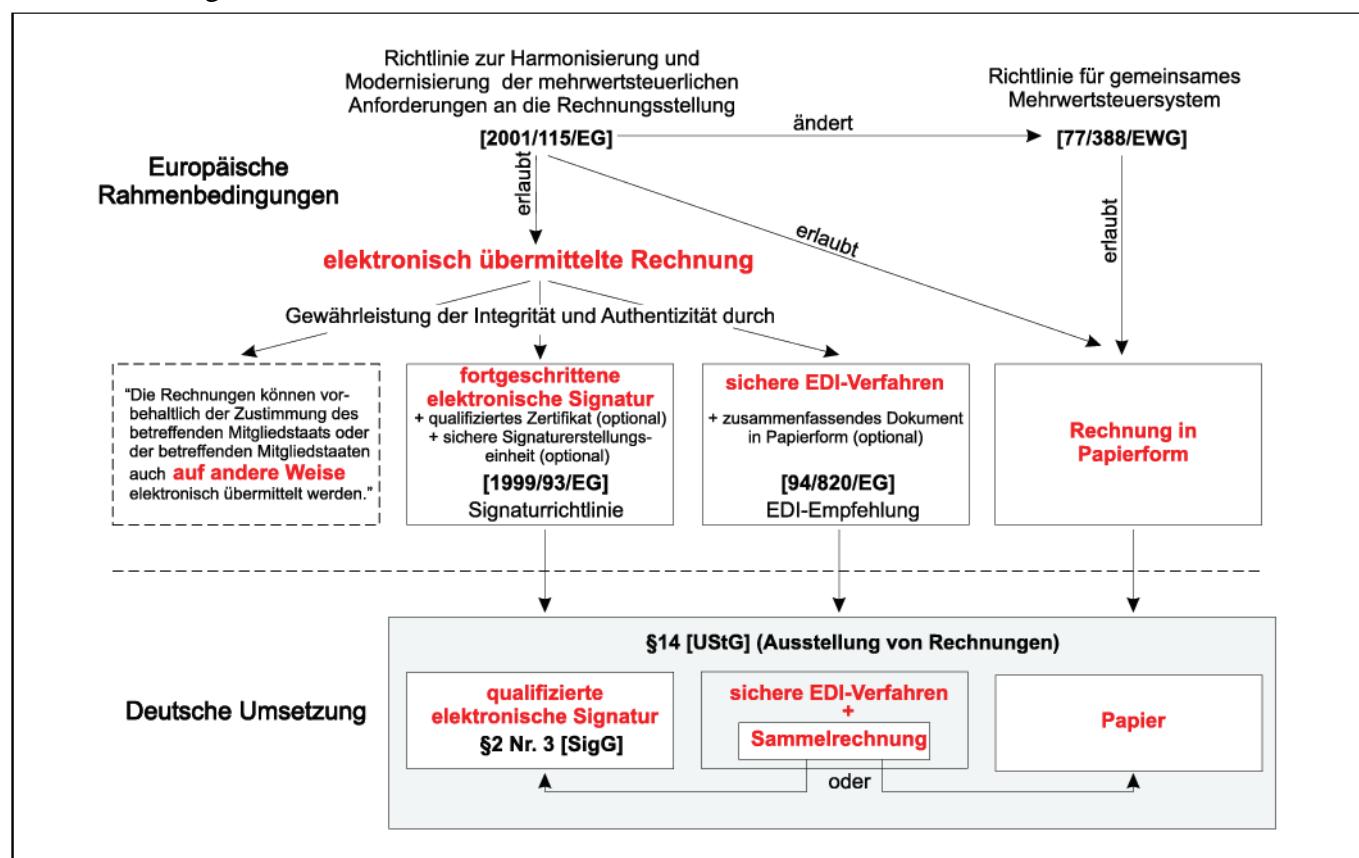


Abbildung 41: Formale Anforderungen für Rechnungen

In *Abschnitt 5.2.1.1* wird näher auf die europäische Richtlinie [2001/115/EG] eingegangen, durch die der rechtliche Rahmen für die elektronische Übermittlung von Rechnungen in den Mitgliedstaaten der Europäischen Union definiert ist. In *Abschnitt 5.2.1.2* werden die in § 14 [UStG] definierten Anforderungen an Rechnungen näher erläutert. *Abschnitt 5.2.1.3* behandelt die Rahmenbedingungen für die Aufbewahrung und die Prüfbarkeit elektronischer Rechnungen. Schließlich werden in *Abschnitt 5.2.1.4* einige weitere Aspekte der

## 5 Anwendungsbereiche

---

Rechnungsstellung beleuchtet.

### 5.2.1.1 Europäische Rechnungsrichtlinie

Den Europäischen Rahmen für die elektronische Abrechnung bildet die Richtlinie [2001/115/EG], die gemäß Artikel 5 bis zum 01.01.2004 in nationales Recht umgesetzt werden musste. Diese Richtlinie sieht in Artikel 2 eine Änderung des Artikels 22 der Richtlinie [77/388/EWG] vor, der u. a. die elektronische Abrechnung betrifft. Im geänderten Artikel 22 Absatz 3 c) dieser Richtlinie wird festgelegt, dass Rechnungen „auf Papier oder vorbehaltlich der Zustimmung des Empfängers auf elektronischem Weg übermittelt werden“ können.

Zur Anerkennung elektronisch übermittelter Rechnungen legt Artikel 22 Absatz 3 c) der Richtlinie [77/388/EWG] fest:

„Elektronisch übermittelte Rechnungen werden von den Mitgliedstaaten unter der Voraussetzung akzeptiert, dass die Echtheit der Herkunft und die Unversehrtheit des Inhalts gewährleistet werden

- entweder durch eine fortgeschrittene elektronische Signatur im Sinne des Artikels 2 Nummer 2 der Richtlinie [1999/93/EG] des Europäischen Parlaments und des Rates vom 13. Dezember 1999 über gemeinschaftliche Rahmenbedingungen für elektronische Signaturen; die Mitgliedstaaten können allerdings verlangen, dass die fortgeschrittene elektronische Signatur auf einem qualifizierten Zertifikat beruht und von einer sicheren Signaturerstellungseinheit erstellt wird (Artikel 2 Nummern 6 und 10 der genannten Richtlinie);
- oder durch elektronischen Datenaustausch (EDI) gemäß Artikel 2 der Empfehlung [94/820/EG] der Kommission vom 19. Oktober 1994 über die rechtlichen Aspekte des elektronischen Datenaustauschs, wenn in der Vereinbarung über diesen Datenaustausch der Einsatz von Verfahren vorgesehen ist, die die Echtheit der Herkunft und die Unversehrtheit der Daten gewährleisten; die Mitgliedstaaten können allerdings unter von ihnen festzulegenden Bedingungen verlangen, dass zusätzlich ein zusammenfassendes Dokument in Papierform erforderlich ist.“

Für die Definition der detaillierten Anforderungen an elektronische Rechnungen lässt die Richtlinie [2001/115/EG] den Mitgliedstaaten also beträchtlichen Gestaltungsspielraum. So existieren Umsetzungen dieser Richtlinie (vgl. [CEN-eIFG]), bei denen elektronische Rechnungen, wie in Finnland, nur durch entsprechend gesicherte *EDI*-Verfahren und deshalb möglicherweise gänzlich ohne Signaturen realisiert werden können. Auf der anderen Seite existieren EU-Mitgliedstaaten, wie beispielsweise Deutschland, Slowenien oder die Slowakei, die den Einsatz von *qualifizierten elektronischen Signaturen* vorschreiben. Viele Mitgliedstaaten wählten einen Weg, der zwischen diesen beiden Extrempositionen lag und für die elektronische Rechnungsstellung den Einsatz von *fortgeschrittenen elektronischen Signaturen*, teilweise mit weiteren Auflagen, verlangt. Beispielsweise muss für die elektronische Rechnung in Österreich [OeV583/03] eine *fortgeschritte elektronische Signatur* eingesetzt werden, die auf einem Zertifikat eines Zertifizierungsdiensteanbieters gemäß dem österreichischen Signaturgesetz [OeSigG] beruht. Vor diesem Hintergrund erscheint es fraglich, ob die Richtlinie [2001/115/EG] ihr Ziel der Harmonisierung der mehrwertsteuerlichen Anforderungen an die Rechnungstellung wirklich erreicht hat.

Grundsätzlich sind bei der Ausstellung von Rechnungen die Anforderungen des Staates zu berücksichtigen, in denen der Aussteller ansässig ist. Um nicht zwischen den spezifischen Regelungen der einzelnen EU-Mitgliedstaaten unterscheiden zu müssen, empfiehlt sich der Einsatz von *qualifizierten elektronischen Signaturen*.

## 5 Anwendungsbereiche

---

### 5.2.1.2 Anforderungen an Rechnungen gemäß § 14 UStG

Die Anforderungen an Rechnungen, die gemäß § 15 [UStG] zum Vorsteuerabzug herangezogen werden dürfen, sind in § 14 [UStG] definiert. Nach § 14 Abs. 1 [UStG] ist eine Rechnung „jedes Dokument, mit dem über eine Lieferung oder sonstige Leistung abgerechnet wird, gleichgültig, wie dieses Dokument im Geschäftsverkehr bezeichnet wird. Rechnungen sind auf Papier oder vorbehaltlich der Zustimmung des Empfängers auf elektronischem Weg zu übermitteln.“ Wie in [BMF04, Rz. 10] klar gestellt, bedarf die Zustimmung des Empfängers keiner besonderen Form. Es muss lediglich Einvernehmen darüber existieren, dass die Rechnung elektronisch übermittelt werden soll. Beispielsweise genügt es, dass die Beteiligten die Verfahrensweise praktizieren und damit stillschweigend billigen.

Die detaillierten Anforderungen an elektronisch übermittelte Rechnungen sind heute<sup>36</sup> in § 14 Abs. 3 [UStG] geregelt. Dieser Paragraph ist eng an die Richtlinie [2001/115/EG] angelehnt und folgendermaßen gefasst:

„(3) Bei einer auf elektronischem Weg übermittelten Rechnung müssen die Echtheit der Herkunft und die Unversehrtheit des Inhalts gewährleistet sein durch

1. eine *qualifizierte elektronische Signatur* oder eine qualifizierte elektronische Signatur mit Anbieter-Akkreditierung nach dem Signaturgesetz vom 16. Mai 2001 (BGBI. I S. 876), das durch Artikel 2 des Gesetzes vom 16. Mai 2001 (BGBI. I S. 876) geändert worden ist, in der jeweils geltenden Fassung, oder
2. elektronischen Datenaustausch (*EDI*) nach Artikel 2 der Empfehlung [94/820/EG] der Kommission vom 19. Oktober 1994 über die rechtlichen Aspekte des elektronischen Datenaustausches (ABl. EG Nr. L 338 S. 98), wenn in der Vereinbarung über diesen Datenaustausch der Einsatz von Verfahren vorgesehen ist, die die Echtheit der Herkunft und die Unversehrtheit der Daten gewährleisten, und zusätzlich eine zusammenfassende Rechnung auf Papier oder unter den Voraussetzungen der Nummer 1 auf elektronischem Weg übermittelt wird.“

In [BMF04, Rz. 18] wurde verdeutlicht, dass zur Erstellung der qualifizierten elektronischen Signatur alle technischen Verfahren zulässig sind, die den Anforderungen des Signaturgesetzes entsprechen. „Der Unternehmer hat die Voraussetzungen auf Anforderung nachzuweisen.“ Deshalb empfiehlt es sich, im Rahmen der Verfahrensdokumentation gemäß [GoBS, Abschnitt VIIIb) 1.] (vgl. § 145 AO und [BMF04, Rz. 12]) insbesondere auch auf die Erfüllung der Anforderungen gemäß Signaturgesetz einzugehen. Wie in Abschnitt 2.3.2 erläutert, umfasst dies auch die Existenz entsprechender Bestätigungen und Herstellererklärungen für die eingesetzten Komponenten.

Außerdem wurde in [BMF04, Rz. 23-24] klar gestellt, dass bei der Übermittlung von Rechnungen per Telefax danach unterschieden werden muss, welche Art von Faxgerät eingesetzt wird. Es handelt sich nur dann nicht um eine elektronische Übermittlung der Rechnung im Sinne von § 14 Abs. 3 [UStG], wenn die Übertragung von einem Standard-Fax zu einem Standard-Fax erfolgt. In allen anderen Fällen, in denen auf Seite des Senders oder Empfängers beispielsweise ein Fax-Server eingesetzt wird, handelt es sich um eine elektronische Übermittlung der Rechnung im Sinne von § 14 Abs. 3 [UStG] und die Echtheit der Herkunft und die Unversehrtheit des Inhalts müssen durch eine *qualifizierte elektronische Signatur* gewährleistet werden. Hierbei kann ein 2D-Barcode-Verfahren eingesetzt werden [OeKr03, OeKr04].

Außerdem sind in § 14 Abs. 4 [UStG] die Mindestangaben für Rechnungen definiert:

---

<sup>36</sup>Ein historischer Überblick zur Entwicklung des § 14 [UStG] im Hinblick auf die elektronische Übermittlung von Rechnungen findet sich in [HuTe04, Abschnitt 3.1.2].

## 5 Anwendungsbereiche

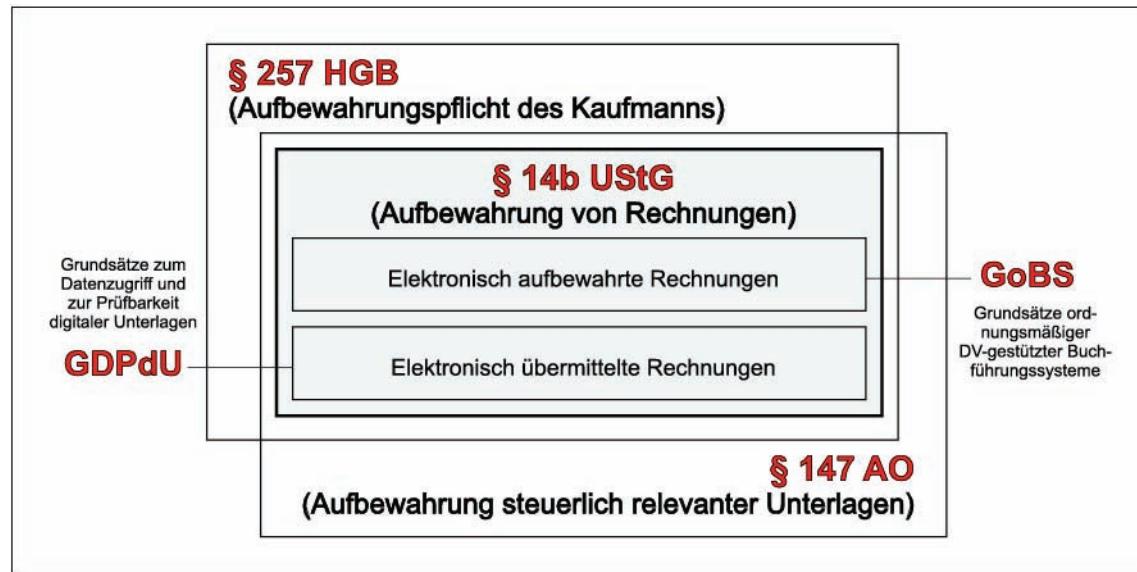
1. Name und Anschrift des Rechnungsstellers und Rechnungsempfängers,
2. Steuernummer oder Umsatzsteuer-Identifikationsnummer,
3. Ausstellungsdatum,
4. Rechnungsnummer,
5. Bezeichnung der Lieferung oder Leistung,
6. Lieferzeitpunkt, sofern nicht mit Ausstellungsdatum identisch,
7. Entgelt nach Steuersätzen,
8. Steuersatz,
9. bei einer steuerpflichtigen Werklieferung oder sonstigen Leistung im Zusammenhang mit einem Grundstück einen Hinweis auf die zweijährige Aufbewahrungspflicht für den privaten Leistungsempfänger.

Wie in § 31 Abs. 2 [UStDV] erläutert, ist es ausreichend, „wenn sich auf Grund der in die Rechnung aufgenommenen Bezeichnungen der Name und die Anschrift sowohl des leistenden Unternehmers als auch des Leistungsempfängers eindeutig feststellen lassen.“ Außerdem können nach § 31 Abs. 3 [UStDV] für die gemäß Nr. 1 und 5 notwendigen Angaben „Abkürzungen, Buchstaben, Zahlen oder Symbole verwendet werden, wenn ihre Bedeutung in der Rechnung oder in anderen Unterlagen eindeutig festgelegt ist. Die erforderlichen anderen Unterlagen müssen sowohl beim Aussteller als auch beim Empfänger der Rechnung vorhanden sein.“

Bei Rechnungen unter 100 Euro (vgl. § 33 [UStDV]) oder bei Fahrausweisen (vgl. § 34 [UStDV]) sind einige Angaben entbehrlich. Umgekehrt sind bei einigen der in § 14a [UStG] aufgeführten Sonderfälle zusätzliche Angaben notwendig.

### 5.2.1.3 Aufbewahrung und Prüfbarkeit von Rechnungen

Wie in *Abbildung 42* visualisiert, sind bei der Aufbewahrung und der Prüfbarkeit von Rechnungen verschiedene Aspekte zu berücksichtigen.



**Abbildung 42: Rahmenbedingungen für die Aufbewahrung von Rechnungen**

Neben den generellen Aufbewahrungspflichten des Kaufmanns (§ 257 [HGB]) und der Aufbewahrungspflicht für steuerlich relevante Unterlagen (§ 147 [AO]) sind für die Aufbewahrung von Rechnungen die spezifischen Regelungen des § 14b [UStG] zu

## 5 Anwendungsbereiche

---

berücksichtigen. Die Aufbewahrung elektronischer Unterlagen für steuerliche Zwecke ist in den Grundsätzen ordnungsmäßiger DV-gestützter Buchführungssysteme [GoBS] geregelt. Diese Anforderungen sind auch bei der *Aufbewahrung* von Rechnungen zu berücksichtigen. Außerdem müssen bei der elektronischen *Übermittlung* von Rechnungen die Grundsätze zum Datenzugriff und zur Prüfbarkeit digitaler Unterlagen [GDPdU] berücksichtigt werden.

### **Generelle Aufbewahrungspflichten für kaufmännische und steuerlich relevante Unterlagen**

Gemäß § 257 [HGB] ist der Kaufmann zur Aufbewahrung bestimmter Unterlagen (u.a. Handelsbriefe) verpflichtet. In ähnlicher Weise definiert § 147 [AO] die generellen Anforderungen an die Aufbewahrung von Unterlagen, die für die Besteuerung von Bedeutung sind. Dies umfasst beispielsweise

- empfangene Handels- und Geschäftsbriebe (vgl. § 257 Abs. 1 Nr. 2 [HGB] und § 147 Abs. 1 Nr. 2 [AO]), sowie
- Wiedergaben von abgesandten Handels- und Geschäftsbrieften (vgl. § 257 Abs. 1 Nr. 2 [HGB] und § 147 Abs. 1 Nr. 2 [AO])

und demnach insbesondere eingehende und ausgehende Rechnungen.

Nach § 147 Abs. 2 [AO] können solche Unterlagen „auch als Wiedergabe auf einem Bildträger oder auf anderen Datenträgern aufbewahrt werden, wenn dies den Grundsätzen ordnungsmäßiger Buchführung entspricht und sichergestellt ist, dass die Wiedergabe oder die Daten

1. mit den empfangenen Handels- oder Geschäftsbrieften und den Buchungsbelegen bildlich und mit den anderen Unterlagen inhaltlich übereinstimmen, wenn sie lesbar gemacht werden,
2. während der Dauer der Aufbewahrungsfrist jederzeit verfügbar sind, unverzüglich lesbar gemacht und maschinell ausgewertet werden können.“

### **Spezifische Regelungen für die Aufbewahrung von Rechnungen gemäß § 14b [UStG]**

Wie in § 14b Abs. 1 [UStG] ausgeführt, „muss der Unternehmer ein Doppel der Rechnung, die er selbst oder ein Dritter in seinem Namen und für seine Rechnung ausgestellt hat, sowie alle Rechnungen, die er erhalten hat, zehn Jahre aufbewahren.“

Allerdings läuft diese Aufbewahrungsfrist nicht ab, „soweit und solange die Unterlagen für Steuern von Bedeutung sind, für welche die Festsetzungsfrist noch nicht abgelaufen ist“ (vgl. § 147 Abs. 3 Satz 3 [AO] und [BMF05, Rz. 68]).

Gemäß [UStR, Abschnitt 190b, Abs. 6] müssen die Rechnungen über den gesamten Aufbewahrungszeitraum lesbar sein. Das bedeutet, dass je nach dem eingesetzten Datenträger Erhaltungsmaßnahmen notwendig werden können. So müssen elektronische Rechnungen auf zuverlässigen Speichermedien aufbewahrt und auf Thermopapier ausgedruckte Rechnungen durch einen nochmaligen Kopiervorgang auf alterungsbeständigem Papier konserviert werden.

Nach § 14b Abs. 2 [UStG] hat die Aufbewahrung der Rechnungen regelmäßig im Inland zu erfolgen. Sofern die zuständigen Finanzbehörden elektronisch aufbewahrte Rechnungen über Online-Zugriff unverzüglich einsehen, herunterladen und verwenden können, dürfen die Daten aber auch im übrigen Gemeinschaftsgebiet aufbewahrt werden (vgl. § 14 Abs. 4 [UStG]).

Außerdem wird in [BMF04, Rz. 70] klar gestellt, dass sich die Aufbewahrungspflicht bei einer elektronisch übermittelten Rechnung auch auf die *qualifizierte elektronische Signatur* erstreckt.

## 5 Anwendungsbereiche

---

### **Anforderungen der GoBS bei der elektronischen Aufbewahrung von Rechnungen**

Weitere Details bezüglich der elektronischen Aufbewahrung von Unterlagen sind in Abschnitt VIII der Grundsätze ordnungsmäßiger DV-gestützter Buchführungssysteme [GoBS] geregelt. Hierbei wird grundsätzlich zwischen originär in Papierform verkörperten und originär digitalen Unterlagen unterschieden:

#### 1. Originär in Papierform verkörperte Unterlagen

In diesem Fall wird das Schriftgut eingescannt und die bildliche Wiedergabe, beispielsweise in Form einer *TIFF*- oder *PDF*-Datei, elektronisch gespeichert. Wie in [GoBS, Abschnitt VIIIb) 1.] erläutert, bedarf der Scannvorgang „einer Organisationsanweisung darüber,

- wer scannen darf,
- zu welchem Zeitpunkt gescannt wird,
- welches Schriftgut gescannt wird,
- ob eine bildliche oder inhaltliche Übereinstimmung mit dem Original erforderlich ist (§ 147 Abs. 1 Nr. 2 oder 3 [AO]),
- wie die Qualitätskontrolle auf Lesbarkeit und Vollständigkeit und
- wie die Protokollierung von Fehlern zu erfolgen hat.“

Außerdem muss die eingesetzte Hard- und Software sicher stellen, „dass das Scannergebnis unveränderbar ist.“ Dies kann unter Verwendung von *digitalen Signaturen* oder *Zeitstempeln* erfolgen. Anders als beim Rechnungswesen in der Sozialversicherung (vgl. Abschnitt 5.1.4) ist der Einsatz von *qualifizierten elektronischen Signaturen* zur Dokumentation der Übereinstimmung von Original und Wiedergabe hier aber nicht zwingend notwendig. Sind die Anforderungen der [GoBS] erfüllt, so können die Originale der Rechnungen vernichtet werden (vgl. [BMF04, RZ. 72] und [UStR, Abschnitt 255 Abs. 2]).

#### 2. Originär digitale Unterlagen

„Originär digitale Dokumente werden durch Übertragung der Inhalts- und Formatierungsdaten auf einem digitalen Datenträger archiviert. Bei originär digitalen Dokumenten muss hard- und softwaremäßig sichergestellt sein, dass während des Übertragungsvorgangs auf das Speichermedium eine Bearbeitung nicht möglich ist. Die Indexierung hat wie bei gescannten Dokumenten zu erfolgen.“

In beiden Fällen benötigt man zur Erfüllung der Anforderungen der [GoBS] im Hinblick auf die elektronische Aufbewahrung von steuerlich relevanten Unterlagen nicht unbedingt *digitale Signaturen*, sondern lediglich ein so genanntes „revisionssicheres Archiv“, das die in [KaRo97] näher erläuterten Grundsätze der elektronischen Archivierung berücksichtigt.

### **Anforderungen der GDPdU für die Prüfbarkeit elektronisch übermittelter Rechnungen**

Im Schreiben des Bundesfinanzministeriums [GDPdU] zu den Grundsätzen der Prüfbarkeit digitaler Unterlagen sind eine Reihe von Anforderungen definiert, die insbesondere vom Empfänger einer elektronisch übermittelten Rechnung beachtet werden müssen. Neben den oben erläuterten Anforderungen der [GoBS] für die elektronische Aufbewahrung von Rechnungen sehen die [GDPdU] in Abschnitt II. 1. eine Reihe von speziellen Anforderungen für die Behandlung elektronischer Rechnungen und Signaturen vor:

#### 1. Pflicht zur Prüfung der Signatur

Vor einer weiteren Verarbeitung der elektronischen Abrechnung (z.B. Verbuchen) muss die *qualifizierte elektronische Signatur* im Hinblick auf die *Integrität* der Daten und die Signaturberechtigung geprüft werden. Das Ergebnis der Prüfung ist zu dokumentieren.

## 5 Anwendungsbereiche

---

### 2. Speicherung auf einem unveränderbaren Datenträger

Die elektronische Abrechnung muss auf einem Datenträger gespeichert werden, der Änderungen nicht mehr zulässt. Bei einer temporären Speicherung auf einem änderbaren Datenträger muss das DV-System sicherstellen, dass Änderungen nicht möglich sind.

### 3. Speicherung jeder Version bei Konvertierung

Bei der Umwandlung (Konvertierung) der elektronischen Abrechnung in ein unternehmenseigenes Format (sog. Inhouse-Format) müssen beide Versionen archiviert und nach den [GoBS] mit demselben Index verwaltet werden. Eine konvertierte Version muss als solche gekennzeichnet werden.

### 4. Aufbewahrung des Signaturprüfsschlüssels

Der Signaturprüfsschlüssel muss aufbewahrt werden.

### 5. Aufbewahrung von Verschlüsselungsschlüsseln

Bei Einsatz von Verschlüsselungstechniken müssen die verschlüsselte und die entschlüsselte Abrechnung sowie der Schlüssel zur Entschlüsselung der elektronischen Abrechnung aufbewahrt werden.

### 6. Protokollierung der Verarbeitungsschritte

Der Eingang der elektronischen Abrechnung, ihre Archivierung und ggf. Konvertierung sowie die weiteren Verarbeitungsschritte müssen protokolliert werden.

### 7. Erfüllung der Anforderungen der GoBS

Die Übertragungs-, Archivierungs- und Konvertierungssysteme müssen den Anforderungen der GoBS [GoBS] entsprechen. Hierbei sind insbesondere die Dokumentation, das interne Kontrollsystem, das Sicherungskonzept und die Aufbewahrung zu berücksichtigen.

Außerdem wird (vgl. [GDPdU, S. 6, oben]) verlangt, dass „das qualifizierte Zertifikat des Empfängers<sup>37</sup> aufbewahrt wird.“

Besondere Bedeutung kommt der im ersten Punkt geforderten Prüfungspflicht zu. Dies ist darin begründet, dass ein elektronisches Dokument, das nicht mit einer gültigen qualifizierten elektronischen Signatur versehen ist, insbesondere nicht die Anforderungen des § 14 Abs. 3 [UStG] erfüllt und deshalb eine solche Abrechnung *nicht* zum Vorsteuerabzug gemäß § 15 [UStG] herangezogen werden darf.

Wie in *Abschnitt 4.2* erläutert, ist die Prüfung einer *qualifizierten elektronischen Signatur* keine triviale Aufgabe. Neben der mathematischen Gültigkeit der Signatur muss auch der Zertifikatspfad bis hin zu einer vertrauenswürdigen Wurzel geprüft werden. Dabei werden insbesondere die folgenden Punkte geprüft:

- Mathematische Gültigkeit der Signaturen,
- Gültigkeit der Zertifikate gemäß Gültigkeitsmodell,
- Korrektheit des Verwendungszwecks der Zertifikate.

Beim letzten Punkt ist für das Zertifikat des Rechnungsstellers (oder des beauftragten Dritten) zu prüfen, ob nicht etwa der Verwendungszweck nach Art und Umfang eingeschränkt ist und ob es sich jeweils um *qualifizierte Zertifikate* handelt.

---

<sup>37</sup>Da der Empfänger der elektronisch übermittelten Rechnung *kein* qualifiziertes Zertifikat benötigt, bleibt der Zweck dieser Forderung unklar. Vermutlich handelt es sich hier um ein Redaktionsversehen, wobei das Zertifikat des Zertifizierungsdiensteanbieters gemeint sein könnte.

## 5 Anwendungsbereiche

---

Die Forderung der [GDPdU], dass man das Ergebnis der Signaturprüfung dokumentieren muss, kann<sup>38</sup> durch einen elektronischen Prüfbericht realisiert werden.

Ein solcher Prüfbericht umfasst mindestens die Information

1. dass geprüft wurde und
2. das Ergebnis der Prüfung.

Da der Prüfbericht vor allem als Entscheidungsgrundlage für einen möglichen Vorsteuerabzug dient, genügt streng genommen ein Prüfungsergebnis der Form „gültig“ oder „nicht gültig“. Es scheint also nicht unbedingt nötig, dass alle Teilergebnisse der Prüfung (vgl. [CCES-API, Kapitel 4.2.1.3]) einzeln im Prüfbericht ausgewiesen werden. Allerdings muss in der Verfahrensdokumentation gemäß [GoBS, Abschnitt VI] festgelegt und nachprüfbar sein, wer die Prüfung nach welchem Verfahren ausgeführt hat.

Außerdem ist es eine wesentliche Eigenschaft der *qualifizierten elektronischen Signatur* gemäß dem deutschen Signaturgesetz, dass sich ihre Gültigkeit im Laufe der Zeit *nicht* mehr ändert. Sofern sie zu irgend einem Zeitpunkt nach Ihrer Erstellung als gültig geprüft werden konnte, bleibt sie für alle Zeiten gültig. Deshalb kann ein Steuerprüfer auch zu einem späteren<sup>39</sup> Zeitpunkt die Signaturprüfung selbst vornehmen und würde zum gleichen Ergebnis kommen.

Soll ein elektronischer Prüfbericht erstellt werden, so muss dieser nicht zwingend vom Rechnungsempfänger selbst erstellt werden. Gemäß [BMF04, Rz. 30] ist es vielmehr möglich, die Prüfungsschritte auch von einem beauftragten Dritten durchführen zu lassen. Wie durch das BMF-Schreiben [BMF05] klar gestellt wurde, darf ein spezialisierter Dienstleister im Auftrag des Rechnungsstellers eine Signatur erzeugen und im darauf folgenden Prozessschritt diese Signatur im Auftrag des Rechnungsempfängers prüfen, sofern durch organisatorische Maßnahmen sicher gestellt ist, „dass die Verantwortung für Signaturerstellung und Verifikation administrativ in unterschiedlichen Händen liegt“.

Für die Beauftragung des Dritten zur Prüfung der Signatur an der Rechnung ist keine besondere Form notwendig. Die Beauftragung des Dritten könnte also zusammen mit der formfreien Einwilligung des Rechnungsempfängers zur elektronischen Übermittlung der Rechnung erfolgen. Soweit die Tätigkeit des Dritten die Verarbeitung personenbezogener Daten umfasst, ist allerdings die Einhaltung der Vorschriften des [BDSG] sicherzustellen.

### 5.2.1.4 Weitere ausgewählte Aspekte der Rechnungsstellung

Neben den oben erläuterten Rahmenbedingungen sind bei der elektronischen Übermittlung von Rechnungen weitere Aspekte zu berücksichtigen, auf die im Folgenden näher eingegangen werden soll.

#### Sammelrechnung

Grundsätzlich muss nicht für jede einzelne Lieferung von Gegenständen oder Dienstleistung

---

<sup>38</sup>Erfolgt eine manuelle Signaturprüfung, so wäre auch eine konventionelle, nicht-elektronische Dokumentation des Prüfungsergebnisses denkbar, sofern die Grundsätze ordnungsmäßiger Buchführung erfüllt sind.

<sup>39</sup>Bei angezeigten Zertifizierungsdiensteanbietern ist die Prüfbarkeit der Zertifikate bis zu 5 Jahre nach Ablauf der Gültigkeit gewährleistet; bei akkreditierten Zertifizierungsdiensteanbietern verlängert sich dieser Zeitraum auf 30 Jahre. Zusätzlich sind die in *Abschnitt 4.6* diskutierten Aspekte der langfristigen Archivierung von signierten Daten zu beachten.

## 5 Anwendungsbereiche

---

eine einzelne Rechnung gestellt werden. Wie bereits in [BMF92] und in [BMF04, Rz. 21] klargestellt, kann periodisch (täglich, wöchentlich, monatlich etc.) eine Sammelrechnung erstellt werden.

Während die laufenden Abrechnungsdaten in einem gesicherten *EDI*-Verfahren übermittelt werden können, muss die Sammelrechnung gemäß § 14 Abs. 3 Nr. 2 [UStG] entweder auf Papier ausgestellt oder mit einer *qualifizierten elektronischen Signatur* versehen sein.

Hierbei muss über den elektronischen Datenaustausch eine Vereinbarung nach Artikel 2 der Empfehlung [94/820/EG] bestehen, in der der Einsatz von Verfahren vorgesehen ist, die die *Authentizität* und die *Integrität* der übermittelten Daten gewährleisten (vgl. [BMF04, Rz. 20]).

Die Sammelrechnung muss die Summen der Umsätze und Steuerbeträge, sowie die in § 14 Abs. 4 und § 14a [UStG] aufgeführten Merkmale enthalten, wobei jedoch gemäß § 31 Abs. 1 [UStDV] auf ergänzende Dokumente verwiesen werden kann (vgl. [BMF04, Rz. 21]).

Bemerkenswert ist außerdem, dass die Voraussetzung für den Vorsteuerabzug durch den Leistungsempfänger der Besitz einer Rechnung ist, die *alle* Anforderungen der §§ 14 und 14a [UStG] erfüllt (vgl. [BMF04, Rz. 87]). Wie in [BMF04, Rz. 88 und 90] erläutert, muss der Rechnungsempfänger die Angaben in der Sammelrechnung auf Vollständigkeit und Richtigkeit prüfen. Insbesondere darf der Vorsteuerabzug also erst nach dem Empfang und der Prüfung der Sammelrechnung vorgenommen werden.

### Abrechnung durch Gutschrift

Gemäß § 14 Abs. 2 Satz 3 [UStG] ist eine Gutschrift eine Rechnung, die vom Leistungsempfänger ausgestellt wird. Die am Leistungsaustausch Beteiligten können vor der Abrechnung formfrei<sup>40</sup> vereinbaren, dass mittels Gutschriften abgerechnet werden soll.

Nach § 14 Abs. 2 Satz 4 [UStG] ist die Voraussetzung für die Wirksamkeit einer Gutschrift, dass die Gutschrift dem leistenden Unternehmer übermittelt worden ist und dieser dem ihm zugeleiteten Dokument nicht widerspricht (vgl. [BMF04, Rz. 7 und 8]).

Gemäß § 14 Abs. 2 Satz 5 [UStG] kann der Leistungsempfänger mit der Ausstellung einer Gutschrift auch einen Dritten beauftragen, der im Namen und für Rechnung des Leistungsempfängers abrechnet.

Wichtig ist, dass auch bei der Abrechnung durch Gutschrift entweder ein Papier-basiertes oder mit einer *qualifizierten elektronischen Signatur* versehenes Dokument übermittelt werden muss. Insofern bestehen keine Unterschiede zwischen der konventionellen Rechnungsstellung und der Abrechnung im Gutschriftenverfahren (vgl. [BMF04, Rz. 26]).

### Abrechnung durch Dritte

Gemäß § 14 Abs. 2 Satz 5 [UStG] kann eine Rechnung im Namen und für Rechnung des leistenden Unternehmers oder – bei der Abrechnung durch Gutschrift – des Leistungsempfängers von einem Dritten ausgestellt werden. Wie in [BMF04, Rz. 27] klargestellt wurde, gilt dies auch für elektronisch übermittelte Rechnungen und Gutschriften. Allerdings wird in [BMF04, Rz. 3] verdeutlicht, dass der Leistungsempfänger nicht als Dritter fungieren und somit Rechnungen an sich selbst erstellen darf.

Um die Anforderungen des § 14 Abs. 3 [UStG] zu erfüllen, werden hierbei vom Ersteller des Abrechnungsdokuments (Rechnung oder Gutschrift) eine oder mehrere natürliche Personen

---

<sup>40</sup>Wie in [BMF04, Rz. 6] klar gestellt wurde, kann sich die Vereinbarung zur Abrechnung mit Gutschrift aus Verträgen oder sonstigen Geschäftsunterlagen ergeben oder beispielsweise auch mündlich getroffen werden.

## 5 Anwendungsbereiche

---

beim Dritten bevollmächtigt, die Abrechnungsdokumente mit einer *qualifizierten elektronischen Signatur* zu versehen (vgl. § 167 ff. [BGB] und [BMF04, Rz. 28]).

Wie in [BMF04, Rz. 29] klar gestellt wurde, gelten die Anforderungen des § 14 Abs. 3 [UStG] nicht für die Übermittlung der Daten vom leistenden Unternehmer oder vom Leistungsempfänger zum Zweck der Rechnungserstellung an den Dritten. Für die Übertragung der Daten zum Dritten muss also insbesondere keine *qualifizierte elektronische Signatur* verwendet werden. Allerdings ist der Dritte nach § 93 ff. [AO] verpflichtet, dem Finanzamt die Prüfung des Verfahrens durch Erteilung von Auskünften und Vorlage von Unterlagen in seinen Räumen zu gestatten.

Außerdem wurde in [BMF04, Rz. 29] erläutert, dass der Empfänger einer elektronisch übermittelten Rechnung die gemäß [IGDPdU] notwendige Prüfung der Rechnung auch durch einen Dritten erledigen lassen kann. Wie oben erläutert, darf ein Dritter im Auftrag des leistenden Unternehmers eine Rechnung erstellen und die so erstellte Signatur im Auftrag der Rechnungsempfänger prüfen, sofern die Systeme für die Erstellung und Prüfung administrativ in unterschiedlichen Händen liegen (vgl. [BMF05]).

### 5.2.2 Abrechnungsszenarien

Wie in *Abschnitt 5.2.1* ausführlich erläutert, sieht der Gesetzgeber verschiedene Optionen vor, die zu unterschiedlichen Abrechnungsszenarien führen. Im Einzelnen existieren folgende Wahlmöglichkeiten:

1. Papier oder elektronische Übermittlung (vgl. *Abschnitt 5.2.1.2*),
2. Einzel- oder Sammelrechnung (vgl. *Abschnitt 5.2.1.4*),
3. Rechnung oder Gutschrift (vgl. *Abschnitt 5.2.1.4*),
4. Abrechnung selbst oder durch Dritte (vgl. *Abschnitt 5.2.1.4*).

Neben diesen direkt aus den rechtlichen Rahmenbedingungen abgeleiteten Merkmalen kann bei der elektronischen Übermittlung außerdem<sup>41</sup> danach unterschieden werden, ob die Rechnungsdaten in strukturierter Form, z.B. als XML- oder EDIFACT-Daten, oder aber als Bilddaten, z.B. im PDF- oder TIFF-Format, übermittelt werden. Diese Unterscheidung ist in der Praxis wichtig, da strukturiert übermittelte Daten automatisiert und deshalb meist kostengünstiger weiter verarbeitet werden können.

Somit erhält man durch die Kombination dieser unabhängigen Merkmale insgesamt  $(3 \cdot 2 \cdot 2 \cdot 2) = 24$  verschiedene Abrechnungsszenarien. Von diesen generell möglichen Szenarien werden im Folgenden die in der Praxis besonders wichtigen Klassen näher beleuchtet:

- Papiergebundene Rechnung (vgl. *Abschnitt 5.2.2.1*),
- Elektronische Rechnung in strukturiertem Datenformat (vgl. *Abschnitt 5.2.2.2*),
- Elektronische Rechnung im Bilddatenformat (vgl. *Abschnitt 5.2.2.3*),
- EDI mit papiergebundener Sammelrechnung (vgl. *Abschnitt 5.2.2.4*),
- Abrechnung mittels Gutschrift (vgl. *Abschnitt 5.2.2.5*),
- Abrechnung über Konsolidator (vgl. *Abschnitt 5.2.2.6*).

#### 5.2.2.1 Papiergebundene Rechnung

Bei der Rechnungsstellung ist der Ursprung der Rechnungsdaten typischerweise ein *Enterprise-Ressource-Planning (ERP)*-System. Nach dem Ausdrucken der Rechnung und der

---

<sup>41</sup>Bei einer in Papierform übermittelten Rechnung handelt es sich in diesem Sinne immer um unstrukturierte Daten, die nicht ohne weiteres automatisiert weiterverarbeitet werden können.

## 5 Anwendungsbereiche

Übermittlung der Rechnung auf dem Postweg muss der Empfänger der Rechnung die Rechnungsdaten wieder erfassen und in sein *ERP*-System einspeisen.

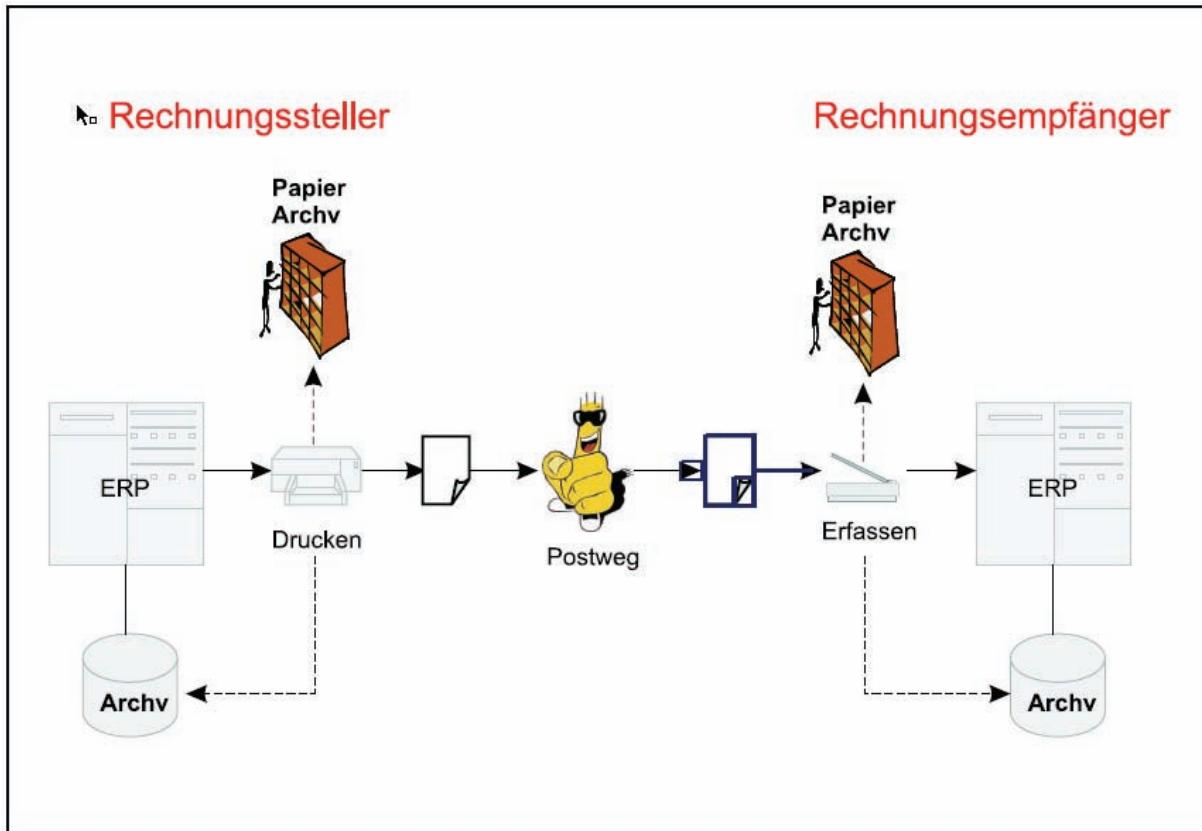


Abbildung 43: Papiergebundene Rechnung

Auf Grund der Aufbewahrungspflichten (vgl. *Abschnitt 5.2.1.3*) muss der Rechnungssteller eine Kopie der Rechnung aufbewahren. Gemäß [GDPdU, Abschnitt III 1] müssen originär digitale Unterlagen regelmäßig<sup>42</sup> auf maschinell verwertbaren Datenträgern archiviert werden. Deshalb bietet es sich an, dass der Rechnungssteller das Doppel der Rechnung nicht in Papierform, sondern auch elektronisch archiviert. Der Empfänger der Rechnung erfassst die Rechnungsdaten manuell und archiviert die Eingangsrechnung in Papierform oder er überführt die Papierrechnung unter Einsatz von spezialisierten Systemen zur Posteingangsbearbeitung in mehr oder weniger manueller Art und Weise in eine elektronische Form. Hierbei wird die Rechnung gescannt, mit *OCR*-Techniken der Text extrahiert und der erkannte Text durch Formularanalyse-Techniken auf die gemäß § 14 Abs. 4 [UStG] notwendigen Datenelemente einer Rechnung abgebildet. Hierbei ist jedoch eine vollständige Automatisierung in der Praxis nicht zu erreichen, so dass die Erfassung von papiergebundenen Eingangsrechnungen oft mit erheblichen Prozesskosten verbunden ist.

### 5.2.2.2 Elektronische Rechnung in strukturiertem Datenformat

In diesem Fall werden die Rechnungsdaten vor dem Versand nicht gedruckt, sondern durch einen *EDI*-Konverter vom Inhouse-Format in ein standardisiertes Datenformat für den

<sup>42</sup>Ausgenommen hiervon sind Unterlagen, wie z.B. Textdokumente, die nicht zur Weiterverarbeitung in einem DV-gestützten Buchführungssystem geeignet sind.

## 5 Anwendungsbereiche

Austausch von Rechnungsdaten, z.B. auf Basis von *XML* [openTRANS] oder *EDIFACT* [INVOIC], konvertiert. Die Übermittlung der strukturierten Rechnungsdaten erfolgt elektronisch.

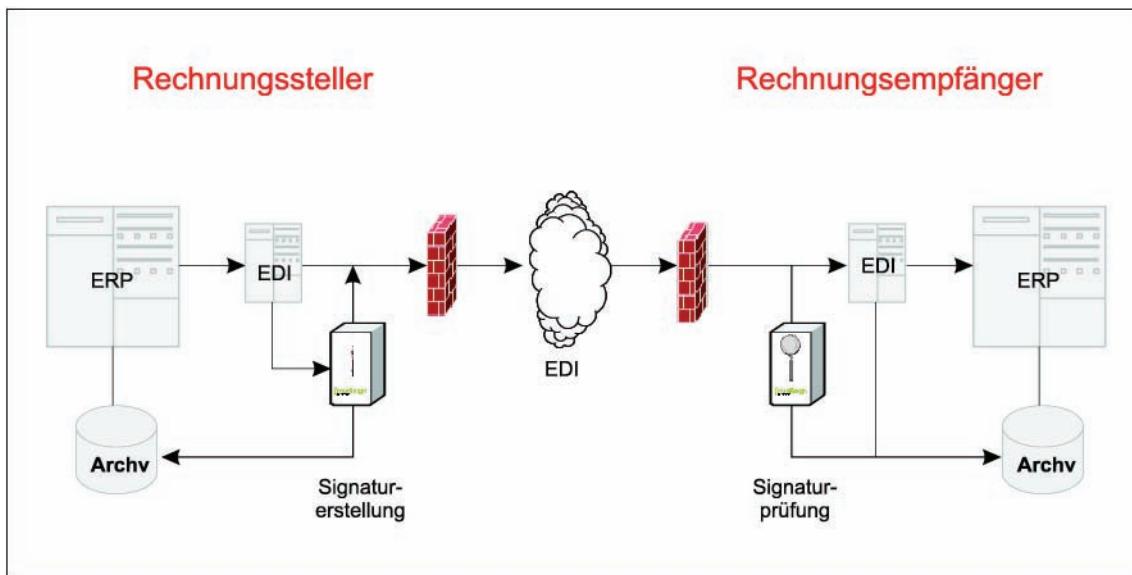


Abbildung 44: EDI-Rechnung mit Signatur

Durch Einsatz der *qualifizierten elektronischen Signatur* kann nun komplett auf kostenintensive papiergebundene Prozesse verzichtet werden. Hierbei verwendet der Rechnungssteller ein Massensignatur-System (vgl. Abschnitt 4.4 und [HuKn03]), mit dem er die Rechnungen in automatisierter Art und Weise signiert. Auch der Rechnungsempfänger kann die empfangenen Signaturen automatisiert prüfen und insbesondere auf die aufwändige Erfassung oder die manuelle Prüfung der (Sammel-) Rechnungen verzichten. Nach erfolgreicher Prüfung der qualifizierten elektronischen Signatur kann automatisch die Konvertierung in das Inhouse-Format, die Verbuchung und schließlich Archivierung der Rechnung erfolgen.

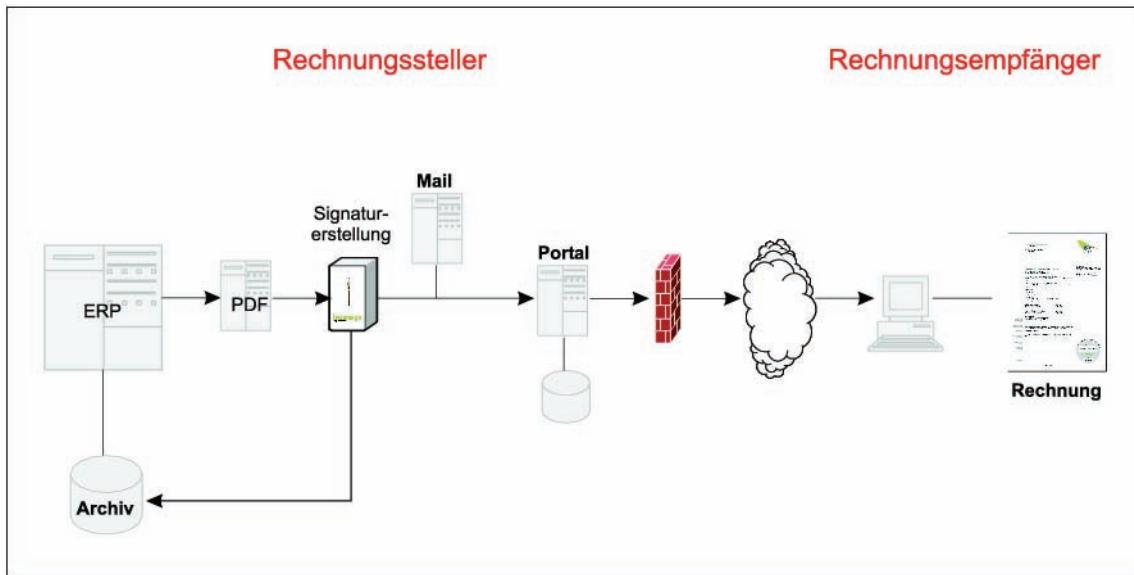
Durch den sehr hohen Automatisierungsgrad bei allen beteiligten Parteien handelt es sich hierbei sehr oft um die (systemweit betrachtet) kostengünstigste Art der Rechnungsstellung.

### 5.2.2.3 Elektronische Rechnung im Bilddatenformat

Ist der Rechnungsempfänger nicht auf die elektronische Weiterverarbeitung von empfangenen Rechnungen eingestellt, so ist die Verwendung von Dokumentenformaten zu bevorzugen, die vor allem eine bildliche Darstellung der Rechnung mit Standard-Viewer-Komponenten erlauben. Die Übermittlung der Rechnung erfolgt in diesem Fall per E-Mail oder durch Download in einem Web-Portal.

## 5 Anwendungsbereiche

---



**Abbildung 45: Elektronische Rechnung im Bilddatenformat**

Da der kostenlose Adobe Reader inzwischen praktisch auf jedem System verfügbar ist und das *PDF*-Format durch die etwaige Kompression des Bildmaterials für die effiziente Übertragung von Dokumenten im Internet gut geeignet ist, bietet sich die Nutzung dieses Formates für die Übermittlung von Rechnungsdokumenten an. Darüber hinaus kann in diesem Fall auch die Signatur in standardisierter Art und Weise in das Dokument integriert werden, so dass keine zusätzliche Signatur-Datei übertragen und bei der Prüfung erst manuell mit der Rechnungsdatei verknüpft werden muss. Außerdem kann die Signatur direkt im kostenlosen Adobe Reader geprüft werden. Die Dokumentation und Archivierung des Prüfungsergebnisses gemäß [GDPdU] muss in diesem Fall aber anderweitig realisiert werden.

Um den Rechnungsempfänger in dieser Hinsicht zu entlasten, kann auch ein Prüfbericht in die zu übertragende *PDF*-Datei integriert werden. In diesem Fall führt ein Dritter zusätzlich – im Auftrag des Rechnungsempfängers – die Prüfung der Signatur durch und übermittelt diesen Prüfbericht zusammen mit der Rechnung an den Empfänger. Da das *PDF*-Format verschiedene, voneinander getrennte Revisionen eines Dokumentes verwalten kann und insbesondere die Extraktion der signierten Version eines Dokumentes möglich ist, kann der Prüfbericht auch direkt an die signierte *PDF*-Datei angehängt werden.

## 5 Anwendungsbereiche

---

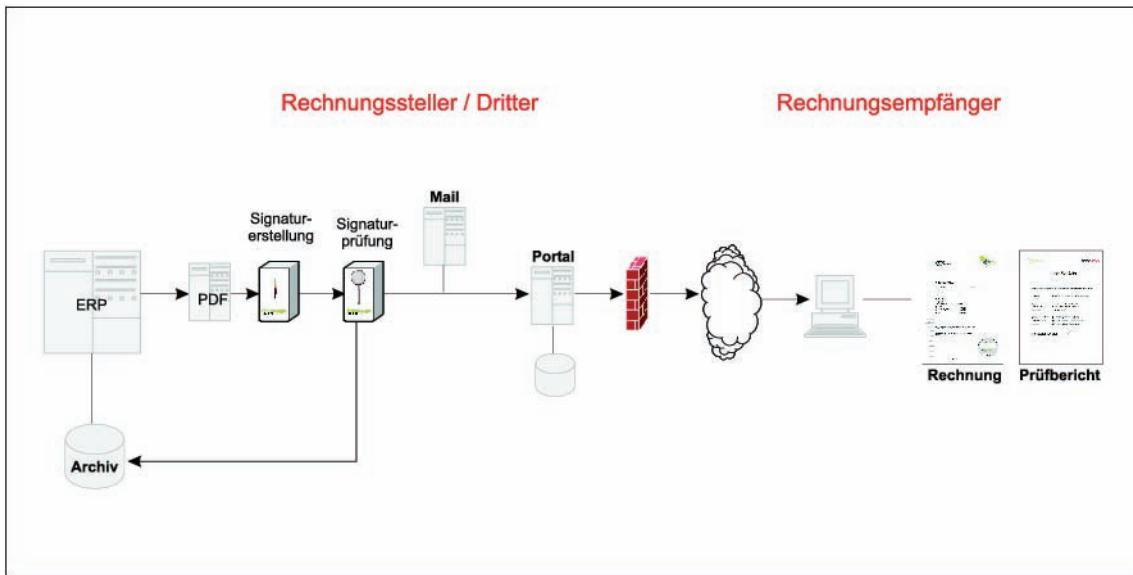


Abbildung 46: PDF-Rechnung mit Signatur und Prüfbericht

Wie in [BMF05] klar gestellt, muss in diesem Fall dafür gesorgt werden, dass das System zur Erstellung der Signatur und das System zur Prüfung der Signatur administrativ in unterschiedlichen Händen liegt.

### 5.2.2.4 EDI mit papiergebundener Sammelrechnung

In diesem Fall werden die Rechnungsdaten wieder, wie beim Szenario in Abschnitt 5.2.2.2, strukturiert elektronisch ausgetauscht. Allerdings werden diese Daten nicht signiert, sondern man erstellt zusätzlich periodisch eine papiergebundene Sammelrechnung, die per Post zugestellt wird. Während die strukturierten EDI-Daten automatisiert in das ERP-System des Empfängers übernommen werden können, ist die eigentliche Rechnung im umsatzsteuerrechtlichen Sinne, die die Anforderungen des § 14 [UStG] erfüllt und deshalb zum Vorsteuerabzug gemäß § 15 [UStG] herangezogen werden darf, jedoch die Sammelrechnung. Deshalb darf der Vorsteuerabzug auch erst nach dem Empfang und der Prüfung der Sammelrechnung erfolgen. Die Sammelrechnung ist entweder in Papierform oder unter Berücksichtigung der Vorgaben der [GoBS] elektronisch zu archivieren. Da der Prozessablauf im Vergleich zur vollständig elektronischen Abwicklung vergleichsweise aufwändig ist, ist die zusätzliche Nutzung von papiergebundenen Sammelrechnungen häufig nicht sinnvoll. Da die vollständig elektronische Abwicklung erst seit Anfang 2002 möglich ist, ist diese Variante der Abrechnung aber zur Zeit, insbesondere im Handel, noch weit verbreitet.

## 5 Anwendungsbereiche

---

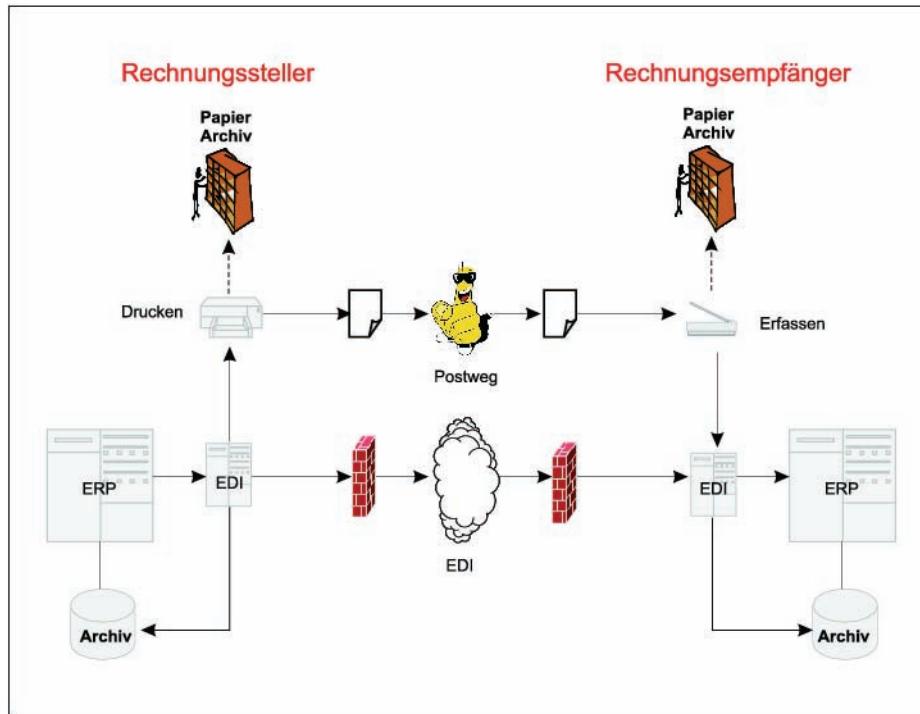


Abbildung 47: EDI mit papiergebundener Sammelrechnung

### 5.2.2.5 Abrechnung mittels Gutschrift

In diesem Fall erstellt der Leistungsempfänger, wie in Abbildung 48 dargestellt, das steuerlich relevante Abrechnungsdokument – die Gutschrift.

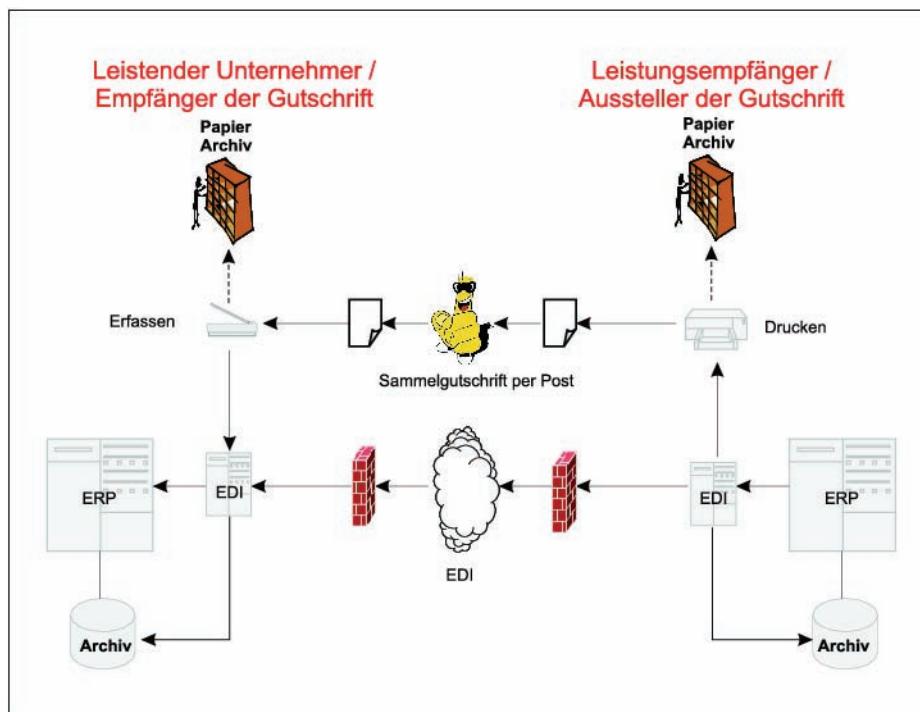


Abbildung 48: EDI mit papiergebundener Sammelgutschrift

## 5 Anwendungsbereiche

Abgesehen davon, dass der leistende Unternehmer gemäß § 14 Abs. 2 Satz 4 [UStG] der empfangenen Gutschrift widersprechen kann, so dass diese ihre steuerliche Wirkung als Rechnung verliert, gelten alle sonstigen Vorschriften für Rechnungen und Gutschriften gleichermaßen. Deshalb muss insbesondere auch eine Gutschrift in Papierform verschickt oder bei elektronischer Übermittlung mit einer qualifizierten elektronischen Signatur versehen werden. Außerdem kann auch in diesem Fall der elektronische Austausch strukturierter Daten in Verbindung mit papiergebundenen Sammelgutschriften erfolgen. Diese Variante der Abrechnung ist derzeit noch relativ häufig in der Automobil-Industrie anzutreffen.

### 5.2.2.6 Abrechnung über Konsolidator

In diesem letzten Fall erfolgt die Übermittlung der Rechnungen nicht direkt vom leistenden Unternehmer zum Empfänger der Lieferung oder Leistung, sondern über einen so genannten Konsolidator, der hier als Mittler auftritt und neben der Übermittlung, Signatur und Prüfung der Rechnung möglicherweise noch weitere Aufgaben, beispielsweise im Umfeld der Bezahlung oder Archivierung der Rechnungen, übernimmt.

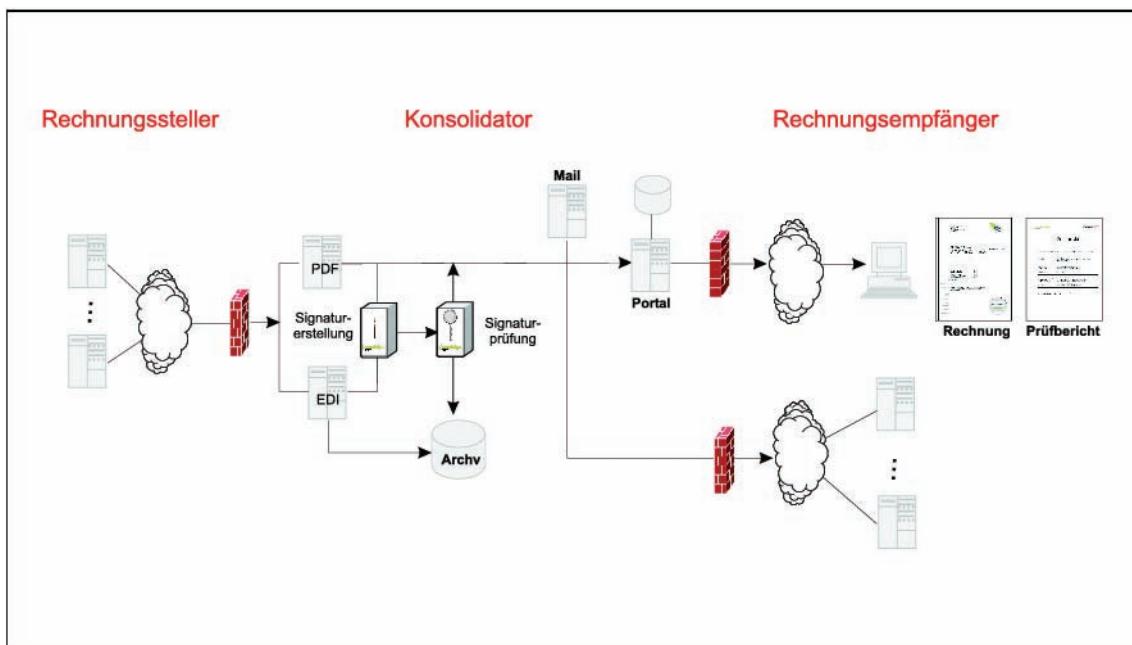


Abbildung 49: Abrechnung über Konsolidator

Häufig werden die Rechnungen im für den jeweiligen Rechnungsempfänger geeigneten Format erzeugt, signiert (vgl. *Abschnitt 4.3*) und über verschiedene Wege übermittelt. Ungeachtet der Implementierungsdetails sind in diesem Fall die in *Abschnitt 5.2.1.4* erläuterten Aspekte der Rechnungsstellung durch Dritte zu berücksichtigen.

### 5.2.3 Wirtschaftlichkeitsbetrachtung

Der elektronischen Übermittlung von Rechnungen wird ein großes Einsparungspotenzial bescheinigt. Beispielsweise prognostiziert eine Studie im Auftrag der EU-Kommission [PWC99] der elektronischen Rechnungsstellung ein Einsparungspotenzial von mehr als 70%. Aktuelle Studien lassen eine Reduktion der Kosten pro ausgehender Rechnung von 16 auf 2 Euro [SKG+04] erwarten; bei eingehenden Rechnungen sollen Einsparungen von 4,80 Euro pro Rechnungsosten und damit durchschnittlich etwa 30 Euro pro Rechnung zu erwarten

## 5 Anwendungsbereiche

---

sein.

Allerdings ist bei derartigen Aussagen eine differenzierte Betrachtung geboten. In Szenarien, in denen der Abrechnungsprozess bereits weitgehend optimiert ist, werden die Einsparungen oft geringer ausfallen. Außerdem ist der große Nutzen beim elektronischen Rechnungsempfang nur dann gegeben, wenn die Rechnungsdaten in strukturierter Form eingehen und deshalb automatisiert weiterverarbeitet werden können. Von der in *Abschnitt 5.2.2.3* erläuterten Übertragung der Rechnungen im Bilddatenformat profitieren also insbesondere die Absender. Geht man davon aus, dass der Empfänger die Eingangsrechnung in Papierform archivieren möchte, so entsteht ihm durch die elektronisch übermittelte Rechnung sogar ein Nachteil.

Deshalb muss für die konkrete Ermittlung des Nutzens der elektronisch übermittelten Rechnung eine Prozesskostenanalyse durchgeführt werden.

Beim Rechnungssteller sind hierbei die folgenden Kostenblöcke zu berücksichtigen:

- Bereitstellung der Rechnungsrohdaten,
- Aufbereiten der Rechnungsrohdaten,
- Verbuchen der Rechnung,
- Produktion der Rechnung,
- Archivierung der Kopie der Rechnung,
- Versand der Rechnung,
- Forderungsmanagement.

Beim Empfänger der Rechnung fallen die folgenden Kostenblöcke an:

- Rechnungseingang,
- Interne Rechnungslogistik,
- Erfassung der Rechnungsdaten,
- Feststellung der Richtigkeit,
- Mängelbearbeitung,
- Archivierung der Rechnung,
- Vorsteuerverzug.

### 5.2.4 Praxisbeispiele

In diesem Abschnitt erläutern schließlich zwei praktische Beispiele die elektronische Übermittlung von Rechnungen.

#### 5.2.4.1 Provisionsabrechnungen bei der Bausparkasse Schwäbisch-Hall

##### Ausgangssituation

Die Bausparkasse Schwäbisch Hall ist mit mehr als 6,6 Millionen Kunden in Deutschland und einem Bestand von rund 179 Mrd. Euro Bausparsumme die größte Bausparkasse Europas und mit einem Marktanteil von 26,1 % Marktführer in der Bundesrepublik. Die gesamte Geschäfts- und Vertriebsleistung betrug in 2004 38,1 Mrd. Euro. Die jährliche Baugeldauszahlungen beläuft sich auf rund 7,2 Mrd. Euro.

Seit 1948 hat die Bausparkasse Schwäbisch Hall AG über 215 Mrd. Euro Bausparmittel zugeteilt. Finanziert wurden damit fast 6,3 Millionen Wohneinheiten (Bau, Kauf, Modernisierung oder Umbau).

Die IT-Systeme der Bausparkasse werden von der Tochtergesellschaft VR Kreditwerk Hamburg - Schwäbisch Hall AG betrieben und mit Unterstützung der syskoplant AG

## 5 Anwendungsbereiche

weiterentwickelt. Zur Vertriebsunterstützung wird von Mitarbeitern im Innen- und Außendienst die Vertriebs-Controlling-Software „BSH-Info“ eingesetzt. Bisher mussten pro Monat etwa 6.000 papiergebundene Provisionskonto-Abrechnungen an fast 4.000 Außendienst-Mitarbeiter versandt werden. Diese Abrechnungen enthalten bei Führungskräften bis zu 1000 Seiten in der Spurze – bis die Papiere bei den Außendienstmitarbeitern eintrafen, vergingen teilweise bis zu zehn Tage.

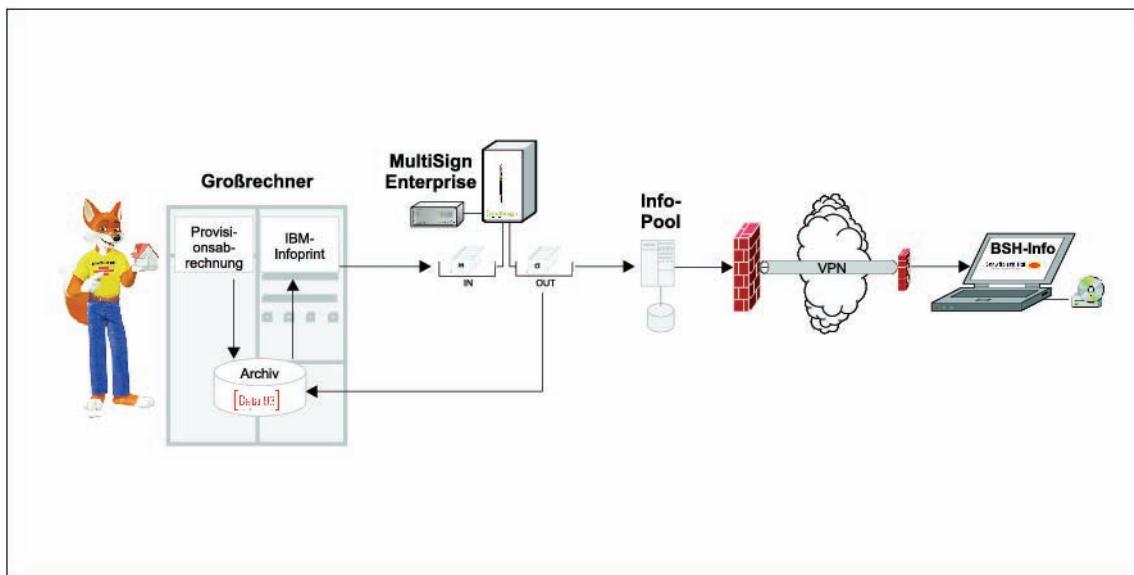


Abbildung 50: Provisionsabrechnungen bei der Bausparkasse Schwäbisch-Hall

### Signurlösung

Um die Zustellung der Abrechnungen zu beschleunigen, den Außendienst-Mitarbeitern bessere Auswertungsmöglichkeiten zu eröffnen und dabei signifikante Kostensenkungen zu realisieren, werden die Provisionsabrechnungen nunmehr elektronisch – in Form von *PDF*-Dateien – übermittelt. Hierbei wurde die elektronische Abrechnung in die bewährten Provisionsabrechnungsprozesse und die „BSH-Info“-Software integriert. Um den gesetzlichen Anforderungen bei der elektronischen Übermittlung von Rechnungen (vgl. Abschnitt 5.2.1) genüge zu tun, werden die *PDF*-basierten Rechnungen mit einer *qualifizierten elektronischen Signatur* versehen. Wie in Abbildung 50 ersichtlich, wird die Provisionsabrechnung am Großrechner erstellt, im Archivsystem [Beta93] abgelegt und schließlich vom [Infoprint]-Server in das *PDF*-Format gewandelt. Die Provisionsabrechnungen werden durch den [multisign] Enterprise Server, der über eine Verzeichnisschnittstelle angesprochen wird, mit einer *qualifizierten elektronischen Signatur* versehen und danach zum Data Warehouse System (Info-Pool) übertragen und zusätzlich im Archivsystem abgelegt. Aus dem Info-Pool werden die Provisionsabrechnungen über eine VPN-Verbindung zum BSH-Info-Client des Außendienstmitarbeiters übertragen. Dort erfolgen auch die Verwaltung der eingegangenen Rechnungen und die Prüfung der Signatur unter Verwendung der [multisign]-Prüfbibliothek. Außerdem wird der Außendienstmitarbeiter spätestens alle sechs Monate aufgefordert, die elektronischen Unterlagen auf einer CD zu archivieren, um damit seinen Verpflichtungen gemäß [GDPdU] gerecht zu werden.

## 5 Anwendungsbereiche

### 5.2.4.2 Buchungsbestätigungen bei der dba Luftfahrtgesellschaft

#### Ausgangssituation

Die dba Luftfahrtgesellschaft mbH ist mit über 4 Millionen Passagieren im Jahr Deutschlands zweitgrößte innerdeutsche Airline. Das flexible, zuverlässige und pünktliche Flugangebot zu fairen Preisen und einem außergewöhnlichen Service ist insbesondere auf die Bedürfnisse von Geschäftsreisenden zugeschnitten. Ein vergleichsweise überdurchschnittlich hoher Anteil der Flugbuchungen wird mittlerweile über das Internet abgewickelt.

Um den Direktvertrieb von Flugtickets über das Internet zu ermöglichen, wurde eine „Internet Booking Engine (IBE)“ entwickelt, die die Buchung einer Flugreise in wenigen, einfachen Schritten ermöglicht:

1. Flugsuche,
2. Auswahl eines verfügbaren Fluges,
3. Buchungsabschluss,
4. Buchungsbestätigung und PDF-Rechnung per E-Mail.

Damit kein zusätzliches Rechnungsdokument auf dem postalischen Weg übermittelt werden muss, soll bereits die PDF-Rechnung alle Anforderungen des § 14 [UStG] erfüllen.

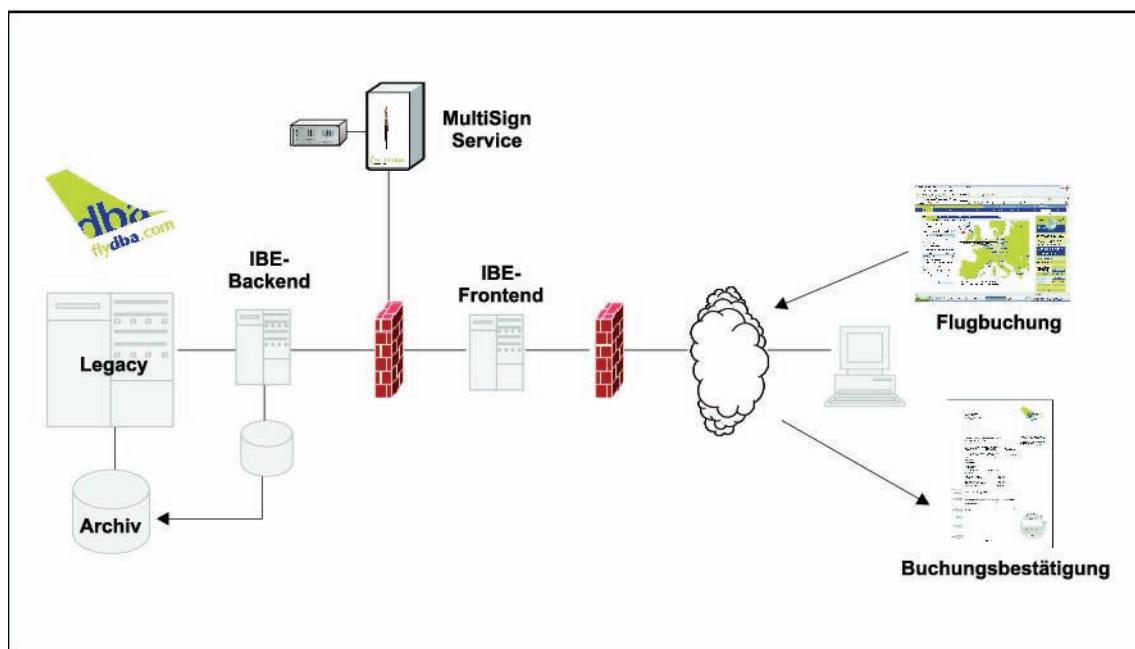


Abbildung 51: Buchungsbestätigung bei der dba Luftfahrtgesellschaft

#### Signurlösung

Wie in Abbildung 51 dargestellt, erfolgt die Buchung eines Fluges unter Verwendung eines Web-Browsers, der mit dem Frontend der Internet Booking Engine (IBE) unter der Adresse [www.flydba.com](http://www.flydba.com) kommuniziert. Das Frontend ist wiederum mit dem IBE-Backend verbunden. Sobald die Buchung vom Fluggäst veranlasst wurde, sorgt das IBE-Backend für die Generierung der Flugbuchung und eine Erfassung der Umsätze in den darunter liegenden Legacy-Systemen (Amadeus, MonaLisa, etc.). Unmittelbar im Anschluss daran erzeugt das Backend eine PDF-basierte Buchungsbestätigung, die dem Fluggäst per E-Mail zugestellt und zusätzlich archiviert wird. Um die Anforderungen des § 14 [UStG] zu erfüllen, wird die

## 5 Anwendungsbereiche

---

Buchungsbestätigung durch den [multisign]-Service mit einer *qualifizierten elektronischen Signatur* versehen, die in Form eines Stempels auf dem Rechnungsdokument angezeigt wird.

Der Massensignatur-Server wird im Rahmen dieser Lösung nicht zusammen mit der Internet Booking Engine betrieben, sondern von der Deutschen Post Com GmbH, die als Application Service Provider für die Massensignatur fungiert und die Rechnungen im Auftrag der dba Luftfahrtgesellschaft mbH signiert. Dazu werden die Signaturen beim Signaturserver über ein Client-Server-Protokoll angefordert. Da die *qualifizierte elektronische Signatur* in die PDF-Datei integriert ist, kann diese direkt im Adobe Reader geprüft werden.

### 5.3 eCard-Strategie der Bundesregierung

Das Bundeskabinett hat am 9. März 2005 die Eckpunkte für eine gemeinsame eCard-Strategie beschlossen [eCard]. Wesentliche Stützpfiler dieser Strategie sind die elektronische *Authentizierung* und die *qualifizierte elektronische Signatur*, die auf *Chipkarten* unterschiedlicher Ausprägung zum Einsatz kommen.

Für die eCard-Strategie, die unter gemeinsamer Federführung des Bundeswirtschaftsministeriums und des Bundesinnenministeriums erarbeitet wurde, sind insbesondere die folgenden Kartenprojekte von Bedeutung:

- Die elektronische Gesundheitskarte

Sie soll ab 2006 die bisherige Krankenversichertenkarte ersetzen. Zusätzlich werden schrittweise zunächst rund 300.000 elektronische Heilberufsausweise eingeführt. Ziel ist es, die Wirtschaftlichkeit und Leistungstransparenz im Gesundheitswesen zu steigern und die Arbeitsprozesse und die Bereitstellung von aktuellen gesundheitsstatistischen Informationen zu optimieren. Zuständig hierfür ist das Bundesministerium für Gesundheit und Soziale Sicherung. Nähere Informationen zur elektronischen Gesundheitskarte finden sich unter [bIT4health].

- Der digitale Personalausweis

Der digitale Personalausweis, für den das Bundesinnenministerium zuständig ist, wird neben den bisherigen Funktionen (Sichtausweis, Identifikationsdokument, Reisedokument) auch eine auf einem Chip gespeicherte elektronische Authentizierungsfunktion beinhalten. Weitere Informationen zum digitalen Personalausweis finden sich in [RRM05].

Beide Karten sollen von vornehmerein technisch so ausgestaltet sein, dass sie auf Wunsch der nutzenden Personen auch für *qualifizierte elektronische Signaturen* zu verwenden sind.

Neben der elektronischen Steuererklärung (ELSTER), die in Abschnitt 5.1.1 näher vorgestellt wurde, nennt die Pressemitteilung [eCard] insbesondere das JobCard-Verfahren [JobCard, HoRo04] als wichtiges Anwendungsgebiet von Signaturkarten.

Ziel des JobCard-Verfahrens, für das das Bundeswirtschaftsministerium zuständig ist, ist die Entlastung der Arbeitgeber von der Ausstellung papierbezogener Bescheinigungen (zum Beispiel Verdienstbescheinigungen) und die Modernisierung von Verwaltungsabläufen. Bestimmte, für die Entscheidung über Ansprüche auf Arbeitslosengeld und andere Sozialleistungen benötigte Daten von Arbeitnehmerinnen und Arbeitnehmern, sollen zukünftig in einer zentralen Stelle gespeichert werden. Um einen Missbrauch der zentral gespeicherten Daten zu verhindern, erfolgt der Zugriff durch den Einsatz einer Signaturkarte der Arbeitnehmerin oder des Arbeitnehmers mit *qualifiziertem Zertifikat*. Hierbei handelt es sich jedoch nicht um eine spezielle Karte (wie die Gesundheitskarte), sondern um ein Verfahren, das mit beliebigen Signaturkarten mit qualifiziertem Zertifikat genutzt werden kann.

Die rechtlichen Aspekte dieser geplanten Kartenprojekte sind in [Horn05] erörtert.

# Glossar

## Abelsche Gruppe

Eine abelsche Gruppe ist eine *Gruppe*  $(G, \cdot)$ , bei der die Reihenfolge der Elemente vertauscht werden darf, so dass für alle  $a, b \in G$  das Kommutativitätsgesetz  $a \cdot b = b \cdot a$  gilt.

## Abstract Syntax Notation One (ASN.1)

Die Abstract Syntax Notation One (ASN.1) ermöglicht es, die Syntax von Daten präzise zu spezifizieren. Sie wurde im Rahmen des [X.408]-Standards entwickelt und danach als X.208-Empfehlung und inzwischen in [X.680] standardisiert. ASN.1 erlaubt die abstrakte Spezifikation von Daten unabhängig von deren tatsächlicher Codierung, die durch spezifische Codierungsregeln festgelegt wird. Die Basic Encoding Rules (BER), die Canonical Encoding Rules (CER) und die Distinguished Encoding Rules (DER) sind in [X.690] definiert. Daneben existieren die Packed Encoding Rules (PER) [X.691] und die XML Encoding Rules (XER) [X.693]. ASN.1 wird beispielsweise beim X.509-Standard und den Standards der PKCS-Reihe eingesetzt. Eine ausführliche Behandlung der Materie findet sich in [Dubu00].

## Akkreditierung

Die Akkreditierung ist gemäß § 2 Nr. 15 [SigG] ein freiwilliges „Verfahren zur Erteilung einer Erlaubnis für den Betrieb eines Zertifizierungsdienstes, mit der besondere Rechte und Pflichten verbunden sind.“ Zu den Pflichten des *Zertifizierungsdiensteanbieters* gehört beispielsweise, dass er *geprüfte und bestätigte Produkte* einsetzen muss (§ 15 Abs. 7 [SigG]), sofern er *Zertifikate* ausstellt, diese mindestens 30 Jahre nach Ablauf der Gültigkeit des Zertifikates nachprüfbar halten muss (§ 4 [SigV]) und sein Sicherheitskonzept von einer von der *BNetzA* anerkannten Stelle auf seine Eignung und praktische Umsetzung hin prüfen und bestätigen lassen muss (§ 15 Abs. 2 [SigG]). Auf der anderen Seite darf sich ein akkreditierter Zertifizierungsdiensteanbieter als solcher bezeichnen und sich im Rechts- und Geschäftsverkehr auf die nachgewiesene Sicherheit berufen (§ 15 Abs. 1 [SigG]).

## American National Standards Institute (ANSI)

Die ANSI (<http://www.ansi.org/>) ist ein privatwirtschaftliches Standardisierungsorgan der Vereinigten Staaten von Amerika und der Vertreter der USA in der ISO.

## Application Programming Interface (API)

Ein Application Programming Interface ist eine dokumentierte Software-Schnittstelle, mit deren Hilfe ein Software-System bestimmte Funktionen eines anderen Software-Systems nutzen kann.

## Asymmetrische Kryptoalgorithmen

Hierbei existiert ein komplementäres Schlüsselpaar (*privater Schlüssel* und *öffentlicher Schlüssel*), das zur Realisierung *digitaler Signaturen*, zur Vereinbarung *geheimer Schlüssel* oder zur asymmetrischen *Verschlüsselung* verwendet werden kann. Das Konzept asymmetrischer Kryptoalgorithmen geht zurück auf Whitfield Diffie und Martin Hellman [DiHe76]. Der heute gebräuchlichste asymmetrische Kryptoalgorithmus ist der *RSA-Algorithmus*.

## **Attribut**

Ein Attribut ist allgemein eine Eigenschaft eines Objekts. Im Umfeld der *elektronischen Signatur* können Attribute als Bestandteil von *Zertifikaten* – „Public-Key-Zertifikaten“ oder spezialisierten *Attributzertifikaten* – oder aber in *High-Level-Signaturen*, z.B. im *PKCS #7/CMS*-Format, vorkommen. Beispielsweise kann ein Zertifikat ein Attribut enthalten, aus dem hervor geht, dass der Zertifikatsinhaber ein Arzt ist.

## **Attributbestätigungsinstanz**

Eine Attributbestätigungsinstanz ist Teil einer *PKI* und bescheinigt, dass der Antragsteller für ein *Zertifikat* eine bestimmte Eigenschaft besitzt, so dass diese als *Attribut* in das beantragte Zertifikat aufgenommen werden kann.

## **Attributzertifikat**

Ein Attributzertifikat ist ein *Zertifikat*, das selbst keinen *öffentlichen Schlüssel* enthält, sondern lediglich in eindeutiger Weise auf ein *Public-Key-Zertifikat* verweist. Es wird verwendet, um dem referenzierten *Public-Key-Zertifikat* weitere *Attribute* zuzuweisen.

## **Augenschein und Augenscheinsbeweis**

Augenschein ist jede unmittelbare sinnliche Wahrnehmung durch eine für das Gericht oder die Behörde tätige Person mit dem Ziel, beweiserhebliche Tatsachen festzustellen (z. B. durch Sehen, Hören, Riechen). Der Beweis durch Augenschein (Augenscheinsbeweis), der im Zivilprozessrecht in §§ 371 ff [ZPO] geregelt ist, umfasst alle Beweismittel, die nicht als Zeugen-, Urkunden-, oder Sachverständigenbeweis gesetzlich besonders geregelt sind. Er besteht darin, dass sich das Gericht durch eine unmittelbare sinnliche Wahrnehmung beispielsweise einen Eindruck von der Beschaffenheit einer Sache, der Lage von Gegenständen oder der Existenz und den Verhaltensweisen eines Menschen macht.

## **AUTACK**

AUTACK ist ein in [ISO9735-6] spezifizierter *EDIFACT*-Nachrichtentyp zur Übermittlung von *Integritäts- und Authentizitätsinformationen* über versendete Nutzdaten. Anwendungsregeln für die Nutzung der AUTACK finden sich auch in [DIN16560-15].

## **Authentifizierung**

Die Authentifizierung bezeichnet den Vorgang, die Identität einer Person oder eines Rechnersystems an Hand eines bestimmten Merkmals zu überprüfen.

## **Authentizität**

Unter dem Nachweis der Authentizität von elektronischen Daten versteht man den Nachweis über die Echtheit der Daten (vgl. *Integrität*) und die eindeutige Zuordnung zum Verfasser, Ersteller und/oder Absender.

## **Betriebsanzeige**

Gemäß § 4 Abs. 3 [SigG] i.V.m. § 3 [SigG] muss die Aufnahme eines Zertifizierungsdienstes bei der *BNetzA* angezeigt werden. Im Rahmen dieser Betriebsanzeige muss der *ZDA* insbesondere sein Sicherheitskonzept vorlegen. Anstatt der Anzeige des Betriebes kann ein *ZDA* auch die *Akkreditierung* gemäß § 15 [SigG] beantragen.

## Biometrie

Die Biometrie beschäftigt sich allgemein mit der Vermessung quantitativer Merkmale von Lebewesen. Zu den populären menschlichen Merkmalen, die zur *Authentifizierung* herangezogen werden, zählt beispielsweise der Fingerabdruck [BioFinger], das Gesicht [BioFace] oder die eigenhändige Unterschrift [LeSc04]. Weitere Informationen zur Biometrie finden sich auch unter <http://www.bsi.de/fachthem/biometrie/index.htm>.

## Bitoperation

Eine Bitoperation bezeichnet den Aufwand eines Algorithmus, um zwei Bits miteinander zu verknüpfen. Beispielsweise benötigt man zur Addition von Zahlen der Bitlänge  $l$   $O(l)$  Bitoperationen.

## Bundesnetzagentur (BNetzA)

Durch das in Kraft treten des zweiten Gesetzes zur Neuregelung des Energiewirtschaftsrechts [EnWGAendG] am 13. Juli 2005 wurde der Aufgabenbereich der *Regulierungsbehörde für Telekommunikation und Post (RegTP)* erweitert und eine Umbenennung der Behörde in „Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen“, mit der amtlichen Kurzbezeichnung „Bundesnetzagentur“ (BNetzA), vorgenommen.

## CEN

Siehe *Comité Européen de Normalisation*.

## Certificate Policy (CP)

Eine Certificate Policy besteht aus einer Menge von Regeln, die bei der Ausstellung des Zertifikates berücksichtigt wurden. Auf Basis der Certificate Policy kann entschieden werden, ob ein Zertifikat für einen bestimmten Einsatzzweck ausreichende Sicherheit bietet. Ein Rahmenwerk für die Entwicklung von Certificate Policies findet sich in [RFC3647].

## Certificate Revocation List (CRL)

Siehe *Sperrliste*.

## Certification Authority (CA)

Siehe *Zertifizierungsinstanz*.

## Charakteristik

Die Charakteristik  $\text{char}(K)$  eines *endlichen Körpers*  $K$  gibt an, wie oft man das neutrale Element der multiplikativen Gruppe (1) aufzaddieren muss, um das neutrale Element der additiven Gruppe (0) zu erhalten. Da alle endlichen Körper endliche Erweiterungen von Primkörpern sind, ist die Charakteristik eines endlichen Körpers immer eine Primzahl  $p$ . Unendlichen Körpern, wie z. B. den rationalen Zahlen  $\mathbb{Q}$  oder den reellen Zahlen  $\mathbb{R}$  weist man die Charakteristik  $\text{char}(K) = 0$  zu.

## Chipkarte

Eine Chipkarte ist eine meist aus Kunststoff bestehende Karte, die ein oder mehrere Halbleiterchips enthält. Man unterscheidet zwischen Speicherkarten, auf denen lediglich Daten abgelegt werden können, und Mikroprozessorkarten, in die ein Prozessor integriert ist, der auch Daten verarbeiten kann. Eine Mikroprozessorkarte nennt man auch auch *Smart Card*.

Umfassende Informationen zur Chipkarte finden sich in [RaEf02].

### **Chipkartenterminal**

Ein Chipkartenterminal ist ein Gerät, das die elektrische Versorgung und den Datenaustausch mit einer *Chipkarte* ermöglicht. Zusätzlich kann dieses Gerät mit einer Tastatur und einem Display ausgestattet sein, um eine sichere *PIN*-Eingabe zu ermöglichen. Im Kontext der *qualifizierten elektronischen Signatur* treten Chipkartenterminals insbesondere als Teil von *Signaturanwendungskomponenten* auf.

### **Comité Européen de Normalisation (CEN)**

CEN ist ein Europäisches Komitee für Normung mit Sitz in Brüssel, das die Ziele der Europäischen Gemeinschaft mit der Entwicklung von freiwilligen, technischen Standards und Regularien zur Konformitätsprüfung unterstützt. Diese Standards werden als „CEN Workshop Agreements (CWA)“ publiziert.

Für die elektronische Signatur in Europa sind die Standards [CWA14167-1, CWA14167-2, CWA14169] von besonderer Bedeutung, da sie durch [2003/511/EG] eine offizielle Präzisierung der technischen Anhänge II f) und III der Europäischen Signaturdirektive [1999/93/EG] darstellen.

### **Common Criteria (CC)**

Mit den Common Criteria for Information Technology Security Evaluation (kurz: Common Criteria) [CC] wurde ein internationaler Standard (ISO 15408) für die Bewertung und Zertifizierung der Sicherheit von Computersystemen geschaffen, so dass Komponenten oder Systeme nicht in verschiedenen Ländern mehrfach zertifiziert werden müssen. Die Common Criteria, in deren Entwicklung unter anderem die europäischen IT-Sicherheitskriterien (ITSEC) eingeflossen sind, sehen verschiedene Vertrauenswürdigkeitsstufen (Evaluation Assurance Level) vor. Hierbei existieren die Stufen „EAL1“ (funktionell getestet) bis „EAL7“ (formal verifizierter Entwurf und getestet), durch die bestimmte Anforderungen im Hinblick auf folgende Aspekte definiert werden:

- Konfigurationsmanagement,
- Auslieferung und Betrieb,
- Entwicklung,
- Handbücher,
- Lebenszyklus-Unterstützung,
- Testen,
- Schwachstellenbewertung.

Die Anforderungen an die Vertrauenswürdigkeit sind derart gestaffelt, dass in einer Stufe EAL ( $n$ ) jeweils mindestens die Anforderungen der Stufe EAL ( $n - 1$ ) gefordert werden. Für Produkte für *qualifizierte elektronische Signaturen* (mit *Anbieterakkreditierung*) ist eine Prüfung gemäß Common Criteria oder ITSEC und eine nachfolgende *Bestätigung* durch eine Stelle gemäß § 18 [SigG] erforderlich.

### **Cross-Zertifikat**

Ein Cross-Zertifikat ist ein *Public-Key-Zertifikat*, das eine *Zertifizierungsinstanz* für eine andere Zertifizierungsinstanz ausstellt.

## Cryptographic Message Syntax (CMS)

Die Cryptographic Message Syntax [RFC2630, RFC3369, RFC3852] ist eine von der *IETF* getragene Weiterentwicklung des *PKCS #7*-Standards. In diesem Standard, der bereits heute von vielen Standardsoftware-Komponenten unterstützt wird, ist unter anderem ein sehr weit verbreitetes *High-Level-Signaturformat* spezifiziert. Außerdem bildet es die Basis für das *S/MIME*-Format zur Verschlüsselung und Signatur von E-Mail-Nachrichten sowie für spezifische Nachrichten zur Zertifikatsverwaltung [RFC2797].

## DCF77

DCF77 ist das von der Physikalisch-Technischen Bundesanstalt (<http://ptb.de>) in Mainflingen – südöstlich von Frankfurt – ausgestrahlte Funksignal, das die gesetzlich festgelegte Zeit gemäß Zeitgesetz [ZeitG] trägt. Dieses Signal wird insbesondere von *Zertifizierungsdiensteanbietern* genutzt, um die Aktualität der Systemzeit der von Ihnen betriebenen *OCSP*- und *TSP*-Responder zu gewährleisten.

## Delta-CRL

Eine Delta-CRL ist eine *Sperrliste*, die nicht alle Sperreinträge enthält, sondern lediglich Updates zu einer von ihr referenzierten Basis-CRL.

## Deterministischer Algorithmus

Ein deterministischer Algorithmus ist ein Algorithmus, der zu einer bestimmten Eingabe immer die gleiche Folge von Operationen ausführt. Zu jedem Zeitpunkt ist der nachfolgende Abarbeitungsschritt des Algorithmus eindeutig festgelegt und nicht vom Zufall abhängig. Das Gegenteil eines deterministischen Algorithmus ist ein *probabilistischer Algorithmus*.

## Digitale Signatur

Eine digitale Signatur ist eine *elektronische Signatur*, die auf *asymmetrischen Kryptoalgorithmen* basiert. Hierbei kann eine digitale Signatur nur mit dem *privaten Schlüssel* erzeugt, aber durch jedermann unter Verwendung des *öffentlichen Schlüssels* geprüft werden. Siehe auch *fortgeschrittene elektronische Signatur* und *qualifizierte elektronische Signatur*

## Directory (DIR)

Siehe *Verzeichnisdienst* und *X.500*.

## Digital Signature Algorithm (DSA)

Der Digital Signature Algorithm (DSA) [FIPS186-2] ist ein Signaturalgorithmus auf Basis des *Diskreten Logarithmus* in der multiplikativen *Gruppe* eines *endlichen Körpers*.

## Diskreter Logarithmus

Neben *asymmetrischen Kryptoalgorithmen*, die auf dem *Faktorisierungsproblem* beruhen, werden in der Praxis zunehmend auch Kryptosysteme eingesetzt, die auf dem Diskreten Logarithmus Problem (DLP) basieren. Das Problem des Diskreten Logarithmus in einer endlichen, *abelschen Gruppe*  $G$  ist die Berechnung des Exponenten  $n$  aus einem gegebenen Gruppenelement  $g^n \in G$ . Wie schwierig die Lösung dieses Problems ist, hängt maßgeblich von der verwendeten Gruppe  $G$  ab. Für kryptographische Zwecke werden beispielsweise multiplikative Gruppen *endlicher Körper* oder Punktegruppen auf *elliptischen Kurven* eingesetzt. Unter Verwendung dieser Gruppen können Signaturverfahren, wie *DSA*

bzw. ECDSA, konstruiert werden.

### Distinguished Name (DN)

Ein Distinguished Name (DN) ist gemäß [X.501] eine Folge von „Relative Distinguished Names (RDN)“, die wiederum aus einem oder mehreren Werten bestehen. Der DN beschreibt den Pfad von einem Verzeichniseintrag zum Wurzelknoten, so dass durch den DN alle Einträge in einem X.500-Verzeichnis eindeutig adressiert werden können. Ein DN kann beispielsweise aus Einträgen für das Land (Country, C), die Organisation (Organization, O), die Organisationseinheit (Organizational Unit, OU) und schließlich den Namen des Objekts (Common Name, CN) bestehen.

### DLP

Siehe *Diskreter Logarithmus*.

### Einfache elektronische Signatur

Unter einer „einfachen elektronischen Signatur“ versteht man eine *elektronische Signatur*, die nicht alle Anforderungen an eine *fortgeschrittene elektronische Signatur* erfüllt.

### Einwegfunktion

Eine Funktion  $f$  nennt man Einwegfunktion (vgl. [Gold01, Definition 2.1]), wenn sie „leicht“ berechnet aber „schwer“ invertiert werden kann. Hierbei bedeutet „leicht“, dass es einen *deterministischen Algorithmus* mit *polynomieller Laufzeit* gibt, der zu einem gegebenen  $x$  den Funktionswert  $f(x)$  berechnet. „Schwer“ bedeutet, dass für hinreichend große Eingabelängen jeder *probabilistische Algorithmus* mit *polynomieller Laufzeit* für die Berechnung der inversen Funktion  $f^{-1}(f(x))$  eine „vernachlässigbare“ Erfolgswahrscheinlichkeit hat. Hierbei nennt man einen Wert „vernachlässigbar“, wenn er für hinreichend große Eingabelänge  $l$  kleiner als  $\frac{1}{p(l)}$  für jedes Polynom  $p$  ist.

Es ist bisher nicht bekannt, ob solche Einwegfunktionen tatsächlich existieren. Allerdings existieren eine Reihe von Funktionen, die die Einweg-Eigenschaft zu besitzen *scheinen*, da für die Berechnung von  $f^{-1}(f(x))$  bislang kein *probabilistischer Algorithmus* mit *polynomieller Laufzeit* bekannt ist, der eine nicht vernachlässigbare Erfolgswahrscheinlichkeit hat. Beispielsweise basiert die Sicherheit von vielen populären Signaturverfahren auf der unbewiesenen Annahme, dass die Multiplikation von großen, zufällig gewählten Primzahlen (vgl. *Faktorisierungsproblem*) oder die Exponentiation in bestimmten endlichen, *abelschen Gruppen* (vgl. *Diskreter Logarithmus*) eine Einwegfunktion sei.

### Electronic Business XML (ebXML)

ebXML (<http://www.ebxml.org>) ist eine 1999 gestartete, gemeinsame Initiative von UN/CEFACT und OASIS, durch die eine Reihe von Spezifikationen für die Nutzung von XML für elektronische Geschäftsprozesse entwickelt wurden.

### Electronic Data Interchange (EDI)

EDI ist ein Sammelbegriff für alle elektronischen Verfahren zum vollautomatischen Versand von strukturierten Nachrichten zwischen Anwendungssystemen unterschiedlicher Institutionen. Zu den möglicherweise wichtigsten Standards für EDI zählen EDIFACT und ebXML.

## **Electronic Data Interchange For Administration, Commerce and Transport (EDIFACT)**

EDIFACT ist ein branchenübergreifender internationaler Standard (*ISO9735*) für den automatisierten Austausch elektronischer Daten im Geschäftsverkehr. Er ist einer von mehreren gebräuchlichen Standards für *EDI*.

## **Elektronische Form**

Für die in § 126a [BGB] definierte elektronische Form, die gemäß § 126 [BGB] im Regelfall die *Schriftform* ersetzen kann, „muss der Aussteller der Erklärung dieser seinen Namen hinzufügen und das elektronische Dokument mit einer qualifizierten elektronischen Signatur nach dem Signaturgesetz versehen.“

## **Elektronische Signatur**

Gemäß § 2 Nr.1 [SigG] sind elektronische Signaturen Daten in elektronischer Form, die der *Authentifizierung* dienen. Die Bandbreite möglicher Ausprägungen reicht von einem sehr leicht fälschbaren Bitmap einer handschriftlichen Unterschrift bis hin zur *qualifizierten elektronischen Signatur* als sehr sichere Form der *digitalen Signatur*.

## **Elliptic Curve Digital Signature Algorithm (ECDSA)**

Der ECDSA [ANSI-X9.62] ist ein Signaturalgorithmus auf Basis des *Diskreten Logarithmus* in der *Gruppe* der Punkte einer *elliptischen Kurve* über einem *endlichen Körper*.

## **Elliptische Kurve**

Eine elliptische Kurve (über einem Körper  $K$  mit Charakteristik  $\text{char}(K) \neq 2, 3$ ) ist die Menge aller Punkte  $P = (x, y)$  auf der „glatten“<sup>43</sup> Kurve  $y^2 = x^3 + ax + b$  zusammen mit dem Punkt  $\mathcal{O}$  im „Unendlichen“. Für die kryptographische Anwendung solcher elliptischer Kurven ist wesentlich, dass die Menge der Punkte eine endliche, *abelsche Gruppe* mit  $\mathcal{O}$  als neutralem Element definiert, in der man effizient rechnen kann, aber das Problem des *Diskreten Logarithmus* äußerst schwer ist. Deshalb benötigen Kryptosysteme auf Basis elliptischer Kurven, wie z.B. *ECDSA*, für das gleiche Maß an Sicherheit deutlich weniger Speicherplatz und Rechenkapazität als herkömmliche Verfahren, wie z.B. *RSA* und *DSA*.

## **Endlicher Körper**

Ein endlicher Körper ist ein *Körper*, der nur endlich viele Elemente besitzt. Bei einem solchen Körper  $K$  gibt die *Charakteristik* des Körpers  $\text{char}(K)$  an, wie oft man das  $1$ -Element aufzaddieren muss, um das  $0$ -Element zu erhalten. Ist  $p$  eine *Primzahl*, so erhält man einen endlichen Körper, wenn man modulo dieser Primzahl  $p$  rechnet, also bei der Addition und Multiplikation nur den verbleibenden Rest (abzüglich Vielfacher von  $p$ ) betrachtet. Ein solcher Körper hat die Charakteristik  $\text{char}(K) = p$ . Alle endlichen Körper erhält man, wenn man endliche Körpererweiterungen von Primkörpern betrachtet. Diese endlichen Körper besitzen  $q = p^k$  Elemente, wobei  $k$  der Grad der Körpererweiterung ist. In der Kryptographie spielen endliche Körper eine wichtige Rolle, da die Berechnung *Diskreter Logarithmen* in ihrer multiplikativen Gruppe nach heutigem Kenntnisstand ein schwieriges Problem ist.

## **Enterprise-Ressource-Planning (ERP)**

Enterprise-Resource-Planning bezeichnet die unternehmerische Aufgabe, die in einem

---

<sup>43</sup>Die Kurve ist genau dann glatt, wenn  $4a^3 + 27b^2 \neq 0$ .

Unternehmen vorhandenen Ressourcen (wie z.B. Kapital, Betriebsmittel, Personal, ...) möglichst effizient für den betrieblichen Ablauf einzuplanen und dadurch die Geschäftsprozesse unternehmensweit zu optimieren. Zur Unterstützung des ERP-Prozesses werden heute relativ komplexe Software-Systeme (ERP-Systeme) eingesetzt. Zu den bekanntesten Herstellern von ERP-Systemen zählt die SAP AG.

### **Entschlüsselung**

Vorgang, bei dem unter Verwendung mathematischer Algorithmen und *privater* oder *geheimer Schlüssel* elektronische Daten wieder les- bzw. verarbeitbar gemacht werden. In verschlüsselter Form sind die Daten von unbefugten Dritten nicht einsehbar. Die Daten können nur vom Besitzer des entsprechenden privaten oder geheimen Schlüssels wieder in die Originalform überführt werden.

### **Erweiterter Euklidischer Algorithmus**

Mit dem erweiterten Euklidischen Algorithmus berechnet man für zwei ganze Zahlen  $a$  und  $b$  eine Linearkombination des größten gemeinsamen Teilers  $d = \text{ggT}(a, b) = s \cdot a + t \cdot b$  der eingegebenen Zahlen. Ruft man den Algorithmus für ein zum Modul  $n$  teilerfremdes  $x$  auf und erhält die Linearkombination  $1 = \text{ggT}(n, x) = s \cdot n + t \cdot x$  so ist  $t \equiv x^{-1} \pmod{n}$  das inverse Element von  $x$  modulo  $n$ .

### **Exponentialzeit**

In der Komplexitätstheorie bezeichnet man ein Problem als in Exponentialzeit lösbar, wenn es einen Algorithmus gibt, der für die Eingabelänge  $l$  und ein konstantes  $c > 1$   $O(c^l)$  Operationen benötigt. Man sagt, der Algorithmus hat exponentielle Laufzeit. Jedes Problem aus der Komplexitätsklasse  $\mathcal{NP}$  kann auch mit *exponentieller Laufzeit* gelöst werden (vgl. [Wagn94, Satz 5.10]).

### **Extensible Markup Language (XML)**

XML ist ein flexibler, von einer Arbeitsgruppe des W3C entwickelter, Standard [XML(v1.0), XML(v1.1)] zur Erstellung strukturierter, maschinen- und menschenlesbarer Dateien.

### **Evaluationsgegenstand (EVG)**

Bei einer Evaluation gemäß *ITSEC* oder *Common Criteria* nennt man das zu bewertende Produkt oder System „Evaluationsgegenstand“ (EVG).

Ein EVG kann aus mehreren Komponenten bestehen. Von besonderer Bedeutung für die Evaluation sind die *sicherheitsspezifischen* und *sicherheitsrelevanten* Komponenten. Hierbei ist eine Komponente sicherheitsspezifisch, wenn sie unmittelbar zum Erreichen der Sicherheitsziele beiträgt. Eine Komponente, die zwar nicht sicherheitsspezifisch ist, aber dennoch korrekt arbeiten muss, um die Sicherheit des EVG zu gewährleisten, nennt man sicherheitsrelevant.

### **Faktorisierungsproblem**

Die Sicherheit aller heute bekannten *asymmetrischen Kryptoalgorithmen* beruht auf der vermeintlichen Schwierigkeit von bestimmten mathematischen Problemen. Ein in der Kryptographie sehr populäres Problem ist das Faktorisierungsproblem, das darin besteht, eine große zusammengesetzte Zahl  $n = pq$  in ihre Primfaktoren  $p$  und  $q$  zu zerlegen. Wählt man die *Primzahlen*  $p$  und  $q$  zufällig, so dass  $n$  etwa 300 Dezimalstellen hat, so ist das

Faktorisierungsproblem nach derzeitigem Stand der Wissenschaft und Technik in der Praxis nicht lösbar und ein darauf basierendes kryptographisches Verfahren, wie z.B. der *RSA-Algorithmus*, sicher. Der derzeitige Faktorisierungsrekord für solche Zahlen liegt bei 200 Dezimalstellen [RSA200] und wurde kürzlich von Wissenschaftlern der Universität Bonn in Zusammenarbeit mit dem Bundesamt für Sicherheit in der Informationstechnik aufgestellt.

### **Fortgeschrittene elektronische Signatur**

Eine fortgeschrittene elektronische Signatur ist gemäß § 2 Nr. 2 [SigG] eine elektronische Signatur mit besonderen Eigenschaften, durch die zumindest ein grundlegendes Maß an *Authentizität* und *Integrität* sicher gestellt werden kann. Anders als bei der *qualifizierten elektronischen Signatur* kann aber eine lediglich fortgeschrittene elektronische Signatur nicht die *Schriftform* gemäß § 126 [BGB] ersetzen und hat geringere Beweiskraft vor Gericht (vgl. § 371a [ZPO]). Zumeist verwendet man *digitale Signaturen* und *Zertifikate*, um fortgeschrittene elektronische Signaturen zu realisieren.

### **Geheimer Schlüssel**

Geheime Schlüssel werden im Zusammenhang mit symmetrischen Kryptoalgorithmen verwendet. Im Gegensatz zu den bei *asymmetrischen Kryptoalgorithmen* eingesetzten *privaten Schlüsseln* ist das gesamte Schlüsselmaterial allen Kommunikationspartnern bekannt. Da der Unterzeichner das Schlüsselmaterial nicht wie in § 2 Nr. 2 a) und c) [SigG] unter seiner alleinigen Kontrolle halten kann, kann mit symmetrischen Kryptoalgorithmen auch keine *fortgeschrittene elektronische Signatur* realisiert werden – es kann zwar die *Authentizität* und *Integrität* sicher gestellt werden, aber keine *Nichtabstreitbarkeit* erreicht werden.

### **Geprüfte und bestätigte Produkte**

Das Signaturgesetz definiert in § 17 [SigG] und § 15 [SigV] eine Reihe von Anforderungen für Produkte, die im Umfeld der *qualifizierten elektronischen Signatur* zum Einsatz kommen. Diese Anforderungen müssen von *Zertifizierungsdiensteanbietern* streng beachtet werden – für den Anwender der elektronischen Signatur haben diese Anforderungen empfehlenden Charakter (vgl. § 17 Abs. 2 [SigG]). Ein *akkreditierter Zertifizierungsdiensteanbieter* muss ausschließlich gemäß *ITSEC* oder *Common Criteria* geprüfte und bestätigte Produkte einsetzen (vgl. § 15 Abs. 7 [SigG]). Die detaillierten Anforderungen an die Prüfung der einzelnen Produkte sind in Anlage 1 zur Signaturverordnung [SigV] definiert. Im *angezeigten (nicht-akkreditierten) Betrieb* genügt bei den meisten Produkten auch eine *Herstellererklärung* gemäß § 17 Abs. 4 [SigG].

### **Gruppe**

Eine Gruppe  $G$  ist in der Mathematik eine Menge von Elementen, für die eine Verknüpfung von zwei Elementen definiert ist (z.B.  $a \cdot b$ ), die folgende Eigenschaften erfüllt:

- **Abgeschlossenheit:**  
Ist  $a$  und  $b$  ein Element der Menge  $G$ , kurz  $a, b \in G$ , so ist auch  $a \cdot b$  ein Element von  $G$ .
- **Assoziativität:**  
Die Reihenfolge beim Ausrechnen ist egal. Es gilt  $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ .
- **Neutrales Element:**  
Es gibt ein Element  $1$  in der Gruppe, das nichts tut:  $a \cdot 1 = 1 \cdot a = a$ .

- Inverses Element:

Für jedes Element gibt es ein Spiegelbild. Das Inverse von  $a$  ist  $1/a$  und hat die Eigenschaft, dass die Verknüpfung mit  $a$  das neutrale Element  $1$  ergibt:  
$$a \cdot 1/a = 1/a \cdot a = 1.$$

Wenn man außerdem die Reihenfolge der Elemente vertauschen kann, so dass  $a \cdot b = b \cdot a$  gilt (Kommutativität), so spricht man von einer *abelschen* Gruppe. Eine Gruppe, die nur endlich viele Elemente hat, nennt man eine *endliche* Gruppe. Endliche, abelsche Gruppen spielen in der Kryptographie eine große Rolle, da sie die Konstruktion von Kryptosystemen auf Basis des *diskreten Logarithmus* erlauben.

### **Gruppenordnung**

Die Mächtigkeit  $|G|$  der Trägermenge einer *Gruppe*  $G$  bezeichnet man als Gruppenordnung.

### **Gültigkeitsmodell**

Durch das Gültigkeitsmodell wird festgelegt, unter welchen Voraussetzungen eine *digitale Signatur* als gültig erachtet wird. Zu den populärsten Gültigkeitsmodellen zählen das *Schalenmodell*, das *Hybridmodell* und das *Kettenmodell*.

### **Hardware Security Module (HSM)**

Ein HSM ist eine spezialisierte Hardware, die es ermöglicht, kryptographische Schlüssel in besonders sicherer Form aufzubewahren und performant anzuwenden.

### **Hashfunktion**

Eine Hashfunktion ist ein kryptographischer Algorithmus, bei dem Nachrichten beliebiger Länge auf einen *Hashwert* fester Länge (z.B. 160 Bit) abgebildet werden. Bei kryptographisch geeigneten Hashfunktionen ist es praktisch unmöglich, zwei Nachrichten mit dem gleichen Hashwert zu finden (Kollisionsresistenz) und bei einem gegebenen Hashwert eine Nachricht zu finden, die durch die Hashfunktion auf den Hashwert abgebildet wird (Einwegeigenschaft). Meist werden hierfür die vom Bundesamt für Sicherheit in der Informationstechnik (BSI) und der Bundesnetzagentur (BNetzA) als geeignet angesehenen Hashalgorithmen verwendet. In der aktuellen Übersicht über geeignete Algorithmen [BNetzA-Alg05] werden *SHA-1* und *RIPEMD-160* für den Einsatz bis 2010 als geeignet erachtet.

### **Hashwert**

Ein Hashwert ist eine mathematische Prüfsumme, die durch Anwendung einer *Hashfunktion* aus einer elektronischen Nachricht erzeugt wird. Da es bei einer kryptographisch geeigneten Hashfunktion praktisch unmöglich ist, zwei Nachrichten zu finden, deren Hashwert identisch ist, bezeichnet man den Hashwert auch als „digitalen Fingerabdruck“ einer Nachricht. Da man auf Grund des so genannten Geburtstagsparadoxon mit großer Wahrscheinlichkeit eine Kollision bei einer  $l$ -Bit-Hashfunktion findet, wenn man etwa  $2^{l/2}$  zufällige Nachrichten wählt, sollte eine Hashfunktion, die für elektronische Signaturen eingesetzt werden soll, mindestens 160 Bit Hashwerte produzieren.

### **Herstellererklärung**

Eine Herstellererklärung gemäß § 17 Abs. 4 [SigG] ist eine Bescheinigung eines Produktherstellers darüber, welche Anforderungen des Signaturgesetzes durch sein Produkt im

Einzelnen erfüllt werden.

### **High-Level-Signaturformat**

Ein High-Level-Signaturformat spezifiziert, wie eine rohe Signatur in einem *Low-Level-Signaturformat* um weitere für die Prüfung der Signatur relevante Informationen, wie z.B. den Zeitpunkt der Signaturerstellung, das zur Prüfung der Signatur notwendige *Zertifikat* oder entsprechende Zertifikatstatusinformationen, ergänzt werden kann und wie die Signatur mit den Nutzdaten verknüpft, oder in diese eingebettet, wird. Gebräuchliche High-Level-Signaturformate sind beispielsweise *PKCS #7 / CMS, S/MIME, PDF* (embedded *PKCS #7* und *PKCS #1*), *XML-DSig* oder das in *ISO9735-5/6* standardisierte Format zur Signatur von *EDIFACT*-Daten.

### **Hybridmodell**

Das Hybridmodell ist ein *Gültigkeitsmodell* für *digitale Signaturen*, bei dem man fordert, dass alle *Zertifikate* im *Zertifizierungspfad* zum Zeitpunkt der Erzeugung der zu prüfenden Signatur gültig sind. Siehe auch *Schalenmodell* und *Kettenmodell*.

### **Hypertext Transport Protocol (HTTP)**

Das in [RFC1945, RFC2616] spezifizierte HTTP ist ein Protokoll zur Übertragung von Daten, das insbesondere im Rahmen des World Wide Web zum Einsatz kommt und sich meist auf das verbindungsorientierte *TCP* stützt. HTTP kann beispielsweise auch zum Transport von *OCSP*-Nachrichten eingesetzt werden.

### **Identifikationsdaten**

Das Signaturgesetz fordert in § 17 Abs. 1 [SigG], dass die *sichere Signaturerstellungseinheit* vor unberechtigter Nutzung zu schützen ist. § 15 Abs. 1 [SigV] fordert hierfür eine Identifikation „durch Besitz und Wissen oder durch Besitz und ein oder mehrere *biometrische Merkmale*“. Da bislang keine Implementierungen biometrischer Verfahren bekannt sind, die die Anforderungen des Signaturgesetzes (vgl. [SigV]) nachweislich erfüllen, werden für *qualifizierte elektronische Signaturen* in der Praxis immer *Personal Identification Numbers* als Identifikationsdaten eingesetzt. Ein möglicher Ersatz der PIN durch das biometrische Merkmal „Fingerabdruck“ wurde beispielsweise in [DaTe05] untersucht.

### **Information Technology Security Evaluation Criteria (ITSEC)**

[ITSEC] ist ein europäischer Standard für die Prüfung und Zertifizierung von Produkten und Systemen im Hinblick auf ihre Vertrauenswürdigkeit. Hierbei betrachtet man die Wirksamkeit und Korrektheit der eingesetzten Sicherheitsmechanismen. Bei der Wirksamkeit spielt insbesondere die Mindeststärke der kritischen Sicherheitsmechanismen, die man in die Klassen „niedrig“, „mittel“ und „hoch“ einteilt, eine wichtige Rolle. Im Hinblick auf die Korrektheit unterscheidet man die Evaluationsstufen „E1“ bis „E6“ mit jeweils steigender Vertrauenswürdigkeit. Beispielsweise erfolgt ab der Evaluationsstufe E3 auch eine Prüfung des Quellcodes. Die europäischen ITSEC-Kriterien sind in die Entwicklung der international harmonisierten *Common Criteria (CC)* eingeflossen. Für Produkte für *qualifizierte elektronische Signaturen* (mit Anbieterakkreditierung) ist eine Prüfung gemäß ITSEC oder CC und eine nachfolgende *Bestätigung* durch eine Stelle gemäß § 18 [SigG] erforderlich.

### **Informations- und Kommunikationsdienste-Gesetz (IuKD)**

Das IuKD [IuKD] ist ein aus mehreren Artikeln bestehendes Gesetz, in dem neben dem

[SigG97] (Artikel 3 des IuKDG) auch das Teledienstegesetz und das Teledienstedatenschutzgesetz eingeführt wurde.

### **Integrität**

Unter dem Nachweis der Integrität elektronischer Daten versteht man den Nachweis, dass diese vollständig und unverändert sind.

### **International Organization for Standardization (ISO)**

Die ISO (<http://www.iso.org>) ist eine internationale Vereinigung der Standardisierungsgremien von 151 Ländern. Sie verabschiedet internationale Standards in allen technischen Bereichen. Deutschland ist durch das Deutsche Institut für Normung (DIN) (<http://www.din.de>) und die USA durch ANSI in der ISO vertreten.

### **International Telecommunication Union (ITU)**

Die ITU ist eine weltweite Organisation, die sich mit technischen Aspekten der Telekommunikation beschäftigt. In ihrem Telecommunication Standardization Bureau (ITU-T) – früher Comité Consultatif International Téléphonique et Télégraphique (CCITT) – werden technische Normen erarbeitet und als Empfehlung veröffentlicht. Von herausragender Bedeutung für die *elektronische Signatur* ist die Empfehlung X.509, in der u.a. ein weit verbreitetes Format für *Zertifikate* spezifiziert ist.

### **Internet Engineering Task Force (IETF)**

Die Internet Engineering Task Force (IETF) ist eine große, offene, internationale Gemeinschaft, die sich um den reibungslosen Betrieb und die Weiterentwicklung der Internet-Architektur bemüht. Die in der IETF entwickelten Standards und Empfehlungen werden als Request for Comments (RFC) mit einer bestimmten laufenden Nummer unter <http://www.ietf.org> veröffentlicht.

### **Internet Protocol Security (IPSec)**

IPsec ist eine von der *IETF* entwickelte Sicherheitsarchitektur zur Gewährleistung von *Authentizität*, *Integrität* und *Vertraulichkeit* in IP-Netzen. Beispielsweise basiert die Sichere Inter-Netzwerk-Architektur (SINA) [www.bsi.de/fachthem/sina/](http://www.bsi.de/fachthem/sina/) auf IPSec. °

### **Intervall-qualifizierte Zeitstempel**

Intervall-qualifizierte Zeitstempel sind selbst erzeugte *Zeitstempel*, die so mit zwei *qualifizierten Zeitstempeln* verknüpft sind, dass mathematisch bewiesen werden kann (vgl. [Hueh04a]), dass sie nach dem ersten qualifizierten Zeitstempel, aber vor dem zweiten qualifizierten Zeitstempel erstellt wurden.

### **ISIS-MTT**

[ISIS-MTT] ist eine gemeinsame Spezifikation von TeleTrusT e.V. (<http://www.teletrust.de>) und T7 e.V. (<http://www.t7-isis.de>) für *digitale Signaturen*, *Verschlüsselung* und *PKI*. Wesentliches Ziel ist es, durch ISIS-MTT die Voraussetzung für eine internationale Standardisierung und Interoperabilität für Anwendungen auf den genannten Gebieten zu schaffen.

## ISO 9735

Dieser Standard wurde von einer gemeinsamen Arbeitsgruppe von *ISO* und *UN/CEFACT* entwickelt und spezifiziert ein Datenaustauschformat für den elektronischen Austausch strukturierter Daten (*EDIFACT*). ISO9735 besteht aus zehn Teilen [ISO9735-1, ISO9735-2, ISO9735-3, ISO9735-4, ISO9735-5, ISO9735-6, ISO9735-7, ISO9735-8, ISO9735-9, ISO9735-10]. Für die *elektronische Signatur* sind insbesondere die Teile 5 und 6 [ISO9735-5, ISO9735-6] relevant, da in ihnen festgelegt wird, wie Signaturen in *EDIFACT*-Nachrichten integriert werden können und wie spezialisierte *AUTACK*-Nachrichten aufzubauen sind, die zum Schutz der *Authentizität* und *Integrität* sowie zur Bestätigung des Empfangs von *EDIFACT*-Nachrichten dienen.

## ISO 9796

Dieser Standard spezifiziert *Low-Level-Signaturformate*, bei denen ein Teil der Nachricht aus der Signatur rekonstruiert werden kann. Er besteht aus verschiedenen Teilen. Während [ISO9796-1] nach den in [CNS99, CHJ99] präsentierten Attacken komplett zurückgezogen wurde, hat man den Standard [ISO9796-2], in dem auch Signaturen auf Basis des *RSA-Algorithmus* festgelegt sind, entsprechend überarbeitet. Daneben existiert [ISO9796-3], in dem Signaturformate auf Basis *Diskreter Logarithmen* spezifiziert sind.

## Kettenmodell

Das Kettenmodell ist ein *Gültigkeitsmodell* für *digitale Signaturen*, bei dem gefordert ist, dass jede Signatur (die zu prüfende Signatur des Anwenders und alle Signaturen an den *Zertifikaten* im *Zertifizierungspfad*) zu Ihrem Erstellungszeitpunkt auf einem gültigen Zertifikat beruht. Die spätere Sperrung eines *Zertifikates* würde nichts an der Gültigkeit der Signatur ändern. Das Signaturgesetz geht vom Kettenmodell aus, indem es in § 19 Abs. 5 [SigG] vorschreibt, dass die von einem *Zertifizierungsdiensteanbieter* ausgestellten *Zertifikate* von der Einstellung seines Betriebes (und der damit verbundenen Sperrung der CA-Zertifikate) unberührt bleiben. Siehe auch *Schalenmodell* und *Hybridmodell*.

## Körper

Ein Körper ist in der Mathematik eine Menge von Elementen  $K$ , für die zwei Operationen  $+$  und  $\cdot$  definiert sind, so dass  $K$  bezüglich  $+$  eine (additive) *Gruppe* mit neutralem Element  $(\emptyset)$  und die Elemente von  $K$  ohne das Element  $(\emptyset)$  bezüglich  $\cdot$  eine (multiplikative) *Gruppe* bilden, so dass man „ausmultiplizieren“ darf  $a \cdot (b + c) = a \cdot b + a \cdot c$ , also das Distributivgesetz gilt. Weithin bekannte Körper sind die rationalen Zahlen  $\mathbb{Q}$  (Brüche) oder die reellen Zahlen  $\mathbb{R}$  (Brüche und Wurzeln). Ein Körper, der nur endlich viele Elemente hat, wird *endlicher Körper* genannt. In der Kryptographie spielen diese *endlichen Körper* eine wichtige Rolle, da die Berechnung *Diskreter Logarithmen* in ihrer multiplikativen *Gruppe* nach heutigem Kenntnisstand ein schwieriges Problem ist.

## Komplexitätsklasse $\mathcal{P}$

Die Komplexitätsklasse  $\mathcal{P}$  enthält die Menge der Probleme, die mit einem *deterministischen Algorithmus* in *polynomieller Laufzeit* gelöst werden können.

## Komplexitätsklasse $\mathcal{NP}$

Die Komplexitätsklasse  $\mathcal{NP}$  enthält die Menge der Probleme, die mit einem *probabilistischen Algorithmus* in *polynomieller Laufzeit* gelöst werden können. Eine gegebene

Lösung zu einem Problem in der Klasse  $\mathcal{NP}$  kann mit einem *deterministischen Algorithmus in Polynomialzeit* geprüft werden. Es gilt  $\mathcal{P} \subseteq \mathcal{NP}$ . Außerdem ist bekannt (vgl. [Wagn94]), dass jedes Problem aus  $\mathcal{NP}$  auch mit *exponentieller Laufzeit* gelöst werden kann.

### **$L_n[u, v]$ -Funktion**

Zur Beschreibung der Laufzeit von *subexponentiellen Algorithmen* verwendet man die folgendermaßen definierte  $L_n[u, v]$ -Funktion:

$$L_n[u, v] = e^{v(\log(n))^u(\log(\log(n)))^{1-u}}$$

wobei  $\log(n)$  den natürlichen Logarithmus einer Zahl  $n$  bezeichnet, die größer als die Euler'sche Zahl  $e$  ist und  $0 \leq u \leq 1$  sowie  $v > 0$  gilt.

Da  $L_n[0, v] = e^{v(\log(n))^0 \log(\log(n))} = e^{v \log(\log(n))} = \log(n)^v = O(\log(n)^c)$  für eine Konstante  $c > 0$  entspricht,  $L_n[0, v]$  der *Polynomialzeit*. In ähnlicher Weise entspricht  $L_n[1, v] = e^{v \log(n)} = O(n^v)$  der *Exponentialzeit*, und für  $0 < u < 1$  liegt der Aufwand zwischen diesen beiden Extremen.

### **Low-Level-Signaturformat**

Bei Low-Level-Signaturformaten ist bitgenau spezifiziert, wie die zu signierenden Daten, oder ein *Hashwert* derselben, vor der eigentlichen Anwendung des *asymmetrischen Kryptoalgorithmus*, z.B. durch Füllmechanismen (Padding), aufzubereiten sind. Das Format selbst ist maßgeblich vom eingesetzten Kryptoalgorithmus geprägt. Für das auf dem *Faktorisierungsproblem* basierende *RSA*-Verfahren sind beispielsweise die beiden Signaturformate *PKCS #1* und *ISO9796-2* gebräuchlich. Weit verbreitete Signaturformate auf Basis des *Diskreten Logarithmus* sind beispielsweise *DSA* und *ECDSA*.

### **Lightweight Directory Access Protocol (LDAP)**

Mit dem Lightweight Directory Access Protocol (LDAP) [RFC2251] können Informationen, die in einem *Verzeichnisdienst* gespeichert sind, abgerufen oder modifiziert werden. Weitere Details zur Implementierung von LDAP sind auch in [RFC2252, RFC2253, RFC2254, RFC2255, RFC2256] spezifiziert.

### **Message Authentication Code (MAC)**

Ein Message Authentication Code (MAC) dient zur Sicherung der *Integrität* und *Authentizität* einer Nachricht. Anders als bei einer *digitalen Signatur* werden hier aber keine *asymmetrischen Kryptoalgorithmen*, sondern symmetrische Algorithmen und *geheime Schlüssel* zur Erstellung und Prüfung des MACs eingesetzt. Da zur Erstellung und Prüfung der gleiche geheime Schlüssel eingesetzt wird, kann durch einen MAC das Sicherheitsziel der *Nichtabstreitbarkeit* nicht erreicht werden.

### **Multipurpose Internet Mail Extensions (MIME)**

MIME ist ein in [RFC1521] definierter Kodierstandard, der die Struktur und den Aufbau von E-Mails und anderer Internetnachrichten festlegt.

### **Nachsingatur**

Die Sicherheit aller populären Signatursysteme beruht auf der unbewiesenen Annahme, dass die Lösung des zu Grunde liegenden Problems, wie z.B. das *Faktorisierungsproblem* oder das

*Diskrete-Logarithmus Problem*, „schwer“ ist (vgl. *Einwegfunktion*). Die eingesetzten Algorithmen und Parameter können aber durch größere Rechenleistung oder verbesserte Algorithmen ihre Sicherheitseignung verlieren, so dass die mit einer *digitalen Signatur* verbundene Beweiskraft mit der Zeit schwinden würde. Deshalb sieht § 17 [SigV] vor, dass Daten mit einer *qualifizierten elektronischen Signatur* neu zu signieren sind, „wenn diese für längere Zeit in signierter Form benötigt werden, als die für ihre Erzeugung und Prüfung eingesetzten Algorithmen und zugehörigen Parameter als geeignet beurteilt sind. In diesem Falle sind die Daten vor dem Zeitpunkt des Ablaufs der Eignung der Algorithmen oder der zugehörigen Parameter mit einer neuen *qualifizierten elektronischen Signatur* zu versehen. Diese muss mit geeigneten neuen Algorithmen oder zugehörigen Parametern erfolgen, frühere Signaturen einschließen und einen *qualifizierten Zeitstempel* tragen.“

### Nichtabstreitbarkeit

Unter Nichtabstreitbarkeit versteht man die Gewährleistung, dass die Urheberschaft, der Versand oder der Empfang von Daten und Informationen nicht in Abrede gestellt werden können. Die Nichtabstreitbarkeit ist eine Voraussetzung für die *Verbindlichkeit*.

### National Institute for Standards and Technology (NIST)

Das NIST ist ein staatliches Standardisierungsinstitut in den USA. Zu den vom NIST publizierten Standards zählt beispielsweise *DSA* und *SHA-1*.

### O-Notation

Die *O*-Notation verwendet man zur asymptotischen Abschätzung der Laufzeit von Algorithmen. Für zwei Funktionen  $f, g$  schreibt man  $f = O(g)$ , wenn für hinreichend große Werte von  $x$  die Ungleichung  $f(x) \leq c \cdot g(x)$  für einen konstanten Wert  $c$  gilt.

### Optical Character Recognition (OCR)

Unter OCR versteht man die optische Erkennung von Zeichen in Bildern. Beispielsweise verwendet man OCR zur Extraktion von Textinhalten aus eingescannten, ursprünglich papiergebundenen Unterlagen.

### Öffentlicher Schlüssel

Ein öffentlicher Schlüssel ist der Teil eines kryptographischen Schlüsselpaares, der öffentlich bekannt und frei zugänglich ist. Er ist meist Teil eines *Zertifikates* und wird neben der Prüfung *digitaler Signaturen* auch verwendet, um Daten für eine bestimmte Person zu *verschlüsseln*. Nur diese Person kann im Anschluss mit dem zugehörigen, nur ihr bekannten, *privaten Schlüssel* die Daten wieder *entschlüsseln*.

### Online Certificate Status Protocol (OCSP)

OCSP ist ein in [RFC2560] von der *IETF* standardisiertes Client-Server-Protokoll zur Abfrage des Status von *Zertifikaten*. Mittels dieser Online Abfrage kann beispielsweise geprüft werden, ob ein Zertifikat durch den Benutzer gesperrt worden ist.

### Organization for the Advancement of Structured Information Standards (OASIS)

OASIS (<http://www.oasis-open.org/>) ist ein nicht-kommerzielles, globales Konsortium für die Entwicklung und Umsetzung von Standards für eBusiness und *XML*.

### **Online Services Computer Interface (OSCI)**

[OSCI] besteht aus einer Menge von Protokollen und Nachrichtenstrukturen, die insbesondere im E-Government-Umfeld eingesetzt werden. Neben dem generischen OSCI-Transport-Mechanismus, der die *Verschlüsselung* und *digitale Signatur* von Daten sowie einen Laufzettel-Mechanismus vorsieht, wurden eine Reihe von XML-basierten Nachrichtenformaten, wie z.B. [XMeld] oder [XJustiz], standardisiert.

### **Padding**

Unter Padding versteht man allgemein das Ergänzen einer Zeichenfolge um zusätzliche Zeichen, damit eine bestimmte Gesamtlänge erreicht wird. Beispielsweise wird der *Hashwert* einer Nachricht beim RSA-Verfahren aus Sicherheitsgründen um bestimmte Füllzeichen ergänzt, bevor die Signaturerzeugung durch Exponentiation mit dem *privaten Schlüssel* vorgenommen wird. Das Padding kann auf *deterministische* (vgl. [PKCS1(v1.5), ANSI-X9.31]) oder *probabilistische* (vgl. [BeRo96, BeRo98, PKCS1(v2.1)]) Weise geschehen.

### **Personal Security Environment (PSE)**

Ein PSE ist ein Aufbewahrungsmedium für *private Schlüssel* und vertrauenswürdige *Zertifikate*. Ein PSE kann entweder als Software-Lösung, z.B. als mittels Passwort geschützte Datei im *PKCS #12*-Format, oder als Hardware-Lösung, beispielsweise in Form einer *Smart Card*, realisiert sein. In diesem Fall kann das PSE gleichzeitig als *Signaturerstellungseinheit* dienen.

### **Personal Identification Number (PIN)**

Durch eine PIN, eine in der Regel vier- bis achtstellige persönliche Geheimzahl, kann eine *Signaturerstellungseinheit* vor unberechtigtem Zugriff geschützt werden. Eine sechsstellige PIN bietet ausreichend Sicherheit, um die im Signaturgesetz definierten Anforderungen an *Identifikationsdaten* zu erfüllen.

### **Polynomialzeit**

In der Komplexitätstheorie bezeichnet man ein Problem als in Polynomialzeit lösbar, wenn die Rechenzeit mit wachsender Problemgröße höchstens wie eine Polynomfunktion wächst. Man sagt, der Algorithmus besitzt polynomielle Laufzeit. Für die Eingabelänge  $l$  benötigt der Algorithmus nur  $O(l^c)$  Operationen, für ein konstantes  $c$ .

### **Portable Document Format (PDF)**

Das von Adobe Inc. entwickelte Portable Document Format (PDF) [PDF(v1.3), PDF(v1.4), PDF(v1.5), PDF(v1.6)] ist ein Dokumentenformat, das insbesondere für den Austausch von Dokumenten im Internet genutzt wird. Hierbei können Signaturen im *PKCS #1*- und *PKCS #7*-Format so in PDF-Dokumente eingebettet werden, dass sie mit dem kostenlosen Acrobat Reader verifiziert werden können.

### **Pretty Good Privacy (PGP)**

PGP ist ein von Phil Zimmermann entwickeltes Programm zur *Verschlüsselung* und zur Erzeugung *digitaler Signaturen* von Daten unter Verwendung von *asymmetrischen Kryptoalgorithmen*. Die *Authentizität* der *öffentlichen Schlüssel* wird im Regelfall durch ein so genanntes „*Web of Trust*“ sichergestellt – alternativ ist auch die Verwendung von X.509-Zertifikaten möglich. Das PGP-Nachrichtenformat ist in [RFC2440] spezifiziert.

## Primzahl

Eine Primzahl ist eine Zahl, die nur durch die Zahl 1 oder sich selbst teilbar ist. In der Kryptographie spielen Primzahlen insbesondere bei der Konstruktion von Kryptosystemen auf Basis des *Faktorisierungsproblems* oder des *Diskreten Logarithmus* in der multiplikativen *Gruppe endlicher Körper* eine wichtige Rolle.

## Privater Schlüssel

Der private Schlüssel ist der Teil eines kryptographischen Schlüsselpaares, auf den nur der Inhaber des Schlüsselpaares zugreifen kann. Er wird in einem *Personal Security Environment* aufbewahrt und verwendet, um *digitale Signaturen* zu erstellen oder Daten zu *entschlüsseln*.

## Probabilistischer Algorithmus

Bei einem probabilistischen Algorithmus ist die Folge der Schritte bei der Ausführung nicht nur von der Eingabe, sondern auch vom Zufall abhängig. Ist ein Algorithmus nicht vom Zufall abhängig, so spricht man von einem *deterministischen Algorithmus*.

## Public Key Cryptography Standards (PKCS)

PKCS ist eine von den Laboratorien der US-amerikanischen Firma RSA Security Inc. entwickelte Reihe von Standards für Technologien auf Basis von *asymmetrischen Kryptoalgorithmen*. Zu den in der Praxis wichtigsten Standards in dieser Reihe zählen

- PKCS #1: RSA Cryptography Standard

In Version 1.5 dieses Standards [PKCS1(v1.5), RFC2313] ist ein in der Praxis sehr häufig eingesetztes *Low-Level-Signaturformat* auf Basis des *RSA-Algorithmus* spezifiziert. Die aktuelle Version dieses Standards ist [PKCS1(v2.1), RFC3447].

- PKCS #7: Cryptographic Message Syntax Standard

In diesem Standard [PKCS7(v1.5), RFC2315] ist ein sehr weit verbreitetes *High-Level-Signaturformat* spezifiziert. Es wird heute bereits von vielen Standardsoftware-Komponenten unterstützt. Die CMS-Spezifikation der IETF [RFC2630, RFC3369, RFC3852] basiert auf PKCS #7.

- PKCS #11: Cryptographic Token Interface Standard

Der Standard [PKCS11] definiert eine Programmierschnittstelle für den standardisierten Zugriff auf Chipkartenfunktionen.

- PKCS #12: Personal Information Exchange Syntax Standard

[PKCS12] standardisiert ein Datenformat für den Austausch von mittels Passwort verschlüsselten *privaten Schlüsseln*.

## Public-Key-Infrastruktur (PKI)

Eine PKI ist eine technische und organisatorische Infrastruktur, die es ermöglicht, kryptographische Schlüsselpaare (*private Schlüssel* in Form von *PSEs* und *öffentliche Schlüssel* in Form von *Zertifikaten*) auszurollen und zu verwalten. Zu den wesentlichen Kernkomponenten einer PKI zählt die *Registrierungsinstanz*, die *Zertifizierungsinstanz* und der *Verzeichnisdienst*. Unter Umständen umfasst eine PKI auch einen *Zeitstempeldienst* und *Attributbestätigungsinstanzen*.

## Public-Key-Kryptosystem

Public-Key-Kryptosysteme verwenden *asymmetrische Kryptoalgorithmen*.

### **Public-Key-Zertifikat**

Ein Public-Key-Zertifikat ist ein *Zertifikat*, das insbesondere den Namen des Zertifikatsinhabers und den *öffentlichen Schlüssel* enthält.

### **Qualifizierte elektronische Signatur**

Eine qualifizierte elektronische Signatur ist gemäß § 2 Nr. 3 [SigG] eine *fortgeschrittene elektronische Signatur*, die unter Verwendung einer *sicheren Signaturerstellungseinheit* erzeugt wurde und zum Zeitpunkt der Signaturerstellung auf einem gültigen *qualifizierten Zertifikat* beruht. Durch die qualifizierte elektronische Signatur kann die *Schriftform* ersetzt und somit auf kostenintensive Papierprozesse verzichtet werden. Außerdem hat eine qualifizierte elektronische Signatur gemäß § 371a [ZPO] eine sehr hohe Beweiskraft vor Gericht.

### **Qualifizierter Zeitstempel**

Ein qualifizierter Zeitstempel ist gemäß § 2 Nr. 14 [SigG] ein *Zeitstempel*, der von einem *Zertifizierungsdiensteanbieter* gemäß Signaturgesetz ausgestellt wird. Ein solcher Zeitstempel hat eine sehr hohe Beweiskraft vor Gericht. Durch einen qualifizierten Zeitstempel werden die zeitgestempelten Daten quasi „rechtssicher eingefroren“.

### **Qualifiziertes Zertifikat**

Ein qualifiziertes Zertifikat ist gemäß § 2 Nr. 7 [SigG] ein *Zertifikat*, das von einem *Zertifizierungsdiensteanbieter* gemäß Signaturgesetz für natürliche Personen ausgestellt wird. Die detaillierten Inhalte eines qualifizierten Zertifikats ergeben sich aus § 7 [SigG]. Bei der Ausgabe von qualifizierten Zertifikaten müssen die Anforderungen des Signaturgesetzes berücksichtigt werden. Insbesondere muss eine Identifizierung des Signaturschlüsselhabers anhand eines amtlichen Ausweises erfolgen.

### **Random Oracle Model**

Beim Random Oracle Model [BeRo93] geht man von der idealisierten Annahme aus, dass sich eine *Hashfunktion* wie ein „zufälliges Orakel“ verhält. Für jede Frage, die man an das Orakel richtet, erhält man eine scheinbar zufällige Antwort. Unter dieser idealisierten Annahme können Beweise für die Sicherheit populärer Kryptosysteme erbracht werden.

### **Rechtsgeschäft**

Ein Rechtsgeschäft besteht aus mindestens einer *Willenserklärung*. Ein Rechtsgeschäft kann einseitig (z.B. Kündigung) oder mehrseitig (z.B. Vertrag) sein.

### **Registrierungsinstanz**

Eine Registrierungsinstanz (engl. Registration Authority, RA) ist der Bestandteil einer *PKI*, bei dem ein Benutzer ein *Zertifikat* beantragen und ggf. dessen Sperrung veranlassen kann. Im Zuge des erstmaligen Registrierungsprozesses werden die Identität des Antragstellers und möglicherweise zusätzliche *Attribute* überprüft, so dass die Korrektheit der Angaben im *Zertifikat* gewährleistet ist.

### **Regulierungsbehörde für Telekommunikation und Post (RegTP)**

Die RegTP, inzwischen umbenannt in *Bundesnetzagentur (BNetzA)*, ist gemäß § 19 [SigG] zuständig für die Aufsicht über die Einhaltung des Signaturgesetzes. Ein

*Zertifizierungsdiensteanbieter* muss die Aufnahme des Betriebes bei der RegTP anzeigen oder sich von ihr akkreditieren lassen. Analog dazu müssen Produkthersteller spätestens mit dem Inverkehrbringen des Produktes gemäß § 17 Abs. 4 [SigG] eine *Herstellererklärung* bei der RegTP hinterlegen oder ihr Produkt prüfen und bestätigen lassen.

### **Restklasse**

Die Restklasse einer Zahl  $a$  modulo einer als Modul bezeichneten Zahl  $m$  enthält die Menge aller Zahlen, die bei Division durch  $m$  den selben Rest lassen wie  $a$ . Alle in einer Restklasse enthaltenen Zahlen unterscheiden sich also um ein ganzzahliges Vielfaches des Modul voneinander. Deshalb schreibt man  $a + m\mathbb{Z}$  für die Restklasse von  $a$  modulo  $m$ .

### **RIPEMD-160**

RIPEMD-160 [DoBP96, ISO10118-3] ist ein im Rahmen des EU-geförderten RIPE-Projektes (RACE Integrity Primitives Evaluation, 1988-1992) von Hans Dobbertin, Antoon Bosselaers und Bart Preneel entwickelter *Hashalgorithmus*.

### **Root-CA**

Siehe *Wurzel-Zertifizierungsinstanz*.

### **RSA**

Der nach seinen Erfindern (Rivest, Shamir und Adleman) benannte RSA-Algorithmus [RSA78] ist ein *asymmetrischer Kryptoalgorithmus*, der zur *Verschlüsselung* und zur Realisierung *digitaler Signaturen* verwendet werden kann. Die Sicherheit dieses Verfahrens basiert auf der kryptographischen Annahme, dass das *Faktorisierungsproblem* für große Zahlen nicht effizient gelöst werden kann.

### **Schalenmodell**

Das Schalenmodell ist ein *Gültigkeitsmodell* für *digitale Signaturen*, bei dem eine Signatur genau dann als gültig erachtet wird, wenn zum Zeitpunkt der *Prüfung* der Signatur alle *Zertifikate* im *Zertifizierungspfad* gültig waren. Das Schalenmodell ist insbesondere für die Prüfung von digitalen Signaturen zum Zweck der *Authentifizierung* geeignet. Da aber bei einer *qualifizierten elektronischen Signatur* gemäß § 2 Nr. 3 [SigG] nicht der Zeitpunkt der Prüfung, sondern der Zeitpunkt der Erstellung der Signatur entscheidend ist, ist das Schalenmodell für qualifizierte elektronische Signaturen nicht ohne weiteres geeignet. Siehe auch *Hybridmodell* und *Kettenmodell*.

### **Schriftform**

Die Schriftform ist im privatrechtlichen Bereich in § 126 [BGB] normiert. Sie setzt voraus, dass eine schriftliche Urkunde vom Aussteller eigenhändig unterschrieben wird. Die Schriftform kann nach § 126 Satz 3 [BGB] regelmäßig durch die so genannte *elektronische Form* ersetzt werden, die gemäß § 126a [BGB] mittels *qualifizierter elektronischer Signaturen* erreicht wird. Somit kann durch den Einsatz der qualifizierten elektronischen Signatur die Schriftform ersetzt und auf kostenintensive Papierprozesse verzichtet werden.

### **Secure Socket Layer (SSL)**

SSL ist ein ursprünglich von Netscape entwickeltes Protokoll zur sicheren Übertragung von Daten, das vor allem für die sichere Übertragung von Webseiten zwischen Web-Server und

Browser eingesetzt wird.

### **SHA-1**

Der Secure Hash Algorithm (SHA-1) [FIPS180-2] ist ein von der US-amerikanischen Sicherheitsbehörde NSA entwickelter *Hashalgorithmus*, der 160 Bit *Hashwerte* produziert.

### **Sichere Signaturerstellungseinheit (SSEE)**

Eine sichere Signaturerstellungseinheit ist gemäß § 2 Nr. 10 [SigG] eine *Signaturerstellungseinheit*, die den anspruchsvollen Anforderungen des Signaturgesetzes, insbesondere § 17 Abs. 1 [SigG] und § 15 Abs. 1 [SigV], genügt. Dies beinhaltet beispielsweise, dass selbst der Signaturschlüsselhaber seinen *privaten Schlüssel* nicht aus der Signaturerstellungseinheit auslesen und veröffentlichen kann, um das Sicherheitsziel der *Nichtabstreichbarkeit* zu umgehen. Dass die Anforderungen des Signaturgesetzes von einer sicheren Signaturerstellungseinheit erfüllt werden, muss durch eine anspruchsvolle Prüfung nach international anerkannten Sicherheitskriterien ([ITSEC] E3 hoch, [CC] EAL4+) und eine Bestätigung gemäß Signaturgesetz nachgewiesen werden (vgl. *geprüfte und bestätigte Produkte*). Alle heute verfügbaren sicheren Signaturerstellungseinheiten sind *Smart Cards*.

### **Signaturalgorithmus**

Ein Signaturalgorithmus ist ein *asymmetrischer Kryptoalgorithmus*, der zur Erzeugung *digitaler Signaturen* verwendet wird. Zu den populärsten Signaturalgorithmen zählen *RSA*, *DSA* und *ECDSA*.

### **Signaturanwendungskomponente**

Signaturanwendungskomponenten sind gemäß § 2 Nr. 11 [SigG] Software- und Hardwareprodukte, die dazu bestimmt sind, Daten dem Prozess der Erzeugung oder Prüfung *qualifizierter elektronischer Signaturen* zuzuführen oder *qualifizierte elektronische Signaturen* zu prüfen oder *qualifizierte Zertifikate* nachzuprüfen und die Ergebnisse anzuzeigen.

### **Signaturerstellungseinheit**

Eine Signaturerstellungseinheit ist eine Hardware oder Software, in der *private Schlüssel*, die zur Erstellung von Signaturen erforderlich sind, wie in einem *PSE* aufbewahrt und darüber hinaus auch angewandt werden können. Als Signaturerstellungseinheit kommen *Smart Cards*, *HSMs* oder Standard-Rechner-Systeme in Frage, wobei der *private Schlüssel* beispielsweise in einer mittels Passwort verschlüsselten Datei im *PKCS #12*-Format gespeichert wird. Zur Erstellung von *qualifizierten elektronischen Signaturen* sind *sichere Signaturerstellungseinheiten* nötig.

### **Signaturformat**

Damit die Prüfung einer *digitalen Signatur* durch den Empfänger einer signierten Nachricht geschehen kann, muss der Absender die Signatur unter Verwendung eines standardisierten Signaturformats erstellen. Hierbei unterscheidet man zwischen rohen *Low-Level-Signaturformaten* und erweiterten *High-Level-Signaturformaten*. Der grobe Unterschied ist, dass die *High-Level-Signaturformate* neben den rohen Signaturdaten auch weitere Informationen enthalten können: z.B. den Zeitpunkt der Signaturerstellung und das zur Prüfung der Signatur notwendige *Zertifikat* oder einen Verweis darauf.

## Signaturprüfung

Die Signaturprüfung umfasst zwei verschiedene Prüfschritte. Beim ersten Prüfschritt wird die mathematische Gültigkeit der *Low-Level-Signatur* zum Nachweis der *Integrität* geprüft. Im zweiten Schritt wird für den Nachweis der *Authentizität* die Gültigkeit der gesamten Signatur im Hinblick auf das zugrundeliegende Gültigkeitsmodell geprüft. Dieser unter Umständen komplexe Vorgang umfasst die Prüfung, ob das zur Signaturerstellung verwendete *Zertifikat* zum Referenzzeitpunkt – d.h. bei einer *qualifizierten elektronischen Signatur* der Zeitpunkt der Signaturerstellung bzw. bei einer zur *Authentifizierung* verwendeten Signatur der aktuelle Zeitpunkt – gültig ist. Bei der Prüfung der Gültigkeit eines Zertifikates wird hinterfragt, ob die durch die ausstellende Instanz erstellte *Low-Level-Signatur* mathematisch gültig ist, ob die in [RFC3280] definierten Zertifikatserweiterungen richtig gesetzt sind, ob letztlich ein Zertifikatspfad zu einem vertrauenswürdigen Wurzelzertifikat gebildet werden kann und ob das *Zertifikat* nicht gesperrt wurde.

## Smart Card

Eine Chipkarte mit integriertem Prozessor wird auch als Smart Card bezeichnet. Sie kann als *PSE* dienen, um vertrauenswürdige *Zertifikate* und *private Schlüssel* sicher aufzubewahren, und darüber hinaus auch als (*sichere*) *Signaturerstellungseinheit* fungieren.

## S/MIME

Secure Multipurpose Internet Mail Extensions (S/MIME) ist ein ursprünglich von den Laboratorien der RSA Security Inc. entwickeltes, nunmehr in der IETF in [RFC2632, RFC2633] standardisiertes, auf *PKCS #7/CMS* aufbauendes, Format für die *Verschlüsselung* und *Signatur* von E-Mails und E-Mail-Anhängen im MIME-Format [RFC1521].

## Software Publishing Certificate (SPC)

Ein SPC ist eine im Rahmen der Microsoft Authenticode Technologie verwendete Datenstruktur, die bei der Signatur von Programm-Code eingesetzt wird. Es besteht aus einer *PKCS #7*-Struktur, in der sich ein oder mehrere X.509-Zertifikate befinden.

## Sperrliste

Eine Sperrliste wird durch eine *Zertifizierungsinstanz* erstellt und in einem *Verzeichnisdienst* veröffentlicht. Sie beinhaltet Informationen darüber, welche *Zertifikate* durch den Zertifikatsinhaber oder andere berechtigte Stellen gesperrt (revoziert) worden sind. Ein weithin akzeptiertes Format für Sperrlisten wurde in X.509 spezifiziert und in [RFC3280] näher profiliert.

## Subexponentielle Laufzeit

Ein Algorithmus, dessen asymptotische Laufzeit geringer als *exponentiell* ist, besitzt subexponentielle Laufzeit. Für die Beschreibung der Laufzeit dieser Algorithmen verwendet man häufig die  $L_n[u, v]$ -Funktion.

## Tagged Image File Format (TIFF)

TIFF ist ein Dateiformat zur Speicherung von Bilddaten, das häufig beim Einscannen von Schriftgut eingesetzt wird. Die Details des Formates finden sich in der Spezifikation [TIFF(v6.0)].

### **Target of Evaluation (TOE)**

Siehe *Evaluationsgegenstand (EVG)*.

### **Textform**

Für die in § 126b [BGB] definierte Textform „muss die Erklärung in einer Urkunde oder auf andere zur dauerhaften Wiedergabe in Schriftzeichen geeignete Weise abgegeben, die Person des Erklärenden genannt und der Abschluss der Erklärung durch Nachbildung der Namensunterschrift oder anders erkennbar gemacht werden.“ Beispielsweise kann die Textform durch ein unsigniertes elektronisches Dokument, z.B. im *PDF*-Format, erreicht werden.

### **Time - Stamp Protocol (TSP)**

TSP ist ein in [RFC3161] von der *IETF* standardisiertes Client-Server-Protokoll zur Ausstellung von *Zeitstempeln*.

### **Transmission Control Protocol (TCP)**

Das in [RFC793] spezifizierte TCP ist ein zuverlässiges, verbindungsorientiertes Transportprotokoll in Rechnernetzen, das auch im Internet zum Einsatz kommt.

### **Trust-Center**

Im Umfeld der *elektronischen Signatur* wird der Begriff „Trust-Center“ häufig als Synonym für die von einem *Zertifizierungsdiensteanbieter* betriebenen Infrastrukturen verwendet.

### **United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT)**

Das von den Vereinten Nationen getragene CEFACHT <http://www.unece.org/cefact/> zielt auf die Förderung, Vereinfachung und Harmonisierung des internationalen Handels ab. Es ist u.a. verantwortlich für den internationalen Datenstandard *EDIFACT* und einer der Initiatoren von *ebXML*.

### **Untergruppe**

Eine Untergruppe  $U$  einer Gruppe  $G$  ist eine Teilmenge von  $G$ , die mit der Gruppenverknüpfung von  $G$  wiederum eine Gruppe bildet. Die Ordnung einer solchen Untergruppe ist immer ein Teiler der *Gruppenordnung* (vgl. [Buch99, Theorem 2.10.2]).

### **Urkunde**

Eine Urkunde ist eine verkörperte Gedankenerklärung – in der Regel ein Schriftstück –, die zum Beweis im Rechtsverkehr geeignet und bestimmt ist und einen Aussteller erkennen lässt. Stimmt der aus der Urkunde ersichtliche Aussteller der Erklärung nicht mit dem tatsächlichen Hersteller der Urkunde überein, so ist der Tatbestand der Urkundenfälschung gemäß § 267 [StGB] erfüllt. Für den Beweis mittels Urkunden, den so genannten *Urkundsbeweis*, sieht die Zivilprozessordnung bestimmte Regelungen vor, die ein beschleunigtes Verfahren ermöglichen.

### **Urkundsbeweis**

Im Zivilprozess wird hinsichtlich des Beweiswerts zwischen privaten und öffentlichen Urkunden unterschieden. Die private Urkunde (§ 416 [ZPO]) erbringt nur den Beweis, dass der Aussteller die in ihr enthaltene Erklärung abgegeben hat. Dagegen beweist die öffentliche

Urkunde (§ 415 [ZPO]) auch den in ihr beurkundeten Vorgang.

Der Beweis wird jeweils durch Vorlegen der Urkunde angetreten (§ 420 [ZPO]). Bei einer privaten Urkunde muss der Beweisgegner eine Erklärung zur Echtheit der Urkunde und ggf. zur Echtheit der Unterschrift auf der Urkunde abgeben (§ 439 [ZPO]). Wird die Echtheit der Urkunde anerkannt, so ist die diesbezügliche Beweisaufnahme abgeschlossen. Wird die Echtheit der Urkunde bestritten, so muss der Beweisgegner die Echtheit – möglicherweise durch Schriftvergleich (§ 441 [ZPO]) – beweisen (§ 440 [ZPO]).

Bei öffentlichen Urkunden ist die Beweisaufnahme in der Regel noch einfacher, da diese die Vermutung der Echtheit für sich haben (§ 437 [ZPO]).

### **Verbindlichkeit**

Unter Verbindlichkeit versteht man, dass ein *Rechtsgeschäft* seine rechtliche Wirkung entfaltet. Voraussetzungen hierfür ist teilweise die Einhaltung von Formerfordernissen (z.B. *Schriftform*). Zur Wirksamkeit ist weiterhin das Vorhandensein von Beweismitteln erforderlich.

### **Verschlüsselung**

Bei der Verschlüsselung wird ein Klartext unter Verwendung eines symmetrischen oder asymmetrischen *Kryptoalgorithmus* und *geheimen* bzw. *öffentlichen Schlüsseln* in einen Geheimtext umgewandelt, so dass die ursprüngliche Nachricht vor unbefugter Einsicht geschützt ist. Der Empfänger der Nachricht kann diese *entschlüsseln*, um sie wieder lesbar zu machen.

### **Vertraulichkeit**

Vertraulichkeit zielt darauf ab, die unberechtigte Kenntnisnahme und Preisgabe von Informationen zu verhindern.

### **Verzeichnisdienst**

Ein Verzeichnisdienst ist Bestandteil einer *PKI* und wird zur Veröffentlichung von *Zertifikaten* und Zertifikatstatusinformationen in Form von *Sperrlisten* oder *OCSP*-Antworten verwendet.

### **Virtuelles Privates Netz (VPN)**

Bei einem VPN wird unter Verwendung kryptographischer Mechanismen und öffentlicher Transportnetze (z.B. Internet) ein virtuelles privates Netz geschaffen, in dem die Teilnehmer so sicher wie in einem lokalen Netz kommunizieren können. Für hohe Sicherheitsansprüche empfiehlt sich der Einsatz des *IPSec*-basierten, im Auftrag des Bundesamtes für Sicherheit in der Informationstechnik entwickelten SINA-VPN (vgl. [www.bsi.de/fachthem/sina/](http://www.bsi.de/fachthem/sina/)).

### **Virtuelle Zertifizierungsstelle**

Tritt jemand als *Zertifizierungsdiensteanbieter* auf, ohne selbst die dazu notwendigen *PKI*-Komponenten zu betreiben, so spricht man von einer „virtuellen Zertifizierungsstelle“ oder einem „virtuellen Trust-Center“.

### **Web of Trust**

Das dem *PGP*-Verfahren zu Grunde liegende „Web of Trust“ [Stal95] soll die *Authentizität* öffentlicher Schlüssel sicher stellen. Anders als bei einer herkömmlichen *Public-Key*-

*Infrastruktur* wird die Echtheit der öffentlichen Schlüssel allerdings nicht durch eine zentrale *Zertifizierungsinstanz*, sondern durch die *PGP*-Benutzer selbst beglaubigt.

### Willenserklärung

Eine Willenserklärung ist eine Äußerung eines auf die Herbeiführung einer Rechtswirkung gerichteten Willens (siehe [Pala04, Einführung vor §116] und [Bert01, Kapitel 2.2]). Sie kann als ausdrückliche Erklärung, durch schlüssiges Handeln oder sogar durch Schweigen kund getan werden. Während im deutschen Recht grundsätzlich die Formfreiheit gilt, so bedürfen bestimmte Rechtsgeschäfte zu ihrer Gültigkeit einer bestimmten Form, wie z.B. der *Schriftform*, da sie sonst gemäß § 125 [BGB] nichtig wären.

### World Wide Web Consortium (W3C)

Das World Wide Web Consortium (W3C) (<http://www.w3.org>) entwickelt Spezifikationen, Leitfäden, Software und Werkzeuge, die förderlich sind das Potenzial des Webs zu erschließen.

### Wurzel-Zertifizierungsinstanz

Eine Wurzel-Zertifizierungsinstanz (engl. Root-CA) ist eine *Zertifizierungsinstanz*, deren Zertifikat als vertrauenswürdig gilt.

### X.500

[X.500] ist eine von der *ITU* entwickelte Empfehlung für einen (globalen) *Verzeichnisdienst*, bei dem die Einträge in einem hierarchischen Verzeichnisbaum, dem so genannten „*Directory Information Tree (DIT)*“, angeordnet sind und durch ihren *Distinguished Name* adressiert werden. Für den Zugriff auf die Einträge in diesem Verzeichnis ist das in [X.519] spezifizierte „*Directory Access Protocol (DAP)*“ vorgesehen. Da dieses Protokoll vergleichsweise komplex ist, verwendet man für den Zugriff auf Einträge in Verzeichnisdiensten heute meist das einfachere *Lightweight Directory Access Protocol (LDAP)*.

### X.509

X.509 ist eine von der *ITU* entwickelte Empfehlung [X.509:97, X.509:00] für ein Rahmenwerk zur *Authentifizierung* unter Verwendung *asymmetrischer Kryptoalgorithmen*. In diesem Standard werden insbesondere auch sehr weit verbreitete Formate für *Zertifikate* und *Sperrlisten* spezifiziert.

### XML Digital Signature (XML-DSig)

Für die *digitale Signatur* von Daten im *XML*-Format wurde von einer Arbeitsgruppe des *W3C* ein spezifisches Signaturformat entwickelt [XML-DSig, RFC3275]. Im Vergleich zum generischen Signaturformat *PKCS #7*, mit dem Daten beliebigen Formats signiert werden können, bietet XML-DSig ein höheres Maß an Flexibilität, das notwendig ist, um das volle Potenzial von XML auch im Bereich der digitalen Signatur ausnutzen zu können.

### Zahlkörpersieb

Das Zahlkörpersieb ist der leistungsfähigste heute bekannte Algorithmus zur *Faktorisierung* großer Zahlen und zur Berechnung *Diskreter Logarithmen* in multiplikativen *endlichen Körpern*. Er hat eine erwartete Laufzeit von  $L_n[1/3, (64/9)^{1/3} + o(1)]$ .

## **Zeitstempel**

Zeitstempel sind gemäß [ISO18014-1] digitale Daten, mit denen die Existenz bestimmter Daten vor einem bestimmten Zeitpunkt bewiesen werden kann. Häufig, wie z.B. beim *Time Stamp Protocol* aus [RFC3161], werden Zeitstempel unter Einsatz *digitaler Signaturen* erstellt. Somit sind Zeitstempel elektronische Bescheinigung darüber, dass die mit dem Zeitstempel signierten Daten zum Zeitpunkt der Signatur in der signierten Form vorgelegen haben. Im Hinblick auf die mit einem Zeitstempel verbundene Beweiskraft unterscheidet man einfache (selbst erzeugte) Zeitstempel und *qualifizierte Zeitstempel*, die von *Zertifizierungsdiensteanbietern* gemäß Signaturgesetz ausgestellt werden. Darauf hinaus kann man *Intervall-qualifizierte Zeitstempel* durch die geschickte Kombination von einfachen und qualifizierten Zeitstempeln konstruieren.

## **Zeitstempeldienst**

Ein Zeitstempeldienst stellt *Zeitstempel* aus. Oft wird hierbei das in der IETF spezifizierte *Time Stamp Protocol* verwendet.

## **Zertifikat**

Zertifikate sind elektronische Bescheinigungen, die von einer *Zertifizierungsinstanz* ausgestellt (signiert) werden, mit denen dem Zertifikatsinhaber bestimmte Informationen zugeordnet werden. Hierbei unterscheidet man zwischen *Public-Key-Zertifikaten*, bei denen dem Zertifikatsinhaber insbesondere ein *öffentlicher Schlüssel* zugeordnet wird und *Attributzertifikaten*. Das gebräuchlichste Format für Zertifikate ist X.509.

## **Zertifizierungsinstanz**

Eine Zertifizierungsinstanz (engl. Certification Authority, CA) stellt *Zertifikate* aus, indem sie die Zertifikatsinhalte mit einer *digitalen Signatur* versieht. Meist stellt eine Zertifizierungsinstanz auch *Sperrlisten* aus, die in ähnlicher Art und Weise signiert werden.

## **Zertifizierungsstelle**

Der Begriff der Zertifizierungsstelle war in § 2 Abs. 2 [SigG97] definiert als eine „natürliche oder juristische Person, die die Zuordnung von *öffentlichen Signaturschlüsseln* zu natürlichen Personen bescheinigt und dafür eine Genehmigung gemäß § 4 [SigG97] besitzt.“ Im Zuge der Überarbeitung des Signaturgesetzes wurde dieser Begriff durch den Begriff des *Zertifizierungsdiensteanbieters* ersetzt.

## **Zertifizierungsdiensteanbieter (ZDA)**

Ein Zertifizierungsdiensteanbieter ist gemäß § 2 Nr. 8 [SigG] eine natürliche oder juristische Person, die *qualifizierte Zertifikate* oder *qualifizierte Zeitstempel* ausstellt. Ein ZDA muss die Aufnahme des Betriebes bei der *BNetzA anzeigen* oder sich *akkreditieren* lassen.

## **Zertifizierungspfad $i$**

Ein Zertifizierungspfad besteht aus einer Kette von Zertifikaten  $Z_1 - Z_2 - \dots - Z_n$ , wobei für alle  $i$  von 1 bis  $n - 1$  der Eigentümer von  $Z_{i+1}$  das Zertifikat  $Z_i$  ausgestellt hat und  $Z_n$  das Zertifikat einer vertrauenswürdigen *Wurzel-Zertifizierungsinstanz* ist.



## Literatur

- [77/388/EWG] Sechste Richtlinie 77/388/EWG des Rates vom 17. Mai 1977 zur Harmonisierung der Rechtsvorschriften der Mitgliedstaaten über die Umsatzsteuern - Gemeinsames Mehrwertsteuersystem: einheitliche steuerpflichtige Bemessungsgrundlage. Abl. EG Nr. L 145 vom 13.06.1977.
- [94/820/EG] Empfehlung der Kommission vom 19. Oktober 1994 über die rechtlichen Aspekte des elektronischen Datenaustausches (Text von Bedeutung für den EWR). Abl. EG Nr. L 338 vom 28.12.1994.
- [98/0191(COD)] Vorschlag für eine Richtlinie des Europäischen Parlaments und des Rates über gemeinsame Rahmenbedingungen für elektronische Signaturen.  
[http://europa.eu.int/eur-lex/pri/de/oj/dat/1998/c\\_325/c\\_32519981023de00050011.pdf](http://europa.eu.int/eur-lex/pri/de/oj/dat/1998/c_325/c_32519981023de00050011.pdf), 1998.
- [1999/93/EG] *Richtlinie 1999/93/EG des Europäischen Parlaments und des Rates vom 13. Dezember 1999 über gemeinschaftliche Rahmenbedingungen für elektronische Signaturen.* [http://europa.eu.int/eur-lex/pri/de/oj/dat/2000/l\\_013/l\\_01320000119de00120020.pdf](http://europa.eu.int/eur-lex/pri/de/oj/dat/2000/l_013/l_01320000119de00120020.pdf), 2000.
- [1999/C-243/02] *Gemeinsamer Standpunkt (EG) Nr. 28/1999 vom Rat festgelegt am 28. Juni 1999 im Hinblick auf den Erlaß der Richtlinie 1999/.../EG des Europäischen Parlaments und des Rates über gemeinsame Rahmenbedingungen für elektronische Signaturen.* [http://europa.eu.int/eur-lex/pri/de/oj/dat/1999/c\\_243/c\\_24319990827de00330046.pdf](http://europa.eu.int/eur-lex/pri/de/oj/dat/1999/c_243/c_24319990827de00330046.pdf), 1999.
- [2001/115/EG] *Richtlinie 2001/115/EG des Rates vom 20. Dezember 2001 zur Änderung der Richtlinie 77/388/EWG mit dem Ziel der Vereinfachung, Modernisierung und Harmonisierung der mehrwertsteuerlichen Anforderungen an die Rechnungstellung.* Abl. EG v. 17.01.02 Nr. L 15/24. [http://europa.eu.int/eur-lex/pri/de/oj/dat/2002/l\\_015/l\\_01520020117de00240028.pdf](http://europa.eu.int/eur-lex/pri/de/oj/dat/2002/l_015/l_01520020117de00240028.pdf), 2001.
- [2003/511/EG] *Entscheidung der Kommission vom 14. Juli 2003 über die Veröffentlichung von Referenznummern für allgemein anerkannte Normen für Produkte für elektronische Signaturen gemäß der Richtlinie 1999/93/EG des Europäischen Parlaments und des Rates.* [http://www.dfn-pca.de/bibliothek/sigg/europe/l\\_17520030715de00450046.pdf](http://www.dfn-pca.de/bibliothek/sigg/europe/l_17520030715de00450046.pdf), 2003.
- [AKS04] MANINDRA AGRAWAL, NEERAJ KAYAL, NITIN SAXENA. *PRIMES is in P.* In *Annals of Mathematics*, Band 160(2):781–793.  
[http://www.cse.iitk.ac.in/users/manindra/algebra/primality\\_v6.pdf](http://www.cse.iitk.ac.in/users/manindra/algebra/primality_v6.pdf)
- [Albr03] ASTRID ALBRECHT. *Biometrische Verfahren im Spannungsfeld von Authentizität im elektronischen Rechtsverkehr und Persönlichkeitsschutz*. (Nomos Verlag, 2001).
- [AMV90] G. AGNEW, R.MULLIN, S.VANSTONE. *Improved digital signature scheme based on discrete exponentiation.* In *Electronic Letters*, Band 26:1024–1025, 1990.
- [ANSI-X9.31] AMERICAN NATIONAL STANDARDS INSTITUTE (ANSI). *Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA).* Public Key Cryptography for the Financial Services Industry – X9.31, September 1998.

- [ANSI-X9.62] AMERICAN NATIONAL STANDARDS INSTITUTE (ANSI). *The Elliptic Curve Digital Signature Algorithm (ECDSA)*. Public Key Cryptography for the Financial Services Industry – X9.62, 1998.
- [ANSI-X9.68] AMERICAN NATIONAL STANDARDS INSTITUTE (ANSI). *Digital Certificates for Mobile/Wireless and High Transaction Volume Financial Systems: Part 2: Domain Certificate Syntax*. Public Key Cryptography for the Financial Services Industry – X9.68, 2001.
- [AO] *Abgabenordnung*. vom 16. März 1976, BGBI I 1976, 613 (1977, 269), zuletzt geändert durch Art. 13 G v. 22. 3.2005 I 837.  
[http://bundesrecht.juris.de/bundesrecht/ao\\_1977/](http://bundesrecht.juris.de/bundesrecht/ao_1977/), 1976.
- [BaSh96] ERIC BACH JEFFREY SHALLIT. *Algorithmic Number Theory*, 1 (MIT Press, 1996).
- [BCCN01] E. BRIER, C. CLAVIER, J.S. CORON, D. NACCACHE. *Cryptanalysis of RSA signatures with fixed pattern padding*. In JOE KILIAN (Herausgeber), *Advances in Cryptology – CRYPTO 2001*, Band 2139 von *Lecture Notes in Computer Science*, Seiten 276–292 (Springer-Verlag, 2001).
- [BDSG] *Bundesdatenschutzgesetz*. vom 20. Dezember 1990, BGBI I 1990, 2954, 2955, zuletzt geändert durch G v. 5. 9.2005 I 2722.  
[http://bundesrecht.juris.de/bundesrecht/bdsg\\_1990/](http://bundesrecht.juris.de/bundesrecht/bdsg_1990/), 1990.
- [BeRo93] MIHIR BELLARE PAUL ROGAWAY. *Random Oracles are Practical: a Paradigm for Designing Efficient Protocols*. In *1st ACM Conference on Computer and Communications Security*, Seiten 62–73 (1993).
- [BeRo96] MIHIR BELLARE PAUL ROGAWAY. *The Exact Security of Digital Signatures - How to Sign with RSA and Rabin*. In *Advances in Cryptology – EUROCRYPT ’96*, Band 1070 von *Lecture Notes in Computer Science*, Seiten 399–416 (Springer-Verlag, 1996).
- [BeRo98] MIHIR BELLARE PAUL ROGAWAY. *PSS: Provably Secure Encoding Method for Digital Signatures*. Einreichung zur IEEE P1363 Arbeitsgruppe.  
<http://grouper.ieee.org/groups/1363/>, 1998.
- [Bert01] ANDREAS BERTSCH. *Digitale Signaturen*. Xpert.press (Springer Verlag, 2001). ISBN 3-540-42351-6.
- [Beta93] BETA SYSTEMS. *Beta 93 - Hochperformante Archivierung und Verteilung für unternehmensweite Verfügbarkeit der Dokumente*. Webseite.  
<http://www2.betasystems.com/de/portfolio/outputmanagement/beta93.html>, 2005.
- [BGB] *Bürgerliches Gesetzbuch*. RGBI 1896, 195, Neugefasst durch Bek. v. 2. 1.2002 I 42, 2909; 2003, 738; zuletzt geändert durch Art. 1 G v. 21. 4.2005 I 1073. <http://bundesrecht.juris.de/bundesrecht/bgb/>, 1896.
- [BGH-ERV] BUNDESGERICHTSHOF. *Elektronischer Rechtsverkehr*. Webseite.  
[http://www.bundesgerichtshof.de/presse/elek\\_rechtsverkehr.php](http://www.bundesgerichtshof.de/presse/elek_rechtsverkehr.php), 2005.
- [BioFace] BUNDESKRIMINALAMT (BKA) UND FRAUNHOFER GESELLSCHAFT BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK (BSI). *BioFace – Vergleichende Untersuchung von Gesichtserkennungssystemen*. <http://www.bsi.de/literat/studien/BioFace/BioFaceIIBericht.pdf>, Juni 2003.

- [BioFinger] BUNDESKRIMINALAMT (BKA) UND FRAUNHOFER GESELLSCHAFT  
BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK (BSI).  
*Evaluierung biometrischer Systeme Fingerabdrucktechnologien – BioFinger.* [http://www.bsi.de/literat/studien/BioFinger/BioFinger\\_I\\_I.pdf](http://www.bsi.de/literat/studien/BioFinger/BioFinger_I_I.pdf), August 2004.
- [bIT4health] BUNDESMINISTERIUM FÜR GESUNDHEIT UND SOZIALE SICHERUNG. *Die elektronische Gesundheitskarte.* <http://www.bit4health.de>, 2005.
- [Blei96] DANIEL BLEICHENBACHER. *Generating EIGamal Signatures Without Knowing the Secret Key.* In *Advances in Cryptology – EUROCRYPT ’96*, Band 1070 von *Lecture Notes in Computer Science*, Seiten 10–18 (Springer-Verlag, 1996).
- [BLP93] JOE BUHLER, HENDRIK LENSTRA, CARL POMERANCE. *Factoring integers with the number fields sieve.* In A.K. LENSTRA H.W. LENSTRA (Herausgeber), *The Developement of the Number Field Sieve*, Band 1554, Seiten 50–94 (Springer-Verlag, 1993).
- [BMF04] BUNDESFINANZMINISTERIUM. *BMF-Schreiben vom 29. Januar 2004, IV B7 - S7280 - 19/04.* BStBl. I 2004.  
[http://www.secunet.com/download/k/pki\\_bmf-29-01-2004.pdf](http://www.secunet.com/download/k/pki_bmf-29-01-2004.pdf), 2004.
- [BMF04b] BUNDESFINANZMINISTERIUM. *BMF-Schreiben vom 29. November 2004, IV A 6 - S 7340 - 37/04, IV C 5 - S 2377 - 24/04.* BStBl. I 2004.  
<https://www.elster.de/eportal/download/28014.pdf>, 2004.
- [BMF05] BUNDESFINANZMINISTERIUM. *BMF-Schreiben vom 19. Juli 2005, IV A5 - S 7287a - 23/05*, 2005.
- [BMF92] BUNDESFINANZMINISTERIUM. *BMF-Schreiben vom 25. Mai 1992, IV A 2 - S 7280 - 8/92.* BStBl. I 1992 S. 376, 1992.
- [BMI95] BUNDESMINISTERIUM DES INNEREN (BMI). *Verordnung über die Anerkennung von Verfahren zur elektronischen Unterschrift (Verordnung über die elektronische Unterschrift – VEU).* Computerrecht (CR), Seiten 319–324, 1996.
- [BMI96] BUNDESMINISTERIUM DES INNEREN (BMI). *Signaturgesetz – Vorentwurf.* Computerrecht (CR), 578–580, 1996.
- [BNetzA-Alg05] BUNDESNETZAGENTUR (FRÜHER REGULIERUNGSBEHÖRDE FÜR TELEKOMMUNIKATION UND POST). *Bekanntmachung zur elektronischen Signatur nach dem Signaturgesetz und der Signaturverordnung (Übersicht über geeignete Algorithmen).* veröffentlicht am 30. März 2005 im Bundesanzeiger Nr. 59, S. 4695-4696.  
<http://www.bundesnetzagentur.de/media/archive/1507.pdf>, 2005.
- [BNetzA-GMod] BUNDESNETZAGENTUR (FRÜHER REGULIERUNGSBEHÖRDE FÜR TELEKOMMUNIKATION UND POST). *Gültigkeitsmodell: Ketten-Schalenmodell.* Powerpoint Präsentation.  
<http://www.bundesnetzagentur.de/media/archive/1343.pps>.
- [BNetzA-Prod] BUNDESNETZAGENTUR. *Produkte für qualifizierte elektronische Signaturen.* [http://www.bundesnetzagentur.de/enid/Elektronische\\_Signatur/Produkte\\_pi.html](http://www.bundesnetzagentur.de/enid/Elektronische_Signatur/Produkte_pi.html).
- [BNetzA-ZDA] BUNDESNETZAGENTUR. *Zertifizierungsdiensteanbieter.* [http://www.bundesnetzagentur.de/enid/Elektronische\\_Signatur/Zertifizierungsdiensstanbieter\\_ph.html](http://www.bundesnetzagentur.de/enid/Elektronische_Signatur/Zertifizierungsdiensstanbieter_ph.html).

- [BNK95b] *Vorschlag zum §126a BGB.*  
<http://www.fitug.de/archiv/presse/para126a.html>, 1995.
- [BoDu99] DAN BONEH und GLEN DURFEE in Cryptanalysis of RSA with private key d less than  $N^{0.292}$ . In Advances in Cryptology – EUROCRYPT ’99, Band 1592 von Lecture Notes in Computer Science, Seiten 213–229 (Springer-Verlag, 1999).
- [Bone99] DAN BONEH. *Twenty years of attacks on the RSA cryptosystem. Notices of the American Mathematical Society (AMS)*, Band 46(2):203–213, 1999.
- [Born02] FOLKMAR BORNEMANN. *Ein Durchbruch für „Jedermann“*. DMV-Mitteilungen.  
<http://www.mathematik.de/mde/presse/pressestimmen/pdf/bornemann.pdf>, 2002.
- [BOS-DPMA] BREMEN ONLINE SERVICES GMBH & Co. KG. *Elektronischer Antrag zur Patent- und Markenanmeldung*. Projektreferenz.  
[http://www.governikus.de/fastmedia/22/Referenzbericht\\_DPMA.pdf](http://www.governikus.de/fastmedia/22/Referenzbericht_DPMA.pdf), 2003.
- [BReg97] BUNDESREGIERUNG. *Bericht der Bundesregierung über die Erfahrungen und Entwicklungen bei den neuen Informations- und Kommunikationsdiensten im Zusammenhang mit der Umsetzung des Informations- und Kommunikationsdienste-Gesetzes vom 18. Juni 1999*. BT-Drs. 14/1191. <http://dip.bundestag.de/btd/14/011/1401191.pdf>, 1997.
- [BrTe01] G. BRÖHL und A. TETTENBORN. *Das neue Recht der elektronischen Signaturen: kommentierende Darstellung von Signaturgesetz und Signaturverordnung* (Bundesanzeiger-Verlag, 2001). ISBN 3-89817-045-4.
- [BrPo05] R. BRANDNER U. PORDESCH. *Evidence Record Syntax (ERS)*. Internet Draft, LTANS working group, draft-ietf-ltans-ers-02.txt.  
<http://ltans.edelweb.fr/draft-ietf-ltans-ers-02.txt>, April 2005.
- [BSI-VPS] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK (BSI). *Die Virtuelle Poststelle*. <http://www.bsi.bund.de/fachthem/vps/index.htm>, 2005.
- [BSI97] BUNDESAMT FÜR SICHERHEIT IN DER INFORMATIONSTECHNIK (BSI). *BSI-Handbuch für digitale Signaturen – auf Grundlage von SigG und SigV von 1997, Version 1.1, Stand: 18.11.1997*.  
<http://www.bsi.de/esig/basics/techbas/masskat/bsikat.pdf>, 1997.
- [BSL04] D. BONEH, H. SHACHAM, B. LYNN. *Short signatures from the Weil pairing*. In *Journal of Cryptology*, Band 17(4):297–319, 2004.
- [Buch99] JOHANNES BUCHMANN. *Einführung in die Kryptographie* (Springer-Verlag, 1999). ISBN 3-540-66059-3.
- [BNK95a] BUNDESNOTARKAMMER (Herausgeber). *Elektronischer Rechtsverkehr: digitale Signaturverfahren und Rahmenbedingungen* (Verlag Dr. Otto Schmidt KG, 1995). ISBN 3-504-56032-0, 1995.
- [BuWi88] JOHANNES BUCHMANN HUGH C. WILLIAMS. *A key-exchange system based on imaginary quadratic fields*. In *Journal of Cryptology*, Band 1(3):107–118, 1988.
- [CC] *Common Criteria for Information Technology Security Evaluation (CC), Version 2.1.* <http://www.bsi.bund.de/cc/index.htm>, August 1999.

- [CCES-API] COMPETENCE CENTER ELEKTRONISCHE SIGNATUREN (CCES) IM VOI E.V. *CCES-Signature-API*. Version 1.0 vom 21.01.2005.  
[http://www.voi.de/media/custom/561\\_398\\_1.PDF](http://www.voi.de/media/custom/561_398_1.PDF), 2005.
- [CEG87] DAVID CHAUM, JAN-HENDRIK EVERTSE, JEROEN VAN DE GRAAF. *An Improved Protocol for Demonstrating Possession of Discrete Logarithms and some Generalizations*. In *Advances in Cryptology – EUROCRYPT ’87*, Band 304 von *Lecture Notes in Computer Science*, Seiten 127–141 (Springer-Verlag, 1988).
- [CEN-eIFG] CEN/ISSS E INVOICING FOCUS GROUP. *Report and Recommendations of CEN/ISSS e-Invoicing Focus Group on Standards and Developments on electronic invoicing relating to VAT Directive 2001/115/EC*.  
<http://www.cenorm.be/cenorm/businessdomains/businessdomains/issss/activity/finalreporteifg1.zip>, 2003.
- [CHJ99] D. COPPERSMITH, S. HALEVI, C. JUTLA. *ISO 9796-1 and the new forgery strategy*. Research contribution to IEEE P1363.  
<http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf>, 1999.
- [CITES] BUNDESAMT FÜR NATURSCHUTZ. *Ein- und Ausfuhr geschützter Tiere und Pflanzen nach dem Washingtoner Artenschutzübereinkommen (Convention on International Trade in Endangered Species of Wild Fauna and Flora – CITES)*. <http://www.cites-online.de>, 2004.
- [CNS99] J.S. CORON, D. NACCACHE, J.P. STERN. *On the security of RSA Padding*. In MICHAEL WIENER (Herausgeber), *Advances in Cryptology – CRYPTO ’99*, Band 1666 von *Lecture Notes in Computer Science*, Seiten 1–18 (Springer-Verlag, 1999).
- [COM(1999)195] Geänderter Vorschlag für eine Richtlinie des Europäischen Parlaments und des Rates über gemeinsame Rahmenbedingungen für elektronische Signaturen. <http://www.dfn-pca.de/bibliothek/sigg/europe/signamde.pdf>, 1999.
- [CWA14167-1] COMITE EUROPEEN DE NORMALISATION (CEN). *Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures - Part 1: System Security Requirements*. CEN Workshop Agreement CWA-14167-1. <ftp://ftp.cenorm.be/PUBLIC/CWAs/e-Europe/eSign/cwa14167-01-2003-Jun.pdf>, März 2003.
- [CWA14167-2] COMITE EUROPEEN DE NORMALISATION (CEN). *Security Requirements for Trustworthy Systems Managing Certificates for Electronic Signatures - Part 2: Cryptographic Module for CSP signing operations with backup - Protection profile (CMCSOB-PP)*. CEN Workshop Agreement CWA-14167-2. <ftp://ftp.cenorm.be/PUBLIC/CWAs/e-Europe/eSign/cwa14167-02-2004-May.pdf>, März 2002.
- [CWA14169] COMITE EUROPEEN DE NORMALISATION (CEN). *Secure Signature-creation devices (EAL 4+)*. CEN Workshop Agreement CWA-14169.  
<ftp://ftp.cenorm.be/PUBLIC/CWAs/e-Europe/eSign/cwa14169-00-2004-Mar.pdf>, März 2002.
- [DaLu05] MAGNUS DAUM STEFAN LUCKS. *The Story of Alice and her Boss: Hash Functions and the Blind Passenger Attack*. Beitrag zur Rump-Session der Eurocrypt 2005.  
[http://www.cits.rub.de/imperia/md/content/magnus/rump\\_ec05.pdf](http://www.cits.rub.de/imperia/md/content/magnus/rump_ec05.pdf), 2005.

- [Damg89] IVAN DAMGÅRD. *A Design Principle for Hash Functions*. In *Advances in Cryptology – CRYPTO '89*, Band 435 von *Lecture Notes in Computer Science*, Seiten 416–427 (Springer-Verlag, 1990).
- [DaTe05] GOTTFRIED DAIMER TILL TEICHMANN. *Elektronische Signatur - Eignung des biometrischen Merkmals Fingerabdruck als möglicher PIN-Ersatz*. In HANNES FEDERRATH (Herausgeber), *Sicherheit 2005 : Sicherheit - Schutz und Zuverlässigkeit, Beiträge der 2. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.v. (GI)*,, Band P-62 von *Lecture Notes in Information (LNI)*, Seiten 385–396 (GI, 2005).
- [DeOd85] YVO DESMEDT ANDREW ODLYZKO. *A chosen text attack on the RSA cryptosystem and some discrete logarithm schemes*. In *Advances in Cryptology – CRYPTO '85*, Band 218 von *Lecture Notes in Computer Science*, Seiten 516–522 (Springer-Verlag, 1986).
- [DGJ05] LARS DIETZE, ERNST-GÜNTHER GIEßMANN, LUIGI LO IACONO. *Gültigkeitsmodelle revisited. Datenschutz und Datensicherheit (DuD), Heft 4*, 2005.
- [DHLR05] LARS DIETZE, BERND HOLZNAGEL, LUIGI LO LACONO, CHRISTOPH RULAND. *Qualifizierte Signatur im elektronischen Messdatenaustausch*. In HANNES FEDERRATH (Herausgeber), *Sicherheit 2005: Sicherheit - Schutz und Zuverlässigkeit, Beiträge der 2. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.v. (GI)*, 5.-8. April 2005 in Regensburg, Band P-62 *LNI*, Seiten 335–348 (GI, 2005).
- [DiHe76] WHITFIELD DIFFIE MARTIN E. HELLMAN. *New Directions in Cryptography. IEEE Transactions on Information Theory*, Band 22(6):644–654, 1976.
- [DIN16560-15] DEUTSCHES INSTITUT FÜR NORMUNG (DIN). *EDIFACT - Anwendungsregeln - Teil 15: Anwendung des Service-Nachrichtentyps AUTACK zur Übermittlung von Integritäts- und Authentizitätsinformationen über versendete Nutzdaten*. DIN 16560-15, Beuth Verlag, Juli 2003.
- [DKN+03] JOS DUMORTIER, STEFAN KELM, HANS NILSSON, GEORGIA SKOUMA, PATRICK VAN EECKE. *Legal and market aspects of the application of Directive 1999/93/EC and practical applications of electronic signatures in the Member States, the EEA, the Candidate and the Accession countries*. <http://ec.europa.eu/idabc/servlets/Doc?id=18446>, 2003.
- [DoBP96] H. DOBBERTIN, A. BOSSELAERS, B. PRENEEL. *RIPEMD-160, a strengthened version of RIPEMD*. Band 1039:71–82. <http://www.esat.kuleuven.ac.be/~cosicart/pdf/AB-9601/AB-9601.pdf>, 1996.
- [Dono01] N. ANDREW DONOFRIO. *Differences between ANSI X9.31 and RSA PKCS #1*. Corsec. [http://www.corsec.com/copy/pdf/X931\\_PKCS1.pdf](http://www.corsec.com/copy/pdf/X931_PKCS1.pdf), 2001.
- [Dubu00] DAVID A. OLIVIER DUBUISSONCOX. *ASN.1 – Communication between Heterogeneous Systems* (OSS Nokalva, 2000). ISBN:0-12-6333361-0, <http://asn1.elibel.tm.fr/en/book/>.
- [DuRi01] JOS DUMORTIER REGINA RINDERLE. *Umsetzung der Signaturrichtlinie in den europäischen Mitgliedsstaaten*, 2001.

- [EbFa96] S. EBER-FALLER. *Gesetzgebungsvorschläge der Bundesnotarkammer zur Einführung elektronischer Unterschriften. Computerrecht (CR)*, Seiten 375–380, 1996.
- [eCard] BUNDESREGIERUNG. *eCard-Strategie der Bundesregierung*. Pressemitteilung vom 09.03.2005.  
<http://www.bmwi.de/Navigation/Presse/pressemittelungen,did=60006.htm>  
1, 2005.
- [EDI-AK-Handel] EDI ANWENDERKREIS HANDEL. *Empfehlung des AK-Handel zur digitalen Signatur von EDIFACT-INVOIC-Daten*. <http://www.edi-ak-handel.de/download/AKH-AgDISI.pdf>, 2004.
- [ElGa85] TAHER ELGAMAL. *A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms*. *IEEE Transactions on Information Theory*, Band 31(4):469–472, 1985.
- [Elster] BAYERISCHES LANDESAMT FÜR STEUERN. *Elster – die elektronische Steuererklärung*. <https://www.elster.de>, 2005.
- [ElsterOnline] BAYERISCHES LANDESAMT FÜR STEUERN. *ElsterOnline – das Dienstleistungsportal der deutschen Steuerverwaltung*.  
<https://www.elster.de/eportal>, 2005.
- [EnWGAendG] *Zweites Gesetz zur Neuregelung des Energiewirtschaftsrechts*. 7. Juli 2005, BGBl I 42, 1970. <http://217.160.60.235/BGBL/bgb1f/bgb105s1970.pdf>, 2005.
- [EStG] *Einkommensteuergesetz*. 16. Oktober 1934, RGBI I 1934, 1005, zuletzt geändert durch Art. 28 G v. 21. 6.2005 I 1818.  
<http://bundesrecht.juris.de/bundesrecht/estg/>, 1934.
- [ETSI-101733] *Electronic Signatures and Infrastructures (ESI) – Electronic Signature Formats – TS 101 733 – V1.5.1*, Dezember 2003.
- [ETSI-101862] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE (ETSI). *Qualified Certificate Profile*. ETSI Technical Standard TS 101 862, Version 1.3.1, März 2004.
- [EU-SigNot] EUROPEAN UNION. *Status of notification of legal acts implementing the electronic signatures directive*.  
[http://europa.eu.int/information\\_society/eeurope/2005/all\\_about/trust/esignatures/index\\_en.htm](http://europa.eu.int/information_society/eeurope/2005/all_about/trust/esignatures/index_en.htm).
- [eVergabe] BESCHAFFUNGSAMT DES BUNDESMINISTERIUMS DES INNERN. *e-Vergabe – die Vergabeplattform des Bundes*. <http://www.evergabe-online.de/>, 2005.
- [FGHH-ERV] FINANZGERICHT HAMBURG. *Elektronischer Rechtsverkehr*.  
<http://fhh.hamburg.de/stadt/Aktuell/justiz/gerichte/finanzgericht/elektronischer-rechtsverkehr/start.html>, 2005.
- [FGPS02] STEFANIE FISCHER-DIESKAU, ROTRAUD GITTER, SANDRA PAUL, ROLAND STEIDLE. *Elektronisch signierte Dokumente als Beweismittel im Zivilprozess. MultiMedia und Recht*, Seiten 709–713. [http://www.uni-kassel.de/fb7/oeff\\_recht/publikationen/pubOrdner/Beweissicherheit\\_elektronischer\\_Dokumente.pdf](http://www.uni-kassel.de/fb7/oeff_recht/publikationen/pubOrdner/Beweissicherheit_elektronischer_Dokumente.pdf), 2002.
- [FIPS180-1] UNITED STATES OF AMERICA NATIONAL INSTITUTE FOR STANDARDS TECHNOLOGY (NIST). *Secure Hash Standard (SHS)*. Federal Information Processing Standard (FIPS) Publication 180-1, April 1993.

- [FIPS180-2] UNITED STATES OF AMERICA NATIONAL INSTITUTE FOR STANDARDS TECHNOLOGY (NIST). *Secure Hash Standard (SHS)*. Federal Information Processing Standard (FIPS) Publication 180-2, 01.08.2002.  
<http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf>, August 2002.
- [FIPS186] UNITED STATES OF AMERICA NATIONAL INSTITUTE FOR STANDARDS TECHNOLOGY (NIST). *Digital Signature Standard (DSS)*.  
<http://www.itl.nist.gov/fipspubs/fip186.htm>, Mai 1994.
- [FIPS186-2] UNITED STATES OF AMERICA NATIONAL INSTITUTE FOR STANDARDS TECHNOLOGY (NIST). *Digital Signature Standard (DSS)*.  
<http://csrc.nist.gov/publications/fips/fips186-2/fips186-2-change1.pdf>, Januar 2000.
- [FiSh86] AMOS FIAT ADI SHAMIR. *How To Prove Yourself: Practical Solutions to Identification and Signature Problems*. In *Advances in Cryptology – CRYPTO '86*, Band 263 von *Lecture Notes in Computer Science*, Seiten 186–194 (Springer-Verlag, 1987).
- [FMH99] G. FREY, M. MÜLLER, H.G. RÜCK. *The Tate Pairing and the Discrete Logarithm Applied to Elliptic Curve Cryptosystems*. *IEEE Transactions on Information Theory*, (45):1717–1719, 1999.
- [FormAnpG] *Gesetz zur Anpassung der Formvorschriften des Privatrechts und anderer Vorschriften an den modernen Rechtsgeschäftsverkehr, vom 13. Juli 2001*. BGBl. I Nr. 35, S. 1542-1549.  
<http://www.dud.de/documents/formvorschriften-010713.pdf>, 2001.
- [FormAnpGBeg] *Entwurf eines Gesetzes zur Anpassung der Formvorschriften des Privatrechts und anderer Vorschriften an den modernen Rechtsgeschäftsverkehr*. BT-Drucksache 14/4987 vom 14.12.2000.  
<http://dip.bundestag.de/btd/14/049/1404987.pdf>, 2000.
- [Gaus01] CARL FRIEDRICH GAUß. *Disquisitiones Arithmeticae* (Springer-Verlag, 1801). Nachdruck, 1986, ISBN 0-387-96254-9.
- [GDPdU] *Grundsätze zum Datenzugriff und zur Prüfbarkeit digitaler Unterlagen (GDPdU) – BMF-Schreiben vom 16. Juli 2001 - IV D 2 - S 0316 - 136/01 -*. BStBl. 2001 I. [http://www.secunet.com/download/k/pki\\_bmf\\_gdpdu.pdf](http://www.secunet.com/download/k/pki_bmf_gdpdu.pdf), 2001.
- [GIS05] MAX GEBHARDT, GEORG ILLIES, WERNER SCHINDLER. *A Note on the Practical Value of Single Hash Collisions for Special File Formats*. Beitrag zum “Cryptographic Hash Workshop” des NIST, 31. Oktober - 1. November 2005, Gaithersburg, Maryland.  
[http://www.csrc.nist.gov/pki/HashWorkshop/2005/Oct31\\_Presentations/Illies\\_NIST\\_05.pdf](http://www.csrc.nist.gov/pki/HashWorkshop/2005/Oct31_Presentations/Illies_NIST_05.pdf), 2005.
- [GoBS] *Grundsätze ordnungsmäßiger DV-gestützter Buchführungssysteme (GoBS) – BMF-Schreiben vom 7. November 1995 - IV A 8 - S 0316 - 52/95-*. BStBl 1995 I S. 738. [http://www.secunet.com/download/k/pki\\_bmf\\_gobs.pdf](http://www.secunet.com/download/k/pki_bmf_gobs.pdf), 1995.
- [Gold01] ODED GOLDREICH. *Foundations of Cryptography – Volume 1* (Cambridge University Press, 2001). ISBN 0-521-79172-3.
- [Gord93] D. GORDON. *Discrete logarithms in  $\mathbb{F}_p$  using the number field sieve*. *SIAM J. Discrete Math*, Band 6:124–138, 1993.

- [Gord98] DANIEL M. GORDON. *A Survey of Fast Exponentiation Methods.* *J. Algorithms*, Band 27(1):129–146, 1998.
- [HeHu02] JOHAN HESSE DETLEF HÜHNLEIN. *Public-Key Infrastrukturen für Sozialversicherungsträger.* In PATRICK HORSTER (Herausgeber), *Tagungsband “Elektronische Geschäftsprozesse”*, Seiten 177–198 (IT-Verlag, 2002). ISBN 3-936052-07-7, [http://www.secunet.com/download/fachartikel/ebp2002\\_pki-svt.pdf](http://www.secunet.com/download/fachartikel/ebp2002_pki-svt.pdf).
- [HGB] *Handelsgesetzbuch.* vom 10. Mai 1897, RGBl. 1897, 219, zuletzt geändert durch Art. 1 G v. 3. 8.2005 I 2267. <http://bundesrecht.juris.de/bundesrecht/hgb/>, 1897.
- [HH03] HANSESTADT HAMBURG. *HamburgGateway - das digitale Tor zur Hamburg Verwaltung - startet mit Melderegisterauskünften.* <http://fhh.hamburg.de/stadt/Aktuell/pressemeldungen/2003/oktober/20/pressemeldung-2003-10-20-fb-gateway>, 2003.
- [HmbMDUeV] *Verordnung über regelmäßige Datenübermittlungen und automatisierte Abrufe aus dem Melderegister (Meldedatenübermittlungsverordnung - MDÜV).* vom 9. September 1997, HmbGVBl. 1997, S. 453, zuletzt geändert durch die Verordnung vom 7.12.2004, HmbGVBl. 2004, S. 467. [http://hh.juris.de/hh/MeldDUeV\\_HA\\_rahmen.htm](http://hh.juris.de/hh/MeldDUeV_HA_rahmen.htm), 1997.
- [HmbMG] *Hamburgisches Meldegesetz (HmbMG).* vom 3. September 1996, zuletzt geändert durch das Gesetz vom 28.12.2004, HmbGVBl. 2004, S. 527. [http://hh.juris.de/hh/MeldeG\\_HA\\_1996\\_rahmen.htm](http://hh.juris.de/hh/MeldeG_HA_1996_rahmen.htm), 1996.
- [HMP94] PATRICK HORSTER, MARKUS MICHELS, HOLGER PETERSEN. *Meta-ElGamal signature schemes.* *2nd ACM Conference on Computers and Communications Security*, Seiten 96–107 (ACM Press, 1994).
- [HMP95] PATRICK HORSTER, MARKUS MICHELS, HOLGER PETERSEN. *Das Meta-ElGamal Signaturverfahren und seine Anwendungen.* In *Verlässliche Informationssysteme, VIS '95*, Seiten 207–228 (Vieweg Verlag, 1995).
- [Horn05] GERRIT HORNUNG. *Die digitale Identität*, Band 10 von *Der Elektronische Rechtsverkehr* (Nomos Verlagsgesellschaft, Baden-Baden, 2005). ISBN: 3-8329-1455-2.
- [HoRo04] GERRIT HORNUNG ALEXANDER ROßNAGEL. *Die JobCard “Killer-Applikation” für die elektronische Signatur?* In *Kommunikation und Recht*, Band 7(6):263–269. [http://www.uni-kassel.de/fb7/oeff\\_recht/publikationen/pubOrdner/Jobkarte-KuR.pdf](http://www.uni-kassel.de/fb7/oeff_recht/publikationen/pubOrdner/Jobkarte-KuR.pdf), 2004.
- [Hueh01] DETLEF HÜHNLEIN. *Faster Generation of NICE-Schnorr-type Signatures.* In *The Cryptographers’ Track at RSA Conference 2001*, Band 2020 von *Lecture Notes in Computer Science*, Seiten 1–12 (Springer-Verlag, 2001).
- [Hueh04a] DETLEF HÜHNLEIN. *How to Qualify Electronic Signatures and Time Stamps.* In SOKRATIS K. KATSIKAS, STEFANOS GRITZALIS, JAVIER LOPEZ (Herausgeber), *Public Key Infrastructure, First European PKI Workshop: Research and Applications, EuroPKI 2004, Samos Island, Greece, June 25-26, 2004, Proceedings*, Band 3093 von *Lecture Notes in Computer Science*, Seiten 314–321 (Springer Verlag, 2004).
- [Hueh04b] DETLEF HÜHNLEIN. *Kryptosysteme auf Basis imaginärquadratischer Nicht-Maximalordnungen.* Dissertation. <http://elib.tu-darmstadt.de/diss/000521/>, 2004.

- [Hueh04c] DETLEF HÜHNLEIN. *Intervall-qualifizierte Zeitstempel*. In PATRICK HORSTER (Herausgeber), *Elektronische Geschäftsprozesse*, Seiten 431–445 (IT-Verlag, 2004). ISBN 3-00-014186-3, <http://www.secunet.com/download/fachartikel/iq-zeitstempel.pdf>.
- [Hueh05] DETLEF HÜHNLEIN. *Die CCES-Signature-API - Eine offene Programmierschnittstelle für langfristig beweiskräftige elektronische Signaturen*. In HANNES FEDERRATH (Herausgeber), *Sicherheit 2005: Sicherheit - Schutz und Zuverlässigkeit, Beiträge der 2. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.v. (GI), 5.-8. April 2005 in Regensburg*, Band P-62 LNI, Seiten 361–374 (GI, 2005). ISBN 3-88579-391-1.
- [HuKn03] DETLEF HÜHNLEIN YVONNE KNOSOWSKI. *Aspekte der Massensignatur*. In PATRICK HORSTER (Herausgeber), *DACH-Security 2003*, Seiten 293–307 (IT-Verlag, 2003). ISBN 3-00-010941-2, [http://www.secunet.com/download/fachartikel/dach2003\\_aspekte-der-massensignatur.pdf](http://www.secunet.com/download/fachartikel/dach2003_aspekte-der-massensignatur.pdf).
- [HuMe00] DETLEF HÜHNLEIN JOHANNES MERKLE. *An efficient NICE-Schnorr-type cryptosystem*. In *Practice and Theory in Public Key Cryptography, PKC 2000*, Band 1751 von *Lecture Notes in Computer Science*, Seiten 14–27 (Springer-Verlag, 2000).
- [HuTe04] DETLEF HÜHNLEIN RAGNA TERN. *Rechtliche Aspekte der elektronischen Abrechnung durch Dritte*. In PATRICK HORSTER (Herausgeber), *Elektronische Geschäftsprozesse*, Seiten 134–143 (IT-Verlag, 2004). ISBN 3-00-014186-3, <http://www.secunet.de/download/fachartikel/juristische-aspekte.pdf>.
- [IEEE-P1363] IEEE-P1363: *Standard Specifications for Public Key Cryptography*, August 2000.
- [Infoprint] IBM. *Infoprint Server and Infoprint Transforms for z/OS*. [http://www.printers.ibm.com/internet/wwsites.nsf/vwwebpublished/ipservrhome\\_z\\_ww](http://www.printers.ibm.com/internet/wwsites.nsf/vwwebpublished/ipservrhome_z_ww), 2005.
- [INVOIC] CENTRALE FÜR COORGANISATION (CCG). *Übermittlung von Abrechnungsdaten (Rechnungslistensummen) mit EANCOM INVOIC 008*. [http://www.edi-ak-handel.de/ak\\_handel\\_en/Guides/reli.pdf](http://www.edi-ak-handel.de/ak_handel_en/Guides/reli.pdf), Mai 2000.
- [ISIS-MTT] T7 E.V. UND TELETRUST E.V. *ISIS-MTT-Spezifikation, Version 1.1*. <http://www.isis-mtt.de/>, März 2004.
- [ISIS-MTT-SigG] T7 E.V. UND TELETRUST E.V. *ISIS-MTT-Spezifikation – Optional Profile – SigG-Profile, Version 1.1*. <http://www.isis-mtt.de/>, März 2004.
- [ISO10118-3] ISO/IEC 10118-3: *Information Technology Security Techniques Hash functions Part 3: Dedicated hash functions*. Final Draft International Standard, 2004.
- [ISO14888-3] ISO/IEC 14888-3: *Information Technology Security Techniques Digital Signatures with Appendix Part 3: Certificate Based-Mechanisms*. International Standard, 1998.
- [ISO18014-1] ISO/IEC 18014-1: *Information technology – Security techniques – Time-stamping services – Part 1: Framework*. International Standard, Oktober 2002.

- [ISO9735-1] JOINT ISO/TC 154 UN/CEFACT SYNTAX WORKING GROUP (JSWG).  
*Electronic data interchange for administration, commerce and transport (EDIFACT) Application level syntax rules (Syntax version number: 4, Syntax release number: 1) Part 1: Syntax rules common to all parts.* ISO 9735-1 (Second edition 2002-07-01).  
<http://www.gefeg.com/jswg/v41/data/V41-9735-1.zip>, Juli 2002.
- [ISO9735-2] JOINT ISO/TC 154 UN/CEFACT SYNTAX WORKING GROUP (JSWG).  
*Electronic data interchange for administration, commerce and transport (EDIFACT) Application level syntax rules (Syntax version number: 4, Syntax release number: 1) Part 2: Syntax rules specific to batch EDI.* ISO 9735-2 (Second edition 2002-07-01).  
<http://www.gefeg.com/jswg/v41/data/V41-9735-2.zip>, Juli 2002.
- [ISO9735-3] JOINT ISO/TC 154 UN/CEFACT SYNTAX WORKING GROUP (JSWG).  
*Electronic data interchange for administration, commerce and transport (EDIFACT) Application level syntax rules (Syntax version number: 4, Syntax release number: 1) Part 3: Syntax rules specific to interactive EDI.* ISO 9735-3 (Second edition 2002-07-01).  
<http://www.gefeg.com/jswg/v41/data/V41-9735-3.zip>, Juli 2002.
- [ISO9735-4] JOINT ISO/TC 154 UN/CEFACT SYNTAX WORKING GROUP (JSWG).  
*Electronic data interchange for administration, commerce and transport (EDIFACT) Application level syntax rules (Syntax version number: 4, Syntax release number: 1) Part 4: Syntax and service report message for batch EDI (message type CONTRL).* ISO 9735-4 (Second edition 2002-07-01). <http://www.gefeg.com/jswg/v41/data/V41-9735-4.zip>, Juli 2002.
- [ISO9735-5] JOINT ISO/TC 154 UN/CEFACT SYNTAX WORKING GROUP (JSWG).  
*Electronic data interchange for administration, commerce and transport (EDIFACT) Application level syntax rules (Syntax version number: 4, Syntax release number: 1) Part 5: Security rules for batch EDI (authenticity, integrity and non-repudiation of origin).* ISO 9735-5 (Second edition 2002-07-01). <http://www.gefeg.com/jswg/v41/data/V41-9735-5.zip>, Juli 2002.
- [ISO9735-6] JOINT ISO/TC 154 UN/CEFACT SYNTAX WORKING GROUP (JSWG).  
*Electronic data interchange for administration, commerce and transport (EDIFACT) Application level syntax rules (Syntax version number: 4, Syntax release number: 1) Part 6: Secure authentication and acknowledgement message (message type AUTACK).*  
<http://www.gefeg.com/jswg/v41/data/V41-9735-6.zip>, Juli 2002.
- [ISO9735-7] JOINT ISO/TC 154 UN/CEFACT SYNTAX WORKING GROUP (JSWG).  
*Electronic data interchange for administration, commerce and transport (EDIFACT) Application level syntax rules (Syntax version number: 4, Syntax release number: 1) Part 7: Security rules for batch EDI (confidentiality).* <http://www.gefeg.com/jswg/v41/data/V41-9735-7.zip>, Juli 2002.
- [ISO9735-8] JOINT ISO/TC 154 UN/CEFACT SYNTAX WORKING GROUP (JSWG).  
*Electronic data interchange for administration, commerce and transport (EDIFACT) Application level syntax rules (Syntax version number: 4, Syntax release number: 1) Part 8: Associated data in EDI.*  
<http://www.gefeg.com/jswg/v41/data/V41-9735-8.zip>, Juli 2002.

- [ISO9735-9] JOINT ISO/TC 154 UN/CEFACT SYNTAX WORKING GROUP (JSWG). *Electronic data interchange for administration, commerce and transport (EDIFACT) Application level syntax rules (Syntax version number: 4, Syntax release number: 1) Part 9: Security key and certificate management message (message type KEYMAN)*. <http://www.gefeg.com/jswg/v41/data/V41-9735-9.zip>, Juli 2002.
- [ISO9735-10] JOINT ISO/TC 154 UN/CEFACT SYNTAX WORKING GROUP (JSWG). *Electronic data interchange for administration, commerce and transport (EDIFACT) Application level syntax rules (Syntax version number: 4, Syntax release number: 1) Part 10: Syntax service directories*. <http://www.gefeg.com/jswg/v41/data/V41-9735-10.zip>, Juli 2002.
- [ISO9796-1] ISO-IEC 9796-1: *Information Technology Security Techniques Digital Signature Schemes Giving Message Recovery Part 1: Mechanisms using Redundancy*. International Standard, 1999.
- [ISO9796-2] ISO-IEC 9796-2: *Information Technology Security Techniques Digital Signature Schemes Giving Message Recovery Part 2: Integer Factorization Based Mechanisms*. International Standard, Oktober 2002.
- [ISO9796-3] ISO-IEC 9796-3: *Information Technology Security Techniques Digital Signature Schemes Giving Message Recovery Part 3: Discrete Logarithm Based Mechanisms*. International Standard, April 2000.
- [ISTEV97] ISTITUTO PER LO STUDIO DELLA VULNERABILITÁ DELLE SOCIETÁ TECNOLOGICAMENTE EVOLUTE (ISTEV). *Legal and Regulatory Issues for the European Trusted Services Infrastructure - ETS*. <ftp://ftp.cordis.lu/pub/infosec/docs/lrfets.doc>, 1997.
- [ITSEC] *Kriterien für die Bewertung der Sicherheit von Systemen der Informationstechnik*. <http://www.bsi.de/zertifiz/itkrit/itsec-dt.pdf>, August 1992.
- [IuKDG] *Gesetz zur Regelung der Rahmenbedingungen für Informations- und Kommunikationsdienste (Informations- und Kommunikationsdienste-Gesetz - IuKDG)*, vom 22. Juli 1997. BGBl. I, S. 1870. [http://www.iid.de/\\_img/article/iukdgbt.pdf](http://www.iid.de/_img/article/iukdgbt.pdf), 1997.
- [JKomG] *Gesetz über die Verwendung elektronischer Kommunikationsformen in der Justiz (Justizkommunikationsgesetz - JKOMG)*, vom 22. März 2005. BGBl. I Nr. 18, S. 837-858. <http://217.160.60.235/BGBL/bgb1f/bgb1105s0837.pdf>, 2005.
- [JobCard] ITSG INFORMATIONSTECHNISCHE SERVICESTELLE DER GESETZLICHEN KRANKENVERSICHERUNG GMBH. *Das JobCard-Verfahren*. <http://info.imsd.uni-mainz.de/AGDatenschutz/Sitzungen/Jobcard.pdf>, 2004.
- [JoLe03] ANTOINE JOUX REYNALD LERCIER. *Improvements to the general number field sieve for discrete logarithms in prime fields. A comparison with the gaussian integer method*. In *Mathematics of Computation*, Band 72:953–967, 2003.
- [JoMe99] D. JOHNSON A. MENEZES. *The Elliptic Curve Digital Signature Algorithm (ECDSA)*. Technical Report CORR 99-34, Dept. of C&O, University of Waterloo, Canada. <http://citeseer.ist.psu.edu/johnson99elliptic.html>, August 1999.

- [Jung02] SEBASTIAN JUNGERMANN. *Der Beweiswert elektronischer Signaturen – Eine Studie zur Verlässlichkeit elektronischer Signaturen und zu den Voraussetzungen und Rechtsfolgen des §292a ZPO* (Peter Lang Verlag, 2002).
- [KaRo97] ULRICH KAMPFFMEYER JÖRG ROGALLA. *Grundsätze der elektronischen Archivierung „Code of Practice“ zum Einsatz von Dokumenten-Management- und elektronischen Archivsystemen* (1997). ISBN 3-932898-03-6.
- [KCDSA98] KCDSA TASK FORCE TEAM. *The Korean Certificate-based Digital Signature Algorithm*.  
<http://grouper.ieee.org/groups/1363/P1363a/contributions/kcdsa1363.pdf>, August 1998.
- [Klim05] VLASTIMIL KLIMA. *Finding MD5 Collisions a Toy For a Notebook*. Cryptology ePrint Archive: Report 2005/075.  
<http://eprint.iacr.org/2005/075>, März 2005.
- [Kobl87] NEAL KOBLITZ. *Elliptic Curve Cryptosystems*. In *Mathematics of Computation*, Band 48(177):203–209, 1987.
- [Kobl89] NEAL KOBLITZ. *Hyperelliptic cryptosystems*. In *Journal of Cryptology*, Band 1(3):139–150, 1989.
- [Kobl94] NEIL KOBLITZ. *A Course in Number Theory and Cryptography*, Band 114 von *Graduate Texts in Mathematics* (Springer–Verlag, 1994), 2. Auflage
- [Laub99] THOMAS LAUBROCK. *Krypto-Verfahren basierend auf elliptischen Kurven – HTML-Tutorial mit Java™-Applet*. <http://www.elliptische-kurven.de/>, 1999.
- [Lens87] HENDRIK W. LENSTRA. *Factoring integers with elliptic curves*. *Annals of Mathematics*, Band 126:649–673, 1987.
- [LeSc04] JÖRG M. LENZ CHRISTIANE SCHMIDT. *Die elektronische Signatur - eine Analogie zur eigenhändigen Unterschrift?* (Deutscher Sparkassen Verlag, 2004). ISBN 3-09-3057058-1.
- [LeVe01] ARJEN K. LENSTRA ERIC R. VERHEUL. *Selecting Cryptographic Keysizes*. *Journal of Cryptology*, Band 14(4):255–293, 2001.
- [LeWe05] ARJEN K. LENSTRA BENNE DE WEGER. *On the possibility of constructing meaningful hash collisions for public keys*. In COLIN BOYD JUAN MANUEL GONZÁLEZ NIETO (Herausgeber), *Information Security and Privacy, 10th Australasian Conference, ACISP 2005, Brisbane, Australia, July 4-6, 2005, Proceedings*, Band 3574 von *Lecture Notes in Computer Science*, 267–279 (Springer, 2005).
- [Ley04] RUDOLF LEY. *CITES-Online Antragstellung*. Vortragsfolien, 15.09.2004.
- [LiNi86] RUDOLF LIDL HARALD NIEDERREITER. *Introduction to finite fields and their applications* (Cambridge University Press, 1986).
- [LiLe98] CHAE HOON LIM PIL JOONG LEE. *A Study on the Proposed Korean Digital Signature Algorithm*. In KAZUO OHTA PEI DINGYI (Herausgeber), *Advances in Cryptology – ASIACRYPT ’98*, Band 1514 von *Lecture Notes in Computer Science*, Seiten 175–186 (Springer-Verlag, 1998).

- [LLMP93] ARJEN K. LENSTRA, HENDRIK W. LENSTRA, MARK S. MANASSE, JOHN M. POLLARD. *The number field sieve*. In A.K. LENSTRA H.W. LENSTRA (Herausgeber), *The Developement of the Number Field Sieve*, Band 1554 *Lecture Notes in Mathematics* (Springer-Verlag, 1993).
- [May05] ALEXANDER MAY. *Computing the RSA Secret Key Is Deterministic Polynomial Time Equivalent to Factoring*. In MATTHEW K. FRANKLIN (Herausgeber), *Advances in Cryptology – CRYPTO 2004*, Band 3152 von *Lecture Notes in Computer Science*, Seiten 213–219 (Springer-Verlag, 2004).
- [Mene93] ALFRED J. MENEZES. *Elliptic Curve Public Key Cryptosystems* (Kluwer Academic Publishers, 1993).
- [MOV97] ALFRED J. MENEZES, PAUL C. VAN OORSCHOT, SCOTT A. VANSTONE. *Handbook of Applied Cryptography* (CRC Press, 1997).  
<http://www.cacr.math.uwaterloo.ca/hac/>.
- [Merk80] RALPH C. MERKLE. *Protocols for public key cryptosystems*. In *Symposium on Security and Privacy, Oakland, CA, USA*, Seiten 122–134 (1980).
- [Merk87] RALPH C. MERKLE. *A Digital Signature Based on a Conventional Encryption Function*. In *A Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology*, Seiten 369–378 (Springer-Verlag, 1987). ISBN 3-540-18796-0.
- [Merk89] RALPH C. MERKLE. *One way hash functions and DES*. In *Advances in Cryptology – CRYPTO ’89*, Band 435 von *Lecture Notes in Computer Science*, Seiten 428–446 (Springer-Verlag, 1990).
- [Mill85] VICTOR S. MILLER. *Use of Elliptic Curves in Cryptography*. In *Advances in Cryptology – CRYPTO ’85*, Band 218 von *Lecture Notes in Computer Science*, Seiten 417–426 (Springer-Verlag, 1986).
- [Misa98] J. MISARSKY. *How (Not) to Design RSA Signature Schemes*. In *Practice and Theory in Public Key Cryptography, PKC ’98*, Band 1431 von *Lecture Notes in Computer Science*, Seiten 14–28 (Springer-Verlag, 1998).
- [MKatTK] REGULIERUNGSBEHÖRDE FÜR TELEKOMMUNIKATION UND POST (REGTP). *Maßnahmenkatalog für technische Komponenten nach dem Signaturgesetz – Stand 15. Juli 1998*.  
<http://www.bsi.de/esig/basics/techbas/masskat/techkomp.pdf>, 1998.
- [MKatZS] REGULIERUNGSBEHÖRDE FÜR TELEKOMMUNIKATION UND POST (REGTP). *Maßnahmenkatalog für Zertifizierungsstellen nach dem Signaturgesetz – Stand 15. Juli 1998*.  
<http://www.bsi.de/esig/basics/techbas/masskat/zertst.pdf>, 1998.
- [MNP96] MARKUS MICHELS, DAVID NACCACHE, HOLGER PETERSEN. *GOST 34.10 – A Brief Overview of Russias,’ DSA*. In *Computers & Security*, Band 15(8):725–732.  
<http://www.gemplus.com/smart/rd/publications/pdf/MNP96gos.pdf>, 1996.
- [MOV91] ALFRED MENEZES, TATSUAKI OKAMOTO, SCOTT VANSTONE. *Reducing elliptic curve logarithms to logarithms in a finite field*. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, Seiten 80–89 (ACM Press, 1991). ISBN 0-89791-397-3.

- [MRRG] *Melderechtsrahmengesetz (MRRG), vom 16. August 1980.* BGBI I 1980, 1429, zuletzt geändert durch Art. 12 G v. 21. 6.2005 I 1818.  
<http://bundesrecht.juris.de/bundesrecht/mrrg/>, 1980.
- [MSAC] MICROSOFT CORPORATION. *Authenticode*. Microsoft Developer Network Eintrag.  
[http://msdn.microsoft.com/workshop/security/authcode/authenticate\\_node\\_entry.asp](http://msdn.microsoft.com/workshop/security/authcode/authenticate_node_entry.asp), 2005.
- [MTW04] ALFRED MENEZES, EDLYN TESKE, ANNEGRET WENG. *Weak Fields for ECC*. In TATSUAKI OKAMOTO (Herausgeber), *Topics in Cryptology - RSA 2004, The Cryptographers' Track at the RSA Conference 2004*, Band 2964 von *Lecture Notes in Computer Science*, Seiten 366–386 (Springer Verlag, 2004). ISBN 3-540-20996-4.
- [multisign] SECUNET SECURITY NETWORKS AG. *multisign – eine skalierbare Lösungsfamilie zur Massensignatur*. Webseite.  
<http://multisign.secunet.com/>, 2005.
- [Nebe00] G. NEBE. *Faktorisieren großer Zahlen. Jahresbericht der Deutschen Mathematiker-Vereinigung*, Band 102, Seiten 1–14 (B.G. Teubner, Stuttgart, Leipzig, 2000).
- [NgSh02] P. Q. NGUYEN I. E. SHPARLINSKI. *The insecurity of the Digital Signature Algorithm with partially known nonces*. In *Journal of Cryptology*, Band 15(3):151–156, 2002.
- [NgSh03] P. Q. NGUYEN I. E. SHPARLINSKI. *The Insecurity of the Elliptic Curve Digital Signature Algorithm with Partially Known Nonces*. In *Designs, Codes and Cryptography*, Band 30(2):201–217, 2003.
- [Nguy04] PHONG Q. NGUYEN. *Can We Trust Cryptographic Software? Cryptographic Flaws in GNU Privacy Guard v1.2.3*. In *Advances in Cryptology – EUROCRYPT 2000*, Band 3027 von *Lecture Notes in Computer Science*, Seiten 555–570 (Springer, 2004).
- [NyRu94] KAISA NYBERG RAINER A. RUEPPEL. *Message recovery for signature schemes based on the discrete logarithm problem*. In *Advances in Cryptology – EUROCRYPT '94*, Band 950 von *Lecture Notes in Computer Science*, Seiten 182–193 (Springer-Verlag, 1995).
- [OeKr03] URSULA OESING ROLAND KRÜGER. *Das Signatursiegelverfahren*. In PATRICK HORSTER (Herausgeber), *DACH-SECURITY 2003*, 106–116 (IT-Verlag, 2003).
- [OeKr04] URSULA OESING ROLAND KRÜGER. *Gesetzeskonformer Massenversand elektronischer Rechnungen*. In PATRICK HORSTER (Herausgeber), *Elektronische Geschäftsprozesse 2004* (IT-Verlag, 2004).  
[http://www.mediasec.de/downloads/veroeffentlichungen/elektronische\\_rechnungsstellung2004.pdf](http://www.mediasec.de/downloads/veroeffentlichungen/elektronische_rechnungsstellung2004.pdf).
- [OeSigG] REPUBLIK ÖSTERREICH. *Signaturgesetz – SigG*. BGBI. I Nr. 190/1999, zuletzt geändert durch Art. II des Gesetzes vom 21.12.2001.  
<http://www.signatur.rtr.at/de/legal/sigg.html>, 1999.
- [OeV583/03] REPUBLIK ÖSTERREICH. 583. *Verordnung: Bestimmung der Anforderungen an eine auf elektronischem Weg übermittelte Rechnung*. BGBI. II, S. 3645 vom 23. Dezember 2003. <http://www.signatur.rtr.at/repository/legal-ebilling-20031223-de.pdf>, 2003.

- [OkUc98] TATSUAKI OKAMOTO SHIGENORI UCHIYAMA. *A New Public-Key Cryptosystem as secure as Factoring*. In *Advances in Cryptology – EUROCRYPT ’98*, Band 1403 von *Lecture Notes in Computer Science*, Seiten 308–318 (Springer-Verlag, 1998).
- [openTRANS] FRAUNHOFER IAO UNIVERSITÄT ESSEN BLI. *Spezifikation openTRANS*. Version 1.0 vom 07.09.2001. <http://www.opentrans.org/>, 2001.
- [OptiMahn] ABIT AG. *OptiMahnOffice*. Webseite. <http://www.optimahnoffice.de/>, 2005.
- [OSCI] OSCI-LEITSTELLE. *Online Services Computer Interface (OSCI)*. Webseite. <http://www.osci.de/>, 2005.
- [PaBo99] PAVLOVSKI C. BOYD C. *Efficient Batch Signature Generation using Tree Structures*. In *International Workshop on Cryptographic Techniques and E-Commerce – CrypTEC9*, 9, Seiten 70–77 (City University of Hong Kong Press, 1999). <http://sky.fit.qut.edu.au/~boydc/papers/treefinal.ps>.
- [Pail99] PASCAL PAILLIER. *A Trapdoor Permutation Equivalent to Factoring*. In HIDEKI IMAI YULIANG ZHENG (Herausgeber), *Public Key Cryptography, Second International Workshop on Practice and Theory in Public Key Cryptography, PKC ’99, Kamakura, Japan, March 1-3, 1999, Proceedings*, Band 1560 von *Lecture Notes in Computer Science*, Seiten 219–222 (Springer-Verlag, 1999).
- [Pala04] OTTO PALANDT. *Bürgerliches Gesetzbuch – Mit Einführungsgesetz (Auszug), Produkthaftungsgesetz, ErbbaurechtsVO, Wohnungseigentumsgesetz, HausratsVO*, Band 63. Auflage (Verlag C.H. Beck, 2004). ISBN 3-406-51035-3.
- [PDF(v1.3)] ADOBE SYSTEMS INCORPORATED. *PDF Reference – Second Edition – Adobe Portable Document Format Version 1.3*. Addison Wesley, ISBN 0-201-61588-6.  
<http://partners.adobe.com/public/developer/en/pdf/PDFReference13.pdf>, Juli 2000.
- [PDF(v1.4)] ADOBE SYSTEMS INCORPORATED. *PDF Reference – Third Edition – Adobe Portable Document Format Version 1.4*. Addison-Wesley, ISBN 0-201-75839-3.  
<http://partners.adobe.com/public/developer/en/pdf/PDFReference.pdf>, November 2001.
- [PDF(v1.5)] ADOBE SYSTEMS INCORPORATED. *PDF Reference – Fourth Edition – Adobe Portable Document Format Version 1.5*.  
[http://partners.adobe.com/public/developer/en/pdf/PDFReference15\\_v6.pdf](http://partners.adobe.com/public/developer/en/pdf/PDFReference15_v6.pdf), August 2003.
- [PDF(v1.6)] ADOBE SYSTEMS INCORPORATED. *PDF Reference – Fifth Edition – Adobe Portable Document Format Version 1.6*.  
<http://partners.adobe.com/public/developer/en/pdf/PDFReference16.pdf>, November 2004.
- [PKCS1(v1.5)] RSA LABORATORIES. *PKCS #1: RSA Encryption Standard - Version 1.5*. Public Key Cryptography Standards – PKCS #1 v1.5.  
<http://www.rsalabs.com>, November 1993.

- [PKCS1(v2.1)] RSA LABORATORIES. *PKCS #1: RSA Encryption Standard - Version 2.1.* Public Key Cryptography Standards – PKCS #1 v2.1. <http://www.rsalabs.com>, Juni 2002.
- [PKCS6] RSA LABORATORIES. *PKCS #6: Extended-Certificate Syntax Standard - Version 1.5.* Public Key Cryptography Standards – PKCS #6. <http://www.rsalabs.com>, November 1993.
- [PKCS7(v1.5)] RSA LABORATORIES. *PKCS #7: Cryptographic Message Syntax Standard - Version 1.5.* Public Key Cryptography Standards – PKCS #7. <http://www.rsalabs.com>, November 1993.
- [PKCS9] RSA LABORATORIES. : *PKCS #9: Selected Object Classes and Attribute Types - Version 2.0.* Public Key Cryptography Standards – PKCS #9. <ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-9-v2/pkcs-9.pdf>, Februar 2000.
- [PKCS11] RSA LABORATORIES. : *PKCS #11: Cryptographic Token Interface Standard - Version 2.2.* Public Key Cryptography Standards – PKCS #11. <ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-11/v2-20/pkcs-11v2-20.pdf>, Juni 2004.
- [PKCS12] RSA LABORATORIES. : *PKCS #12: Personal Information Exchange Syntax Standard - Version 1.0.* Public Key Cryptography Standards – PKCS #12. <ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-12/pkcs-12v1.pdf>, Juni 1999.
- [Poll74] JOHN M. POLLARD. *Theorems on Factorization and Primality Testing.* In *Proceedings of Cambridge Philosophy Society*, Band 76 (Cambridge University Press, 1974).
- [Poll75] JOHN M. POLLARD. *A Monte Carlo method for factorization.* *BIT*, Band 15:331–334, 1975.
- [Poll78] JOHN M. POLLARD. *Monte Carlo methods for index computation (mod p).* *Mathematics of Computation*, Band 32(143):918–924, 1978.
- [Pome85] CARL POMERANCE. *The Quadratic Sieve Factoring Algorithm.* In *Proceedings of Eurocrypt 84*, Paris, 1984, T. Beth. N. Cot, and I. Ingemarsson (Herausgeber), Band 209 von *Lecture Notes in Computer Sci.*, Seiten 169–182., 1985.
- [ProdHG] *Gesetz über die Haftung für fehlerhafte Produkte, vom 15. Dezember 1989.* BGBI I 1989, 2198, zuletzt geändert durch Art. 9 Abs. 3 G v. 19. 7.2002 I 2674. <http://bundesrecht.juris.de/bundesrecht/prodhaftg/>, 2002.
- [ProfiMahn] BREMEN ONLINE SERVICES GMBH & Co. KG. *ProfiMahn.* Webseite. <http://www.profimahn.de>, 2005.
- [PTB05a] PHYSIKALISCHE TECHNISCHE BUNDESANSTALT. *ArchiSafe – Webseite.* <http://www.archisafe.de>, 2005.
- [PTB05b] PHYSIKALISCHE TECHNISCHE BUNDESANSTALT. *ArchiSafe – Fachkonzept.* [http://www.archisafe.de/s/c/ai\\_e0CVe/ArchiSafe\\_Dokumente/2005-08-29\\_Fachkonzept\\_V1.0.pdf](http://www.archisafe.de/s/c/ai_e0CVe/ArchiSafe_Dokumente/2005-08-29_Fachkonzept_V1.0.pdf), 2005.
- [PWC99] PRICE WATERHOUSE COOPERS. *Study on the requirements imposed by the Member States, for the purpose of charging taxes, for invoices produced by electronic or other means,* 1999.
- [RaEf02] WOLFGANG RANKL WOLFGANG EFFING. *Handbuch der Chipkarten* (Carl Hanser Verlag, München, Wien, 2002). 4. Auflage, ISBN 3-446-22036-4.

- [RRM05] HERBERT REICHL, ALEXANDER ROßNAGEL, GÜNTER MÜLLER (HRSG.). *Digitaler Personalausweis – Eine Machbarkeitsstudie* (Deutscher Universitätsverlag, 2005). ISBN: 3835000543.
- [RFC793] UNIVERSITY OF SOUTHERN CALIFORNIA INFORMATION SCIENCES INSTITUTE. *Transmission Control Protocol*. Request For Comments – RFC 793. <http://www.ietf.org/rfc/rfc793.txt>, September 1981.
- [RFC1321] RON RIVEST. *The MD5 Message-Digest Algorithm*. Request For Comments – RFC 1321. <http://www.ietf.org/rfc/rfc1321.txt>, April 1992.
- [RFC1521] N. BORENSTEIN N. FREED. *MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies*. Request For Comments – RFC 1521. <http://www.ietf.org/rfc/rfc1521.txt>, September 1993.
- [RFC1847] J. GALVIN, S. MURPHY, S. CROCKER, N. FREED. *Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted*. Request For Comments – RFC 1847. <http://www.ietf.org/rfc/rfc1847.txt>, Oktober 1995.
- [RFC1945] T. BERNERS-LEE, R. FIELDING, H. FRYSTYK. *Hypertext Transfer Protocol – HTTP/1.0*. Request For Comments – RFC 1945. <http://www.ietf.org/rfc/rfc1945.txt>, Mai 1996.
- [RFC1950] P. DEUTSCH J-L. GAILLY. *ZLIB Compressed Data Format Specification version 3.3*. Request For Comments – RFC 1950. <http://www.ietf.org/rfc/rfc1950.txt>, Mai 1996.
- [RFC1951] P. DEUTSCH. *DEFLATE Compressed Data Format Specification version 1.3*. Request For Comments – RFC 1951. <http://www.ietf.org/rfc/rfc1951.txt>, Mai 1996.
- [RFC2231] N. FREED K. MOORE. *MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations*. Request For Comments – RFC 2231. <http://www.ietf.org/rfc/rfc2231.txt>, November 1997.
- [RFC2251] M. WAHL, T. HOWES, S. KILLE. *Lightweight Directory Access Protocol (v3)*. Request For Comments – RFC 2251, Dezember 1997.
- [RFC2252] M. WAHL, A. COULBECK, T. HOWES, S. KILLE. *Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions*. Request For Comments – RFC 2252, Dezember 1997.
- [RFC2253] M. WAHL, S. KILLE, T. HOWES. *Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names*. Request For Comments – RFC 2253, Dezember 1997.
- [RFC2254] T. HOWES. *The String Representation of LDAP Search Filters*. Request For Comments – RFC 2254, Dezember 1997.
- [RFC2255] T. HOWES M. SMITH. *The LDAP URL Format*. Request For Comments – RFC 2255, Dezember 1997.
- [RFC2256] M. WAHL. *A Summary of the X.500(96) User Schema for use with LDAPv3*. Request For Comments – RFC 2256, Dezember 1997.
- [RFC2313] B. KALISKI. *PKCS #1: RSA Encryption - Version 1.5*. Request For Comments – RFC 2313. <http://www.ietf.org/rfc/rfc2313.txt>, 1998.

- [RFC2315] B. KALISKI. *PKCS #7: Cryptographic Message Syntax - Version 1.5.* Request For Comments – RFC 2315. <http://www.ietf.org/rfc/rfc2315.txt>, 1998.
- [RFC2440] J. CALLAS, L. DONNERHACKE, H. FINNEY, R. THAYER. *OpenPGP Message Format.* Request For Comments – RFC 2440. <http://www.ietf.org/rfc/rfc2440.txt>, 1998.
- [RFC2459] R. HOUSLEY, W. FORD, W. POLK, D. SOLO. *OpenPGP Message Format.* Request For Comments – RFC 2459. <http://www.ietf.org/rfc/rfc2459.txt>, Januar 1999.
- [RFC2510] C. ADAMS S. FARRELL. *X.509 Internet Public Key Infrastructure – Certificate Management Protocols.* Request For Comments – RFC 2510. <http://www.ietf.org/rfc/rfc2510.txt>, 1999.
- [RFC2511] M. MYERS, C. ADAMS, D. SOLO, D. KEMP. *Internet X.509 Certificate Request Message Format.* Request For Comments – RFC 2511. <http://www.ietf.org/rfc/rfc2511.txt>, 1999.
- [RFC2560] M. MYERS, R. ANKNEY, A. MALPANI, S. GALPERIN, C. ADAMS. *X.509 Internet Public Key Infrastructure – Online Certificate Status Protocol - OCSP.* Request For Comments – RFC 2560. <http://www.ietf.org/rfc/rfc2560.txt>, 1999.
- [RFC2616] R. FIELDING, J. GETTYS, J. MOGUL, H. FRYSTYK, L. MASINTER, P. LEACH, T. BERNERS-LEE. *Hypertext Transfer Protocol – HTTP/1.1.* Request For Comments – RFC 2616. <http://www.ietf.org/rfc/rfc2616.txt>, Juni 1999.
- [RFC2630] R. HOUSLEY. *Cryptographic Message Syntax (CMS).* Request For Comments – RFC 2630. <http://www.ietf.org/rfc/rfc2630.txt>, Juni 1999.
- [RFC2632] B. RAMSDELL. *S/MIME Version 3 Certificate Handling.* Request For Comments – RFC 2632. <http://www.ietf.org/rfc/rfc2632.txt>, Juni 1999.
- [RFC2633] B. RAMSDELL. *S/MIME Version 3 Message Specification.* Request For Comments – RFC 2633. <http://www.ietf.org/rfc/rfc2633.txt>, Juni 1999.
- [RFC2634] P. HOFFMAN. *Enhanced Security Services for S/MIME.* Request For Comments – RFC 2630. <http://www.ietf.org/rfc/rfc2634.txt>, Juni 1999.
- [RFC2797] M. MYERS, X. LIU, J. SCHAAD, J. WEINSTEIN. *Certificate Management Messages over CMS.* Request For Comments – RFC 2797. <http://www.ietf.org/rfc/rfc2797.txt>, April 2000.
- [RFC3039] S. SANTESSON, W. POLK, P. BARZIN, M. NYSTROM. *Internet X.509 Public Key Infrastructure – Qualified Certificates Profile.* Request For Comments – RFC 3126. <http://www.ietf.org/rfc/rfc3039.txt>, Januar 2001.
- [RFC3126] D. PINKAS, J. ROSS, N. POPE. *Electronic Signature Formats for long term electronic signatures.* Request For Comments – RFC 3126. <http://www.ietf.org/rfc/rfc3126.txt>, September 2001.
- [RFC3156] M. ELKINS, D. DEL TORTO, R. LEVIEN, T. ROESSLER. *MIME Security with OpenPGP.* Request For Comments – RFC 3156. <http://www.ietf.org/rfc/rfc3156.txt>, August 2001.
- [RFC3161] C. ADAMS, P. CAIN, D. PINKAS, R. ZUCCHERATO. *Internet X.509 Public Key Infrastructure – Time-Stamp Protocol (TSP).* Request For Comments – RFC 3161. <http://www.ietf.org/rfc/rfc3161.txt>, August 2001.

- [RFC3274] P. GUTMANN. *Compressed Data Content Type for Cryptographic Message Syntax (CMS)*. Request For Comments – RFC 3274. <http://www.ietf.org/rfc/rfc3274.txt>, Juni 2002.
- [RFC3275] D. EASTLAKE, J. REAGLE, D. SOLO. *(Extensible Markup Language) XML-Signature Syntax and Processing*. Request For Comments – RFC 3275. <http://www.ietf.org/rfc/rfc3275.txt>, März 2002.
- [RFC3280] R. HOUSLEY, W. POLK, W. FORD, D. SOLO. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. Request For Comments – RFC 3280. <http://www.ietf.org/rfc/rfc3280.txt>, April 2002.
- [RFC3281] S. FARREL R. HOUSLEY. *An Internet Attribute Certificate Profile for Authorization*. Request For Comments – RFC 3281. <http://www.ietf.org/rfc/rfc3281.txt>, April 2002.
- [RFC3369] R. HOUSLEY. *Cryptographic Message Syntax (CMS)*. Request For Comments – RFC 3369. <http://www.ietf.org/rfc/rfc3369.txt>, August 2002.
- [RFC3447] J. JONSSON B. KALISKI. *Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1*. Request For Comments – RFC 3447. <http://www.ietf.org/rfc/rfc3447.txt>, 2003.
- [RFC3647] S. CHOKHANI, W. FORD, R. SABETT, C. MERRILL, S. WU. *Internet X.509 Public Key Infrastructure – Certificate Policy and Certification Practices Framework*. Request For Comments – RFC 3647. <http://www.ietf.org/rfc/rfc3647.txt>, November 2003.
- [RFC3852] R. HOUSLEY. *Cryptographic Message Syntax (CMS)*. Request For Comments – RFC 3852. <http://www.ietf.org/rfc/rfc3852.txt>, Juli 2004.
- [Ries94] HANS RIESEL. *Prime Numbers and Computer Methods for Factorization*, Band 126 *Progress in Mathematics* (Birkhäuser, 1994), 2.
- [RLP-JP] MINISTERIUM DER JUSTIZ RHEINLAND-PFALZ. *Justizportal Rheinland-Pfalz*. Webseite. <http://www.justiz.rlp.de/justiz/nav/919/>, 2005.
- [Romp90] J. ROMPEL. *One-way functions are necessary and sufficient for secure signatures*. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, Seiten 387 – 394 (Association for Computing Machinery (ACM), 1990). ISBN:0-89791-361-2.
- [Ross00] ALEXANDER ROSSNAGEL. *Der europäische Standard: Die elektronische Signatur der europäischen Richtlinie*. In IVO GEIS (Herausgeber), *Die digitale Signatur – eine Sicherheitstechnik für die Informationsgesellschaft*, Seiten 195–230 (AWV-Eigenverlag, Eschborn, 2000).
- [Ross03a] ALEXANDER ROSSNAGEL. *Das elektronische Verwaltungsverfahren – Das Dritte Verwaltungsverfahrensänderungsgesetz*. In *Neue Juristische Wochenschrift*, Seiten 469–476. [http://www.uni-kassel.de/fb7/oefl\\_recht/publikationen/pubOrdner/NJW2003-VwVfG-2.pdf](http://www.uni-kassel.de/fb7/oefl_recht/publikationen/pubOrdner/NJW2003-VwVfG-2.pdf), 2003.
- [Ross03b] ALEXANDER ROSSNAGEL. *Die qualifizierte elektronische Signaturen mit Einschränkungen im Besteuerungsverfahren*. In *Kommunikation und Recht*, Band 8:379–385. [http://www.uni-kassel.de/fb7/oefl\\_recht/publikationen/pubOrdner/NJW2003-VwVfG-2.pdf](http://www.uni-kassel.de/fb7/oefl_recht/publikationen/pubOrdner/NJW2003-VwVfG-2.pdf), 2003.

- [Ross98] ALEXANDER ROSSNAGEL. *Die Sicherheitsvermutung des Signaturgesetzes.* In *Neue Juristische Wochenschrift*, Seite 3312, 1998.
- [RoSc05] ALEXANDER ROSSNAGEL PAUL SCHMÜCKER (Herausgeber), *Beweiskräftige elektronische Archivierung - Bieten elektronische Signaturen Rechtssicherheit?*, In *Gesundheitswesen in der Praxis* ISBN 978-3-87081-427-4 (Verlagsgruppe Hüthig, Jehle, Rehm, 2005), 2005.
- [RSA200] JENS FRANKE, FRIEDRICH BAHR, M. BÖHM, THORSTEN KLEINJUNG, PETER L. MONTGOMERY, HERMAN TE RIELE. *Announcement: Factorization of RSA200.* <http://www.loria.fr/~zimmerma/records/rsa200>, 2005.
- [RSA640] HEISE. *RSA-640 geknackt.* Meldung vom 09.11.2005. <http://www.heise.de/security/news/meldung/65957>, 2005.
- [RSA78] RONALD L. RIVEST, ADI SHAMIR, LEONARD ADLEMAN. *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.* In *Communications of the ACM*, Band 21(2):120–126, 1978.
- [SaAr98] T. SATOH K. ARAKI. *Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves.* In *Comm. Math. Univ. Sancti. Pauli*, Band 47:81–92, 1998.
- [Schn89] CLAUS P. SCHNORR. *Efficient identification and signatures for smart cards.* In *Advances in Cryptology – CRYPTO '89*, Band 435 von *Lecture Notes in Computer Science*, Seiten 239–252 (Springer-Verlag, 1990).
- [Schn91] CLAUS P. SCHNORR. *Efficient Signature Generation by Smart Cards.* In *Journal of Cryptology*, Band 4(3):161–174, 1991.
- [ScLe84] CLAUS PETER SCHNORR HENDRIK W. LENSTRA, JR. *A Monte Carlo Factoring Algorithm With Linear Storage.* In *Mathematics of Computation*, Band 43(167):289–311, 1984.
- [SGBI] *Sozialgesetzbuch - Erstes Buch (I) Allgemeiner Teil.* Artikel I des Gesetzes vom 11. Dezember 1975, BGBI. I S. 3015, zuletzt geändert durch Art. 2 G v. 21. 3.2005 I 818. [http://bundesrecht.juris.de/bundesrecht/sgb\\_1/](http://bundesrecht.juris.de/bundesrecht/sgb_1/), 2005.
- [SGBIV] *Sozialgesetzbuch - Viertes Buch (IV) Gemeinsame Vorschriften für die Sozialversicherung.* Artikel I des Gesetzes vom 23. Dezember 1976, BGBI. I S. 3845, zuletzt geändert durch Art. 22 G. v. 21.3.2005 I 818. [http://bundesrecht.juris.de/bundesrecht/sgb\\_4/](http://bundesrecht.juris.de/bundesrecht/sgb_4/), 2005.
- [SGBX] *Sozialgesetzbuch - Zehntes Buch (X) Sozialverwaltungsverfahren und Sozialdatenschutz.* Gesetz vom 18. August 1980, BGBI. I 1980, 1469, 2218 BGBI I 1982, 1450, zuletzt geändert durch Art. 10 G. v. 21.4.2005 I 1073. [http://bundesrecht.juris.de/bundesrecht/sgb\\_10/](http://bundesrecht.juris.de/bundesrecht/sgb_10/), 2005.
- [Shan72] DANIEL SHANKS. *The infrastructure of a real quadratic field and its applications.* In *Proceedings of Number Theory Conference, Boulder 1972*, Seiten 217–224 (1972).
- [SigB05] ULRIKE LINDE SIGNATURBÜNDNIS, UAG „ORGANISATIONSZERTIFIKATE“. *Anforderungen an Organisationszertifikate.* Version 0.1 vom 29.06.2005, 2005.
- [SigG] *Gesetz über Rahmenbedingungen für elektronische Signaturen und zur Änderung weiterer Vorschriften, vom 16.05.2001.* BGBI. 2001 Teil I Nr. 22, S. 876 ff, geändert durch Art. 1 G v. 4. 1.2005 I 2. [http://bundesrecht.juris.de/bundesrecht/sigg\\_2001/](http://bundesrecht.juris.de/bundesrecht/sigg_2001/), 2001.

- [SigG97] *Gesetz zur digitalen Signatur (Signaturgesetz, SigG).* Artikel 3 des Informations- und Kommunikationsdienste-Gesetz (IuKDG) vom 22. Juli 1997, BGBl. I S. 1870-1872.  
<http://www.bundesnetzagentur.de/media/archive/894.pdf>, 1997.
- [SigGAendG] *Erstes Gesetz zur Änderung des Signaturgesetzes (1. SigGÄndG), vom 04. Januar 2005.* BGBl. I Nr. 1 S. 2-3, 10.01.2005.  
<http://217.160.60.235/BGBL/bgb1f/bgb105s0002.pdf>, 2005.
- [SigGBeg] *Begründung zum Entwurf eines Gesetzes über Rahmenbedingungen für elektronische Signaturen und zur Änderung weiterer Vorschriften (SigG) in der Fassung des Kabinettsbeschlusses vom 16. August 2000.*  
<http://www.pca.dfn.de/bibliothek/sigg/germany/begrueindung-zum-signaturgesetz-2001-05-16.pdf>, 2000.
- [SigV] *Verordnung zur elektronischen Signatur, vom 16.11.2001.* BGBl. 2001 Teil I Nr. 59, S. 3074 ff, geändert durch Art. 2 G v. 4. 1.2005 I 2.  
[http://bundesrecht.juris.de/bundesrecht/sigv\\_2001/](http://bundesrecht.juris.de/bundesrecht/sigv_2001/), 2001.
- [SigV97] *Verordnung zur digitalen Signatur (Signaturverordnung - SigV), vom 22. Oktober 1997.* BGBl. I Nr. 70, 27. Oktober 1997, S. 2498.  
<http://www.bundesnetzagentur.de/media/archive/895.pdf>, 1997.
- [SigVAendV] *Erste Verordnung zur Änderung der Signaturverordnung (1. SigVÄndV).* BGBl. I S. 981, 22.06.2000.  
<http://217.160.60.235/BGBL/bgb1f/b100029f.pdf>, 2000.
- [SigVBeg] *Begründung zur Verordnung zur elektronischen Signatur (SigV).*  
<http://www.pca.dfn.de/bibliothek/sigg/germany/begrueindung-zur-signaturverordnung-2001-11-16.pdf>, 2001.
- [Silv87] ROBERT D. SILVERMAN. *The multiple polynomial quadratic sieve. Mathematics of Computation*, Band 48:329–339, 1987.
- [SiSu98] JOSEPH H. SILVERMAN JOE SUZUKI. *Elliptic Curve Discrete Logarithms and the Index Calculus.* In *Advances in Cryptology – Proceedings of Asiacrypt’98*, Band 1514 von *Lecture Notes in Computer Science*, Seiten 110–125 (Springer-Verlag, 1998).
- [SKG+04] B. SKIERA, W. KÖNIG, S. GENSLER, T. WEITZEL, D. BEIMBORN, S. BLUMENBERG, J. FRANKE, D. PFAFF. *Financial Chain Management. Prozessanalyse, Effizienzpotenziale und Outsourcing.* Norderstedt, 2004.
- [Smar99] NIGEL P. SMART. *The Discrete Logarithm Problem on Elliptic Curves of Trace One.* In *Journal of Cryptology*, Band 12(3):193–196, 1999.
- [SPKC-ID] CARL M. ELLISON, BILL FRANTZ, BUTLER LAMPSON, RON RIVEST, BRIAN M. THOMAS, TATU YLONEN. *Simple Public Key Certificate.* IETF-Internet-Draft – draft-ietf-spki-cert-structure-06.txt.  
<http://world.std.com/~cme/spki.txt>, Juli 1999.
- [SRVwV] *Allgemeine Verwaltungsvorschrift über das Rechnungswesen in der Sozialversicherung (SRVwV).* vom 15. Juli 1999, BAnz. Nr. 145 a vom 06.08.1999, zuletzt geändert durch die Dritte Allgemeine Verwaltungsvorschrift zur Änderung der Allgemeinen Verwaltungsvorschrift über das Rechnungswesen in der Sozialversicherung vom 10. Juni 2005, Bundesanzeiger vom 16. Juni 2005, Jahrgang 57, S. 9087. <http://www.hvbg.de/d/revision/gruen/kap3a/kap32.doc>, 1999.

- [Stal95] WILLIAM STALLINGS. *The PGP Web of Trust*. BYTE.  
<http://www.byte.com/art/9502/sec13/art4.htm>, Februar 1995.
- [StDUeV] *Verordnung zur elektronischen Übermittlung von Steuererklärungen und sonstigen für das Besteuerungsverfahren erforderlichen Daten (Steuerdaten-Übermittlungsverordnung – StDÜV)*, vom 28.01.2003. BGBl. I, Nr. 5, S. 139. <http://217.160.60.235/BGBL/bgb1f/bgb103s0139.pdf>, 2003.
- [StGB] *Strafgesetzbuch*. RGBl 1871, 127, neugefasst durch Bek. v. 13.11.1998 I 3322, zuletzt geändert durch Art. 5 G v. 24. 6.2005 I 1841.  
<http://bundesrecht.juris.de/bundesrecht/stgb/>, 1871.
- [SunJCS] SUN CORPORATION. *Java Code Signing*. Sun Developer Network Artikel.  
<http://java.sun.com/j2se/1.4.2/docs/guide/jws/developersguide/development.html#security>, September 1999.
- [SVRV] *Verordnung über den Zahlungsverkehr, die Buchführung und die Rechnungslegung in der Sozialversicherung (Sozialversicherungs-Rechnungsverordnung - SVRV)*. vom 15. Juli 1999 (BGBl. I S. 1627), zuletzt geändert durch Artikel 3 des Gesetzes der Rahmenbedingungen für elektronische Signaturen und zur Änderung weiterer Vorschriften vom 16.05.2001. <http://www.hvbg.de/d/revision/gruen/kap3a/kap31.doc>, 1999.
- [Taka98] TSUYOSHI TAKAGI. *Fast RSA-Type Cryptosystem Modulo  $p^kq$* . In HUGO KRAWCZYK (Herausgeber), *Advances in Cryptology – CRYPTO '98*, Band 1462 von *Lecture Notes in Computer Science*, Seiten 318–326 (Springer-Verlag, 1998).
- [Tesk98] EDLYN TESKE. *New Algorithms for Finite Abelian Groups*. ISBN 3-8265-4045-X, 1998.
- [Tett00] ALEXANDER TETTENBORN. *Die Evaluierung des Signaturgesetzes und Umsetzung der EG-Signaturrichtlinie*. In IVO GEIS (Herausgeber), *Die digitale Signatur – eine Sicherheitstechnik für die Informationsgesellschaft*, Seiten 231–252 (AWV-Eigenverlag, Eschborn, 2000).
- [TIFF(v6.0)] ADOBE SYSTEMS INCORPORATED. *TIFF – Revision 6.0*. vom 3. Juni 1992.  
<http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf>.
- [UStDV] *Umsatzsteuer-Durchführungsverordnung (UStDV)*. vom 21. Dezember 1979, BGBl I 1979, 2359, neugefasst durch Bek. v. 21. 2.2005 I 434.  
[http://bundesrecht.juris.de/bundesrecht/ustdv\\_1980/](http://bundesrecht.juris.de/bundesrecht/ustdv_1980/), 1979.
- [UStG] *Umsatzsteuergesetz*. vom 26. November 1979, BGBl I 1979, 1953, Neugefasst durch Bek. v. 21.2.2005 I 386.  
[http://bundesrecht.juris.de/bundesrecht/ustg\\_1980/gesamt.pdf](http://bundesrecht.juris.de/bundesrecht/ustg_1980/gesamt.pdf), 1979.
- [UStR] *Allgemeine Verwaltungsvorschrift zur Ausführung des Umsatzsteuergesetzes (Umsatzsteuer-Richtlinien 2005 - UStR 2005 -)*. BAnz. Az. 696/04, 2004.
- [Utah-DSA] UTAH STATE. *Utah Digital Signature Act*. Utah Code – Title 46 – Chapter 03. [http://www.le.state.ut.us/~code/TITLE46/46\\_02.htm](http://www.le.state.ut.us/~code/TITLE46/46_02.htm), Mai 1995.
- [Vaud96] SERGE VAUDENAY. *Hidden Collisions on DSS*. In *Advances in Cryptology – CRYPTO '96*, Band 1109 von *Lecture Notes in Computer Science*, Seiten 83–88 (Springer-Verlag, 1996).

- [VerwVfAendG] *Drittes Gesetz zur Änderung verwaltungsverfahrensrechtlicher Vorschriften, vom 21. August 2002.* BGBI. I Nr. 60, S. 3322-3343.  
<https://www.dfn-pca.de/bibliothek/sigg/germany/verwaltungsverfahrensgesetz-2002-08-21.pdf>, 2002.
- [VgV] *Verordnung über die Vergabe öffentlicher Aufträge.* 9. Januar 2001, BGBI I 2001, 110, zuletzt geändert durch Art. 2 G v. 1. 9.2005 I 2676.  
[http://bundesrecht.juris.de/bundesrecht/vgv\\_2001/](http://bundesrecht.juris.de/bundesrecht/vgv_2001/), 2001.
- [Vief05] WOLFRAM VIEFHUES. *Das Gesetz über die Verwendung elektronischer Kommunikationsformen in der Justiz. Neue Juristische Wochenschrift*, Seiten 1009–1016, 2005.
- [VOI05] VERBAND ORGANISATIONS UND INFORMATIONSSYSTEME (VOI).  
*Dokumenten-Management vom Archiv zum EnterpriseContentManagement.* Code of Practice. <http://www.voi.de>, 2005.
- [VwVfG] *Verwaltungsverfahrensgesetz* vom 25. Mai 1976, BGBI 1976, 1253, zuletzt geändert durch Art. 4 Abs. 8 G v. 5.5.2004 I 718.  
<http://bundesrecht.juris.de/bundesrecht/vwvfg/>, 2004.
- [Wagn94] KLAUS WAGNER. *Einführung in die Theoretische Informatik* (Springer-Verlag, 1994). ISBN 3-540-58139-1.
- [WAP-Cert] OPEN MOBILE ALLIANCE. *WAP Certificate profile Specification.* WAP 2.0 Specification – Wireless Security.  
<http://www1.wapforum.org/tech/terms.asp?doc=WAP-211-WAPCert-20010522-a.pdf>, 2001.
- [WaYu05] XIAOYUN WANG HONGBO YU. *How to Break MD5 and Other Hash Functions.* In *Advances in Cryptology – EUROCRYPT 2005*, Band 3494 von *Lecture Notes in Computer Science*, Seiten 19–35 (Springer, 2005).
- [Webe96] DAMIAN WEBER. *Computing discrete logarithms with the number field sieve.* In HENRI COHEN (Herausgeber), *Algorithmic Number Theory, ANTS-II*, Band 1122 von *Lecture Notes in Computer Science* (Springer-Verlag, 1996).
- [Will82] HUGH C. WILLIAMS. *A  $p + 1$  Method of Factoring. Mathematics of Computation*, Band 39:225–234, 1982.
- [WYY05a] XIAOYUN WANG, YIQUN LISA YIN, HONGBO YU. *Finding Collisions in the Full SHA-1.* In *Advances in Cryptology – CRYPTO 2005*, Lecture Notes in Computer Science (Springer, 2005).  
<http://www.astalavista.com/index.php?section=directory&linkid=4713>.
- [X.408] ITU-T. *ITU-T Recommendation X.408.* Message Handling Systems: Encoded Information Type Conversion Rules, 1988.
- [X.500] ITU-T. *ITU-T Recommendation X.500 (1993) - ISO-IEC 9594-1:1993.* Information technology Open Systems Interconnection The Directory: Overview of Concepts, Models, and Services, 1993.
- [X.501] ITU-T. *ITU-T Recommendation X.501 (1993) - ISO-IEC 9594-2:1993.* Information technology Open Systems Interconnection The Directory: Models, 1993.

- [X.509:97] ITU-T. *ITU-T Recommendation X.509 (1997) - ISO-IEC 9594-8:1997.* Information technology Open Systems Interconnection The Directory: Authentication Framework, August 1997.
- [X.509:00] ITU-T. *ITU-T Recommendation X.509 (2000) - ISO-IEC 9594-8:2000.* Information technology Open Systems Interconnection The Directory: Public-key and attribute certificate frameworks, März 2000.
- [X.519] ITU-T. *ITU-T Recommendation X.519 (1993) - ISO-IEC 9594-5:1993.* Information technology Open Systems Interconnection The Directory: Protocol Specifications, 1993.
- [X.680] ITU-T. *ITU-T Recommendation X.680 (2002) — ISO/IEC 8824-1:2002.* Information Technology - Abstract Syntax Notation One (ASN.1): Specification of Basic Notation, 2002.
- [X.690] ITU-T. *ITU-T Recommendation X.690 (2002) — ISO/IEC 8825-1:2002.* Information Technology - ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER), 2002.
- [X.691] ITU-T. *ITU-T Recommendation X.691 (2002) — ISO/IEC 8825-2:2002.* Information Technology - ASN.1 Encoding Rules: Specification of Packed Encoding Rules (PER), 2002.
- [X.693] ITU-T. *ITU-T Recommendation X.693 (2002) — ISO/IEC 8825-4:2002.* Information Technology - ASN.1 Encoding Rules: XML Encoding Rules (XER), 2002.
- [XJustiz] OSCI-LEITSTELLE. *XJustiz – Elektronischer Rechtsverkehr mit XML.* Webseite. <http://www.xjustiz.de>, 2005.
- [XMeld] OSCI-LEITSTELLE. *XMeld – Standardisierter Datenaustausch im Meldewesen.* Webseite. <http://www1.osci.de/sixcms/detail.php?id=1181> 2005.
- [XML-DSig] D. EASTLAKE, J. REAGLE, D. SOLO. *XML-Signature Syntax and Processing.* W3C Recommendation. <http://www.w3.org/TR/xmldsig-core/>, Februar 2002.
- [XML(v1.0)] FRANÇOIS YERGEAU, TIM BRAY, JEAN PAOLI, C. M. SPERBERG-MCQUEEN, EVE MALER. *Extensible Markup Language (XML) 1.0 (Third Edition).* W3C Recommendation. <http://www.w3.org/TR/2004/REC-xml-20040204/>, Februar 2004.
- [XML(v1.1)] FRANÇOIS YERGEAU, JOHN COWAN, TIM BRAY, JEAN PAOLI, C. M. SPERBERG-MCQUEEN, EVE MALER. *Extensible Markup Language (XML) 1.1.* W3C Recommendation. <http://www.w3.org/TR/2004/REC-xml11-20040204/>, Februar 2004.
- [XPath] J. CLARK S. DEROSSE. *XML Path Language (XPath) Version 1.0.* W3C Recommendation. <http://www.w3.org/TR/1999/REC-xpath-19991116>, Oktober 1999.
- [XSL] S. ADLER, A. BERGL, J. CARUSO, S. DEACH, P. GROSSO, E. GUTENTAG, A. MILOWSKI, S. PARRELL, J. RICHMAN, S. ZILLES. *Extensible Stylesheet Language (XSL).* W3C Proposed Recommendation. <http://www.w3.org/TR/2001/PR-xsl-20010828/>, August 2001.

- [XSLT] J. CLARK. *XSL Transforms (XSLT) Version 1.0*. W3C Recommendation.  
<http://www.w3.org/TR/1999/REC-xslt-19991116.html>, November 1999.
- [ZeitG] *Gesetz über die Zeitbestimmung – Zeitgesetz (ZeitG)*. vom 25.7.1978,  
BGBl I 1978, 1110, zuletzt geändert durch Gesetz zur Änderung des  
Zeitgesetzes vom 13.9.1994, BGBl I, 2322.  
[http://www.rechtliches.de/info\\_ZeitG.html](http://www.rechtliches.de/info_ZeitG.html), 1978.
- [Zeun00] VOLKER ZEUNER. *Erfahrungen mit der Umsetzung des deutschen  
Signaturgesetzes*. In IVO GEIS (Herausgeber), *Die digitale Signatur – eine  
Sicherheitstechnik für die Informationsgesellschaft*, Seiten 120–126  
(AWV-Eigenverlag, Eschborn, 2000).
- [ZPO] *Zivilprozeßordnung*. BGBl 1950, 455, 512, 533, Zuletzt geändert durch  
Art. 4 G v. 21. 4.2005 I 1073. <http://bundesrecht.juris.de/bundesrecht/zpo/>,  
1950.