In: Horster, P. (Hrsg.): *Trust Center*. Proceedings der Arbeitstagung Trust Center 95, Vieweg-Verlag, Braunschweig 1995, S. 165-172.

Neue Zertifikate für asymmetrische Sicherheitsprotokolle

Dirk Fox, Maik Müller

Universität Siegen
Institut für Nachrichtenübermittlung
Postfach 10 12 40, D-57068 Siegen
{fox, mueller}@nue.et-inf.uni-siegen.de

Zusammenfassung

Zertifikate spielen eine wichtige Rolle in vielen asymmetrischen Protokollen für Authentisierung und Schlüsselaustausch. Verbreitete Protokolle wie beispielsweise die in X.509 und ISO 9798-3 genormten verwenden dabei dasselbe asymmetrische Schlüsselsystem für die Erbringung der Sicherheitsdienste Integrität, Vertraulichkeit und Authentisierung. Daher setzen diese Protokolle die Existenz asymmetrischer Verfahren voraus, die sich sowohl zum Signieren als auch zum Verschlüsseln von Daten eignen – und sind damit derzeit auf das RSA-Verfahren festgelegt.

Das für diese Protokolle spezifizierte und inzwischen vielgenutzte Zertifikatformat nach X.509 ist jedoch für asymmetrische Protokolle und Kryptoverfahren ungeeignet, die diese spezielle „Symmetrie“-Eigenschaft nicht nutzen oder besitzen.

An einigen wichtigen asymmetrischen Protokollen wird diese Schwierigkeit exemplarisch herausgearbeitet. Es werden Zertifikatspezifikationen vorgeschlagen und diskutiert, die in zertifikatsbasierten Sicherheitsprotokollen die Verwendung eines separaten Schlüsselsystems für jeden Sicherheitsdienst erlauben.

1 Einleitung

Klassische symmetrische Authentisierungs- und Schlüsselaustauschprotokolle wie z.B. das auf den von Needham und Schroeder vorgestellten Protokollen aufbauende Kerberos-System erfordern *online* verfügbare Authentisierungsserver [NeSc_78, StNS_88, NeTs_94]. Sie sind daher für viele Anwendungen, insbesondere solche, die auf eine möglicherweise weltweite Sicherheitsinfrastruktur zurückgreifen müssen, nicht geeignet. Auch die Einführung von Tikkets und speziellen *Ticket Granting Servern* wie in jüngeren Kerveros-Versionen vorgesehen löst das Problem nur partiell: Zwar müssen zentrale Authentisierungsserver nur noch beim Login eines Benutzers *online* verfügbar sein; eine ständige Verfügbarkeit von dezentralen *Ticket Granting Servern* ist jedoch auch weiterhin erforderlich [BeMe_91].

Eine Alternative stellen asymmetrische Protokolle dar. Die Authentisierung eines Kommunikationspartners und die Vereinbarung eines gemeinsamen *Session Keys* erfolgt dabei mit Hilfe

öffentlicher Schlüssel: Ein Teilnehmer kann selbst – ohne eine unabhängige, dritte Instanz – mit dem öffentlichen Schlüssel des Kommunikationspartners die Prüfung von Authentizität und Integrität der Protokollelemente vornehmen. Die Originalität dieses öffentlichen Schlüssels wiederum wird durch ein sogenanntes Zertifikat gewährleistet. Dieser Gültigkeitsnachweis wird *offline* zu einem früheren Zeitpunkt von einer unabhängigen Zertifizierungsinstanz ausgestellt, der beide Kommunikationsteilnehmer vertrauen [Rula_93].

Asymmetrische Protokolle zur Authentisierung kommen daher ohne einen *online* verfügbaren Server aus. Einzelne Protokolle, wie beispielsweise SPX oder X.509, erfordern für bestimmte Dienste wie die Vereinbarung eines gemeinsamen *Session Keys* einen *online*-Zugriff auf ein Verzeichnis, in dem die öffentlichen Schlüssel der Kommunikationspartner mit Zertifikat abgelegt sind. Dieser passive *Directory Service* kann allerdings leicht dezentral erbracht werden.

Werden die Zertifizierungsinstanzen hierarchisch oder vernetzt angeordnet, kann so eine weltweite Sicherheitsinfrastruktur geschaffen werden, die auch zwischen einander vollständig unbekannten Teilnehmern authentischen und vertraulichen Datenaustausch ermöglicht. Damit zwei Teilnehmer einander authentisieren können, genügt das gemeinsame Vertrauen in eine oder eine Hierarchie von Zertifizierungsinstanzen.

2 Zertifikatformate

Die Verwendung von Zertifikaten in einer Sicherheitsinfrastruktur erfordert die Vereinbarung eines einheitlichen Zertifikataufbaus. Von CCITT und ISO wurde im Zusammenhang mit dem X.509-Protokoll ein solches Format festgelegt [CCITT_89, ISO_92]. Dessen ASN.1-Spezifikation wurde in PKCS von *Public Key Partners* (PKP Inc.) und dem Internet-Standard *Privacy Enhancement for Electronic Mail* (PEM, RFC 1422, Appendix A) übernommen [Kali_91, Kent_93]. Danach hat ein Zertifikat die folgende Gestalt:

```
Certificate ::= SIGNED SEQUENCE {
      version [0]             Version DEFAULT v1988,
      serialNumber            CertificateSerialNumber,
      signature               AlgorithmIdentifier,
      issuer                  Name,
      validity                Validity,
      subject                 Name,
      subjectPublicKeyInfo    SubjectPublicKeyInfo
}
```

Neben der Versionsnummer des Zertifikatformats werden eine von der Zertifizierungsinstanz vergebene Seriennummer, der Bezeichner des Signieralgorithmus (ggf. mit zusätzlichen Parametern), der Name der Zertifizierungsinstanz und ein Gültigkeitszeitraum (von/bis) eingetragen. Der zu zertifizierende öffentliche Schlüssel wird seinem Besitzer (`subject`) und einem asymmetrischen Algorithmus zugeordnet:

```
SubjectPublicKey Info ::= SEQUENCE {
      algorithm               AlgorithmIdentifier,
      subjectPublicKey        BIT STRING
}
```

2.1 Beschränkte Allgemeinheit genormter Protokolle

Die im X.509-Standard vorgeschlagenen und in die ISO-Normung eingegangenen Protokolle zur gegenseitigen Authentisierung eines Kommunikationspartners mit gleichzeitiger Vereinbarung eines gemeinsamen *Session Keys* erfordern ein asymmetrisches Kryptoverfahren, das mit demselben Schlüsselsystem sowohl digitale Signaturen erzeugen bzw. prüfen als auch ver- und entschlüsseln kann.

So werden beispielsweise im *Three Way*-Protokoll nach X.509 die öffentlichen Schlüssel von Alice und Bob jeweils wechselseitig zum Verschlüsseln des *Session Key*-Teils (K_A bzw. K_B), d.h. zur Sicherstellung von *Vertraulichkeit*, und zum Prüfen der Signatur des übermittelten Protokolltokens, d.h. zur Feststellung der *Integrität* der übertragenen Daten verwendet (Bild 2.1; [CCITT_89]).

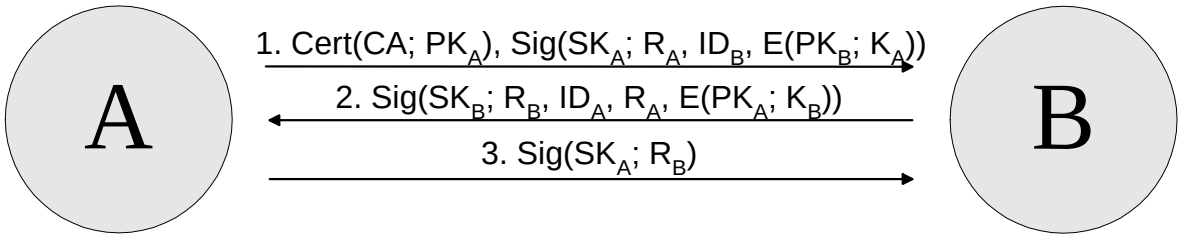

Bild 2.1: Drei-Wege-Authentisierung mit Schlüsselvereinbarung nach X.509

Auch in dem von der ISO genormten *Two Pass Parallel*-Protokoll (ISO 9798-3), das sich wegen der gegenseitigen Unabhängigkeit und Symmetrie der Protokollschritte 1/2 und 3/4 besonders für praktische Anwendungen eignet, werden die öffentlichen Schlüssel von Alice und Bob zum Verschlüsseln und zum Prüfen der Signaturen verwendet (Bild 2.2; [ISO_92]).

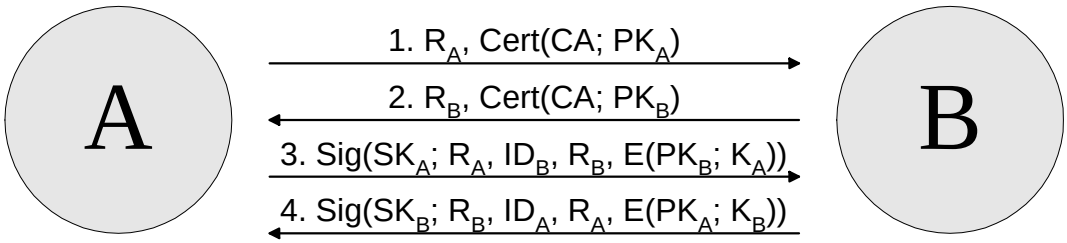

Bild 2.2: Two Pass Parallel-Authentisierungsprotokoll mit Schlüsselvereinbarung nach ISO 9798-3

Damit sind diese Protokolle auf das einzige derzeit bekannte asymmetrische Kryptoverfahren festgelegt, das eine solche „Symmetrie“ aufweist: das RSA-Verfahren [RSA_78]. Im Anhang des X.509 bzw. ISO-Standards wird daher auch RSA als empfohlenes Kryptoverfahren beschrieben. Die Sicherheit der Protokolle hängt damit von der praktischen Unlösbarkeit eines bestimmten mathematischen Problems (der Primfaktorzerlegung sehr großer Zahlen) ab – und

dies nicht einmal bewiesenermaßen.[1] Digitale Signatursysteme, deren Sicherheit auf der praktischen Unlösbarkeit anderer mathematischer Probleme beruht wie z.B. dem Diskreten Logarithmusproblem [ElGa_85, NIST_94, HoPe_94], können in diesen Protokollen ebensowenig eingesetzt werden wie Verfahren, für die die Äquivalenz zum Faktorisierungsproblem nachgewiesen ist (z.B. das GMR-Signatursystem [GoMR_88, FoPf_91]) oder solche Signaturverfahren, die besonders interessante Eigenschaften besitzen (z.B. Fail-Stop-Signaturen [PfWa_91]).

Dabei besteht grundsätzlich keine Notwendigkeit, die unterschiedlichen Sicherheitsdienste mit demselben asymmetrischen Schlüsselsystem zu erbringen: So könnte Alice in den oben skizzierten Protokollen den von ihr generierten *Session Key*-Teil K_A mit einem *Privacy Public Key* (PPK_B) von Bob verschlüsseln, um *Vertraulichkeit* zu erreichen, und ihre Daten anschließend mit ihrem *Integrity Secret Key* (ISK_A) zur Gewährleistung von *Integrität* signieren. Bob verführe analog. Beide prüften anschließend gegenseitig ihre Signaturen mit dem *Integrity Public Key* des Partners (IPK_i) und entschlüsselten den empfangenen Teil K_i des *Session Key* mit ihrem eigenen *Privacy Secret Key* (PSK_i).

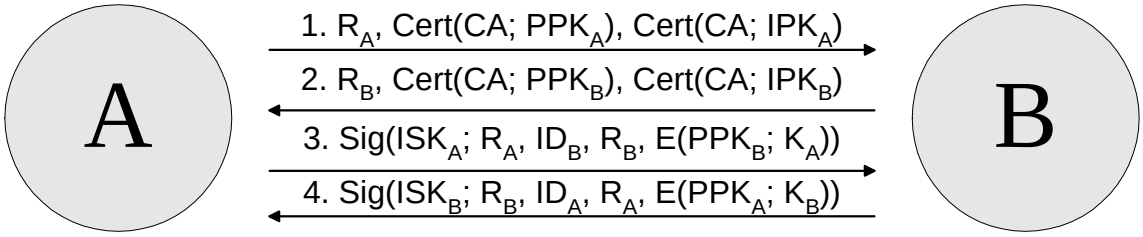

Bild 2.3: Two Pass Parallel-Protokoll mit getrennten Schlüsselsystemen

Dazu sind allerdings zusätzliche Zertifikate oder solche erforderlich, die zwei öffentliche Schlüssel umfassen, da für jeden Sicherheitsdienst die Authentizität eines öffentlichen Schlüssels sichergestellt werden muß (Bild 2.3).

2.2 Anforderungen weiterer asymmetrischer Protokolle

Das im Rahmen der *Distributed System Security Architecture* (DSSA) bei DEC entwickelte asymmetrische SPX-Protokoll zur Authentisierung und Schlüsselvereinbarung arbeitet im Unterschied zu den X.509- und ISO-Protokollen mit separaten Zertifikaten für Verschlüsselungs- und Signaturschlüssel [WoLa_92].

Nach Anforderung des Zertifikats von einem *Certificate Distribution Center* (CDC) wird in Protokollschritt 3 von Alice der generierte *Session Key* K_{AB} an Bob geschickt, verschlüsselt mit dessen *Public Key* (PK_B), den sie in Schritt 2 erhalten hat (Sicherstellung von Vertraulich-

[1] Bis heute ist es eine nicht bewiese (wenn auch gut begründete) Annahme, daß das Brechen des RSA-Verfahrens äquivalent dem Faktorisierungsproblem ist.

keit). Die Authentizität und Integrität dieses verschlüsselten *Session Key* K_{AB} wird von Alice mit einer Digitalen Signatur sichergestellt, die sie mit einem *Secret Delegation Key* (SDK_A) berechnet. Den zugehörigen *Public Delegation Key* (PDK_A) verschickt sie in Gestalt eines Tickets, signiert mit ihrem *Secret Key* und der Angabe eines Gültigkeitszeitraums (Bild 2.4; [GoBD_93]).

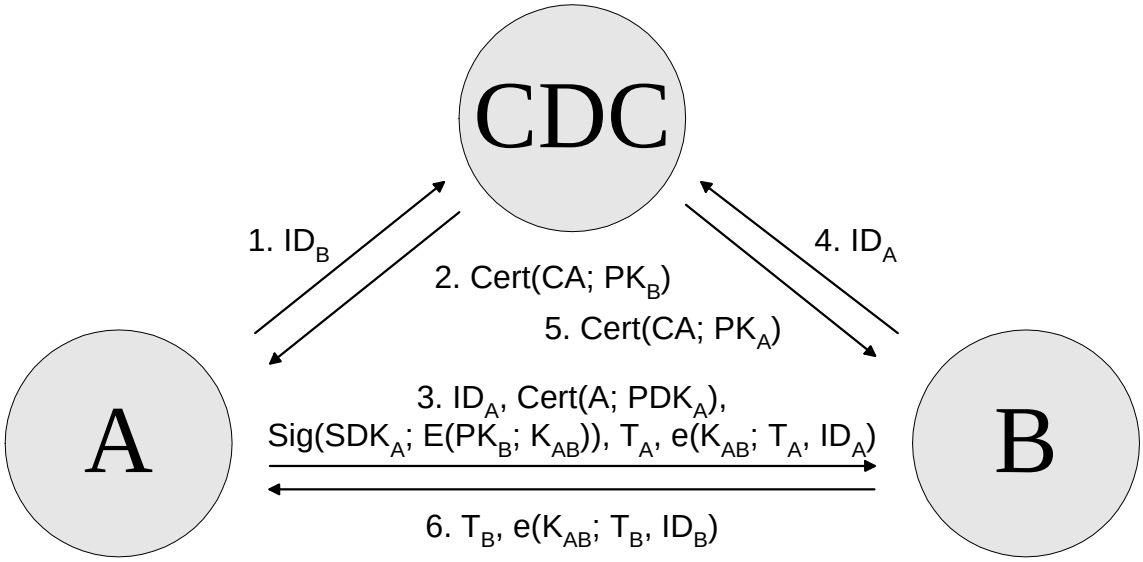

Bild 2.4: Ablauf des SPX-Protokolls

Bob kann die von Alice erzeugte Signatur (und damit die Authentizität des PDK_A) mit dem in Schritt 4 angeforderten *Public Key* von Alice (PK_A) prüfen (*Integrität*) und den *Session Key* mit seinem *Secret Key* (SK_B) entschlüsseln (*Vertraulichkeit*).

Die Symmetrie der Protokollschritte 1/2 und 4/5 legt jedoch nahe, in einer Implementierung jeweils die gleichen Zertifikate zu verwenden – und damit auch dieses Protokoll auf das asymmetrische Kryptosystem RSA festzulegen. Die Verwendung unterschiedlicher Schlüsselsysteme für die Sicherheitsdienste *Integrität* und *Vertraulickeit* würde zudem eine Ergänzung der Zertifikatsanfragen beim CDC in den Protokollschritten 1 und 4 erfordern.

2.3 Anforderungen neuartiger asymmetrischer Protokolle

Für neuartige Authentisierungs- und Schlüsselaustauschprotokolle wie beispielsweise das *Station-to-Station Protocol* (STS) ist das in X.509 spezifizierte Zertifikatformat ungeeignet. STS ist im Kern eine Erweiterung des Diffie-Hellman-Schlüsselaustauschprotokolls (DH) um einen Authentizitätsmechanismus. Durch Zertifikate wird dabei die Authentizität der DH-Parameter und öffentlichen Signatur-Prüfschlüssel sichergestellt. Eine digitale Signatur sichert die Integrität der *Session Key*-Teile zu (Bild 2.5; [DiHe_76, DiOW_92]).

STS erfordert daher entweder ein Zertifikatformat, das neben den öffentlichen Schlüsseln des Signatursystems auch die DH-Parameter enthält, oder separate Zertifikate für die Authentisierung und den Integritätsnachweis der ausgetauschten DH-Schlüsselteile.

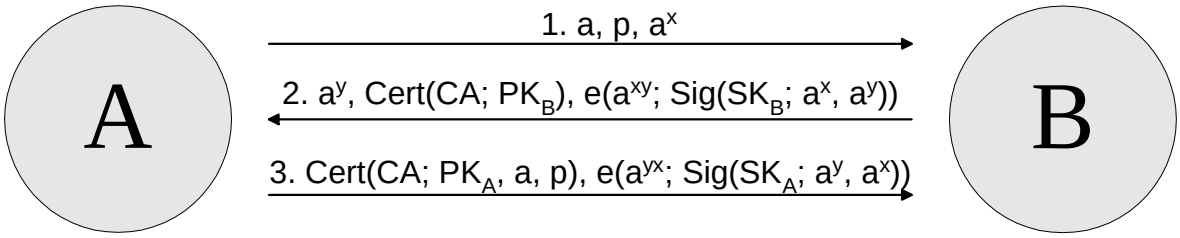

Bild 2.5: STS-Basisprotokoll

3 Neue Zertifikatformate

Die Beispiele in Kapitel 2 zeigen die Unzulänglichkeit des verbreiteten X.509-Zertifikatformats. Eine Spezifikation, die die Verwendung der Zertifikate in unterschiedlichen Authentisierungs- und Schlüsselaustauschprotokollen erlaubt, muß wenigstens die folgenden beiden Eigenschaften besitzen: Es müssen

- Schlüssel für verschiedene Sicherheitsdienste unterschieden und
- unterschiedliche Digitale Signatursysteme und asymmetrische Verschlüsselungsverfahren unterstützt werden (weitgehende „Algorithmenunabhängigkeit“).

Zwei Ansätze bieten sich an:

(1) Spezifikation spezieller Zertifikate für einzelne Sicherheitsprotokolle.

Bei einer solchen Lösung können besondere Eigenschaften eines Sicherheitsprotokolls im Aufbau des Zertifikats berücksichtigt werden. Es ist keine Modifikation existierender Protokolle erforderlich. Allerdings können verschiedene Schlüssel-Gültigkeitszeiträume innerhalb eines Zertifikats nicht unterschieden werden; und Einträge in *Certificate Revocation Lists* (CRLs) beziehen sich immer auf alle im Zertifikat befindlichen Schlüssel.

(2) Erweiterung der Zertifikatformatspezifikation um einen Eintrag *Security Service*.

Ein solcher zusätzlicher Eintrag ermöglicht die Verwendung unterschiedlicher Zertifikate für verschiedene Sicherheitsdienste:

```
Certificate ::= SIGNED SEQUENCE {
      version [0]             Version DEFAULT v1988,
      serialNumber            CertificateSerialNumber,
      signature               AlgorithmIdentifier,
      issuer                  Name,
      validity                Validity,
      securityservice         SecurityServiceIdentifier,
      subject                 Name,
      subjectPublicKeyInfo    SubjectPublicKeyInfo
}
```

Dieser Ansatz ist im Hinblick auf zukünftige Erweiterungen wie die Zertifizierung von öffentlichen Schlüsseln weiterer Sicherheitsdienste (z.B. Nicht-Abstreitbarkeit) offen. Außerdem können zertifizierte Schlüssel in unterschiedlichen Protokollen verwendet und öffentliche Schlüssel einzeln bei unterschiedlichen Zertifizierungsinstanzen zertifiziert werden. Der An-

satz vereinfacht auch den Wechsel einzelner Schlüssel, da davon nur jeweils ein Zertifikat betoffen ist.

Der Speicherbedarf für die Haltung der Zertifikate in Zertifizierungsinstanzen, öffentlichen Zertifikats-Verzeichnissen und bei den Benutzern steigt bei dieser Lösung jedoch, da jeder verwendete öffentliche Schlüssel ein eigenes Zertifikat benötigt, ggf. mit langer Zertifikatskette.

Fazit

Eine Ersetzung des in X.509 spezifizierten und inzwischen von vielen Sicherheitslösungen (wie beispielsweise PEM) übernommene Zertifikatformat ist überfällig. Da es für ein Protokoll entwickelt wurde, das Integrität und Vertraulichkeit mit demselben asymmetrischen Schlüsselsystem sicherstellt, legt es asymmetrische Authentisierungs- und Schlüsselaustauschprotokolle mit X.509-Zertifikaten auf das RSA-Verfahren fest. Es wurde gezeigt, daß dieses Format für Protokolle wie beispielsweise STS ungeeignet ist.

Von den skizzierten Lösungsvarianten erscheint die Erweiterung der Spezifikation um ein Feld *Security Service* am flexibelsten, um auch zukünftigen Protokollanforderungen zu genügen. Dabei sind jedoch kleine Modifikationen existierender Protokollspezifikationen unvermeidlich.

Literatur

BeMe_91 Bellovin, Steven M.; Merrit, Michael: *Limitations of the Kerberos Authentication System*. USENIX, Winter '91, Dallas, 1991, S. 1-15.

CCITT_89 CCITT Recommendation X.509: *The Directory: Authentication Framework*. Genf 1989.

DiHe_76 Diffie, Whitfield; Hellman, Martin E.: *New Directions in Cryptography*. IEEE Transactions on Information Theory, Bd. IT-22, Nr. 6, 1976, S. 644-654.

DiOW_92 Diffie, Whitfield; Oorschot, Paul C. van; Wiener, Michael J.: *Authentication and Authenticated Key Exchange*. Designs, Codes & Cryptography, Nr. 2, 1992, S. 107-125.

ElGa_85 El Gamal, Taher: *A Public Key Cryptosystem an Signature Scheme Based on Discrete Logarithms*. IEEE Trans. on Inform. Theory, Bd. IT-31, Nr. 4, 7/1985, S. 469-472.

FoPf_91 Fox, Dirk; Pfitzmann, Birgit: *Effiziente Software-Implementierung des GMR-Signatursystems*. In: Pfitzmann, A.; Raubold, E. (Hrsg.): Verläßliche Informationssysteme. Proceedings der Fachtagung VIS '91, Informatik Fachberichte Nr. 271, Springer, Heidelberg 1991, S. 329-345.

GoBD_93 Gollmann, Dieter; Beth, Thomas; Damm, Frank*: Authentication services in distributed systems*. Computers & Security, Vol. 12, No. 8, 1993, S. 753-764.

GoMR_88 Goldwasser, Shafi; Micali, Silvio; Rivest, Ronald L.: *A Digital Signature Scheme Secure against Adaptive Chosen Message Attacks*. SIAM Journal on Computing, Bd. 17, Nr. 2, 1988, S. 281-308.

HoPe_94 Horster, Patrick; Petersen, Holger: *Verallgemeinerte ElGamal-Signaturen*. In: Bauknecht, K., Teufel, S. (Hrsg.): Sicherheit in Informationssystemen. Proceedings der Fachtagung SIS '94, vdf-Verlag, Zürich 1994, S. 89-106.

ISO_92 International Organisation for Standardization (ISO): *Entity authentication mechanisms - Part 3: Entity authentication using a public-key algorithm*. International Standard ISO 9798-3, Genf 1992.

Kali_91 Kaliski, Burton S.: *An Overview of the PKCS Standards*. PKP Inc., 1991.

Kent_93 Kent, Stephen T.: *Privacy Enhancement for Internet Electronic Mail. Part II: Certificate-Based Key Management*. Request for Comments (RFC) 1422, Februar 1993.

NeSc_78 Needham, Roger M.; Schroeder, Michael D.: *Using Encryption for Authentication in Large Networks of Computers*. Communications of the ACM, Bd. 21, Nr. 12, Dezember 1978, S. 993-999.

NeTs_94 Neumann, Clifford B.; Ts'o, Theodore: *Kerberos: An Authentication Service for Computer Networks*. IEEE Communications Magazine, Sept. 1994, S. 33-38.

NIST_94 National Institute of Standards and Technology (NIST): *Digital Signature Standard (DSS)*. Federal Information Processing Standards Publication 186 (FIPS-PUB), 19. Mai 1994.

PfWa_91 Pfitzmann, Birgit; Waidner, Michael: *Fail-Stop Signaturen und ihre Anwendung*. In: Pfitzmann, A.; Raubold, E. (Hrsg.): Verläßliche Informationssysteme. Proceedings der Fachtagung VIS '91, Informatik Fachberichte Nr. 271, Springer, Heidelberg 1991, S. 289-301.

RSA_78 Rivest, Ronald L.; Shamir, Adi; Adleman, Leonard: *A Method for obtaining Digital Signatures and Public Key Cryptosystems*. Communications of the ACM, Bd. 21, Nr. 2, 1978, S. 120-126.

Rula_93 Ruland, Christoph: *Informationssicherheit in Datennetzen*. DataCom-Verlag, Bergheim 1993.

StNS_88 Steiner, Jennifer G.; Neumann, Clifford B.; Schiller, Jeffrey I.: *Kerberos: An Authentication Service for Open Network Systems*. USENIX, Winter '88, Dallas, 1/1988, S. 1-15.

WoLa_92 Woo, Thomas Y.C.; Lam, Simon S.: *Authentication for Distributed Systems*. IEEE Computer, January 1992, S. 39-52.