E-Mail-Sicherheitslösungen

Secorvo White Paper

Gegenüberstellung von E-Mail-Sicherheitslösungen: Kriterien, Standards und Produkte

Version 1.3
Stand 04. November 2000

Dirk Fox

Secorvo Security Consulting GmbH
Albert-Nestler-Straße 9
D-76131 Karlsruhe

Tel. +49 721 6105-452
Fax +49 721 6105-455

E-Mail info@secorvo.de
Internet http://www.secorvo.de

Inhaltsübersicht

Abkürzungen

ASN.1	Abstract Syntax Notation One
CA	Certification Authority
CCITT	Comité Consultatif International Télégraphique et Téléphonique
CRL	Certificate Revocation List
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
IPsec	Internet Protocol: Security Functionality
ISO	International Organization for Standardization
ITSEC	Information Technology Security Evaluation Criteria
ITU	International Telecommunication Union
LDAP	Lightweight Directory Access Protocol
MHS	Message Handling System
MOSS	MIME Object Security Services
MTT	MailTrusT
OCSP	Online Certificate Status Protocol
PEM	Privacy Enhancement for Internet Mail
PGP	Pretty Good Privacy
PIN	Personal Identification Number
PKCS	Public Key Cryptography Standard(s)
PKI	Public Key Infrastructure
PKIX	Internet X.509 Public Key Infrastructure
POP3	Post Office Protocol
PSE	Personal Security Environment
RFC	Request for Comments
RSA	Rivest, Shamir, Adleman Kryptosystem
SHA	Secure Hash Algorithm
S/MIME	Secure/Multipurpose Internet Mail Extensions
SET	Secure Electronic Transaction Protocol
SMTP	Simple Mail Transfer Protocol
SSH	Secure Shell (Protocol)
SSL	Secure Sockets Layer Protocol

1 Zusammenfassung

Das vorliegende White Paper gibt eine Übersicht über die für die Auswahl einer E-Mail-Sicherheitslösung wesentlichen Standards und Bewertungskriterien.

2 Anforderungen an E-Mail-Sicherheitsprodukte

Vor einer Bewertung existierender E-Mail-Sicherheitslösungen und der Auswahl eines geeigneten Produkts sollten die Anforderungen an eine solche Lösung genau spezifiziert werden.

Grundsätzlich sind dabei neben spezifischen Anforderungen des Unternehmens insbesondere (mit im Einzelfall unterschiedlicher Gewichtung) die *Sicherheit, Interoperabilität*, *Bedienungsfreundlichkeit* und *Verfügbarkeit* von Lösungen und Produkten (für unterschiedliche Betriebssysteme und E-Mail-Clients) zu berücksichtigen.

2.1 Sicherheit

Heutige E-Mail-Sicherheitslösungen verwenden inzwischen ausschließlich

- hybride Verschlüsselungsverfahren zum Schutz der Nachrichten während der Übertragung vor unberechtigter Kenntnisnahme (*Vertraulichkeit*) und
- digitale Signaturen zum Schutz vor Veränderung oder Fälschung (*Integrität* und *Authentizität*).

Die hybride Verschlüsselung verwendet symmetrische Kryptoverfahren zur Verschlüsselung der Nachricht; der dabei verwendete, zuvor zufällig gewählte geheime Schlüssel (message key) wird anschließend mit dem (asymmetrischen) öffentlichen Schlüssel des Empfängers verschlüsselt und der Nachricht angefügt. Der Empfänger kann mit seinem passenden geheimen Entschlüsselungsschlüssel den symmetrischen message key zurückgewinnen und mit diesem wiederum die Nachricht entschlüsseln.

Die Erzeugung digitaler Signaturen erfolgt mit einem geheimen Signierschlüssel des Senders, dessen zugehöriger öffentlicher Prüfschlüssel dem Empfänger die Prüfung der Signatur ermöglicht.

Wichtig ist, daß das Sicherheitsniveau einer E-Mail-Sicherheitslösung hoch genug ist, um die mit der Einführung der benötigten Produkte verbundenen Kosten und Umstellungen zu rechtfertigen. Das Sicherheitsniveau wird dabei von unterschiedlichen Faktoren bestimmt, auf die im folgenden näher eingegangen wird.

2.1.1 Schutz der geheimen Schlüssel

Der oder die geheimen Schlüssel des Benutzers (zur Erzeugung digitaler Signaturen beziehungsweise zum Entschlüsseln von Nachrichten) müssen vor unberechtigtem Zugriff geschützt werden. Dazu werden die Schlüssel üblicherweise in einer sogenannten PSE (Personal Security Environment) gespeichert, die entweder in einer verschlüsselten Datei oder auf einer PIN-geschützten Chipkarte abgelegt wird.

Idealerweise sollten die geheimen Schlüssel eine Passphrase-geschützte Smartcard niemals verlassen, sondern dort direkt zur Verschlüsselung verwendet werden. Die Passphrase sollte so gewählt sein, daß sie auch nicht durch systematische Rateangriffe in vernünftiger Zeit bestimmt werden kann.

2.1.2 Sicherheit des Zertifizierungssystems

Die Lösung sollte mit einem Zertifizierungssystem – in der Regel einer Public Key Infrastruktur (PKI) – interoperieren, in dem die Vorgänge der Generierung und Zertifizierung von Schlüsseln einem definierten Sicherheitsniveau genügen und kontrolliert werden können:

- Der Zertifizierung eines Schlüssels muß eine verläßliche Identitätsprüfung des Schlüsselinhabers vorausgehen. Der Gültigkeitszeitraum eines Zertifikats muß von der zertifizierenden Stelle beschränkt werden.
- Bei PIN-Verlust, Verdacht auf Kompromittierung oder Schlüsseldefekt muß ein Rückruf möglich sein, der umgehend verbreitet wird. Dies läßt sich entweder durch ein Online-Prüfverfahren (z.B. OCSP) oder durch regelmäßig publizierte Sperrlisten (CRLs) realisieren.

2.1.3 Fehlerfreiheit der Implementierung

Die Implementierung der Sicherheitsfunktionen (nicht nur der Ver- und Entschlüsselung, sondern vor allem des Schlüsselmanagements: Von wo werden welche Zertifikate importiert? Woher kommt der Root-Schlüssel? Wie oft werden CRLs geladen? Wie werden sie überprüft? ...) muß fehlerfrei sein. Häufig sind die Verifikationsvorgänge für digitale Signaturen und Zertifikate in Produkten ungenügend oder fehlerhaft: So werden beispielsweise oft wichtige Zertifikatsattribute nicht oder falsch ausgewertet, die Namensgebung in Zertifikatsketten nicht überprüft oder es erfolgt keine Prüfung der gesamten Zertifikatskette respektive des zugehörigen Wurzel-Zertifikats (root key).

Die Fehlerfreiheit der Implementierung sollte idealerweise entweder durch eine Evaluation nachgewiesen oder durch externe Dritte überprüpft worden sein (z.B. ITSEC-Zertifizierung, unabhängiges Gutachten).

2.2 Interoperabilität

Die schnelle und weite Verbreitung der E-Mail-Kommunikation in den letzten Jahren ist vor allem der Tatsache zu verdanken, daß sich die Internet-Protokolle als einheitlicher Kommunikationsstandard für elektronische Nachrichten allgemein durchgesetzt haben. Wie beim Telefon auch kann heute jeder Sender einer elektronischen Nachricht davon ausgehen, daß ein Empfänger, der im Besitz einer Internet-E-Mail-Adresse ist, über das Internet per E-Mail erreicht werden kann.

Alternative Protokollstandards wie X.400 konnten sich wegen der Einfachheit und schnellen Verbreitung der Internet-Protokolle (SMTP, POP3) nicht durchsetzen; existierende X.400-Netze werden daher heute über X.400-SMTP-Gateways an die Internet-E-Mail-Welt angebunden. Proprietäre E-Mail-Protokolle wie das Messaging von Lotus Notes oder MHS von NetWare, die ursprünglich für eine reine Inhouse-Kommunikation entwickelt wurden, wurden inzwischen durch SMTP-kompatible Protokolle ersetzt oder ergänzt.

Auch eine E-Mail-Sicherheitslösung ist nur dann über eine reine Inhouse-Kommunikation hinaus von Nutzen, wenn sie im Business-to-Business-Bereich und für „elektronische Kundenbeziehungen“ eingesetzt werden kann. Dies setzt die Verwendung allgemeiner Standards voraus, die nicht durch proprietäre Lösungen eingeschränkt werden darf.

Eine E-Mail-Sicherheitslösung muß dabei in drei wesentlichen Punkten mit anderen E-Mail-Sicherheitslösungen interoperieren können:

- **Zertifikatsformat**: Die Formate der Zertifikate selbst müssen insoweit vereinheitlicht sein, daß jede ein Zertifikat ausstellende Stelle dieselben Angaben im Zertifikat bestätigt. Zu diesen Angaben gehören mindestens
  - ein eindeutiger Name,
  - der öffentliche Schlüssel und
  - der Gültigkeitszeitraum des Zertifikats.

  Weiter sind Angaben (Attribute) zur Nutzung des Schlüssels und für weitere Einschränkungen sinnvoll, z.B. hinsichtlich einer maximalen Zertifizierungspfadlänge. Damit Anwendungen nicht viele verschiedene, heterogene Formate unterstützen müssen, ist es wünschenswert, daß ein allgemein einheitliches Format verwendet wird.
- **Zertifikats-Austauschformat**: Zertifikate von Kommunikationspartnern müssen vom E-Mail-Client in einem lokalen Schlüsselverzeichnis verwaltet werden. Dazu ist es erforderlich, diese Zertifikate zu importieren. Übliche Wege dazu sind E-Mail, Diskette oder Zugriff auf einen Verzeichnisdienst. Die formatierte Darstellung von Zertifikaten beim Austausch muß festgelegt sein, damit unterschiedliche E-Mail-Clients Zertifikate empfangen und in ihr lokales Schlüsselverzeichnis importieren können.
- **Nachrichten-Austauschformate**: Der Austausch von verschlüsselten oder digital signierten E-Mail-Nachrichten erfolgt in einem speziellen Nachrichtenformat, das von dem Programm des Empfängers ausgewertet werden muß, um die Nachricht entschlüsseln beziehungsweise eine Signatur prüfen zu können.

Derzeit existieren neben proprietären Lösungen mehrere konkurrierende Standards für diese Formate. Bei der Entscheidung für ein E-Mail-Sicherheitsprodukt ist zu berücksichtigen, daß eine sichere E-Mail-Kommunikation mit Produkten, die die gewählten Formate nicht unterstützen, prinzipiell nicht möglich ist. Daher sollte hinsichtlich der Interoperabilität auch die *Verbreitung* (und tatsächlicher *Nutzung*) von Produkten, die den jeweiligen Standard unterstützen, berücksichtigt werden.

2.3 Bedienungsfreundlichkeit

Für die Akzeptanz einer Sicherheitslösung, insbesondere, wenn sie eine Erweiterung der Funktionalität eines existierenden Produkts betrifft, ist ausschlaggebend, daß die Lösung (unabhängig von der dadurch gewonnenen zusätzlichen Funktionalität) bedienungsfreundlich realisiert ist.

Die Bedienungsfreundlichkeit ist weitgehend unabhängig von den vom Produkt unterstützten Standards. Sie hängt größtenteils von der jeweiligen Qualität der Implementierung ab.

2.3.1 Transparente Integration

Die Integration erfolgt in der Praxis in der Regel durch ein „Plug-In“, das die Funktionalität und die Menus eines existierenden E-Mail-Systems um die Sicherheitsfunktionen erweitert.

Eine E-Mail-Sicherheitslösung sollte möglichst weitgehend in die Mail-Anwendung integriert sein. Verschlüsselung und Prüfung einer digitalen Signatur sollten weitgehend transparent erfolgen; lediglich das Prüfergebnis sollte dem Benutzer in geeigneter Weise bekanntgegeben werden. Dies sollte durch eine einfache und klare Visualisierung erfolgen (beispielsweise durch einen roten beziehungsweise grünen Punkt in der Nachrichtenübericht o.ä.).

Die Lösung sollte so konfiguriert werden können, daß bei jeder Entschlüsselung und Erzeugung digitaler Signaturen die Eingabe einer „Passphrase“ oder PIN verlangt wird.[1] Dabei sollte auch die Möglichkeit einer temporären Zwischenspeicherung des Entschlüsselungsschlüssels für eine einstellbare Zeitspanne gegeben sein. Die Auswahl dieser Sicherheitsfunktionen sollte als „default“ voreinstellbar sein, um bei der Einführung der Lösung eine unternehmensweit einheitliche Sicherheits-Policy ohne individuelle und manuelle Einzelkonfiguration durchsetzen zu können.

2.3.2 Unkomplizierte Verwaltung

Die Verwaltung der Zertifikate muß für einen Benutzer einfach und unkompliziert zu handhaben sein, insbesondere, da ein manueller Zugriff des Benutzers auf die Zertifikate bei einer guten Integration der Sicherheitsdienste nur selten erforderlich sein sollte.

Daher sollte ein E-Mail-Sicherheitsprodukt den folgenden Anforderungen genügen:

- weitgehend automatische Zuordnung von empfangenen Zertifikaten zum Zertifikatsinhaber (über E-Mail-Adresse) im lokalen Adreßbuch,
- automatische Auswertung von CRLs im lokalen Schlüsselverzeichnis (durch klare Markierung gesperrter Zertifikate; keine Löschung, da Zertifikate möglicherweise später noch zur Signaturprüfung benötigt werden),
- einfache, möglichst automatisierte Suche und Import neuer Zertifikate von Verzeichnisdiensten oder Schlüssel-Servern.

2.3.3 Verständliche Meldungen

Beim Auftreten von sicherheitskritischen Ereignissen wie dem Fehlschlagen einer Signaturprüfung oder dem Scheitern einer Entschlüsselung werden Meldungen benötigt, die dem jeweiligen Benutzer eine klare Handlungsanweisung geben und den Vorfall verständlich machen.

Um Fehlerfälle nachvollziehen zu können, sollte die Möglichkeit bestehen, alle sicherheitsrelevanten Vorgänge zu protokollieren. Auch sollten alle Angaben eines Zertifikats angezeigt werden können.

2.3.4 Konfigurationsmöglichkeiten

Abhängig vom Anwendungskontext und den konkreten Anforderungen an die E-Mail-Sicherheitslösung ergeben sich in der Praxis unterschiedliche Idealkonfigurationen. So kann es sein, daß

- die Möglichkeit zur Zuordnung von „Direct Trust“ zu einem Zertifikat für bestimmte Benutzer oder Benutzergruppen gesperrt werden soll;
- für bestimmte Adreß-Domänen nicht die gesamte Zertifikatskette, sondern nur das Zertifikat selbst mit der Nachricht verschickt werden soll;
- bei häufigem Zugriff auf archivierte Nachrichten die Durchführung der Signaturprüfung auf eine Einmalprüfung gleich bei Empfang der Nachricht beschränkt werden soll;

---

[1] Das ist für Anwendungen mit hohem Sicherheitsbedarf unverzichtbar, da bei einer Zwischenspeicherung des geheimen Schlüssels dieser unberechtigtem Zugriff ausgesetzt sein kann.

- nur (oder bevorzugt) bestimmte kryptographische Verfahren verwendet werden sollen.

Dies setzt entsprechende Konfigurationsmöglichkeiten voraus.

2.4 Verfügbarkeit

Eine E-Mail-Sicherheitslösung muß angesichts der existierenden heterogenen Systeme für eine Vielzahl unterschiedlicher E-Mail-Clients und Betriebssysteme erhältlich sein. Nicht notwendig muß das jeweilige Produkt identisch realisiert sein, wohl aber müssen Implementierungen für die Betriebssysteme und E-Mail-Anwendungen existieren, die in dem jeweiligen Unternehmen auch tatsächlich eingesetzt werden. Denn der Wechsel eines eingeführten E-Mail-Systems, nur um eine gewünschte Sicherheitslösung einführen zu können, verursacht erhebliche Kosten, führt zu Widerständen der Nutzer und ist daher in größeren Unternehmen praktisch nicht umsetzbar.

Als typisches Anforderungsprofil kann dabei gelten:

- Zu unterstützende Betriebssysteme:
  - MS Windows 3.1, 95/98, NT (Versionen 3.51, 4.x, 5.x), 2000
  - ggf. Unix, spezielle Derivate (z.B. AIX, SUN/OS, HP-UX, Linux, ...)

Die Betriebssysteme DOS, OS/2 und das Macintosh-Betriebssystem können in Einzelfällen ebenfalls noch eine wichtige Rolle spielen.

- Zu unterstützende E-Mail-Anwendungen (große Verbreitung):
  - MS Exchange/Outlook
  - Netscape Mail
  - Lotus Notes/cc:Mail
  - Groupwise (Novell)
  - Eudora, Pegasus

Auch diese Liste muß im Einzelfall häufig um spezielle E-Mail-Systeme erweitert werden. Für einzelne dieser E-Mail-Clients ist auch die jeweilige Versionsnummer relevant: Oft sind für bestimmte Programmversionen gänzlich unterschiedliche Implementierungen eines Sicherheits-Plugins erforderlich.

3 Existierende Standards

Eine zentrale Rolle für die Interoperabilität einer E-Mail-Sicherheitslösung spielen die vom Produkt unterstützten oder verwendeten Standards.

Bei der Betrachtung der im Zusammenhang mit einer E-Mail-Sicherheitslösung relevanten Standards lassen sich zwei Gruppen von Standards unterscheiden: *PKI-bezogene Standards*, die insbesondere Interoperabilitätszwecken innerhalb einer PKI dienen und so nicht auf die Anwendung „sichere E-Mail“ beschränkt sind, sowie spezielle *Standards für E-Mail-Sicherheit*. Beide Gruppen sind im Hinblick auf ihre Verbreitung und technische Eignung zu berücksichtigen.

3.1 Relevante PKI-Standards

Als PKI-Standards werden hier Normen verstanden, die Formate oder Protokolle, die innerhalb einer Public Key Infrastruktur erforderlich sind, spezifizieren.

3.1.1 Zertifikats-Standards

Für den Aufbau und die Kodierung von Zertifikaten wurden unabhängig voneinander mehrere (weitgehend inkompatible) Standards verabschiedet.

- **X.509**: Der von der ITU (früher CCITT) schon 1989 in einer ersten Version genormte und später auch von ISO/IEC übernommene Zertifikatsformat-Standard X.509 (ISO/IEC 9594-8) umfaßt in der aktuellen Version 3 von 1997 (kurz X.509 (97) oder X.509v3) eine Vielzahl von möglichen Attributen, die auch durch die Definition eigener Extensionen ergänzt werden können [ITU_97]. Das Format ist sehr flexibel und wurde wiederholt an Praxiserfordernisse angepaßt. Es hat inzwischen eine sehr große Verbreitung und wird z.B. von den Protokollspezifikationen SET, SSL, SSH, PKCS, PKIX, IPsec, PEM, S/MIME und MTT verwendet.
- **Edifact**: Im Edifact-Standard wurde ein eigenes Zertifikatsformat spezifiziert, das insbesondere in Banken eine wichtige Rolle spielt. Es hat den Vorzug, anders als die ASN.1-Kodierung von X.509-Zertifikaten eine automatische Verarbeitung stark zu erleichtern. Durch die „fest-verdrahtete“ Spezifikation der Feldbelegungen sind im Unterschied zu X.509 Erweiterungen allerdings nur sehr eingeschränkt möglich. Auch der Anwendungsbereich ist durch eine Beschränkung auf Edifact-Nachrichten (-formate) stark eingeschränkt.
- **PGP**: PGP verwendet ein eigenes, sehr spezielles Zertifikatsformat, das in der Struktur ein wenig an Edifact-Zertifikate erinnert. Es ist ebenfalls unflexibel, daher führt jede Erweiterung des Formats zu einem Verlust der Abwärtskompatibilität. Inzwischen wurde das Zertifikatsformat mit neueren PGP-Versionen mehrfach geändert und erweitert; das Format der PGP-Version 5.x wurde 1998 als „OpenPGP“ spezifiziert und als Internet-Standard vorgeschlagen (s. Abschnitt 3.1.2). Seit der Programm-Version 6.5.x können sogar X.509-Zertifikate genutzt werden. Die Formaterweiterungen haben zur Konsequenz, daß die verschiedenen Versionen nicht abwärtskompatibel zueinander sind.

3.1.2 Zertifikatsaustauschformat

Für den Import, Export und Austausch von Zertifikaten wurden die folgenden Zertifikatsformate vereinheitlicht:

- **PGP**: PGP hat drei verschiedene Versionen von Austauschcodierungen spezifiziert, die in RFC 1991 publiziert wurden [AtSZ_96]. In RFC 2015 wurde mit PGP/MIME eine an MOSS angelehnte Multipart-Message-Erweiterung für PGP spezifiziert [Elki_96]. Schließlich wurde mit RFC 2440 OpenPGP auf der Basis von PGP 5.x spezifiziert [CDFT_98].
- **PKCS#7**: Sehr verbreitetes, von RSA spezifiziertes Format für die Speicherung und den Austausch von X.509-Zertifikaten; inzwischen auch als RFC 2315 publiziert.
- **PEM/MTT**: Das Austauschformat von PEM ist durch die MailTrusT-Spezifikation (MTTv1 [Baus_96] und MTTv2 [BiBF_99]) und das Projekt „Sphinx“ [SBMB+_99] in Deutschland verbreitet.

3.1.3 Verzeichniszugriffsprotokoll

Bei den Protokollen, die für den Zugriff auf den Verzeichnisdienst eingesetzt werden, sind neben einem proprietären Zugriffsprotokoll von PGP auf spezielle PGP-Key-Server vor allem zwei standardisierte Protokolle zu nennen:

- **LDAP**: Zugriffsprotokoll der IETF für Verzeichnisdienste, sowohl zu LDAP-Servern als auch auf X.500-Verzeichnisse. Die aktuelle Version ist LDAPv3 (RFC 2253), viele Produkte unterstützen bislang allerdings nur die in PKIX spezifizierte Nutzung der Version 2 (RFC 2559, 2587).
- **OCSP**: Protokoll für Online-Statusabfragen von Zertifikatsgültigkeiten (RFC 2560).

Eine weitere proprietäre (und funktional erheblich eingeschränkte) Implementierung stammt von der Deutschen Telekom („TTP Viewer“); sie wurde im Zusammenhang mit der Einrichtung einer Zertifizierungsstelle nach Signaturgesetz eingeführt.

3.1.4 Zertifikatsperrlisten

Für die Verteilung von Informationen über die Sperrung von Zertifikaten gibt es nur ein standardisiertes Format:

- **CRL**: Certificate Revocation Lists nach X.509 gibt es in zwei Versionen: Version 1, die in X.509 (89) spezifiziert ist, und die unter anderem um CRL Distribution Points erweiterte Version 2, die in X.509 (97) enthalten ist.

Dieses CRL-Format wird sowohl von PKIX als auch in PEM, MTT und S/MIME verwendet. Lösungen, die ausschließlich mit einem Online-Prüfprotokoll (z.B. OCSP) arbeiten, können ohne CRLs auskommen.

3.1.5 Zertifizierungsprotokolle

Für die Anforderung der Zertifizierung eines vorliegenden öffentlichen Schlüssels gibt es zwei verschiedene Ansätze (neben weiteren proprietären Protokollen):

- **PKCS#10**: Standard eines Certification Request-Formats, das sowohl via E-Mail als auch als Datei von Diskette geladen und transportiert werden kann. Publiziert als Internet-RFC (RFC 2314) und in vielen Produkten im Einsatz.

- **PKIX**: Die Spezifikation der (Internet-) PKIX-Protokolle ist derzeit noch in Arbeit; das Certificate Management Protocol und das Certificate Request Message Format wurden im März 1999 als RFC publiziert (RFC 2510-2511). Erste Implementierungen, die Vorversionen des Standards umsetzen, sind bereits erhältlich.

Einige Hersteller setzen hier noch auf proprietäre Protokolle und warten wahrscheinlich den Abschluß der Standardisierung ab.

3.2 E-Mail-Sicherheitsstandards

E-Mail-Sicherheitsstandards werden seit Ende der 80er Jahre entwickelt. Weiterentwicklungen von PEM, dem ersten Standard, wie MOSS (RFC 1847 und 1848) spielen heute praktisch keine Rolle. Auch PEM ist heute nur noch in Gestalt von MTT von Bedeutung. Für alle im folgenden kurz vorgestellten Spezifikationen gilt, daß sie weder ein internationaler (ISO/IEC-) noch ein IETF-Standard sind. Sie sind aber de-facto-Standards insoweit, als sie in Produkten mit großer Verbreitung umgesetzt worden sind und einheitliche Spezifikationen existieren.

Diese „Standards“ legen Interoperabilitätsanforderungen für den Nachrichtenaustausch und das Schlüsselmanagement fest.

3.2.1 PEM

Die PEM-Spezifikation wurde in einer ersten Version Ende der 80er Jahre als Internet-RFC publiziert (RFC 1113-1115) und in einer stark überarbeiteten und erweiterten Fassung 1993 (RFC 1421-1424). PEM nutzt das Zertifikatsformat von ITU (X.509 (89)) [HoPo_94].

3.2.2 MailTrusT v1 (PEM)

Der MailTrusT-Standard entspricht in wesentlichen Teilen der PEM-Spezifikation. Er wurde 1996 im Auftrag von TeleTrusT Deutschland e.V. spezifiziert [Baus_96]. Einige Unzulänglichkeiten von PEM werden darin beseitigt, ohne die Interoperabilität zu beeinträchtigen. Insbesondere wurden die Mindestanforderungen an die Schlüssellängen deutlich höher als bei PEM angesetzt und einige spezielle, für praktische Anwendungen wichtige Dokumentenformate aufgenommen.

PEM/MTTv1	**Standard**
Zertifikatsformat	X.509 (89)
Schlüsseltrennung	nicht unterstützt
Austauschformate	PEM- und MTT-spezifische Formate
Rückruflisten	CRLv1 (nach X.509 (89))
Zertifizierungsanfrage	PEM-spezifisch
Verzeichnisabfrage	nicht spezifiziert

Der MTT-Standard wird in Deutschland inzwischen von mehr als 50 Produkten unterstützt.

3.2.3 OpenPGP

Mit dem Erfolg des Programms Pretty Good Privacy (PGP) entstand das Interesse an einer Standardisierung von PGP als offenem Protokoll-Standard. Die Spezifikation von OpenPGP ist als RFC publiziert (RFC 2440); die Kodierung und Formate von Zertifikaten und PGP-Nachrichten sind in RFC 1991 und 2015 spezifiziert. Alle drei Spezifikationen haben den IETF-Status eines Proposed Standard.

OpenPGP	**Standard**
Zertifikatsformat	RFC 2440 (Proposed Standard)
Schlüsseltrennung	ab v6.0 („subkeys“)
Austauschformate	PGP/MIME (RFC 2015, RFC 1991), OpenPGP (Proposed Standard)
Rückruflisten	nicht unterstützt
Zertifizierungsanfrage	nicht spezifiziert
Verzeichnisabfrage	proprietär, ab v6.0 auch LDAP

Mit dem Einsatz einer PGP-Lösung legt man sich auch auf ein proprietäres Protokoll für die PKI-Funktionen (Zertifizierung) und damit auf den Hersteller fest. Anders als bei PEM/MTT und S/MIME ist es derzeit nicht möglich, zwischen verschiedenen Anbietern und PKI-Produkten auszuwählen, sondern ist man auf den Certificate Server der Firma Network Associates festgelegt.

3.2.4 MailTrusT v2

Um die Spezifikation der internationalen Standardisierung anzunähern und sie um volle PKI-Funktionalität zu erweitern, wurde die MailTrusT-Spezifikation zu einer Version 2 weiterentwickelt [Bies_99, BiBF_99]. Diese Weiterentwicklung schließt auch die Unterstützung von S/MIME ein und orientiert sich in den PKI-bezogenen Protokollen an PKIX.

MTTv2	**Standard**
Zertifikatsformat	X.509 (97)
Schlüsseltrennung	unterstützt
Austauschformate	S/MIME, PEM- und MTT-spezifische Formate
Rückruflisten	CRLv2 (nach X.509 (97))
Zertifizierungsanfrage	PKCS#10, PKIX
Verzeichnisabfrage	LDAPv2/3

Die Version 2 des MTT-Standards wurde im März 1999 abgeschlossen und wird daher bisher nur von sehr wenigen Produkten unterstützt. Allerdings spielt sie beispielsweise im Projekt „Sphinx“ (Sicherer Dokumentenaustausch in der Bundesverwaltung) eine zentrale Rolle.

3.2.5 S/MIME

Die IETF veröffentlichte im März 1998 mit den RFCs 2311 und 2312 die Spezifikation „Secure/Multipurpose Internet Mail Extensions“ in der Version 2. Diese Standardisierungsaktivität wurde von einer großen Zahl von Herstellern, allen voran der Firma RSA Data Security Inc. vorangetrieben. Die Spezifikation der Austauschformate ist auch Teil von MTTv2.

Eine überarbeitete und erweiterte Fassung, S/MIME Version 3, wurde Mitte 1999 in den RFCs 2630-2634 spezifiziert; sie ist als IETF-Standard vorgeschlagen (Proposed Standard).

S/MIME	**Standard**
Zertifikatsformat	X.509 (97); PKCS#7/12 als Austauschformat
Schlüsseltrennung	möglich (keyUsage)
Austauschformate	S/MIMEv2 (RFC 2311, 2312) resp. S/MIMEv3 (RFC 2630, 2633)
Rückruflisten	CRLv2 (X.509 (97))
Zertifizierungsanfrage	PKCS#10
Verzeichnisabfrage	nicht spezifiziert, in der Regel LDAPv2 (v3)

S/MIME wird heute bereits von einer großen Anzahl von Herstellern unterstützt, darunter Microsoft (Exchange, Outlook, Internet Explorer), Netscape (Messenger), Lotus (Notes) und Novell (Groupwise).

4 Sicherheitsbetrachtung

Die Sicherheit der unterschiedlichen Standards (respektive der diesen entsprechenden Produkte) kann anhand der eingesetzten Kryptoverfahren (und Schlüssellängen), der Speicherung des geheimen Schlüssels (Smartcard) und dem Zertifizierungsprozeß bewertet werden.

4.1 Verwendete Kryptoverfahren

In den zu den Standards gehörigen Produkten kommen unterschiedliche kryptographische Verfahren zum Einsatz. Die folgende Tabelle gibt eine Übersicht der jeweils geforderten Algorithmen (in Klammern die Schlüssellänge in bit).

	PEM/MTTv1	**OpenPGP**	**MTTv2**	**S/MIMEv2/v3**
sym. Verschl.	DES (56), Triple-DES (112)	CAST, IDEA (128), Triple-DES (168)	DES (56), Triple-DES (112)	RC2 (v2: 40-128, v3: mind. 128), Triple-DES (v2: 112, v3: 168), v2: DES (56)
Hash-funktion	MD2/5, SHA-1	MD2/5, SHA-1, RIPEMD-160	MD2/5, SHA-1, RIPEMD-160	SHA-1, v2: MD2, MD5
asym. Verschl.	RSA (508-2048)	ElGamal/DH, RSA (PGP: bis 2048)	RSA (508-2048)	RSA (512-1024)
Signatur-system	RSA (508-2048)	DSA, RSA (PGP: bis 2048)	RSA (508-2048)	RSA (512-1024), v3: DSA (512-1024)
Smart-card	möglich, verfügbar	angekündigt für PGP v7	möglich, verfügbar	möglich, verfügbar

In einem von internationalen Kryptologen 1996 publizierten White Paper zur Empfehlung geeigneter Schlüssellängen für symmetrische Kryptoverfahren wird geraten, mindestens 75, besser 90 bit lange Schlüssel und publizierte, gut untersuchte Verfahren einzusetzen [BDRS+_96].

In einer Untersuchung von Lenstra und Verheul vom Dezember 1999 wird für symmetrische Schlüssel, die im Jahr 2020 noch dieselbe Sicherheit vor Brute-Force-Attacken bieten sollen, wie 56 bit im Jahr 1986, eine Schlüssellänge von mindestens 86 bit empfohlen. Für RSA-Schlüssel, die im Jahr 2020 eine dem Sicherheitsniveau von 417 bit in 1982 entsprechenden Schutz vor Kryptoanalysen bieten sollen, werden mindestens 1880 bit Schlüssellänge empfohlen [LeVe_99].

Als Hashfunktionen gelten heute nur noch die Verfahren RIPEMD-160 und SHA-1 als geeignet [Dobb_97]; daher werden MD2 und MD5 in MTTv2 auch nur noch aus Kompatibilitätsgründen unterstützt.

4.2 Schlüsseltrennung

Zertifikate nach X.509 (97) erlauben die Angabe eines Attributs, das die Schlüsselnutzung festlegt (keyUsage). Damit ist es möglich, zwischen Schlüsseln für die Verschlüsselung und solchen zur Prüfung digitaler Signaturen zu unterscheiden und so Angriffen wie den Denning/Moore-Attacken auf RSA vorzubeugen [Denn_84].

Die S/MIME-Spezifikation erlaubt die Trennung von Signier- und Verschlüsselungsschlüsseln durch das keyUsage-Attribut. Allerdings gibt es derzeit nur wenige Produkte, die die Schlüsseltrennung praktisch unterstützen.

Die Spezifikationen PEM und MTTv1 enthalten keine Schlüsseltrennung, da sie X.509 (89)-Zertifikate verwenden, die diese Erweiterungsattibute noch nicht enthalten. Anders bei MTTv2: Hier werden X.509 (97)-Zertifikate verwendet, die eine Schlüsseltrennung ermöglichen.

PGP besitzt ursprünglich keinen Mechanismus zur Schlüsseltrennung. Erst mit Version 6.0 wurde die Möglichkeit von „subkeys“ eingeführt, die es erlaubt, weitere eigene Schlüssel für unterschiedliche Zwecke zu erzeugen. Die Verwendung dieser Erweiterung ist allerdings auf die Programmversionen 6.x beschränkt. Die Erweiterung ist zudem komplett proprietär, d.h. entspricht nicht dem X.509-Standard. Weder kann PGP X.509-Zertifikate nutzen, noch können PGP-Zertifikate von Produkten verwendet werden, die mit X.509-Zertifikaten arbeiten. Seit der Version 6.5.x unterstützt PGP in gewissem Umfang auch X.509-Zertifikate.

4.3 Zertifizierungsverfahren

Die Ausstellung von Zertifikaten zu öffentlichen Schlüsseln erfolgt durch Erzeugung einer digital signierten Bestätigung, daß ein Schlüssel zu einer Identität gehört. Bei allen X.509-basierten Verfahren (PEM, MTTv1/v2 und S/MIME) liegt die Zuständigkeit für die Zertifikatsausstellung bei einer dritten, unabhängigen Instanz (Certification Authority, CA). Diese legt die Gültigkeitsdauer eines Zertifikats fest. Wo der Schlüssel erzeugt wurde, ob vom Schlüsselinhaber selbst oder durch die CA, ist nicht festgelegt.

PGP geht mit dem Konzept des „Web of Trust“ einen anderen Weg: Ausgangspunkt ist bei PGP der „selbst-signierte“ öffentliche Schlüssel eines Schlüsselinhabers. Er enthält den Gültigkeitszeitraum des *Schlüssels*. Digitale Signaturen weiterer Personen können die Zusammengehörigkeit von Identität und Schlüssel bestätigen. Bis zur Version 6.x sind diese Schlüsselsignaturen aber nicht in ihrer Gültigkeitsdauer beschränkt.

Auch erst seit Version 5.x von PGP gibt es die Möglichkeit, Unterschriften unter einem signierten Schlüssel zurückzuziehen. Ein Zertifikatsrückruf ist das allerdings nicht, denn nur der Schlüsselinhaber kann einen eigenen, von einem Dritten signierten Schlüssel auf einem Key-Server löschen oder ersetzen lassen.

Mit Version 6.x von PGP ist es nun möglich, Signaturen unter einem Schlüssel zeitlich in ihrer Gültigkeit (unabhängig von der Schlüsselgültigkeit) zu beschränken. Auch können Dritte zum Rückruf des eigenen signierten Schlüssels ermächtigt werden. Mit diesen Erweiterungen nähert sich PGP den Erfordernissen eines zentralen Zertifikatsmanagements, wie es in größeren Unternehmen benötigt wird. Allerdings sind diese Erweiterungen nicht mit früheren PGP-Versionen kompatibel.

5 Empfehlung

PGP ist ein sehr stabiles Produkt mit hervorragender Eignung für kleinere, überschaubare Gruppen oder Projekte,

- deren Größe eine gegenseitige Überprüfung von Fingerprints (das sind Hashwerte des öffentlichen Schlüssels) erlaubt, oder
- bei denen ein „Trusted Introducer“ existiert, der die Prüfung der öffentlichen Schlüssel stellvertretend für alle Gruppenmitglieder/Projektmitarbeiter übernimmt.

Die existierenden Produktintegrationen sind sehr benutzerfreundlich realisiert. Zudem erlaubt PGP eine Vielzahl von Konfigurationsmöglichkeiten, die in der Praxis hilfreich und nützlich sind.

In größeren Gruppen ab etwa 50 Teilnehmern, in Unternehmen und in Verwaltungen unterliegt PGP jedoch einer Reihe von Einschränkungen:

- konzeptionell wurde PGP für ein „Web of Trust“ und nicht für eine Zertifizierungs-infrastruktur entwickelt;
- bis Version 6.5 unterstützt PGP keine X.509-Zertifikate;
- die sichere Kommunikation ist auf PGP-Nutzer beschränkt, da PGP/MIME nicht mit S/MIME kompatibel ist;
- die Produktpolitik von PGP verfolgt seit Version 5.0 eine wechselhafte und uneinheitliche Linie (z.B. Einschränkung der RSA-Unterstützung in einzelnen Versionen),
- mit der Festlegung auf PGP entsteht eine erhebliche Abhängigkeit von einem einzigen Hersteller.[2]

S/MIME wird in nächster Zukunft aller Voraussicht nach von den meisten Herstellern von E-Mail-Clients unterstützt. S/MIME-Lösungen erlauben das Zusammenspiel mit allen PKI-Komponenten, die X.509 (97)-Zertifikate ausstellen.

Viele S/MIME-Produkte sind heute allerdings amerikanischen oder kanadischen Ursprungs. Sie unterliegen inzwischen keinen Exportrestriktionen mehr, so daß auch in Europa Produktversionen mit hinreichend langen Schlüsseln bezogen werden können. Allerdings bleibt ungewiß, ob exportierte amerikanische Produktversionen möglicherweise Hintertüren für die amerikanischen Nachrichtendienste enthalten.

MailTrusT-Produkte bieten sich als (Zwischen-) Lösung an, obwohl Version 1 keine S/MIME-Formate unterstützt und lediglich mit X.509 (89)-Zertifikaten arbeitet. Die Spezifikation der Version 2 des MailTrusT-Standards hat jedoch eine Weiterentwicklung der Produkte zu vollständigen PKI-Lösungen beziehungsweise S/MIME-kompatiblen E-Mail-Clients (oder Plug-Ins) zur Folge. Wichtiger Vorteil von MTT-Produkten ist, daß sie in Deutschland heute bereits zahlreich verfügbar sind und sich in größeren Pilot-Installationen (wie z.B. im Projekt „Sphinx“) bewähren mußten. Da die Hersteller alle aus Deutschland kommen, arbeiten die Produkte ohne Ausnahme mit starker Kryptographie.

---

[2] Eine „Gegenmaßnahme“ der Deutschen Bundesregierung (BMI) ist das Projekt „Gnu Privacy Guard (GPG)“ (http://www.gpg.org), in dem derzeit eine OpenPGP-Implementierung unter Open Source-Lizenz realisiert wird.

6 Literatur

ANSI_81 American National Standards Institute (ANSI): *Data Encryption Algorithm*. ANSI X3.92, 1981.

ANSI_98 American National Standards Institute (ANSI): *Triple Data Encryption Algorithm Modes of Operation*. American National Standards Institute, ANSI X9.52, 1998.

AtSZ_96 Atkins, D.; Stallings, W.; Zimmermann, P.: *PGP Message Exchange Formats*, Request for Comments (RFC) 1991, August 1996.

Baus_96 Bauspieß, Fritz: *MailTrusT-Spezifikation*. Version 1.1, 18. Dezember 1996. (`http://www.secorvo.de/publikat/mttspc11.pdf`)

BDRS+_96 Blaze, Matt; Diffie, Whitfield; Rivest, Ronald L.; Scheier, Bruce; Shimomura, Tsutomu; Thompson, Eric; Wiener, Michael: *Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security*. BSA Report, Januar 1996.

BiBF_99 Biester, Jobst; Bauspieß, Fritz; Fox, Dirk: *MailTrusT Version 2. März 1999* (`http://www.secorvo.de/publikat/mttspc20.zip`)

Bies_99 Biester, Jobst: *MailTrusT-PKI-Spezifikation*. Datenschutz und Datensicherheit (DuD), 4/1999, S. 218-221.

CDFT_98 Callas, J.; Donnerhacke, Lutz; Finney, H.; Thayer, R.: *OpenPGP Message Format*. Request for Comments (RFC) 2440, November 1998.

Denn_84 Denning, Dorothy E.: *Digital Signatures with RSA and Other Public-Key Cryptosystems*. Communications of the ACM, Vol. 27, No. 4, April 1984, S. 388-392.

Dobb_97 Dobbertin, Hans: *Digitale Fingerabdrücke. Sichere Hashfunktionen für digitale Signatursysteme*. Datenschutz und Datensicherheit (DuD), 2/1997, S. 82-87.

Elki_96 Elkins, M.: *MIME Security with Pretty Good Privacy (PGP)*, Request for Comments (RFC) 2015, Oktober 1996.

HoPo_94 Horster, Patrick; Portz, Michael: *Privacy Enhanced Mail: Ein Standard zur Sicherung des elektronischen Nachrichtenverkehrs im Internet*. Datenschutz und Datensicherung (DuD), 8/1994, S. 434-442.

ITU_97 International Telecommunication Union: *Information Technology – Open Systems Interconnection – The Directory: Authentication Framework*. ITU-T Recommendation X.509 (6/1997).

Kali_92 Kaliski, Burt: *The MD2 Message-Digest Algorithm*. Request for Comments (RFC) 1319, aktualisiert RFC 1115, Network Working Group, 4/1992.

LeVe_99 Lenstra, Arjen K.; Verheul, Eric: *Selecting Cryptographic Key Sizes*. November 24, 1999; http://www.cryptosavvy.com.

NIST_95 National Institute of Standards and Technology (NIST): *Secure Hash Standard (SHS-1)*. Federal Information Processing Standards Publication 180-1 (FIPS-PUB), 17.04.1995.

NIST_98 National Institute of Standards and Technology (NIST): *Digital Signature Standard (DSS)*. Federal Information Processing Standards Publication 186-1 (FIPS-PUB), 15.12.1998.

RiSA_78 Rivest, Ronald L.; Shamir, Adi; Adleman, Leonard: *A Method for obtaining Digital Signatures and Public Key Cryptosystems*. Communications of the ACM, Vol. 21, No. 2, 1978, S. 120-126.

Rive_92 Rivest, Ronald L.: *The MD5 Message-Digest Algorithm*. Request for Comments (RFC) 1321, April 1992.

SBMB+_99 Stark, Claus; Biester, Jobst; Mack, Holger; Bauspieß, Fritz, Fell, Hans-Willi; Wielandt, Klaus; Beckmann, Thomas; Landwehr, Norbert: *PKI Organisationshandbuch*. SPHINX Pilotversuch Ende-zu-Ende-Sicherheit, BMI, Schriftenreihe der KBSt, Band 46, Berlin, November 1999.