S/MIME Version 3.x Certificate Handling

- Ausarbeitung -

Name: Thomas Donner
Studienrichtung: Wirtschaftsinformatik
Semester: VI
Matrikelnummer: 72 10 41
Email: thomas.donner@student.fh-nuernberg.de

Studienfach: Internet Security
Professor: Prof. Dr. Trommler

Agenda

1. Einführung in Zertifikate

    1.1 Notwendigkeit von Zertifikaten
    1.2 Einordnung in die Welt der Kryptographie
    1.3 Prinzipieller Ablauf mittels Zertifikate
    1.4 Aufbau eines Zertifikates
    1.5 Public-Key Infrastruktur
    1.6 Gültigkeit von Zertifikaten

2. S/MIME Zertifikate Handhabung

    2.1 Einleitung
    2.2 Cryptographic Message Syntax (CMS)
    2.3 Inhalt eines Zertifikates
    2.4 Ablaufsteuerung mit Zertifikaten
    2.5 Sperrung von Zertifikaten
    2.6 Sicherheit
    2.7 Änderungen von Version 3.0 zu Version 3.1

3. Literatur-Verweise

1. Einführung in Zertifikate

1.1 Notwendigkeit von Zertifikaten

Warum benötigt man Zertifikate und wozu dienen sie? Diese Frage lässt sich nicht so ohne weiteres beantworten, aber ich werde in dieser Einleitung versuchen, eine Analogie als Erklärungsansatz zu formulieren.
Im täglichen Leben wird es immer wieder nötig sein, dass man sich ausweisen muss. Sei es für einen Behördengang, wie z.B. die Beantragung eines Reisepasses oder für eine Polizei-Kontrolle. Aber auch für die Eröffnung eines Kontos bei einer Bank oder für einen neuen Handy-Vertrag muss irgend eine Art von Dokument vorgelegt werden, welches beweist, dass man auch derjenige ist, für den man sich ausgibt. Als Dokument wird üblicherweise ein Personalausweis oder ein Reisepass verwendet.
Auf das Thema zurückkommend kann man also diese Analogie wie folgt auflösen. Als Ausweis oder Pass dienen uns Zertifikate, die wiederum benötigt werden, um eine Person X davon zu überzeugen, dass z.B. Alice auch wirklich Alice ist.
Pauschal kann man sagen, dass Zertifikate zur eindeutigen Authentifizierung einer Person oder eines Unternehmens dienen.
Damit wäre geklärt, wozu Zertifikate nötig sind. Zu klären bleibt also noch, warum Zertifikate benötigt werden. Auch hierfür wird versucht, dies durch eine Übertragung der Analogie in die Welt des Internets zu verdeutlichen. Theoretisch wäre es denkbar, ein Konto bei einer Direktanlage Bank zu eröffnen, indem man einfach nur sein Zertifikat an diese schickt und nicht wie bisher eine Kopie seines Ausweises oder ein PostId-Verfahren vorlegt. Vorraussetzung hierfür ist natürlich, dass die Bank dem Zertifikat vertrauen muss und kann.

In der Welt des Internets sind Zertifikate - wie gezeigt - gleichwertig wie amtliche Dokumente des täglichen Lebens zu behandeln. Daher sollten Zertifikate auch von einer glaubwürdigen Organisation ausgestellt bzw. gegengezeichnet werden. Genauso wie ein Reisepass enthalten sie damit Angaben zum Aussteller und zum Inhaber.
Für das hier behandelte Thema möchte ich Zertifikate rein als Mittel zur Übertragung von gesicherten Emails verwenden. Möglich wäre es allerdings auch, Zertifikate für Protokolle wie SSL zu verwenden.

1.2 Einordnung in die Welt der Kryptographie

In der Welt der Kryptographie zählen Zertifikate zu dem Bereich „Schlüsselaustausch und Instanzauthentisierung“. Darunter fallen unter anderem folgende Protokolle:

- das Kerberos Prinzip, welches versucht, eine sichere Netzwerk-Kommunikation über einen „Rechte“-Server zu erlangen,
- das Diffie-Hellmann-Schlüssel-Abkommen, welches die Grundlage für jegliche Public-Key-Kryptographie ist,
- der Schlüsselaustausch mit beidseitiger Authentisierung, welches z.B für elektronische Wahlen eingesetzt werden kann,
- das Station-to-Station-Protokoll, welches ebenfalls von Diffie mitentwickelt wurde und der sicheren Kommunikation zwischen ISDN-Telefonen dienen sollte sowie
- die Public-Key-Austausch-Verfahren, zu denen die Zertifikate zählen.

1.3 Prinzipieller Ablauf mittels Zertifikate

Als Grundlage dienen zwei Kommunikationspartner, Alice und Bob. Alice möchte nun entweder Bob eine verschlüsselte Nachricht zukommen lassen, oder aber die Signatur einer Nachricht von Bob prüfen. Dazu fordert Alice - falls sie es nicht bereits besitzt - das Zertifikat von Bob an. In diesem steht der öffentliche Schlüssel von Bob, der zur Verschlüsselung/Verifizierung benötigt wird. Wie kann Alice aber sicher sein, dass sie dem Zertifikat auch wirklich vertrauen kann. In diesem Beispiel wird angenommen, dass Alice eine Einheit/Organisation kennt, welcher sie vertraut und die auch Bob kennt. Nun „fragt“ Alice bei Einheit/Organisation an, ob sie dem erhaltenen Zertifikat trauen kann. Erfolgt von dieser eine Bestätigung, so kann Alice ihre Verschlüsselung/Verifizierung durchführen. Sollte eine negative Rückmeldung erfolgen, so sind mehrere Szenarien denkbar, diese würden allerdings den Rahmen dieses Grundprinzips sprengen und werden daher erst später eingehend behandelt.

1.4 Aufbau eines Zertifikates

Im Internet hat sich der ITU-Standard X.509 [X.509] als Standard durchgesetzt. In diesem werden zusätzlich zu den oben bereits genannten Angaben noch weitere Attribute erlaubt. Dieser Standard existiert derzeit in der Version 3, jedoch ist die Folgeversion bereits in Arbeit.
Der Aufbau dieses Zertifikates sieht wie folgt aus:

- Signatur des Herausgebers
  - Versionsnummer
  - Seriennummer des Zertifikates (1)
  - Signatur- und Hashalgorithmus
  - Herausgeber (1)(2)
  - Gültigkeit
    - Nicht vor
    - Nicht nach
  - Subjekt (2)
  - Öffentlicher Schlüssel des Subjektes
  - Erweiterungen
    - Erweiterung 1
    - ……
    - Erweiterung n

(1) Die beiden Punkte zusammen bilden eine weltweit eindeutige Kombination. Somit ist sichergestellt, dass jeder Herausgeber sich nur um die Eindeutigkeit seiner eigenen Zertifikate kümmern muss und es keine globale Organisation geben muss, welche die Herausgabe dieser überwacht.

(2) Diese beiden werden jeweils wie folgt weiter unterteilt:

- CN = Instanzname oder Domainname
- OU = Bereich innerhalb der Organisation
- O = die Organisation
- L = die Stadt
- S = (amerikanischer) Bundesstaat
- C = Land

1.5 Public-Key-Infrastruktur

In dem Abschnitt zum prinzipiellen Ablauf wurde kurz darauf eingegangen, dass Alice eine vertraute Einheit/Organisation kennt, die sie nutzt, um Bobs Zertifikat zu überprüfen. Auf diesen Ablauf wird wie folgt näher eingegangen. An der Spitze einer Kette aus Zertifikaten steht ein Wurzel(Root)-Element. Dieses Element legt fest, dass man ihm trauen muss, denn der Herausgeber und das Subjekt sind identisch und das Zertifikat ist selbst-signiert. Veröffentlicht werden solche Zertifikate von staatlichen Organisationen, oder aber von Firmen wie Microsoft, Verisign oder Deutsche Telekom. Der öffentliche Schlüssel des Wurzel-Elementes der Regulierungsbehörde für Telekommunikation und Post kann online geprüft werden [REGTP] und ist ebenso im Signatur-Gesetz (SigG) eingetragen.
Von diesem Root-Element ausgehend werden weitere Zertifikate an so genannte Zertifizierungsdienstanbieter (Certification Authority CA) ausgestellt. In diesen weiteren Zertifikaten ist der Herausgeber der Herausgeber des Wurzel-Zertifikates und das Subjekt der jeweilige Anbieter. Diese CAs haben nun wieder das Recht, weitere Zertifikate an Endbenutzer wie Alice oder Bob auszustellen.
Die Prüfung des Zertifikates von Bob funktioniert dann in entgegengesetzter Reihenfolge, bis Alice auf ein Element stößt, dem sie vertraut. In Deutschland kann man Zertifikate z.B. über folgende Seiten [MEDIZON] oder [NRCA] prüfen.

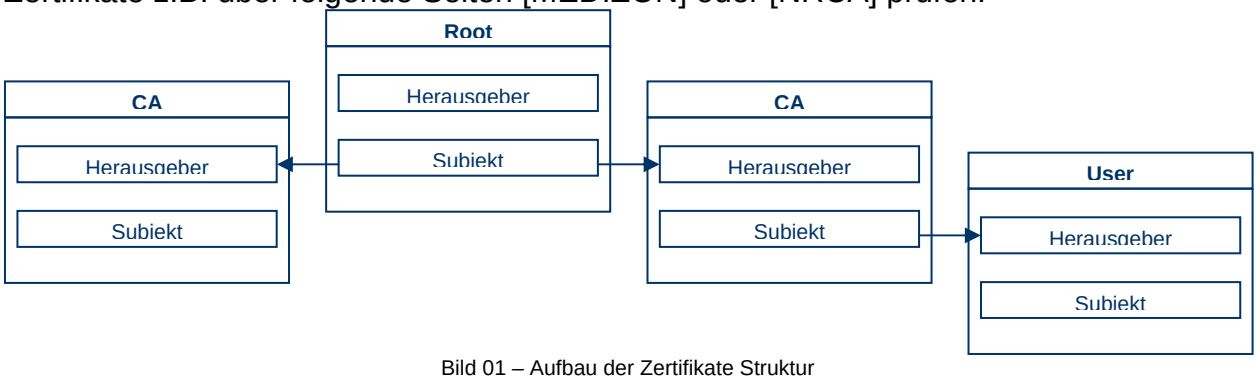


Bild 01 – Aufbau der Zertifikate Struktur

1.6 Gültigkeit von Zertifikaten

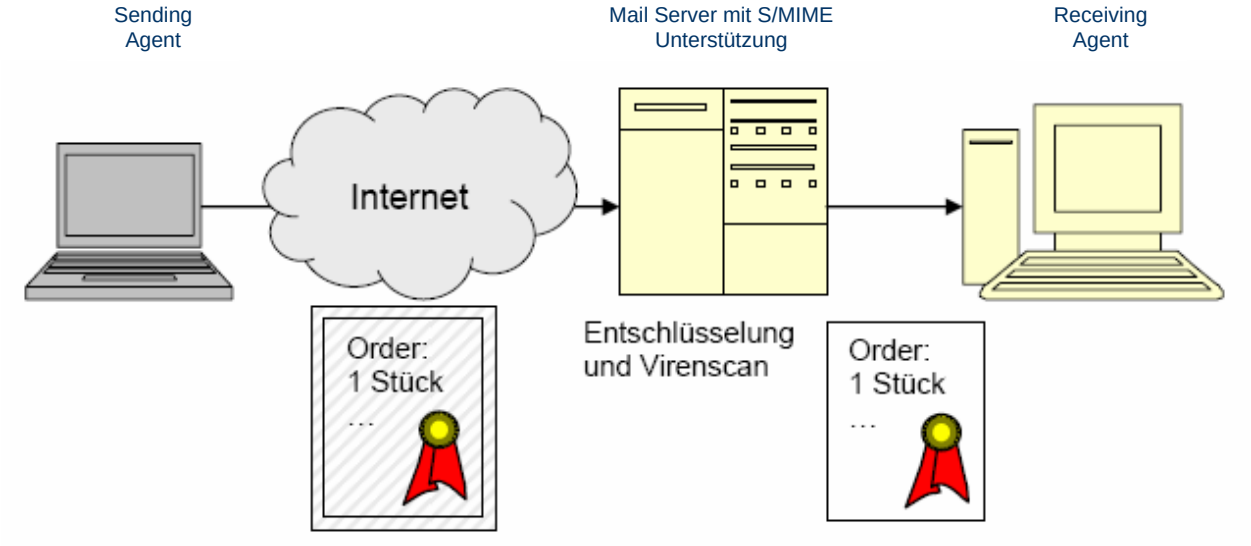
Um dem Endbenutzer möglichst viele Root-Zertifikate bereits „mit auf den Weg“ zu geben, sind viele dieser bereits in den jeweiligen Browsern eingebunden. Um eine solche Einbindung zu erreichen, muss die Wurzel Einheit/Organisation zum einen eine Menge Geld zahlen, zum anderen eine Gültigkeit von 20 Jahren für ihre Zertifikate sicherstellen. Die Überprüfung der Gültigkeit von CAs- oder Endbenutzer-Zertifikaten erfolgt meist automatisch, sobald ein oben beschriebener Baum bis zu einem vertrauten Element aufgespannt werden kann. Andernfalls bekommt der Benutzer eine Meldung, ob er diesem Zertifikat trauen möchte oder nicht.
Die Lebensdauer von CA-Zertifikaten schwankt etwa zwischen fünf und zehn Jahren. Für Endbenutzer ist standardmäßig ein Jahr vorgesehen.
Natürlich können Zertifikate auch bereits vor dem Ablauf ihrer Lebenszeit ungültig werden. So kann z.B. der private Schlüssel verloren oder schlimmer noch geknackt bzw. geklaut worden sein. Es kann sich z.B. die Anschrift oder Email des Subjekts geändert haben oder das Zertifikat kann einfach auch nur falsch ausgestellt worden sein.
Für solche Fälle gibt es so genannte Certificate Revocation Lists (CRLs), in denen ungültige Zertifikate geführt werden, oder das Online Certificate Status Protocol (OCSP), bei dem die Gültigkeit eines Zertifikates ad hoc geprüft werden kann. Auf beide wird später noch detaillierter eingegangen.

2 S/MIME Zertifikate Handhabung

2.1 Einleitung

Das Verfahren für eine sichere Email-Kommunikation beruht auf dem allgemeinen Email-Standard [RFC822] und den Erweiterungen für S/MIME [RFC2311], [RFC2312], [RFC2313], [RFC2314], [RFC2315], [RFC2630], [RFC2632], [RFC2633]. Die Zusammenfassung findet sich in [SMIME]. Im Verlauf dieses zweiten Teils wird primär der [RFC2632] und die dazu passende Neuerung [DRAFT] behandelt.

Die Übermittlung einer Nachricht kann wie folgt statt finden: Alice sendet eine signierte (und verschlüsselte Nachricht) an einen Server Bridget. Ein Zertifikat kann hierbei mitangehängt werden, falls Alice bei Bridget noch nicht bekannt ist. Bridget prüft die eingehende Mail anhand des Zertifikates und scannt sie nach Viren. Wenn beide Aktionen positiv abgeschlossen worden sind, wird die signierte (jetzt unverschlüsselte) Email an Bob weitergeleitet und dieser kann die Signatur verifizieren. Dabei ist sich Bob sicher, dass die Email auch wirklich von Alice kommt und dass sie unverändert gesendet wurde.



[Bild02] – Beispiel für eine S/MIME Anwendung

2.2 Cryptographic Message Syntax (CMS)

Die CMS erlaubt eine Vielzahl von Optionen bezüglich Inhalt und unterstützten Algorithmen. Im Folgenden werden einige grundlegende Dinge erläutert werden, die als Basisvoraussetzung für die Implementierung von S/MIME benötigt werden. Weitere Details können im RFC für die „S/MIME Version 3 Message Specification“ [RFC2633] nachgelesen werden.
Zuerst wird, wie bereits oben erwähnt, eine Unterstützung für CRLs gefordert. Es muss gewährleistet werden, dass sowohl „Sending Agents (SA)“ als auch „Receiving Agents (RA)“ dieses Verfahren unterstützen. Der festgelegte Aufbau dieser CRLs kann in [RFC2459] nachgelesen werden.
Beide Agents müssen in der Lage sein, Gültigkeitsprüfungen anhand der CRLs durchzuführen. Ferner muss der RA in der Lage sein, eine CRL in einer eingehenden Nachricht zu erkennen, zu verwenden und diese dann ggf. auch für den weiteren Gebrauch speichern zu können. Weiterhin müssen beide Agents in der Lage sein,

mehrere gültige CA-Zertifikate mit dem selben Subjekt und gleichen öffentlichen Schlüsseln, aber mit überlappenden Zeitintervallen zu verwalten. Dies kann z.B. der Fall sein, wenn eine Einheit/Organisation bei unterschiedlichen CAs das selbe Zertifikat beantragt und bestätigt bekommen hat. Dies ist beispielsweise erforderlich, um mehrere Bäume von Zertifikaten miteinander zu verknüpfen, oder um die Glaubwürdigkeit in das eigene Zertifikat zu erhöhen, da es auf diese Art von mehreren CAs validiert werden kann.

Weiterhin wird gefordert, dass sowohl die PKIX v1 und v3 Zertifikate unterstützt werden. Im Jahre 1988 wurde von der ITU der erste [X.509] Standard in der Version 1 (v1) herausgegeben. 1993 erfolgte eine Abstimmung auf X.500, welche zur Version 2 (v2) erklärt wurde. Hierbei wurden zwei neue Felder in das Zertifikat eingefügt, zur Unterstützung eines Verzeichnis-Zugriffs. Ein von der Internet Privacy Enhanced Mail (PEM) herausgegebener RFC [RFC1422], ebenso im Jahre 1993, beinhaltet Spezifikationen für ein Public-Key X.509 Zertifikat basierend auf v1. Jedoch stellte sich heraus, dass mehr Felder für die Übertragung der Informationen nach dem PEM Entwurf nötig sind. So wurde durch die Organisationen ISO/IEC/ITU und ANSI X9 (1996) die Version 3 (v3) des X.509 Zertifikates entwickelt, welche v2 um mehrere Erweiterungen ergänzt. Da v2 bis S/MIME Version 3 als untauglich in der Anwendung erachtet wurde, wurde auch von deren Verwendung abgeraten. In S/MIME Version 3.1 wird sie jedoch empfohlen. Eine weitere Erklärung dieser Versionen würde den Rahmen dieser Arbeit übersteigen. Daher sei hier auch auf den [RFC2459] verwiesen.

Als dritter Punkt wird angeregt, dass der SA beim Versenden einer Nachricht zusätzlich zu seinem Zertifikat eine Liste von verketteten Zertifikaten mitanfügt, die bis zu einem Wurzelelement reicht. Der Sinn liegt darin, dass z.B. ein neuer Empfänger, oder ein Empfänger, der keine weitere Möglichkeit der Validierung besitzt, Vertrauen in die übermittelte Nachricht und somit in das Zertifikat des SA legen kann. Auf das Anhängen dieser Liste kann verzichtet werden, wenn die Nachricht in einer Art geschlossenem Netzwerk gesendet wird, bei dem sich alle Nutzer gegenseitig vertrauen, Zugriff auf eine Datenbank oder einen gemeinsamen Zertifikat-Server haben.
Auch das Versenden eines Wurzel-Zertifikates kann erforderlich sein, wenn eine DSA-Signatur verwendet wird und die nötigen Parameter dazu in den Erweiterungen des Wurzel-Zertifikates zu finden sind. Jedoch sollte ein RA nicht ohne weiteres einem selbst-signierten Zertifikat trauen, sondern andere Methoden verwenden, sich von dessen Echtheit zu überzeugen. Auf die anderen Methoden wird weder im RFC noch in dem dazugehörigen DRAFT näher eingegangen. Denkbar ist jedoch evtl. eine telefonische/persönliche Rückfrage oder aber der Abgleich mit einem öffentlich gedruckten Dokument, wie im Falle des Signatur-Gesetzes, in welchem der öffentliche Schlüssel abgedruckt ist (vgl. [REGTP]). Weiterhin muss der RA in der Lage sein, die Verkettung anhand des Herausgeber-Feldes eindeutig nachzuvollziehen.

2.3 Inhalt eines Zertifikates

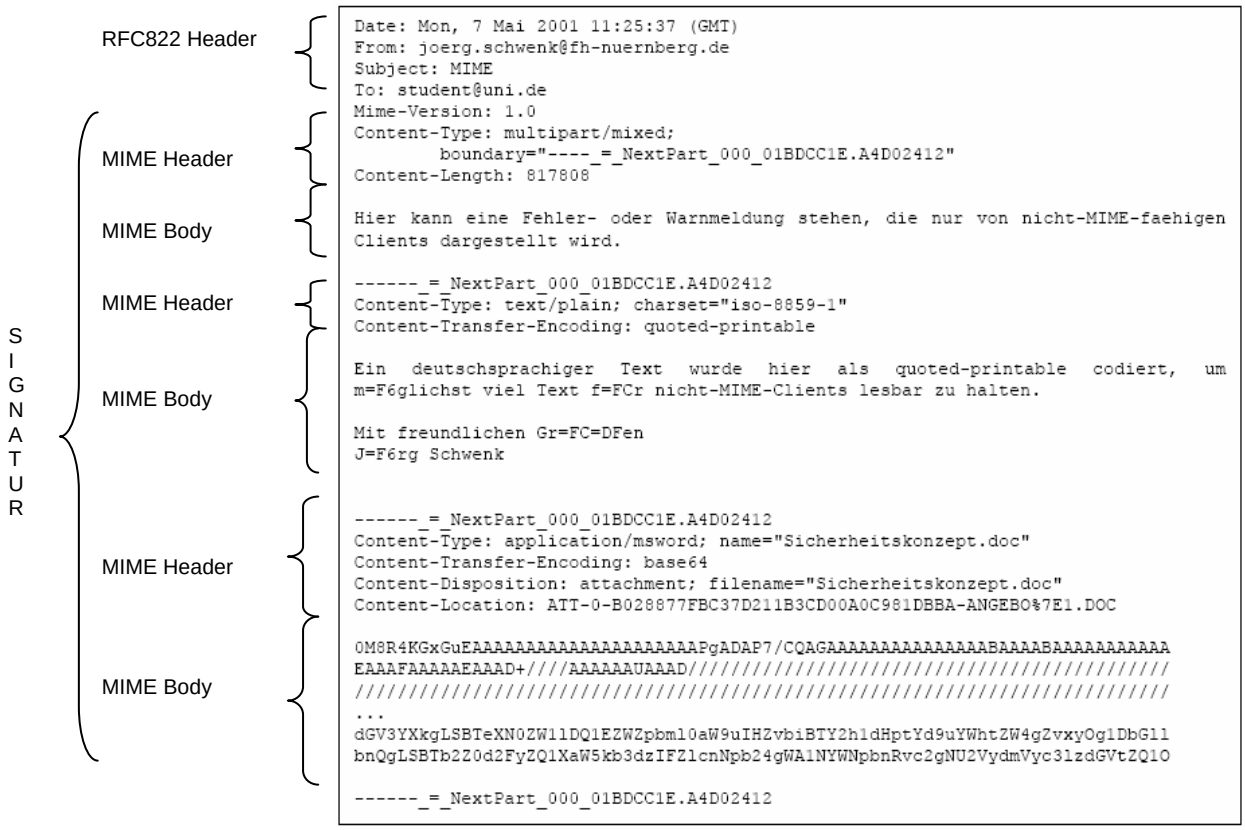
End-Benutzer-Zertifikate können Email-Adressen beinhalten. Diese müssen dem Standard gemäß [RFC822] entsprechen. Weiterhin sollten diese nicht im Subjekt-Feld stehen, sondern in der subjectAltName-Erweiterung. RAs müssen jedoch in der Lage sein, in beiden Feldern Email-Adressen zu erkennen. SAs sollten ihrerseits darauf achten, dass sie als Absende-Adresse in ihrer Nachricht eine Email-Adresse verwenden, welche auch in dem mitgeschickten Zertifikat aufgeführt ist, denn RAs müssen die Absende-Adresse verifizieren können. Sollte dies nicht der Fall sein, so

muss der Benutzer mit einer Warnmeldung darauf aufmerksam gemacht werden, dass die Absende-Adresse nicht verifiziert werden konnte, denn somit kann dieser sich nicht sicher sein, dass die Nachricht auch tatsächlich vom angegebenen Absender stammt. (Ausnahmen sind hier z.B. die Möglichkeit der Nutzung eines LDAP-Servers.)
Das Problem bei einer Nachricht besteht darin, dass zwar die „MIME Headers“ und der „MIME Body“ Teil der Signatur sind, jedoch nicht der „RFC822 Header“ [RFC2633]. Eine Email nach RFC822 umfasst folgende Bestandteile:

Subject: Text
Date: Tue, 20 Apr 2004 12:34:56 +0200
From: name@host.network
To: name2@host.network

Text

In MIME (ohne weitere Verweise, siehe: RFC2045 bis 2049) umfasst eine Email folgende Bestandteile:



[Bild03] – Beispiel für eine MIME Email

2.4 Ablaufsteuerung mit Zertifikaten

Für den Empfang und die Verwaltung von Zertifikaten werden gewisse Mechanismen benötigt, um diese einfach und leicht zu gestalten. Der von der ITU entwickelte, auf der Version 2 (s.o.) basierende X.500 Verzeichnis Service wird als ein perfektes Modell vorgeschlagen. Mit einer Struktur dieser Art können RAs schnelle Abfragen bezüglich der Gültigkeit eines Zertifikates durchführen. Eine andere Entwicklung der PKIX-Gruppe ist ein Directory Server, der im Prinzip die gleiche Aufgabe erfüllt, jedoch eben kein Service, sondern ein Server ist. Eine weitere Entwicklung der IETF ist ein Zertifikate-

Abfrage-Service auf Basis des bestehenden Domain Name Systems (DNS). Als minimale Voraussetzung muss gewährleistet sein, dass eine beabsichtigter Empfänger nach seinem Zertifikat gefragt werden kann und dieser dieses in einer signierten Nachricht zurückschickt. So ist sichergestellt, dass jeder User lokal eine Art Adressbuch pflegt, indem er alle Zertifikate von für ihn interessanten Personen speichern kann. Dafür muss gewährleistet werden, dass jeder Agent fähig ist, Zertifikate zu speichern, ferner zu im- und exportieren. Somit sollte es auch möglich sein, Nachrichten zu verschicken, welche nur Zertifikate beinhalten. Eine genauere Beschreibung dazu kann in [RFC2633] nachgelesen werden.

Als einen weiteren Verfahrenspunkt müssen hier nun die bereits mehrfach genannten CRLs erneut aufgegriffen werden. Um stets den aktuellen Stand einer CRL zu gewährleisten, sollte diese automatisch in einem bestimmten Zeitintervall von einer CA erneuert werden. Dies stellt eine aktuellere und sichere Methode dar, als sich rein auf CRLs in eingehenden Nachrichten zu verlassen. Natürlich sollte ebenso wie für Zertifikate auch dem RA die Möglichkeit gegeben werden, diese bei der Prüfung einer Nachricht mit einzubeziehen, als sie auch für den späteren Gebrauch zu speichern, was wiederum voraussetzt, dass ein RA eine CRL in einer eingehenden Nachricht erkennt.

Im Allgemeinen kann man sagen, dass es sinnvoll ist, bei jeder Überprüfung alle benötigten Zertifikate auf ihre Gültigkeit zu prüfen. Da dies einen größeren Aufwand (Gültigkeit anfragen, Antwort interpretieren, Antwort evtl. mehrfach verifizieren lassen) erfordert, also auch stellenweise für manche Benutzer nicht möglich ist (mangelnde Netzanbindung), wird hierbei vorgeschlagen, diese Überprüfung abhängig von der notwendigen Informationssicherheit der Nachricht zu machen. Meiner Meinung ist dies eine sehr ausweichende Beschreibung, denn wenn ich eine Nachricht empfange und S/MIME verwende, dann will ich auch sichergehen, dass diese korrekt ist und nicht nur zu einer gewissen Wahrscheinlichkeit. Weiter unten wird noch näher auf die Schwächen von CRLs eingegangen, ebenso werden Alternativen aufgezeigt.

Bisher wurde besprochen, wie Zertifikate und CRLs empfangen, gespeichert und benutzt werden können. Im Folgenden wird betrachtet, wie genau mit einer Nachricht verfahren werden soll, die einen RA erreicht. Der eigentliche Benutzer sollte von dem Validierungsprozess im besten Falle nichts mitbekommen und nur bei Fehlern oder schwereren Entscheidungen konsultiert werden. Somit ist es Aufgabe des RA, die erste Behandlung einer eingehenden Nachricht vorzunehmen. Dies kann die Prüfung einer Signatur, die Entschlüsselung einer Nachricht oder die Erstellung eines Diffie-Hellmann-Schlüssels sein. Bevor jedoch irgendeine dieser Aktionen durchzuführen ist, muss der RA die eingehende Nachricht anhand der vorhandenen Zertifikate und CRLs verifizieren. Dies geschieht mittels der wie auch immer verfügbaren Zertifikate/CRLs anhand der oben aufgezeigten Baumstruktur. Dafür muss der RA Unterstützung für die jeweils gewählte Signatur (DSS, DSA) bieten. Genauere Spezifikationen dazu finden sich in [DSS].

Abschließend für diesen Abschnitt wird noch kurz auf die PKIX-Erweiterungen in Zertifikaten eingegangen. Hier werden nur wenige grundlegende Aspekte angesprochen. Eine detaillierte Ausführung kann in [RFC2459] nachgelesen werden. In X.509 v3 ist es möglich, jede Erweiterung als kritisch einzustufen. Zertifikate mit als kritisch markierten Erweiterungen sollten zurückgewiesen werden!

Die wichtigsten Erweiterungen für S/MIME sind die Felder:

- authorityKeyID:
  Dieses Feld identifiziert den öffentlichen Schlüssel passend zu dem privaten Schlüssel, mit welchem das Zertifikat signiert wurde. Dies ist erforderlich, wenn ein Herausgeber mehrere Schlüssel zum Signieren verwendet.
- subjectKeyID:
  Wenn ein Zertifikat einen besonderen öffentlichen Schlüssel verwendet, so muss dieses Feld dem Wert des authorityKeyID-Feldes entsprechen, wenn dieses gesetzt ist. Somit ergänzen sich diese beiden Felder.
- subjectAltName(s):
  Wie bereits oben vorgestellt sollte in diesem Feld die Email-Adresse des Zertifikate-Inhabers eingetragen sein. Es kann aber ebenso durch DNS-, IP- oder URI-Einträge belegt werden.
- basicConstrains:
  Diese Erweiterung kennzeichnet, ob es sich bei dem Zertifikat um eine CA handelt oder um einen Endbenutzer.
- keyUsage:
  In diesem Feld tritt eine Einschränkung auf, wofür der angegebene Public-Key verwendet werden darf. Steht also z.B. eingetragen: sign-only, so darf dieser nur zum Signieren und nicht zum Entschlüsseln verwendet werden. Dieses Feld muss allerdings als kritisch eingestuft werden. Eine Ausnahme stellt hier der Diffie-Hellmann-Schlüssel-Austausch dar. Wird der verwendete Schlüssel zur Erstellung eines paarweisen Schlüssels verwendet, so müssen abhängig von der Verwendung andere Erweiterungen auf 0 bzw. 1 gesetzt sein.

2.5 Sperrung von Zertifikaten

Die Gründe für eine Sperrung von Zertifikaten wurden bereits oben aufgezählt, daher werden hier nur erneut CRLs betrachtet und die bereits erwähnten Probleme näher geschildert.
Ungültige Zertifikate werden, wie bereits bekannt, in CRLs auf dem RA gelagert. Dieser überprüft anhand der vorhanden CRL die Gültigkeit einer eingehenden Nachricht oder eines Zertifikates. Dafür, wie ebenso oben gezeigt, müsste er nun jedoch immer erst eine aktuelle CRL von einer CA anfordern. Sollte er sich gar auf mehrere CAs beziehen, so müsste er jede CRLs von allen ihm bekannten CAs beziehen, diese zusammenfügen und die Nachricht/das Zertifikat dann prüfen, oder die Nachricht/das Zertifikat gegen jede einzelne CRL prüfen. Haben wir es nun mit einem mittleren Firmen-Server zu tun, der als Beispiel 1000 Nachrichten pro Tag bekommt, so ist dies ein sehr großes Datenaufkommen. Weiterhin besteht das Problem, dass der RA über die Ungültigkeit aller im Netz verbreiteten Zertifikate informiert wird, wobei für ihn aber mit grosser Sicherheit viele dieser Zertifikate uninteressant sind, da er sie noch nie benötigt bzw. gesehen hat.

Eine Alternative dazu bietet das Online Certificate Status Protocol (OCSP) [RFC2560]. Dieses erlaubt es, die Gültigkeit eines Zertifikates ad hoc online abzufragen. Dabei wird ein OCSP-Server verwendet. Dieser steht als eine Art vertraute CA zur Verfügung und hat Kenntnis über den jeweiligen aktuellen Gültigkeitsstand aller im Netz existierenden Zertifikate. (Woher er dieses Wissen hat, steht außerhalb dieser Arbeit und kann im oben genannten RFC nachgelesen werden.) Nun ist es möglich, die Gültigkeit einer Nachricht direkt bei diesem Server zu erfragen, was einen bedeutend geringeren

Aufwand als die lokale CRL-Verwaltung hat und auch den entstehenden Traffic erheblich reduziert. Ein anderes Problem dafür ist allerdings, dass jede Nachricht, die vom Server verschickt wird, signiert werden muss, allein um deren zeitliche Aktualität zu gewährleisten. Da dies einen großen Rechenaufwand und damit viel Zeit beansprucht, wurde diese Anwendung nur für Nachrichten mit einem hohen Sicherheitsanspruch vorgeschlagen.

Eine andere Alternative sieht eine Kombination aus beiden genannten Verfahren vor. Hierbei könnten die CRLs in kleinere Listen aufgeteilt werden, welche jeweils nur die Daten enthalten, die für den RA interessant sind. Der RA muss dann selbst für eine regelmäßige Aktualisierung dieser Liste sorgen. Damit wird der Datenverkehr erneut drastisch reduziert, ebenso wie die Serverauslastung, da dieser nur noch die geforderten Listen signieren muss und nicht mehr jede einzelne Anfrage.

2.6 Sicherheit

Hier wird kurz auf die notwendige Konzipierung der RAs eingegangen, sollte die Verifizierung eines Zertifikates fehlschlagen. Dies kann passieren wenn:

- keine Email-Adresse im Zertifikat der Absende-Adresse entspricht,
- keine Email-Adresse im Zertifikat steht und kein LDAP-Server zur Verfügung steht,
- keine Zertifikate-Verkettung zu einer vertrauenswürdigen Wurzel führt,
- keine Abgleichmöglichkeit mit einer CRL (oder ähnlichem) besteht,
- eine ungültige CRL empfangen wurde,
- die CRL, gegen die abgeglichen wird, veraltet/ungültig ist,
- das Zertifikat abgelaufen ist oder
- das Zertifikat zurück annulliert wurde.

Alle dieser möglichen Fehler müssen eine entsprechende Reaktion des Programms beinhalten, die zu großen Teilen standardisiert werden sollten. Nur im Notfall sollte der Benutzer mit einer Meldung oder einer Auswahlentscheidung konfrontiert werden, da in der Regel von einem unversierten Nutzer ausgegangen werden muss.

Weiterhin muss der RA gefeit sein, gegen Angriffe durch Würmer, Viren und Hacker. Der Privat-Key muss vor unberechtigtem Gebrauch geschützt werden und darf nicht manipulierbar sein. Auch sollte die Benutzerführung einfach gehalten sein und der Benutzer davor geschützt werden Fehler zu begehen, wie das versehentliche Verwenden seines Privat-Keys für Verschlüsselung und Signatur (Angriff durch Ausnutzen des isomorphen Rings im Falle einer RSA-Signatur und Verschlüsselung).

Diese und viele weitere denkbare Fehlersituationen gilt es zu bedenken, wenn man mit einem solchen System arbeitet. Bei weitem nicht alle Fehler können einfach behandelt werden, sondern bedürfen komplexer Einzellösungen, die jedoch außerhalb dieser Arbeit stehen.

2.7 Änderungen von Version 3.0 zu Version 3.1

Abschließend wird kurz auf die Neuerungen, welche in der Version 3.1 zum Tragen gekommen sind, eingegangen. Diese können detaillierter in [DRAFT] nachgelesen werden, da sie hier nur stichpunktartig behandelt werden.

- Mehrere gültige CA-Zertifikate mit demselben Subjekt und gleichen öffentlichen Schlüsseln, aber mit überlappenden Zeitintervallen müssen unterstützt werden, bisher mussten die Agents sie nur bearbeiten können.
- V2 Attribut-Zertifikate sollten unterstützt werden, v1 Attribut-Zertifikate dürfen nicht mehr unterstützt werden.
- Verifizierung von MD2-Signaturen werden unterstützt.
- Eindeutiger Gebrauch der Email-Adresse in Zertifikaten vorgeschrieben. Bei Zertifikaten ohne Email-Adresse müssen diese nicht mit der Absende-Adresse abgeglichen werden.
- RAs sollten Informationen zum Zertifikat anzeigen, wenn die Ergebnisse der Signatur-Überprüfung vorliegen.
- RAs dürfen keine erstellten Signaturen mit einem Zertifikat akzeptieren, bei dem das digitalSignature oder nonRepudiation Bit nicht gesetzt ist.
- Weitere Klärung für das Feld „keyUsage“ und dessen Erweiterungen hinzugefügt (Sektion 4.4.4).

3) Literatur Verweise

[Bild02] Aus: „Sicherheit und Kryptographie im Internet“, Jörg Schwenk Vieweg Verlag, 1.Auflage 2002, Seite 63

[Bild03] Aus: „Sicherheit und Kryptographie im Internet“, Jörg Schwenk Vieweg Verlag, 1.Auflage 2002, Seite 60

[DRAFT] Ramsdell, B., "S/MIME Version 3.1 Certificate Handling ", Internet Draft, draft-ietf-smime-rfc2632bis-05.txt, February 2003.

[DSS] NIST FIPS PUB186, „Digital Signature Standard“, 18 May 1994.

[MEDIZON] http://149.239.16.135/mdz/Templates/zertifikatsanfrage_003_german.html

[NRCA] http://www.nrca-ds.de/chchkcert.htm

[PKIX] IETF Public-Key Infrastructure (X.509) (pkix) Working Group: http://www.ietf.org/html.charters/pkix-charter.html

[REGTP] http://www.regtp.de/tech_reg_tele/start/in_06-02-00-00-00_m/fs.html

[RFC 822] Crocker, D., "Standard for the format of ARPA Internet text messages", STD 11, RFC 822, August 1982.

[RFC1422] Kent, S., "Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management," RFC 1422, February 1993.

[RFC2311] Dusse, S., Hoffmann, P., Ramsdell, B., Lundblade, L., Repka, L., "S/MIME Version 2 Message Specification" RFC 2311, March 1998.

[RFC2312] Dusse, S., Ramsdell, B., Hoffmann, P.,Weisten , J., "S/MIME Version 2 Certificate Handling" RFC 2312, March 1998.

[RFC 2313] Kaliski, B., "PKCS #1: RSA Encryption Version 1.5", RFC 2313, March 1998.

[RFC 2314] Kaliski, B., "PKCS #10: Certification Request Syntax Version 1.5”, RFC 2314, March 1998.

[RFC 2315] Kaliski, B., "PKCS #7: Cryptographic Message Syntax Version 1.5”, RFC 2315, March 1998.

[RFC 2459] Housley, R., Ford, W., Polk, W. Solo, D., "Internet X.509 Public Key Infrastructure Certificate and CRL Profile”, RFC 2459, January 1999.

[RFC2560] Myers, M., Myers, R., Malpani, A., Galperin, S., Adams,C., "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP”, June 1999

[RFC 2630] Housley, R."Cryptographic Message Syntax”, RFC 2630, June 1999.

[RFC 2632] Ramsdell, B."S/MIME Version 3 Certificate Handling”, RFC 2632, June 1999.

[RFC 2633] Ramsdell, B."S/MIME Version 3 Message Specification”, RFC 2633, June 1999.

[SMIME] IETF S/MIME Mail Security (smime) Working Group: http://www.ietf.org/html.charters/smime-charter.html

[X.509] ITU-T Recommendation X.509 (1997 E): Information Technology - Open Systems Interconnection - The Directory: Authentication Framework, June 1997.