

Universität Hamburg
Fachbereich Informatik
Vogt-Kölln-Straße 30
22527 Hamburg

WS2002/2003

Seminar-Arbeit

PKI

Stefan Conrad, Thilo Mende und Christian Weitendorf

18. Januar 2003

Betreut durch
H.J. Mück

Inhaltsverzeichnis

1	Die Motivation	1
1.1	Einleitung	1
1.1.1	Das Papierdokument	1
1.1.2	Das elektronische Dokument	1
1.2	Digitale Signatur	1
1.2.1	Die Lücken	2
1.3	Zertifikate	3
1.4	Möglichkeiten der Verteilung	3
1.4.1	Zertifikate von Benutzern	3
1.4.2	Zertifikate von Instanzen	3
1.5	Widerruf von Zertifikaten	4
2	Zertifizierungsinstanzen	4
2.1	Dienstleistungen einer CA	5
2.2	Trustcenter ist mehr als CA	7
3	Public Key Infrastrukturen	7
3.1	Das Problem der Skalierbarkeit	7
3.2	Einfache PKI-Architekturen	8
3.2.1	Einzelne CA	8
3.2.2	Basic Trust List	8
3.2.3	Vertrauenshierarchien	9
3.2.4	Vertrauensnetz	10
3.3	Hybrid-Architekturen	10
3.3.1	Extended Trust List	11
3.3.2	Cross-Zertifizierung	11
3.3.3	Bridge CA	12
3.4	Abschlußbetrachtung der verschiedenen PKI	13
4	X.509-Zertifikate	14
4.1	Wozu Zertifikate?	14
4.2	X.509	14
4.2.1	tamper-evident envelope	14
4.2.2	basic Certificate Content	15
4.2.3	Extensions	15
4.3	Verteilung der Zertifikate	16
4.4	Probleme beim Einsatz von X.509	16
5	Rechtliche Rahmenbedingungen	16
5.1	Das Signaturgesetz von 1997	16
5.2	Die EU-Richtlinie	16
5.3	Das Signaturgesetz von 2001	17
5.4	Die Signaturverordnung	17
5.5	Anpassung der Formvorschriften	17
6	Fallbeispiel 1: DNSSEC	18

7	Fallbeispiel 2: Gesundheitswesen	19
7.1	Eingesetzte Technik	19
7.2	Akzeptanz	19
7.3	Probleme	20

1 Die Motivation

1.1 Einleitung

Dem Thema "Sicherheit im Datenverkehr" muss heutzutage ein immer größerer Stellenwert zugeordnet werden. Immer sensiblere Kommunikation, wie z.B. Finanzaufträge, Verträge, etc. werden über öffentliche Netzwerke erledigt, die jedoch im Normalfall von sich aus keinerlei Sicherheit garantieren.

Hier müssen somit zusätzliche Mechanismen entwickelt und angewendet werden, die diese geforderte Sicherheit nachträglich garantieren können. Dazu muss zunächst einmal geklärt werden, was überhaupt Sicherheit im Datenverkehr bedeutet. Um dies herauszufinden ziehen wir ein konventionelles Papierdokument heran und betrachten was für wünschenswerte Eigenschaften durch ein gleichwertiges elektronisches Pendant anzubieten sind.

1.1.1 Das Papierdokument

Wir betrachten hier ein Papierdokument, das mit einer (beglaubigten) Unterschrift versehen ist. Bei wichtigen Dokumenten liegt das Dokument als Kopie bei einem Notar vor.

Die Unterschrift in Verbindung mit der Beglaubigung garantiert die Authentizität des Dokumentes. Die Unterschrift ist somit von der Person von der sie sein soll.

Zusätzlich wird durch das Hinterlegen einer Kopie die Integrität des Dokumentes sichergestellt, da eine nachträglich geänderte Kopie aufgrund eines Abgleiches erkannt werden kann.

Was die Vertraulichkeit der Inhalte angeht, so wird diese bei einem Papierdokument hauptsächlich durch die Art der Versendung und Lagerung definiert. So geht man bei einer Versendung per Post z.B. davon aus, dass das Briefgeheimnis gewahrt wird und der Inhalt somit vertraulich bleibt.

1.1.2 Das elektronische Dokument

Die Vertraulichkeit eines elektronischen Dokumentes wird durch Kryptographie garantiert. Auf diese wollen wir hier jedoch nicht weiter eingehen, da sie Thema eines früheren Vortrages war.

Um die Punkte der Authentizität und Integrität befriedigen zu können, bedarf es einem entsprechenden Konstrukt zur konventionellen Unterschrift, die wir hier als **digitale Signatur** einführen. Anhand dieser werden wir im folgenden die Notwendigkeit einer PKI herleiten.

1.2 Digitale Signatur

Wie bereits beschrieben ist die Aufgabe der digitalen Signatur die eindeutige Zuordnung eines Dokumentes zu einem Benutzer, sowie die Sicherung der Integrität des Dokumentes.

Das Verfahren das hierfür angewendet wird ist ähnlich dem, das bei der asymmetrischen Verschlüsselung verwendet wird. Hier werden Daten jedoch nicht mit dem Public-Key, sondern mit dem Private-Key verschlüsselt. Ist eine Entschlüsselung mit Hilfe des Public-Keys erfolgreich, so kann die Verschlüsselung nur mit dem Private-Key stattgefunden haben.

Angewendet wird dies nicht auf das eigentliche Dokument, sondern einen Hashwert (z.B. MD5), der aus dem Dokument erstellt wird. Schlägt der Vergleich des entschlüsselten Hashwertes mit dem vom Empfänger erneut berechneten Wert fehl, so wurde entweder das Dokument nachträglich geändert, oder der Wert wurde nicht mit dem entsprechenden Private-Key

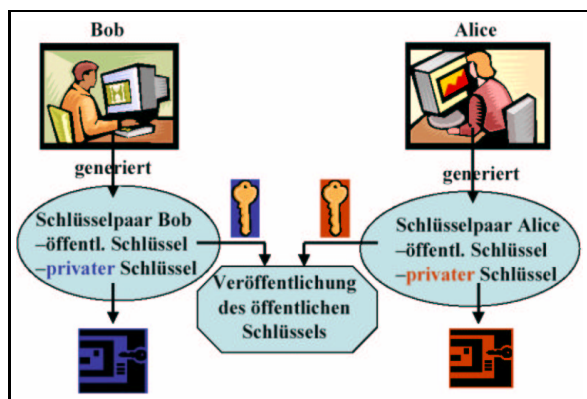


Abbildung 1: Alice und Bob erstellen Schlüsselpaare. Public-Keys werden veröffentlicht

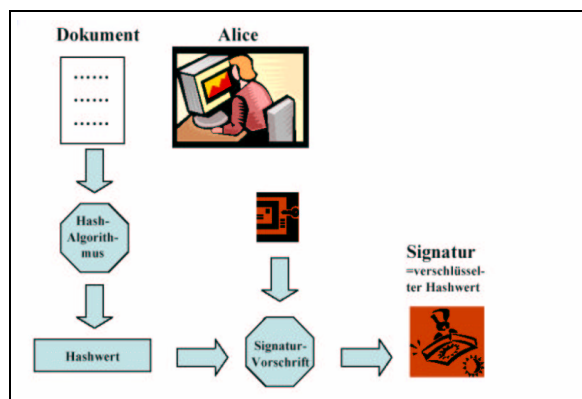


Abbildung 2: Alice verschlüsselt mit ihrem Private Key einen Hashwert des Dokumentes

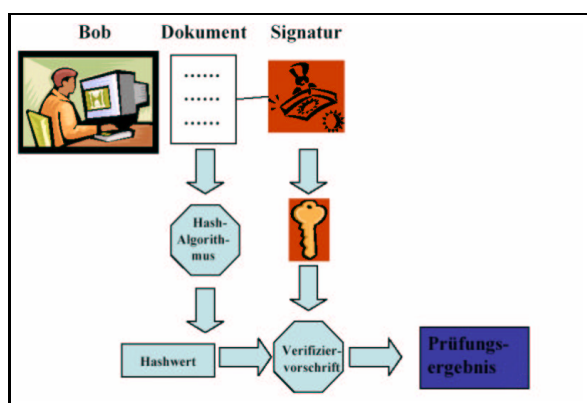


Abbildung 3: Bob entschlüsselt mit Public-Key den Hashwert und vergleicht

verschlüsselt.

Ein anschauliches Beispiel zur digitalen Signatur ist in Abbildung 1 - Abbildung 3 zu sehen.

1.2.1 Die Lücken

An dem Beispiel kann man jedoch Lücken erkennen, die ein nicht unerhebliches Sicherheitsrisiko darstellen können. Hauptsächlich zwei Punkte sind in dem Ablauf nicht genauer spezifiziert.

1. Woher hat bzw. kennt Bob den Public-Key von Alice. Damit einher geht:
2. Woher weiß Bob, dass der verwendete Public-Key tatsächlich zu Alice gehört?

Kann nicht sichergestellt werden, dass der verwendete Public-Key zu der intendierten Person gehört, so ist das Konstrukt der digitalen Signatur hinfällig. Um dies sicherzustellen müssen geeignete Wege der Verteilung von Public-Keys eingeführt werden, die die Forderung der Authentizität und Verifizierbarkeit erfüllen.

1.3 Zertifikate

Ein Zertifikat ist eine Zusage darüber, dass ein öffentlicher Schlüssel A zu einer Person B gehört.

Oft entsteht der Eindruck der Glaubwürdigkeit der zertifizierten Person aufgrund des ausgestellten Zertifikates. Dies ist ein grober Fehler, da ein Zertifikat darüber keinerlei Aussage macht. Es ist lediglich eine Zusammengehörigkeitserklärung.

Durch ein Zertifikat kann somit die Forderung nach Authentizität von Public-Keys befriedigt werden. Es bleibt jedoch noch offen, wer Zertifikate über wen ausstellt und wie Schlüssel bzw. Zertifikate zum Benutzer gelangen.

1.4 Möglichkeiten der Verteilung

Zunächst einmal gibt es ganz offensichtliche Möglichkeiten der Verteilung von Public-Keys. Dazu zählen z.B. der persönliche Kontakt von Personen, sowie die Übertragung über einen beliebigen sicheren Kanal (Post, Standleitung, ...). In beiden Fällen werden die Schlüssel von dem Benutzer selber zertifiziert, da dieser (zurecht?) davon ausgeht, dass die Authentizität gewährleistet ist.

Beide Möglichkeiten haben jedoch auch gravierende Probleme. So ist ein persönlicher Kontakt schwer möglich bei sehr großen Benutzergruppen, d.h. das Verfahren skaliert schlecht. Was den sicheren Kanal angeht, so ist dieser nicht immer verfügbar oder auch teuer.

Eine weitere Möglichkeit zur Verteilung ist ein vertrauenswürdiger Dritter. Dies ist zwar auf den ersten Blick lediglich eine Verlagerung des Problems, sie wird sich jedoch als sehr praktikabel erweisen.

1.4.1 Zertifikate von Benutzern

Zertifizieren sich Benutzer direkt untereinander, so ist dies bei übersichtlichen Benutzergruppen sicherlich die einfachste und günstigste Lösung. Bei beliebigen Zusammenkünften (z.B. Key-Signing-Parties) können Schlüssel oder Fingerprints ausgetauscht werden. Die Authentifizierung geschieht auf direktem Weg durch visuelles (oder sonstiges) Erkennen des Schlüsselbesitzers.

Nachteil dabei ist, dass das Verfahren schlecht auf verstreute und auch große Benutzergruppen anwendbar ist. Wird das Verfahren erweitert, so dass auch indirekte Zertifizierung (über mehrere Benutzer) möglich ist, so wird die Qualität der Zertifikate unklar, da nicht bekannt ist nach welchen Regeln andere Benutzer ihre Zertifikate vergeben haben.

Ein weiteres akutes Problem ist das Widerrufen von öffentlichen Schlüsseln. Ohne feste Verteilungspfade kann dies (wenn überhaupt möglich) sehr lange dauern, und im Falle eines kompromittierten Schlüssels somit ein erhebliches Sicherheitsrisiko darstellen.

1.4.2 Zertifikate von Instanzen

Werden Zertifikate von einer Instanz ausgestellt, so lassen sich die Teilnehmer lediglich von dieser zertifizieren. Die Instanz überprüft die Identität der Zertifikatnehmer und erstellt gegebenenfalls das Zertifikat. Technisch gesehen ist dieses Zertifikat nicht sehr viel mehr als eine Signatur der Zertifizierungsinstanz um den Public-Key des Zertifikatnehmers (dazu siehe auch X.509v3). Somit ist die Verteilung der Zertifikate auch über unsichere Kanäle

möglich. Voraussetzung ist jedoch, dass der Empfänger den Public-Key der Zertifizierungsinstanz kennt und besitzt. Hiervon kann jedoch ausgegangen werden, da fast alle betroffenen Programme von Haus aus eine Liste entsprechender Public-Keys mitbringen.

Ein häufig auftretender Irrtum ist, dass der Zertifikatnehmer der Zertifizierungsinstanz vertrauen muss. Dies wäre zwar schön, ist jedoch prinzipiell nicht notwendig. Vertrauen müssen lediglich Benutzer, die einen Public-Key (Zertifikat) von der Instanz anfordern.

Ein weiterer Vorteil von Zertifizierungsinstanzen ist, dass für jeden Teilnehmer klar ist, wem er vertrauen muss. Dieser Punkt wird zwar bei späterer Betrachtungsweise (mehr als eine Zertifizierungsinstanz) etwas komplizierter, er wird jedoch immer noch gelten.

Zertifizierungsinstanzen unterliegen einer Policy, die von einer Policy Certification Authority (PCA) festgelegt wird. Zertifiziert eine PCA eine Zertifizierungsinstanz (CA), so muss die CA sich mindestens an die entsprechende Policy der PCA halten. In der Policy sind z.B. minimale Anforderungen an die Identitätsüberprüfung bei der CA festgelegt (reicht eine Mail, oder ist ein Ausweis Pflicht?). Die Policy ist öffentlich und somit für jeden Benutzer einsehbar. Das Verfahren der Zertifikatsvergabe ist somit sehr transparent.

Ein korrigierter Ablauf des bereits aufgezeigten Bob-Alice-Beispiels ist in Abbildung 4 - Abbildung 7 zu sehen. Die zuvor genannten Lücken sind nun nicht mehr vorhanden. Bob kann von der Zertifizierungsinstanz ein Zertifikat anfordern. Hat diese ihre Arbeit gewissenhaft gemacht, so gehört der von Bob erhaltene Public-Key garantiert zu Alice. Ist die Entschlüsselung der Signatur erfolgreich, so folgt daraus sicher, dass das Dokument von Alice kommt und nach der Signierung nicht mehr geändert wurde.

1.5 Widerruf von Zertifikaten

Wie bereits erwähnt, ist ein Widerruf von Zertifikaten bei direkter Benutzerzertifizierung schwierig. Bei Zertifikatserstellung von Instanzen ist dies einfacher zu realisieren. So wie die Zertifikate verteilt werden, können auch Widerrufe verteilt werden.

Wurde z.B. ein Private-Key missbraucht / gestohlen, so kann der Besitzer bei seiner CA einen Antrag auf Widerruf stellen. Diese übernimmt dann die Bekanntmachung. Es ist dabei wichtig zu beachten, dass Schlüssel / Zertifikate bloß vom Zertifizierer widerrufen werden können.

Die Verteilung findet dabei über Certificate Revocation Lists (CRL) statt. Diese sind genau wie Zertifikate von der CA signiert, wodurch auch hier eine Verteilung über unsichere Kanäle möglich ist. Wie genau die CRLs nun jedoch verteilt werden, ist Ermessenssache. Vom Postwege bis zu einem Online Revocation Checking ist vieles denkbar. Wünschenswert bei der Verteilung ist eine hohe Aktualität und ein möglichst geringes Verkehrsaufkommen. Dies sind jedoch zwei sich widersprechende Wünsche.

2 Zertifizierungsinstanzen

Während Benutzer sich beliebig gegenseitig zertifizieren können, darf eine Zertifizierungsinstanz, auch Zertifizierungsstelle genannt, nicht so willkürlich Zertifikate erteilen. Die Zertifizierungsinstanz muss sich an einen festgelegten, veröffentlichten Katalog von Richtlinien halten, ihre Policy. Die Policy ist das Regelwerk für die Arbeit der Zertifizierungsstelle - sie legt, die Maßnahmen zur Sicherung des Public Key, die Art und Weise, wie die Identität des Antragsteller zu überprüfen ist, sowie alle weiteren sicherheitsrelevanten Arbeitsweisen

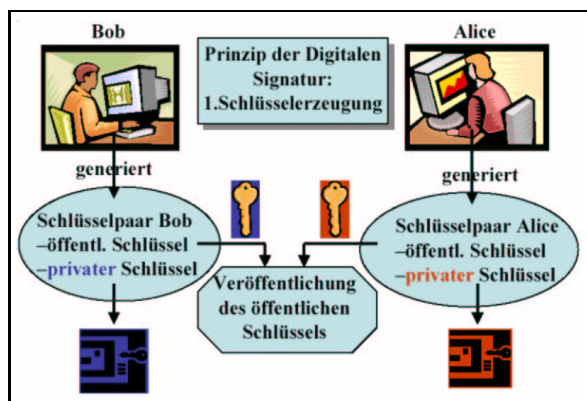


Abbildung 4: Alice und Bob erstellen Schlüsselpaare. Public-Keys werden veröffentlicht

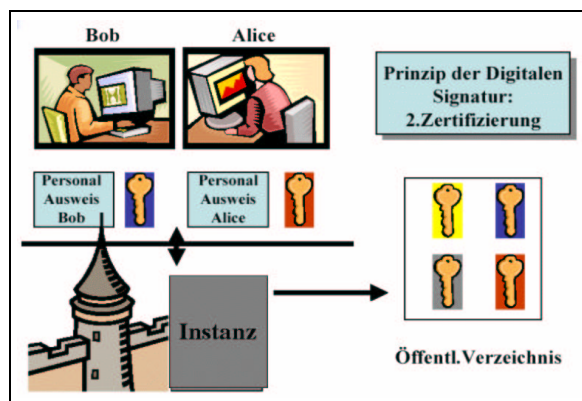


Abbildung 5: Alice und Bob identifizieren sich, Zertifikate werden erstellt und veröffentlicht

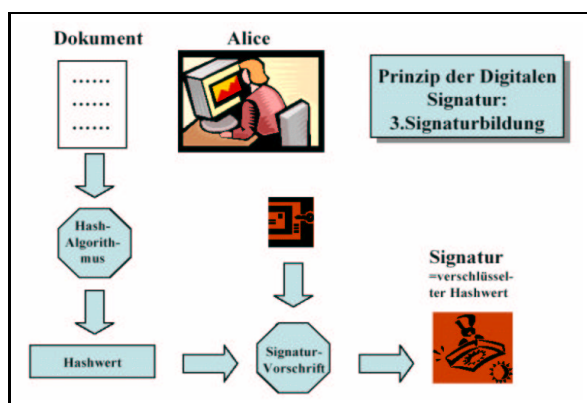


Abbildung 6: Alice verschlüsselt mit ihrem Private Key einen Hashwert des Dokumentes

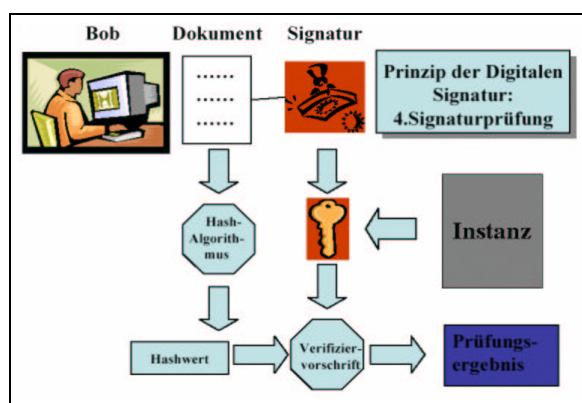


Abbildung 7: Bob entschlüsselt mit dem von der CA signierten Public-Key den Hashwert und vergleicht

fest. Zur Einhaltung ist die „Certification Authority“ (engl. Name für Zertifizierungsstelle, kurz CA) gesetzlich verpflichtet. Die CA übernimmt die Garantie, dass die Daten der von ihr ausgestellten Zertifikate korrekt sind und muss dafür sogar haften.

Die Einhaltung der Policy ist für die CA aber nicht nur aus rechtlichen Gründen notwendig, da die Vertrauenswürdigkeit ihr größtes Kapital ist. Wird bekannt, dass eine CA nicht gewissenhaft arbeitet oder Zertifikate zu Unrecht beglaubigt, so wird sie das Vertrauen der Benutzer verlieren und damit auch ihre Kunden. Wie genau eine Zertifizierungsinstanz arbeitet, kann leicht durch jeden Nutzer überprüft werden, einfach indem er einen Antrag bei der CA stellt und sich die Arbeitsweise der CA dabei anschaut.

2.1 Dienstleistungen einer CA

Damit der Nutzer die Vorteile einer CA, bzw. PKI nutzen kann, muss jede CA einige (Basis-)Dienstleistungen anbieten:

Identitätsprüfung oder auch Registrierung der Zertifikatnehmer Hier werden die zu zertifizierenden Daten direkt von der CA oder durch Dritte für die CA geprüft. Dies kann, je nach Policy, auf verschiedene Weisen geschehen. Im einfachsten Fall kann es der Nachweis sein, dass der Nutzer den privaten Schlüssel zum Public Key besitzt („proof of possession“) und per E-mail erfolgen. Eine seriöse CA wird jedoch auf ein persönliches Erscheinen mit geeignetem Identitätsnachweis (z.B. Personalausweis) nicht verzichten.

Wird die Identitätsprüfung von der CA auf Dritte ausgelagert, spricht man von einer „Registration Authority“ (RA). Eine RA ist i.a. eine von der CA als vertrauenswürdig angesehene Person, die gemäß der Policy der CA arbeitet. Hier kann entweder eine Überprüfung vor Antragstellung durchgeführt werden: Ein Antragsteller kommt zur RA, identifiziert sich und der Antrag wird als überprüft und korrekt zur CA weitergeleitet, die daraufhin das Zertifikate des Antragstellers ausstellt. Oder es kann eine Überprüfung im Auftrag der CA durchgeführt werden, die ist z.B. nötig, wenn jemand ein Zertifikat als Mitarbeiter einer Firma haben will und die RA überprüfen soll, ob der Antragsteller wirklich bei der angegebenen Firma angestellt ist.

Zertifizierung der Benutzer Die CA erstellt ein Zertifikat für den Benutzer, das u.a. seine persönlichen Daten und seinen Public Key enthält, und signiert es, indem sie den Hash Code des Zertifikates bildet und mit ihrem private Key verschlüsselt.

Sperrmanagement für zurückgerufene und abgelaufene Zertifikate Sollte der private Schlüssel eines Zertifikat-Nehmers verloren gehen, oder ein Angestellter das Unternehmen verlassen, so muss die CA die Möglichkeit bereitstellen, bekannt zu machen, dass das Zertifikat nicht mehr gültig ist. Dafür pflegt die CA eine Liste mit ungültigen Zertifikaten, die so genannte „Certificate Revocation List“ (CRL). Wird der CA ein Zertifikat als nicht mehr vertrauenswürdig gemeldet, so wird die eindeutige Seriennummer des Zertifikates auf die CRL gesetzt. Dort ist sie so lange aufgelistet, bis das Zertifikat seine Gültigkeitsdauer überschritten hat.

Bereitstellung und Verteilung der Zertifikate Die von einer CA erstellten Zertifikate und auch die CRL müssen nun noch von der CA zum Endnutzer kommen. Die kann durch Verzeichnisdienste oder auch durch E-mail- oder Post-Abonnements geschehen. Hierbei ist immer das Sicherheitsloch zu beachten. Läuft ein Abonnement wöchentlich oder gar nur monatlich, so sind die Daten, aufgrund deren man einem Zertifikat traut (oder auch nicht), eventuell schon veraltet. Eine Verkürzung der Erscheinungsfrequenz würde zu einem erhöhten Datenverkehr führen, der größtenteils nur redundante Daten enthält. Ein Post-Abonnement käme bei einer Erscheinungshäufigkeit bei weniger als einer Woche schon gar nicht mehr in Frage, da der Versand zu lange dauert und die Daten bis sie angekommen sind schon (fast) wieder veraltet sind.

Eine sinnvolle Alternative bietet da die Online-Statusabfrage (OCSP), die für jedes zu überprüfende Zertifikat die aktuellste Information bietet. Hier muss für jedes zu überprüfende Zertifikat ein Onlinezugriff auf eine Datenbank erfolgen, was bei größeren CAs zu einem hohen Datenverkehr führt.

Damit die Rechner einer CA nicht nur mit solchen Status Abfragen beschäftigt sind, kann eine CA die Verteilung von Zertifikaten und CLR auf andere Rechner auslagern, die auf Performance und nicht, wie die eigentlichen Rechner der CA, auf Sicherheit ausgelegt sind.

Die Korrektheit und Integrität der verteilten Daten wird einzig und allein durch die Signatur der CA gewährleistet. So lange der private Key der CA nicht kompromittiert wurde, ist es nicht möglich die Daten zu fälschen.

2.2 Trustcenter ist mehr als CA

Eine CA, die zu den o.g. Basis-Dienstleistungen weiter Dienstleistungen wie Erzeugung von Schlüsselpaaren für den Zertifikatnehmer, Aufbewahrung einer Kopie seines privaten Schlüssels, für den Fall, dass das „Original“ nicht mehr verfügbar sein sollte oder Personalisierung und Ausgabe von Chipkarten als Speichermedium für die Schlüssel, so spricht man nicht mehr von einer CA, sondern von einem „Trustcenter“ oder auch von einer „Trusted Third Party“ (TTP).

Der Name macht deutlich, worin der Unterschied zu einer „normalen“ CA liegt - der Zertifikatnehmer muss dem Trustcenter deutlich mehr Vertrauen entgegenbringen als es bei einer reinen CA notwendig ist. Einer CA müssen hauptsächlich die Personen vertrauen, die die Korrektheit eines Zertifikates bestätigt bekommen, nicht aber so sehr der Zertifikatnehmer - er hat direkt nichts zu verlieren, da er der CA nur seinen öffentlichen Schlüssel offenbart. Bei einem Trustcenter muss der Zertifikatnehmer schon ein sehr großes Vertrauen zu diesem haben, immerhin überlässt er ihm seinen privaten Schlüssel, mit dem er auch Verträge unterzeichnen könnte.

3 Public Key Infrastrukturen

3.1 Das Problem der Skalierbarkeit

Kehren wir noch einmal zurück zu Alice und Bob - Alice kann ganz einfach das Zertifikat von Bob verifizieren lassen, wenn Alice und Bob die selbe CA benutzen. Alice kennt den Public Key der CA und weiß woher sie CLRs bekommt. Was ist aber wenn Bob sein Zertifikat von einer CA hat, die Alice nicht kennt. Eine einfache Idee ist, dass für alle Nutzer zusammen eine einzige CA existiert. Leider wird das nicht funktionieren können, da eine CA nicht nur einige Hundert Angestellte eines Unternehmens als Nutzer in einem Ort haben können, sondern auch Tausende oder noch mehr Nutzer überall auf der Welt verteilt, wie das z.B. bei einem Kreditkarten-Unternehmen durchaus der Fall sein kann. Das Problem liegt darin, dass die Zertifizierungsinstanz für die Identifizierung der Nutzer immer noch die Nähe zu diesen benötigt. Andererseits ist der Aufwand für die Sicherung des eigenen privaten Schlüssels, die Verteilung der CRL und Zertifikate so hoch, dass sich eine Zentralisierung der Zertifizierungsinstanzen anbieten würde. Um diesem Problem aus dem Weg zu gehen, kann man oben beschriebene RAs einrichten. Diese bieten leider auch nur in begrenztem Maße Abhilfe. Um wirklich eine große Anzahl an Nutzern sinnvoll betreuen zu können, ist es notwendig Infrastrukturen aufzubauen, die in der Lage sind, Zertifikate von Nutzern verschiedener CAs zu verwalten und auf Anfrage zu verifizieren. In folgenden werden wir verschiedene Infrastrukturen vorstellen und jeweils Vor- und Nachteile gegeneinander abwägen.

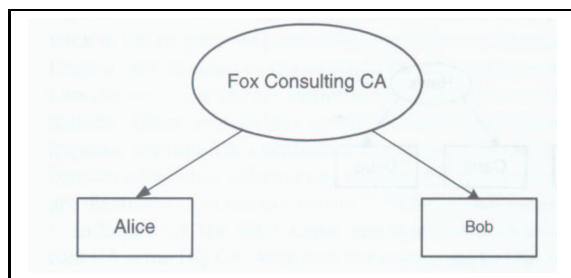


Abbildung 8: Alice und Bob bekommen ihre Zertifikate von der Fox Consulting CA

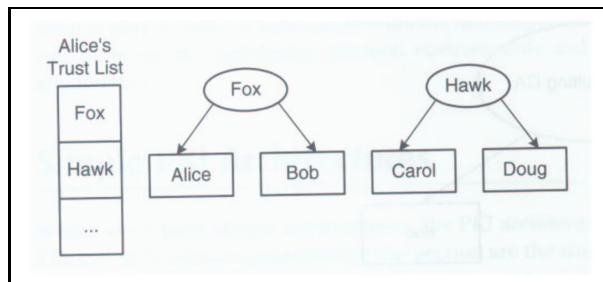


Abbildung 9: Alice vertraut noch anderen CAs

3.2 Einfache PKI-Architekturen

3.2.1 Einzelne CA

Wenn man eine einzelne CA für sich betrachtet, kann man eigentlich nicht von einer Infrastruktur sprechen, wir wollen trotzdem damit beginnen, um dann ,darauf aufbauend, Verbesserungen und Erweiterungen vorzustellen:

Eine einzelne CA vergibt ihre Zertifikate nur an Endnutzer und nicht an andere Zertifizierungsstellen. Eine Nutzer akzeptiert nur Zertifikate, die von der selben CA stammen, wie sein Zertifikat.

Für Alice und Bob bedeutet das, sie müssen ihre Zertifikate beide von der selben CA haben (vgl. Abb. 8), damit sie sich gegenseitig authentifizieren können.

Die Vorteile dieser PKI Architektur ist seine Einfachheit - es muss kein Vertrauenspfad, in dem die Vertrauenswürdigkeit der ausstellenden CA überprüft wird, gebildet werden, es kann einfach mit dem bekannten öffentlichen Schlüssel der CA überprüft werden, ob die Person gegenüber vertrauenswürdig ist, oder nicht. Die Nachteile dieser Architektur sind aber genauso offensichtlich: Als größte Schwäche ist wohl die fehlende Skalierbarkeit zu nennen; eine einzelne CA kann immer nur für einen kleinen Kreis an Endnutzern bestimmt sein. Die eine CA stellt einen „Single Point of Failure“ dar - fällt die CA aus, so sind alle Nutzer davon betroffen, oder noch schlimmer, würde die Kompromittierung des privaten Schlüssels der CA eine Neuausstellung aller Zertifikate nötig machen.

3.2.2 Basic Trust List

Bei der Basic Trust List handelt es sich um die einfachste und direkteste Erweiterung der einzelnen CA. Es gibt mehrere CAs, die jeweils Zertifikate nur an Endnutzer ausstellen - es werden also wieder keine anderen CAs zertifiziert. Jeder Nutzer hat eine Liste von CAs, die er als vertrauenswürdig erachtet, seine Trust List. So vertraut ein Nutzer auch Zertifikaten, die nicht von seiner eigenen CA ausgestellt wurden. Um ein Zertifikat zu validieren, muß nur das Zertifikat und die CRL der entsprechenden CA überprüft werden. Der Nutzer vertraut nun allen gültigen Zertifikaten, die von einer der CAs auf seiner Trust List ausgestellt wurden.

Für Alice, Bob und Carol ist das schon eine große Verbesserung - um jetzt gegenseitig ihre Zertifikate zu bestätigen, müssen sie nicht mehr bei der selben CA sein, es reicht aus, die CA des anderen auf seiner Trust List zu haben (vgl. Abb. 9).

Auch bei dieser Architektur liegt der größte Vorteil in der Einfachheit - wieder müssen keine

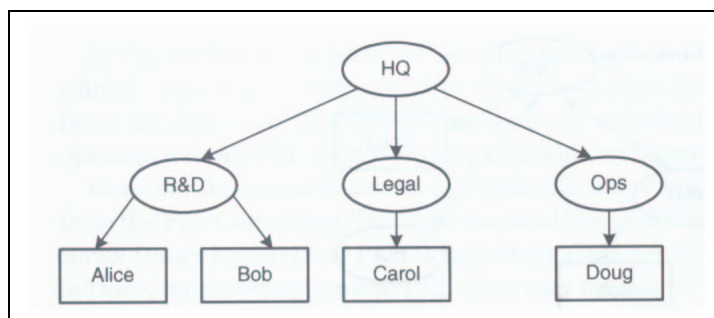


Abbildung 10: Alice' Trustpoint ist HQ als Root-CA dieser Hierarchie

Vertrauenspfade gebildet werden, alle Zertifikate können (fast) direkt validiert werden, es muss nur überprüft werden, ob die ausstellende CA auf der Trust List ist. Wenn dies der Fall ist, kann sofort die Gültigkeit des Zertifikates geprüft werden. Der nächste Vorteil ist die einfache Erweiterbarkeit - ist der Aussteller eines Zertifikates nicht auf der Trust List, so kann er einfach hinzugefügt werden, um in Zukunft alle Zertifikate von ihm zu akzeptieren. Doch genau das ist gleichzeitig ein große Schwäche. Um das erste Mal ein Zertifikat einer bisher unbekannt CA zu verifizieren, sollte der Nutzer genau überlegen, ob die CA vertrauenswürdig genug ist, um auf die Trust List zu kommen. (Denn wenn die CA erst einmal auf der Trust List ist, wird er in Zukunft einfach alle Zertifikate dieser CA akzeptieren, ohne sich weitere Gedanken darüber zu machen.) Dazu müsste er sich als erstes die Policy der CA anschauen und überprüfen, ob diese seriös ist und den Ansprüchen an das Zertifikat gerecht wird. Das ist ein nicht zu unterschätzender Aufwand, da die Policy einer kommerziellen CA leicht mehr als 100 Seiten umfassen kann. Wenn der Nutzer mit vielen unterschiedlichen Geschäftspartnern zu tun hat, wird dies sehr schnell aufwändig und lästig. Das wird oft dazu führen, dass ein Nutzer, der ein Zertifikat einer noch unbekannt CA erhält, diese nicht überprüft, sondern sie einfach auf seine Trust List setzt, ohne sich der daraus entstehenden Sicherheitslücken bewusst zu sein. (siehe auch "Abschlussbetrachtung der versch. PKI")

3.2.3 Vertrauenshierarchien

In dieser Architektur gibt es eine ausgewählte CA, die Wurzel-CA (engl. root-ca), der alle Nutzer und anderen CAs vertrauen. Alle weiteren CAs haben genau eine übergeordnete CA - so dass eine Baumstruktur entsteht. Jede CA kann sowohl Nutzer als auch weitere untergeordnete CAs zertifizieren, wobei eine untergeordnete CA, die Policy der übergeordneten CA anerkennt (, das heißt nicht unbedingt, dass sie genau die selbe Policy haben muss. Ihre Policy darf nur nicht im Widerspruch zur übergeordneten Policy stehen). Jeder Nutzer hat als Ausgangspunkt der Vertrauenspfad-Bildung die Wurzel CA. Da es in dieser Architektur nur genau einen Pfad von Wurzel CA zum Nutzer gibt (wegen der Baumstruktur), ist die Pfadbildung sehr einfach. Es muss bei jedem Zertifikat dessen Gültigkeit und danach das Zertifikat der ausstellenden Instanz geprüft werden und das rekursiv, bis man bei dem von der Root-CA ausgestellten Zertifikat angekommen ist.

Alice und Carol seien nun zwei Nutzer in einer Hierarchie und Alice erhält ein Zertifikat von Carol. Alice stellt fest, dass das Zertifikat von Carol von der Legal CA ausgestellt wurde, außerdem ist die Legal CA von der HQ CA zertifiziert, die die Root-CA in dieser Hierarchie

ist. Da Alice der HQ-CA vertraut kann sie das Zertifikat der Legal CA überprüfen. Stellt Alice fest, dass das Zertifikat OK ist, kann sie mit dem dadurch von der Legal CA erhaltenen public Key das Zertifikat von Carol überprüfen (vgl. Abb. 10).

Die Vorteile dieser Architektur sind vor allem die eindeutigen Vertrauenspfade, die sogar dem Zertifikat angehängt werden können. Die Vertrauenspfade sind zwar länger als bei den vorherigen Architekturen, doch durch ihre Eindeutigkeit und leichte Berechenbarkeit ist dies kein großer Nachteil. Des weiteren ist ihre Länge durch die Tiefe der Hierarchie begrenzt. Die Hierarchie-PKI ist leicht beliebig erweiterbar: Um eine neue CA in die PKI aufzunehmen, muss sie nur von einer CA, die bereits in die Hierarchie eingebunden ist, zertifiziert werden. Fällt eine CA aus oder muss sie aus irgend einem Grund ihre Zertifikate zurücknehmen (z.B. Kompromittierung des private Key), müssen zwar nur ihre Zertifikate neu ausgestellt werden, doch sind dann alle Nutzer, die in der Hierarchie unterhalb dieser CA angeordnet sind, aus der PKI ausgeschlossen - je weiter oben in der Hierarchie das passiert, desto schlimmer sind natürlich die Auswirkungen. Eine Kompromittierung oder ein Ausfall der Root-CA kommt dem Ausfall der CA im einfachen „Single“ CA Modell gleich.

3.2.4 Vertrauensnetz

Das Vertrauensnetz wird z.T. auch als Web of Trust bezeichnet, wobei dieser Ausdruck im allgemeinen Sprachgebrauch eher sich gegenseitig zertifizierenden Benutzern (wie bei PGP) zugeordnet wird.

Im Vertrauensnetz sind CAs direkt miteinander (peer-to-peer) verbunden (nicht notwendigerweise jede mit jeder) und zertifizieren sich untereinander. Eine CA vergibt Zertifikate sowohl an Nutzer als auch an andere CAs. Auch in diesem Fall heißt ein Zertifikat einer CA an eine andere wieder nur, dass ihre Policies sich nicht widersprechen und nicht, dass sie gleich sind. Jeder Nutzer hat als Ausgangspunkt für die Vertrauenspfadbildung die CA, die sein Zertifikat ausgestellt hat. Diese Architektur ist sehr flexibel, was den Ansprüchen der heutigen Wirtschaft sehr entgegen kommt. Es können leicht neue CAs in die PKI aufgenommen werden oder auch wieder entfernt werden, falls eine Kooperation nicht mehr gewünscht ist. Sollte ein CA einmal ausfallen, so sind nur die Nutzer davon betroffen, die ihr Zertifikat von genau dieser CA bekommen haben, da es i.a. nicht nur einen Pfad zwischen zwei verschiedenen CAs gibt.

Leider ist die Vertrauenspfadbildung bei weitem nicht mehr so einfach wie in der Hierarchie, da der Vertrauenspfad weder eindeutig ist, noch deterministisch erzeugt werden kann. Die Vertrauenspfadbildung muss über komplizierte Such-Algorithmen implementiert werden. Im ungünstigsten Fall (bei schlechten Heuristiken) geht ein solcher Vertrauenspfad durch viel zu viele oder sogar sämtlich Knoten im Netz.

3.3 Hybrid-Architekturen

Wenn nun eine Nutzergruppe sich für eine der oben genannten PKI entschieden hat und mit einer anderen Nutzergruppe zusammenarbeiten will, so müssen zwei bestehende, möglicherweise verschiedene, Architekturen miteinander verbunden werden. Die Alternative, dass beide Gruppen sich zusammen eine komplett neue PKI aufbauen, wird schon aus Kostengründen nicht in Fragen kommen, da vor allem die Dauer des Zusammenschlusses u.U. auch gar nicht feststeht. Für solche Aufgaben werden sogenannte „Hybrid-Architekturen“ verwen-

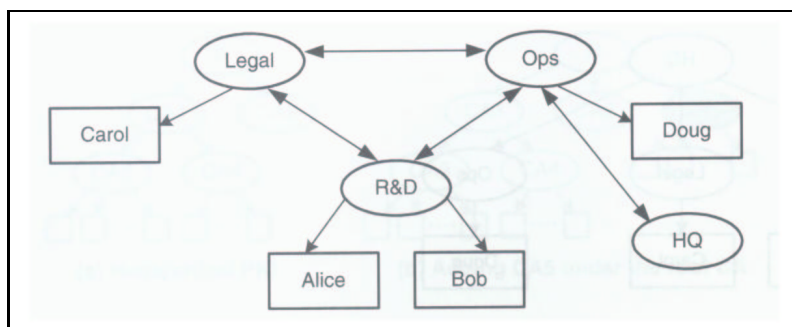


Abbildung 11: Eine Vertrauensnetz

det. Der Preis dafür ist aber immer ein höherer Aufwand für die Vertrauenspfadbildung - vor allem weil man im Vorfeld nicht weiß, aus welcher Art von PKI das vorliegende Zertifikat stammt; es müssen also Algorithmen vorgesehen werden, die mit hierarchischen Strukturen genau so gut zurecht kommen, wie mit PKI mit Netzstruktur.

3.3.1 Extended Trust List

Die einfachste Hybrid-Architektur ist die Extended Trust List. Sie ist eine direkte Erweiterung der Basic Trust List. Hierbei werden aber nicht mehr nur einzelne CAs in die Trust List gesetzt, sondern so genannte TrustPoints. Das sind CAs, die als Stellvertreter für die gesamte PKI stehen. Von diesen TrustPoints wird dann der Vertrauenspfad weiter bis zum einzelnen Nutzer aufgebaut. Dieser TrustPoint ist bei einer hierarchischen PKI die Root-CA, bei einem Vertrauensnetz kann der TrustPoint eine beliebige CA aus dem Netz sein. Dieses Verfahren behält die meisten Eigenschaften der Basic Trust List aufrecht:

- + die Erweiterung auf weitere PKI ist sehr einfach;
- es müsste wieder vor jedem Neueintrag in die Trust List die Vertrauenswürdigkeit der PKI aufwendig geprüft werden;
- aus Gründen der Bequemlichkeit wird auf diese Prüfung oft verzichtet werden

Um ein Zertifikat zu validieren, muss nun von einem Eintrag der Trust List ein Vertrauenspfad zum Nutzer gefunden werden. Es werden also nacheinander alle PKI der Trust List nach diesem Nutzer durchsucht.

3.3.2 Cross-Zertifizierung

Diese Architektur ist dem Vertrauensnetz ähnlich. Auch hier werden wie bei der „Extended Trust List“ nicht einzelne CAs, sondern ganze PKI miteinander verbunden. Wobei die TrustPoints der PKI jeweils direkt (peer-to-peer) miteinander verbunden sind. Im Gegensatz zum Vertrauensnetz ist der Graph des Netzes hier vollständig, d.h. jede PKI tauscht mit jeder anderen PKI Zertifikate aus. Der aufzubauende Vertrauenspfad hat als fixen Anfangspunkt die CA, der der Nutzer vertraut (die CA, die sein Zertifikat ausgestellt hat, bzw. in der hierarchischen PKI die Root-CA), danach läuft die Suche nach dem Zertifikatnehmer aber ähnlich aufwändig ab, wie bei der Trust List.

Hier muss bzw. darf aber nicht der einzelne Nutzer entscheiden, welcher PKI vertraut wird,

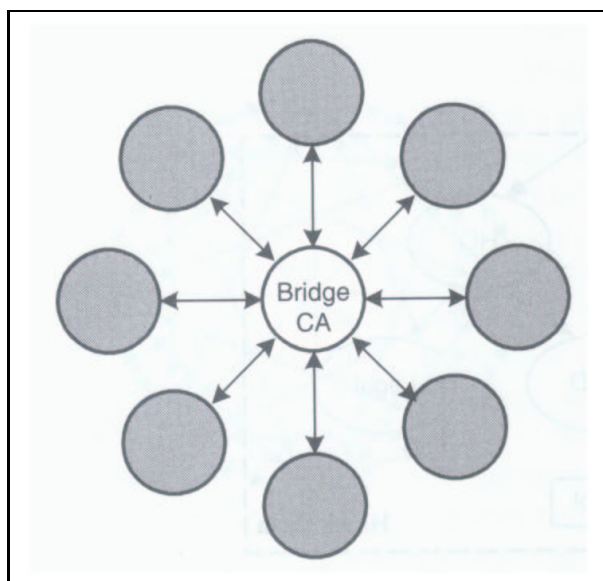
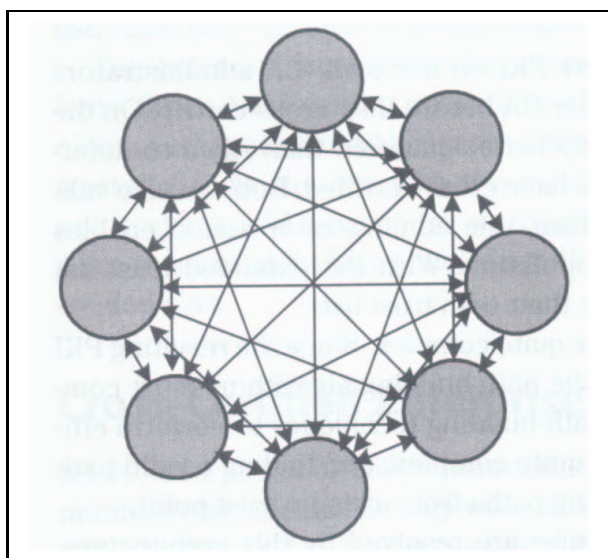


Abbildung 12: Die Verbindung von 8 PKI durch Cross-Zertifizierung
Abbildung 13: Die 8 PKI diesmal über eine Bridge CA verbunden

dies wird vom Administrator entschieden, also nur noch ein Mal pro PKI und nicht mehrfach, wie es vorher möglich war.

Diese Architektur ist aber nur für eine kleine überschaubare Anzahl an PKI geeignet, da es sehr viele peer-to-peer Verbindungen und Zertifikate gibt. Für n PKI, die sich Cross-Zertifizieren wollen, müssen $n(n-1)/2$ Verbindungen eingerichtet und $n(n-1)$ Zertifikate ausgestellt werden. Für nur 8 PKI sind das schon 28 Verbindungen und 56 Zertifikate (vgl. Abb. 12).

3.3.3 Bridge CA

Durch den Einsatz einer Bridge CA lässt sich die starke Vernetzung und die Anzahl der ausgestellten Zertifikate der Cross-Zertifizierung reduzieren. Ein Bridge CA zertifiziert nur andere CAs, aber keine Nutzer. Damit ist sie bei der Bildung von Vertrauenspfaden immer nur Zwischenstation und kein Endpunkt. Eine Bridge CA führt eine Cross-Zertifizierung mit jeder Prinzipal-CA durch. Eine Prinzipal-CA ist die CA einer PKI, die mit der Bridge direkt verbunden ist. Auf diese Weise werden beim Zusammenschluss von n PKI nur n peer-to-peer Verbindungen und $2n$ Zertifikate benötigt (vgl. Abb 13).

Sollte die Bridge-CA ausfallen oder kompromittiert werden, ist es zwar für kurze Zeit nicht möglich, Zertifikate aus anderen PKI zu validieren, aber da die Bridge CA nur Zertifikate an andere CAs ausgestellt hat und damit recht wenige, sollte es möglich sein, sehr bald die Arbeit wieder aufzunehmen. Auch der Fall, dass eine PKI ausfallen sollte, ist in dieser Architektur wenig problematisch: Die Bridge CA widerruft einfach ihr Zertifikat für die entsprechende Prinzipal-CA und damit können keine Vertrauenspfade zu dieser PKI aufgebaut werden. Sobald die PKI wieder vertrauenswürdig ist, stellt die Bridge ein neues Zertifikat aus und schon ist die PKI wieder aufgenommen. Für den selben Vorgang bei n Cross-Zertifizierten PKI müssten $n-1$ Zertifikate widerrufen und dann neu ausgestellt werden und nicht nur eines wie bei der Bridge CA.

3.4 Abschlußbetrachtung der verschiedenen PKI

Wenn man die hierarchische PKI-Architektur für sich nimmt, so bietet diese Architektur sehr viele Vorteile, die einfache Vertrauenspfad-Bildung und die einfache Erweiterbarkeit. Man kann sich fragen, wozu eigentlich die anderen Architekturen notwendig sind - es wäre doch das einfachste und effektivste für alle Beteiligten, wenn es weltweit eine Root-CA gäbe und sich alle anderen CAs, statt sich eine eigene PKI, welcher Form auch immer einfach, aufzubauen, dieser CA direkt oder auch indirekt unterordnen würden. Damit würden sämtliche Hybrid-Architekturen mit ihren aufwendigen Pfadbildungs-Algorithmen hinfällig werden. Das Problem hierbei ist aber kein technisches, sondern vielmehr ein politisches. Bei einer weltweiten Hierarchie müsste man sich weltweit auf eine Root-CA einigen - und da liegt das Problem. . . Es ist leider aus (macht- oder wirtschafts-)politischen Interessen nicht möglich, eine solche internationale Wurzelinstanz einzurichten. Von den sehr unterschiedlichen internationalen Gesetzgebungen und Erwartungen einmal ganz abgesehen, gibt es allein durch sehr divergente Vorstellungen von Anforderungen an Zertifizierungsrichtlinien große Probleme sich auf eine internationale Root-CA zu einigen. Des weiteren wird es im Bereich der Wirtschaft sehr schwer sein, von einem Unternehmen zu verlangen, sich ohne zwingenden Grund sich einer Partei oder noch schlimmer womöglich noch einem konkurrierenden Unternehmen unterzuordnen. So wird es (zumindest auf absehbare Zeit) dabei bleiben, dass jede mehr oder wenig kleine Benutzergruppe für sich seine eigene PKI aufbauen wird.

Also muss jede Nutzergruppe für sich allein entscheiden, welche Art von PKI für sie am Besten ist. So kann eine PKI, die für eine Firma schnell und kostengünstig einzurichten ist, für eine andere ein enormer Kraftakt sein. Ein Großunternehmen mit einer klaren hierarchischen Struktur, kann leicht darauf eine hierarchische PKI aufsetzen. Ist das Unternehmen aber mehr in schnelllebigen Projekten organisiert, die sich mehr oder weniger auf gleicher Ebene befinden, bietet sich eine andere PKI an. Sollte hier versucht werden, eine (starre) hierarchische Struktur einzusetzen, wäre das alles andere als sinnvoll. Durch die ständig neu auftauchende und recht schnell auch wieder verschwindenden Projekte können keine untergeordneten Projekte zertifiziert werden - denn sobald die Projekte wegfallen, besteht für die untergeordneten Projekte keine Verbindung mehr zur PKI. Also müssten alle CAs direkt unter der Root-CA angeordnet sein, was auch nicht der Idee einer Hierarchie entspricht. Für ein derartig strukturiertes Unternehmen bietet sich viel mehr ein Vertrauensnetz an.

Auch die Architektur der Trust List ist sehr verbreitet - mit einer Liste von vertrauenswürdigen CAs ist sie in nahezu jedem Internet Browser zu finden. Die Probleme der Trust List lassen sich an diesem Anwendungsbeispiel veranschaulichen: Wenn ein Nutzer im Internet auf eine Seite mit vertraulichen Inhalten surft, so wird vom Browser überprüft, von wem die Inhalte zertifiziert wurden. Für den Fall, dass sich die ausstellende Instanz nicht in der Trust List des Browsers befindet, wird der Nutzer gefragt, wie mit dem Zertifikat umgegangen werden soll. So wie oben als kritisch beschrieben, werden sich die meisten User verhalten: Der Standard Nutzer wird einfach auf OK klicken, um an die gewünschten Daten zu kommen. Wenn dabei dann noch das Feld „Inhalten von xyz immer vertrauen“ angekreuzt ist, stehen böartigem Code und Trojaner Tür und Tor auf den Rechner so weit offen, wie es nur geht.

4 X.509-Zertifikate

4.1 Wozu Zertifikate?

Um die Signatur eines Dokumentes zu verifizieren benötigt man den öffentlichen Schlüssel des Absenders. Um die Authentizität des Schlüssels zu gewährleisten, wird er meist in Zertifikaten gespeichert, die von einer CA ausgestellt werden. Das Zertifikat wird mit dem privaten Schlüssel der CA unterschrieben, die damit die Integrität des Zertifikates und, je nach Policy der CA, die Identität des Zertifikat-Benutzers sicherstellt. Um die Kompatibilität verschiedener Signierungs/Verschlüsselungs-Software sicherzustellen, ist ein gemeinsamer Standard für das Format von Zertifikaten nötig. Neben PGP-Zertifikaten hat sich besonders der X509-Standard durchgesetzt, auf den wir nun näher eingehen werden.

4.2 X.509

X.509 Zertifikate waren eine CCITT Empfehlung von 1988 zur Zugangssicherung an X.500 Verzeichnissen. Die Zertifikate werden in ASN.1 beschrieben und dann DER (distinguished encoding rules) kodiert. Mit der heute fast ausschließlich genutzten Version 3 des Standards (X.509v3) wurden die Extensions eingeführt, die eine einfache Anpassung der Zertifikate an die eigenen Bedürfnisse ermöglichen. Jede Extension kann als **critical** markiert werden, um das Bearbeiten dieser Extension auf der Client-seite zu erzwingen.

In RFC 2459 werden von der IETF die für den Einsatz im Internet nötigen Extensions und deren Markierung vorgeschlagen, um die Anzahl der möglichen Extensions zu reduzieren.

Im Folgenden wollen wir den Aufbau eines Zertifikates genauer betrachten.

version	← v3
serialNumber	← 12
signature	← md5withRSAEncryption
issuer	← C=DE, ST=Hamburg, O=University of Hamburg ...
validity	← Not Before: Aug 13 13:26:26 2002 GMT Not After...
subject	← C=DE, ST=Hamburg, O=University of Hamburg, ...
subjectPublicKeyInfo	← rsaEncryption,1024 bit,00:e5:ab:d6:a7:77:ff:...
issuerUniqueId	
subjectUniqueId	
extensions	← Subject Alternative Name: email:wwwmaint@inf...
SignatureAlgorithm	← md5WithRSAEncryption
signatureValue	← 39:37:dd:62:f1:7f:d6:3f:ed:9b:85:f5:d8:9b:....

Abbildung 14: X509v3 Zertifikat mit Werten aus dem Zertifikat des Webmailers des FB Informatik

4.2.1 tamper-evident envelope

Der sogenannte “Tamper-evident envelope” enthält das zu signierende Zertifikat (*tbsCertificate*), die ID für den zum Signieren benutzten Algorithmus (*signatureAlgorithm*) und die Signatur (*signatureValue*), das eigentliche Zertifikat wird also “eingepackt”. Die ID ist ein *ObjectIdentifier*, der Teil der ASN.1-Spezifikation ist und von offiziellen Stellen vergeben wird.

Zum Verifizieren wird die Signatur des Zertifikates mit dem angegebenen Algorithmus berechnet und mit dem `signatureValue` verglichen. Der Algorithmus wird im `tlsCertificate` nochmals gespeichert, wo er durch die Signatur gegen Veränderung geschützt ist.

4.2.2 basic Certificate Content

Der Basic Certificate Content enthält die verwendete X.509-Version, die die Syntax bestimmt, z.B. muss Version 3 angegeben werden, wenn Extensions benutzt werden. Die Serial-Number ist eine innerhalb einer CA eindeutige Nummer, die zusammen mit dem CA-Namen einen weltweit eindeutigen Identifier ergeben muss. Eine Kopie des `signatureAlgorithm` aus dem tamper-evident envelope wird im Feld `signature` gespeichert. Der Name der herausgebenden CA wird als X.500-String im Feld `issuer` gespeichert. Die `notBefore` und `notAfter` Daten geben an, in welchem Zeitraum das Zertifikat gültig ist. Ein eindeutiger Name des Zertifikatnehmers wird unter `subject` gespeichert, der öffentliche Schlüssel und der dazugehörige Algorithmus wird unter `subjectPublicKey` gespeichert. Mit X.509v2 wurden `issuerUniqueId` und `subjectUniqueId` eingeführt, die das Wiederverwenden von `subject` und `issuer` Name erlauben sollten, allerdings hat sich gezeigt, dass dieser Ansatz nicht zum gewünschten Ergebnis führt.

4.2.3 Extensions

Mit X.509 Version 3 wurden die Extensions eingeführt; wenn diese genutzt werden, muss im `basic-certificate` Version 3 gesetzt werden.

Subject-type-Extensions geben an, ob der Zertifikat-Nehmer ein Benutzer oder eine CA ist. Außerdem kann eine Pfadlängenbeschränkung angegeben werden, die die maximale Anzahl der zum Verifizieren benutzten CAs angibt. Diese sollte nach den Empfehlungen von RFC2459 jedoch nur für CAs gesetzt sein, um innerhalb einer CA auch tiefe Unterbäume zu ermöglichen.

Name Extensions können alternative Namen definieren, z.B. IP-Adressen bei IPSEC-Routern. Mittels Name-Constraints können gewisse Unterbäume aus X.500-Verzeichnissen ausgeschlossen oder nur bestimmte erlaubt werden.

Key Attributs definieren, wozu der Schlüssel benutzt werden darf, z.B. nur zum Signieren oder nur zum Verschlüsseln. Ein Problem mit der Gültigkeit des Zertifikates aus dem Basic certificate ist, dass ein Schlüssel einen Tag, bevor er abläuft, zum Signieren eines Vertrages genutzt werden kann. Soll der Vertrag 2 Tage später verifiziert werden, muss dies mit einem ungültigen Zertifikat geschehen. Deshalb kann man mit der `PrivateKeyValidity` angeben, bis wann der private Schlüssel gültig ist.

Ein Ort, wo nähere Informationen über die Policy einer CA bezogen werden können, kann in den `PolicyInformation`s angegeben werden. Außerdem kann dort beschrieben werden, welche Policy einer anderen CA äquivalent zu der benutzten ist, ob ein solches Policy Mapping erlaubt ist, und wenn ja, über wie viele verschiedenen CAs dies geschehen darf.

In den `Additional Information` kann angegeben werden, wo CRLs und weitere Informationen über die CA bezogen werden können. Zuletzt kann im Feld `SubjectDirectoryInfo` jedes beliebige X.500-Attribut gespeichert werden.

4.3 Verteilung der Zertifikate

Eine komfortable Verteilung der Zertifikate ist für die Akzeptanz einer PKI sehr wichtig. Die Verteilung per E-Mail mag in kleinen PKIs praktikabel sein, in größeren Strukturen skaliert diese Lösung aber nicht. Eine weitere Möglichkeit ist die Verteilung mittels HTTP und FTP, um die einzelnen Zertifikate durchsuchen zu können, müssen sie aber zunächst auf den eigenen PC geladen werden, was sehr unökonomisch ist. Am besten lassen sich die Zertifikate in X.500 oder LDAP-Verzeichnissen speichern, die meisten PKI benutzenden Programme haben schon einen LDAP-Client integriert. Zur Zeit wird überlegt, DNS zur Verteilung von Zertifikaten zu benutzen, dies würde allerdings die Größe der DNS-Zonendateien und die Last auf den DNS-Servern erheblich erhöhen.

4.4 Probleme beim Einsatz von X.509

Die Anzahl der existierenden Extensions und deren jeweilige Markierung als critical/non-critical führen zu Inkompatibilitäten zwischen den Zertifikaten verschiedener CAs. Auch das PolicyMapping funktioniert in der Regel nicht zufriedenstellend, da die Policies zu unterschiedlich sind.

Auch wenn die Zertifikate DER kodiert werden, können durch die unterschiedlichen Zeichensätze des subject oder issuer Namens beim Vergleichen Probleme entstehen, so dass eine CA als 2 verschiedene CAs betrachtet wird.

Außerdem werden viele Zertifikate für zu lange Zeiträume ausgestellt, so sind einige CA-Zertifikate von z.B. Verisign über 30 Jahre gültig. Niemand kann die Weiterentwicklung auf dem Gebiet der Kryptoanalyse über einen so langen Zeitraum vorhersagen, sodass diese Zertifikate eventuell vorher kompromittiert werden können.

5 Rechtliche Rahmenbedingungen

5.1 Das Signaturgesetz von 1997

Im Jahr 1997 trat das weltweit erste Gesetz zur Regelung digitaler Signaturen in Deutschland in Kraft. Die Zertifizierungsinstanzen brauchten eine Genehmigung der "Regulierungsbehörde für Telekommunikation und Post (RegTP)", wobei die Erteilung einer solchen an sehr strenge Vorschriften für die CAs geknüpft war. Zudem waren weder Haftungsfragen geklärt, noch eine Gleichstellung der digitalen Signatur zur eigenhändigen Unterschrift geregelt, sodass 1999 erst 2 Zertifizierungsinstanzen eine Genehmigung beantragt und bekommen hatten.

5.2 Die EU-Richtlinie

Am 30.11.1999 wurde eine EU-Richtlinie „über gemeinschaftliche Rahmenbedingungen für elektronische Signaturen“ verabschiedet, die für alle EU-Staaten verbindlich gilt. Sie sollte vor allem einer steigenden Akzeptanz der digitalen Signatur und einer Öffnung des Binnenmarktes für CAs dienen. Um Abschottungen einzelner Länder vorzubeugen und den Betrieb einer EU-weiten CA zu ermöglichen, bedarf der Betrieb einer CA keinerlei Genehmigungen, dafür werden aber Haftungsregelungen festgelegt. In der Richtlinie werden keinerlei tech-

nische Regelungen wie Algorithmen oder Schlüssellängen vorgeschrieben, sondern lediglich Begriffe und Eigenschaften definiert.

5.3 Das Signaturgesetz von 2001

Zur Umsetzung der EU-Richtlinie trat 2001 in Deutschland ein neues Signaturgesetz in Kraft. Dieses unterscheidet entsprechend der EU-Richtlinie zwischen der **elektronischen Signatur**, der **fortgeschrittenen elektronischen Signatur**, der **qualifizierten elektronischen Signatur** und der **qualifizierten elektronischen Signatur mit Anbieter-Akkreditierung**. Es unterscheidet **angezeigte** und **akkreditierte** Zertifizierungsdienste. Schon eine eingescannte Unterschrift ist eine elektronische Signatur, die fortgeschrittene elektr. Signatur wäre z.B. ein PGP-Schlüssel. Eine qualifizierte elektr. Signatur ist mit einem **qualifizierten Zertifikat** erstellt, welches von einer CA ausgestellt wurde. Wurde das Zertifikat von einem akkreditierten Zertifizierungsdienst ausgestellt, spricht man von der **qualifizierten Signatur mit Anbieter-Akkreditierung**.

5.4 Die Signaturverordnung

In der Signaturverordnung werden Vorschriften für CAs geregelt, unter anderem werden folgende Themen behandelt:

- Sicherheitskonzept
- Wie erfolgt die Identitätsprüfung
- Inhalt der Zertifikate
- Speicherung und Sperrung
- Einstellung der Tätigkeit
- Umgang mit ausländischen CAs

Die Vorschriften für akkreditierte CAs sind deutlich strenger, so wird z.B. nicht nur das Sicherheitskonzept CA sondern auch dessen tatsächliche Umsetzung überprüft. Die Zertifikate müssen bei einer akkreditierten CA mindestens 30 Jahre gespeichert werden und bei Einstellung der Zertifizierungstätigkeit dürfen die Zertifikate nicht gesperrt werden. Für angezeigte CAs sind diese Regelungen deutlich weniger streng, trotzdem gibt es bislang (Stand März 2003) nur einen angezeigten Anbieter, wohingegen 14 Anbieter ihr Angebot akkreditiert haben. Die Hälfte dieser Anbieter sind Steuerberater, Rechtsanwälte und Notarkammern, unter den anderen ist ein Tochterunternehmen der Bundesdruckerei und ein Gemeinschaftsunternehmen einiger privater Banken.

5.5 Anpassung der Formvorschriften

Das "Gesetz zur Anpassung der Formvorschriften" dient der Anpassung vieler Gesetze an die immer weitere Verbreitung des elektronischen Datenverkehrs. Neben dem Strafgesetzbuch und vielen anderen Gesetzen wurde vor allem das Bürgerliche Gesetzbuch (BGB) im Zuge der Reform zum 1.1.2001 verändert. Neben der Einführung der Textform, die nicht unterschriebene elektronische Texte wie z.B. E-Mails behandelt, wurde die Schriftform erweitert.

Verlangte diese vorher die handschriftliche Unterschrift, reicht seit dem 1.11.2001 auch die elektronische Unterschrift gemäß dem Signaturgesetz von 2001. Vor Gericht wird die qualifizierte digitale Signatur grundsätzlich als Beweis akzeptiert, wobei sie natürlich angefochten werden kann. Bis auf einige Ausnahmen wie z.B. die Eheschließung und die Kündigung eines Arbeitsvertrages kann somit fast alles digital unterschrieben werden.

6 Fallbeispiel 1: DNSSEC

Das Protokoll zum Auflösen von Domain-Namen zu IP-Adressen DNS ist in der Vergangenheit immer wieder durch Probleme aufgefallen. Wenn ein DNS-Server mit dem Auflösen eines Domain-Namens, der nicht in seine Zone gehört, beauftragt wird, versucht er, den für die Zone zuständigen Nameserver zu kontaktieren und die IP-Adresse zu erfragen. Kann ein Angreifer den zuständigen Nameserver außer Kraft setzen und anstelle dessen eine gefälschte Antwort senden, wird der Client auf einen anderen Server umgeleitet, ohne dass er dies merken kann.

DNSSEC ist eine konservative Erweiterung des ursprünglichen Protokolls, das diese Art von Angriffen zu verhindern versucht.

```
domain: nlnetlabs.nl
using open.nlnetlabs.nl..
@      1D IN SOA      open hostmaster (
        2003031100      ; serial
        8H              ; refresh
        2H              ; retry
        1W              ; expiry
        1D )           ; minimum

1D IN SIG      SOA 3 2 86400 20030410101515 20030311101515 63406 @ (
        CHcaDpv9aJeDA04StrA0oR5kbs/XodE6Qp9xQ5H3h3nv3ic1
        DCQYGtU= )

1D IN KEY      0x0100 3 3 (
        COjk0WXRbKbPHbESE4kNLZT12K6RqHrot6T7wnDfTEKH08b8
        . . . .
        XPbLWnq7EqvTy98YH9KwGwPmu0C916dT1h01zj71V90FNibM
        NIWCK38BbNmc ) ; key_tag= 62379

1D IN SIG      KEY 3 2 86400 20030410101515 20030311101515 63406 @ (
        CNjA9UYTFgp/T7bNaulTZd7Pt4Vnxzke5qpgDpBAb5uJKxfd
        yBasTOM= )

1D IN NXT      alpha NS SOA MX TXT SIG KEY NXT
1D IN SIG      NXT 3 2 86400 20030410101515 20030311101515 63406 @ (
        CN62IQ1a3D70p5BkG+tNNwB9pSHFZBHnY8p7tMhQx7h+1N12
        u9yrdPU= )
```

Abbildung 15: Auszüge der DNSSEC-Zone nlnetlabs.nl

Erreicht wird diese Sicherheit durch zusätzliche Records in den Zonendaten, KEY und SIG. In KEY wird der öffentliche Schlüssel eines Hosts oder einer Zone abgelegt, in SIG die Signatur eines Recordsets. Im Record NXT (not existing) werden die Records angegeben, die für eine bestimmten Host oder eine bestimmte Zone nicht existieren. Außer zu den dort angegebenen existiert für jeden Record ein SIG Eintrag. Ein Beispiel einer so signierten Zone ist in Abbildung 15 zu finden. Wird der SIG-Record eines Key-Records mit dem Schlüssel der darüberliegenden Zone signiert, kann ein Client die Chain-of-trust bis zu einem Host durchlaufen, dessen Public-Key ihm bekannt ist. Zum Beispiel könnten die Schlüssel der Root-Server mit dem Betriebssystem mitgeliefert werden.

In den Niederlanden findet zur Zeit ein Test von DNSSEC statt. Jeder Besitzer einer .NL-Second-level-Domain kann sich unter <http://secreg.nlnetlabs.nl/> registrieren und seinen PublicKey signieren lassen. Außerdem testen die Betreiber die Verfahren zum Schlüsselaustausch und zum Signieren großer DNS-Zonen, so wurde die DE-Zone testweise signiert. Mit ca.2 Mio Domains hat das Signieren auf einem 500MHz-PC 13 Stunden gedauert, nähere Informationen sind unter <http://www.nlnetlabs.nl/dnssec/de-signing.txt> zu finden. Das Signieren der Zonen ist also auch praktisch möglich, das Signieren der COM-Zone führte auf Grund ihrer Größe allerdings schon zu deutlich mehr Problemen.

DNSSEC sichert die Namensauflösung nur vor den oben beschriebenen Problemen, eine Verschlüsselung der eigentlichen Domain-Informationen wurde von den Entwicklern absichtlich nicht vorgesehen, da ihrer Meinung nach DNS-Daten öffentlich sein sollen und müssen.

7 Fallbeispiel 2: Gesundheitswesen

1992 wurde die Einführung eines digitalen Abrechnungssystems zwischen den Krankenhäusern und den Krankenkassen vorgeschrieben. Im Sozialgesetzbuch §301 werden die zu übertragenen Daten festgelegt, aufgrund der Vertraulichkeit der Patientendaten müssen die Daten während der Übertragung besonders gut geschützt werden. Die Infrastruktur für diesen Schutz wird in einer Vereinbarung zwischen den Krankenkassen und den Krankenhäusern (Team301) festgelegt und kann den veränderten technischen Möglichkeiten angepasst werden.

7.1 Eingesetzte Technik

Es wird als asymmetrische Verschlüsselung RSA mit 768bit, als Sitzungsschlüssel DES-CBC eingesetzt, zur Übertragung der Daten wird der PEM-Standard benutzt. Die Krankenhäuser, die Krankenkassen, die Rentenversicherer und die BfA haben jeweils eine eigene CA, die sich zu einer PCA zusammengeschlossen haben. Die CAs arbeiten gleichzeitig als RAs und stellen X509v1-Zertifikate aus, die Verteilung der Zertifikate erfolgt über LDAP-Verzeichnisse.

7.2 Akzeptanz

Die Kosten für ein Zertifikat liegen bei ca. 100 EUR, seit neben X.400 auch der Einsatz über das deutlich günstigere Internet möglich ist, steigt die Anzahl der Zertifikate deutlich, im September 2001 waren ca. 7000 Zertifikate im Umlauf.

7.3 Probleme

Die anfänglichen Probleme mit PEM, der Standard bot zu viel Implementationsspielraum, sind mittlerweile behoben, allerdings können noch immer keine Binärdaten ausgetauscht werden.

Auch in der neuesten Fortschreibung der Vereinbarung zum §301 wird an den veralteten Verfahren und Schlüssellängen festgehalten, der Standard widerspricht so den Empfehlungen des BSI.

Literatur

- [HP01] R. Housley, T Polk: *Planning for PKI, Best Practice Guide for Deploying Public Key Infrastructure*, Wiley Computer Publihing 2001
- [MBH99] H.-J. Mück, Carsten Benecke, Stefan Kelm (Hrsg.): *Bericht 224: Sicherheit in vernetzten Systemen*, FB Informatik 1999
- [iX901] iX September 2001 *Abgehört, Public-Key-Infrastrukturen im Gesundheitswesen*, Heise verlag
- [iX1202] X Dezember 2002 *Gesicherte Bindung, Verschlüsselung in BIND 9*, Heise verlag