Elektronische Identität

Gilbert Brands, Bernd Roellgen

Version 1.0 , 10. Juni 2016

Eine sichere elektronische Kommunikation erfordert eine End-2-end-Verschlüsselung und eine Möglichkeit, den Kommunikationspartner identifizieren zu können. SSL/TLS mit X.509-Zertifikaten ermöglicht die Aushandlung geheimer Sitzungsschlüssel und ist für alle Kommunikationsarten einsetzbar. Das X.509-Modell weist bei zunehmender Verbreitung allerdings einige Merkmale auf, die der erforderlichen globalen Anwendbarkeit im Weg stehen. Wir erweitern das X.509-Zertifikatmodell zur „Elektronischen Identität“, um diese Mängel zu beseitigen. In diesem Artikel werden Aufbau und Handhabung der elektronischen Identität beschrieben.

1 X.509 - Zertifikate

1.1 Aufbau

Wir gehen davon aus, dass der Leser mit dem Grundprinzip des X.509-Zertifikatwesen vertraut ist. X.509 – Zertifikate in der Version 3 (aktuelle Definition siehe https://tools.ietf.org/html/rfc5280) enthalten Angaben zum Aussteller, Inhaber, die öffentlichen Schlüssel des Inhabers und Informationen zum Verwendungsbereich. Die aktuelle Definition eines X.509v3-Zertifikats in der ASN.1-Notation (RFC 5280) ist:

```
Certificate  ::=  SEQUENCE  {
        tbsCertificate       TBSCertificate,
        signatureAlgorithm   AlgorithmIdentifier,
        signatureValue       BIT STRING  }

   TBSCertificate  ::=  SEQUENCE  {
        version         [0]  EXPLICIT Version DEFAULT v1,
        serialNumber         CertificateSerialNumber,
        signature            AlgorithmIdentifier,
        issuer               Name,
        validity             Validity,
        subject              Name,
        subjectPublicKeyInfo SubjectPublicKeyInfo,
        issuerUniqueID  [1]  IMPLICIT UniqueIdentifier OPTIONAL,
                             -- If present, version MUST be v2 or v3
        subjectUniqueID [2]  IMPLICIT UniqueIdentifier OPTIONAL,
                             -- If present, version MUST be v2 or v3
        extensions      [3]  EXPLICIT Extensions OPTIONAL
                             -- If present, version MUST be v3
        }

   Version  ::=  INTEGER  {  v1(0), v2(1), v3(2)  }

   CertificateSerialNumber  ::=  INTEGER
```

```
Validity ::= SEQUENCE {
     notBefore      Time,
     notAfter       Time }

Time ::= CHOICE {
     utcTime        UTCTime,
     generalTime    GeneralizedTime }

UniqueIdentifier  ::=  BIT STRING

SubjectPublicKeyInfo  ::=  SEQUENCE  {
     algorithm            AlgorithmIdentifier,
     subjectPublicKey     BIT STRING  }

Extensions  ::=  SEQUENCE SIZE (1..MAX) OF Extension

Extension  ::=  SEQUENCE  {
     extnID      OBJECT IDENTIFIER,
     critical    BOOLEAN DEFAULT FALSE,
     extnValue   OCTET STRING
                 -- contains the DER encoding of an ASN.1 value
                 -- corresponding to the extension type identified
                 -- by extnID
     }
```

Ein Inhaber-Zertifikat wird vom Aussteller, einer „Certificate Authority (CA)“, signiert und ist damit nicht fälschbar. Die CA garantiert, dass der Inhaber die angegebene Identität besitzt und keine Fälschung eines Angreifers vorliegt. Der Nutzer soll der CA-Signatur in diesem Sinn vertrauen. Die CA kann zusätzlich Listen über vorzeitig ungültig gewordene Zertifikate oder deren aktuellen Arbeitsstatus führen und soll damit ebenfalls zur Sicherheit beitragen.

1.2 Grundlegende Mängel

Im Prinzip für alle Arten der Kommunikation zur Schlüsselaushandlung/Verschlüsselung grundsätzlich geeignet, weist das Arbeitsprinzip entscheidende Schwächen auf, die bereits auf der Anwendungsschiene HTTP → E-Mail zum Versagen führen.

- Die CA-Zertifikate müssen zur Überprüfung eines Zertifikats auf dem Nutzerrechner vorhanden sein und sind in der Regel heute vorinstalliert. Gegen 50 oder mehr CA mit teilweise bis zu 50 verschiedenen Stammzertifikaten, die jeweils Unterschiedliches garantieren sollen, sind jedoch selbst für Fachleute nicht mehr überschaubar.

- Grundsätzlich ist es möglich, mit eigenen CA/Zertifikaten zu arbeiten; viele Anwendungen lassen das jedoch selbst bei korrektem Vorgehen nicht oder nur eingeschränkt zu und „warnen“ den Nutzer in übertriebenem Maße vor einer Gefährdung, die sich angesichts der Undurchschaubarkeit der Zertifikate nur graduell von der vom etablierten CA-Wesen ausgehenden Gefährdung unterscheidet.

- CA verlangen für ihre Dienste überproportional hohe Gebühren und wiederkehrenden Aufwand, was insbesondere den Einsatz der Verschlüsselung im Privatsektor wirkungsvoll verhindert.

- Der Nutzer wird im X.509-System nahezu vollständig seiner Eigenverantwortung beraubt. Er hat im Grunde keine Möglichkeit, selbst die Schutzmechanismen festzulegen und deren Einhaltung zu kontrollieren. Diese Entmündigung, zusammen mit den Kosten, dürfte vielen potentiellen privaten Nutzern den Rest an Motivation nehmen, sich überhaupt mit der Sicherheitsfrage auseinander zu setzen.

1.3 Technik

SSL/TLS-Anwendungen nutzen mehrere Komponenten:

- Eine Verschlüsselungsbibliothek stellt die notwendigen Komponenten zur Prüfung von Zertifikaten oder zur Vereinbarung von Schlüsseln / zur Verschlüsselung zur Verfügung. Häufig genutzte Bibliotheken sind OpenSSL oder Botan-SSL. Die Bibliotheken arbeiten in der Regel neutral, d.h. sie werten die Informationen nach den definierten Regeln aus und geben die Ergebnisse zur weiteren Auswertung an die Hauptinstanz weiter.
- Die Hauptinstanz stellt die Schnittstelle zum Nutzer dar. Ergebnisse der Auswertung der Bibliothek werden hier weiter verarbeitet und dem Nutzer nach weiteren Regeln präsentiert.
- Applets erlauben in geringem Umfang eine Einflussnahme des Nutzers (Bestätigung von Ausnahmen, Installation von CA und eigenen Zertifikaten), sind jedoch oft wenig nutzerfreundlich, versteckt oder führen manche Nutzervorgaben trotz Bestätigung nicht aus.

2 Elektronischen Identität

2.1 Erweiterung von X.509

Die breite Verfügbarkeit von SSL/TLS und das an sich sehr gute Hintergrundkonzept bieten an, auf dem Konzept aufzubauen und auf die Definition von etwas vollständig Neuem zu verzichten. Das EI-Konzept nutzt daher konsequent die vorhandene Technik (siehe 1.3):

- Bibliotheken werden ungeändert genutzt. Optionen zur Hinzunahme weiterer Algorithmen (z.B. der quantencomputersichere RVB-Algorithmus) entstehen jedoch zusätzlich.
- Hauptinstanzen werden „as is“ genutzt - in der Erwartung, dass gewisse anfängliche Mängel in der Nutzerfreundlichkeit bei Akzeptanz des EI-Konzepts verschwinden.
- Eigenen Applets werden verwendet, um die Nutzerfreundlichkeit zu vergrößern und die Besonderheiten des EI-Konzeptes umzusetzen.

X.509 und EI sind daher gemeinsam einsetzbar, der Softwareaufwand ist minimal und beschränkt sich auf wenige zusätzliche Komponenten.

2.2 Wesentliche Unterschiede und Ziele

Das EI-Konzept greift die erkannten Mängel des ursprünglich für das Onlineshopping konzipierten X.509-Schemas auf, die einem Einsatz für globale Verschlüsselung derzeit im Weg stehen. Das wesentliche Prinzip besteht darin, dem Nutzer die Kontrolle über seine Sicherheit selbst in die Hand zu geben und die Kosten zu beseitigen, d.h.

- **Es existiert keine zentrale Signaturinstanz**, d.h. es wird nichts garantiert bzw. der Nutzer ist nicht gehalten/gezwungen, Instanzen/Signaturen zu vertrauen, die er nicht versteht.
- **Es existiert eine nutzerindividuelle Vertrauensdatenbank**, d.h. die Kontrolle und auch die Verantwortung liegt komplett beim Nutzer, der selbst bestimmt, welcher EI er vertraut.

Ziele des EI-Schemas sind

- die grundsätzliche Verschlüsselung jeglichen Informationsaustausches, d.h. HTTP, E-Mail, Telefonie, Instant Messaging, Gerätekommunikation, ...
- die skalierbare Authentifizierung des Kommunikationspartners durch automatisierte Verfahren oder unter Mitwirkung des Nutzers für Sicherheitsbereiche,
- die Beschränkung der Kommunikation in Sicherheitsbereichen auf definierte Gruppen von Geräten,
- dem grundlegenden Unterschied zwischen den Bedürfnissen der Anwender beim Onlineshopping und dem Sicherheitsbedürfnis der Anwender beim Führen privater Kommunikation Rechnung zu tragen.

2.3 Nutzerfreundlichkeit

Dies kann natürlich nur erreicht werden, wenn die Nutzerverantwortung mit einer einfachen und übersichtlichen Bedienung verbunden ist. Diese ist gegeben, wenn

- aufwändigere Arbeitsschritte einmalig im Rahmen einer Inbetriebnahme und geführt erfolgen. Nutzer sind es gewohnt, bei der Inbetriebnahme von Geräten bestimmte Inbetriebnahmeschritte auszuführen, so dass einige zusätzliche Schritte für das EI-Konzept sich nicht akzeptanzmindernd auswirken.
- die laufende Funktion weitgehend automatisiert ist und den Nutzer nicht mit häufigen Aktionen belästigt. Ständig notwendige Zusatzaktionen führen in der Regel zum Abschalten der Funktionalität, gelegentliches Nachfragen erinnert den Nutzer an seine Eigenverantwortlichkeit und wird akzeptiert.
- klare und einfache Steuerungselemente vorhanden sind, die immer im Blickfeld des Nutzers liegen, ihn im laufenden Betrieb an seine Eigenverantwortlichkeit erinnern und durch ihre Präsenz verhindern, dass die Funktionalität in Vergessenheit gerät.
- einfache und umfassende Möglichkeiten vorhanden sind, die eigenen Sicherheitsanforderungen in den Systemfunktionen zu definieren.

2.4 Inhalte

Die Inhalte einer EI sind mit dem Bezug auf X.509 weitgehend vorgegeben. Eine Reihe von wichtigen Inhalten für das X.509-Management sind im Bereich **Extensions** untergebracht. Grundsätzlich ist dieser Bereich ein offener Datenbereich, d.h. Datenfelder, die für das EI-Management zusätzlich zu den X.509-Definitionen notwendig sind, können in diesem Bereich untergebracht werden und sollten als unbekannte Optionen vom X.509-Management ignoriert werden.

Einige X.509-Extensions sind für das EI-Management nicht notwendig oder werden ggf. etwas anders interpretiert. Vom Grundsatz her sollte das Fortlassen oder Multiplizieren bestimmter Extension-Felder in den meisten Fällen auch für X.509 unkritisch sein.

Anwendungen genießen jedoch einen gewissen Interpretationsspielraum der Regeln. Wir vereinbaren daher anwendungsabhängig, dass die Inhalte im Zweifelsfall an X.509 ausgerichtet werden, um eine Koexistenz zu ermöglichen. Korrekturen oder Ergänzungen sind nach vorhandenen Möglichkeiten auszuführen.

Elektronische Identitäten besitzen **kein Verfallsdatum** und sind unbegrenzt gültig. Wird eine EI aus irgendeinem Grund unbrauchbar oder durch eine andere ersetzt, wird dies durch das EI-Management erkannt (siehe unten). Zentrale Instanzen wie eine Revocation List oder OCSP sind nicht primärer Bestandteil des EI-Konzeptes, können

aber gleichwohl als externe Dienstleistung angeboten und berücksichtigt werden. Weitere Details werden bei den verschiedenen Anwendungsfällen diskutiert.

2.5 Bindung

EI sind an Einheiten (Geräte, Nutzer, Software) und nicht an Funktionen (Signieren, Schlüsselaushandlung) wie bei X.509 gebunden. Eine EI kann für jeden Zweck eingesetzt werden, kennzeichnet aber immer die Einheit, für die sie eingesetzt wird (zur Gewährleistung der Einsatzmöglichkeit für bestimmte Zwecke siehe Bemerkungen in 2.4). Jede Einheit erhält bei Inbetriebnahme eine EI; die Erzeugung einer EI ist Bestandteil der Inbetriebnahmeprozedur. Da jede Einheit, die in Kommunikationsvorgängen beteiligt sein kann, eine EI besitzt, ist eine generelle und globale Verschlüsselungsmöglichkeit sichergestellt. Maschinen oder Nutzer werden hierdurch wiedererkennbar (Identitätsprinzip).

Jede Einheit besitzt einen **UniqueIdentifier** (extensions v3; ein Zufallsstring mit hinreichender Länge, um globale Eindeutigkeit zu garantieren), der lebenslang gültig ist. Die Seriennummer wird für die Wiedererkennung und Verifizierung während der Lebensdauer verwendet (siehe 5).

> Ein PC besitzt eine Maschinen-EI, für den Inhaber wird bei Inbetriebnahme zusätzlich eine Nutzer-EI erstellt oder eine bereits vorhandene Nutzer-EI verwendet. Bestimmte Softwarekomponenten können mit eigenen EI ausgestattet sein. Auf einer funktionellen Einheit können daher mehrere unterschiedliche EI installiert sein. Welche EI für welchen Zweck genutzt wird, wird durch ein Regelwerk festgelegt. Implizit kann durch die Kopplung einer EI an eine Einheit ein bestimmter Nutzungszweck resultieren.
>
> Ein Nutzer kann aus verschiedenen Gründen mehrere EI besitzen. Die korrekte Zuordnung einer EI zu einem Kommunikationsvorgang liegt in der Verantwortung des Nutzers. Das entspricht dem Prinzip, unterschiedliche Kennungen/Kennworte für verschiedene Serverkonten zu verwenden und ist Nutzern methodisch vertraut.

3 EI-Typen

3.1 Globale EI

Im weltweiten Netz ist Kommunikation mit jedem anderen Teilnehmer möglich. In vielen Fällen kennen sich die Teilnehmer zunächst nicht. Für die Realisierung der Verschlüsselung einer Kommunikation dienen **globale EI**. Globale EI sind **selbstsigniert** und können mit beliebigen anderen globalen EI eine verschlüsselte Verbindung aufbauen.

SSL/TLS sieht zwei Mechanismen der Verbindungsinitiierung vor:

1. Die Verschlüsselung wird nur mit Hilfe der Server-EI realisiert; die Client-EI ist nicht beteiligt. Soll der Server den Client ebenfalls erkennen, werden Login-Prozeduren mit Name/Kennwort-Kombination oder – im Fall bereits bestehender Kontakte – Cookies eingesetzt.
2. Bei der Initiierung werden beide EI ausgetauscht, so dass Server und Client den jeweiligen Partner erkennen. Name/Kennwort entfällt.

Ist anwendungsabhängig eine Authentifizierung des Clients notwendig, wird im Rahmen des EI-Konzeptes die Möglichkeit 2 – Austausch beider EI – angestrebt. Dies führt zu

- Vereinfachung der Anmeldeprozedur, weil ein Nutzer keine Name/Kennwort-Kombination eingeben muss, sowie
- zu einer höheren Sicherheit, weil ein Angreifer im Falle einer Man-in-the-Middle-Attacke keine nutzbaren Informationen erhält, die es ihm ermöglichen würden, sich zu einem späteren Zeitpunkt ohne Beteiligung des Nutzers beim Server anzumelden.

3.2 Lokale EI und Public Key Infrastructure

Ein Kernproblem der Sicherheit ist die Verriegelung eines Netzbereiches gegen fremdes Eindringen. Die Anwendungsfälle steigen mit zunehmender Vernetzung technischer Geräte, während eine Verschlüsselung häufig nicht stattfindet und die Gefahr durch Fehlkonfiguration des Netzwerkmamangements steigt.

Lokale EI stellen eine einfache Möglichkeit der Absicherung gegen Fremdzugriffe dar. Lokale EI sind im Gegensatz zu globalen EI durch eine andere EI **fremdsigniert** und werden immer zusammen mit der EI des Signaturgebers gespeichert, so dass Kontrollen wie bei X.509-CA-Rootzertifikaten möglich sind. Lokale EI unterliegen folgenden Einsatzregeln:

- Lokale EI können nur mit anderen lokalen EI, die von der gleichen EI signiert sind, oder mit der signaturgebenden EI eine verschlüsselte Verbindung aufbauen. Alle anderen Verbindungen werden grundsätzlich abgelehnt.
- Bei Einsatz lokaler EI ist beidseitiger Austausch der EI zwingend vorgesehen, um die Absicherung des Netzwerkbereiches zu garantieren.

Mittels lokaler EI ist das Konzept der Public Key Infrastructure im industriellen und privaten Bereich problemlos realisierbar. Die Absicherung, dass nur Geräte miteinander kommunizieren, die dazu berechtigt sind, erfolgt bereits auf Netzwerkebene und nicht erst auf Anwendungsebene (zusätzliche Authentifizierungsmaßnahmen auf der Anwendungsebene sind dadurch nicht ausgeschlossen).

> Auch im privaten Bereich nimmt die Automatisierung von Hausgeräten zu. Viele Geräte sind auch aus dem Internet erreichbar, um dem Nutzer Steuerungsmöglichkeiten zu geben. Eine sichere Konfiguration ist aber oft schwierig, was die Gefahr beinhaltet, dass Angreifer die Geräte manipulieren (Abschalten der Alarmanlage, des Eisschrankes, usw.). Bei der Inbetriebnahme generiert der Nutzer mittels seiner EI eine lokale Geräte-EI. Das neue Gerät kann nun ausschließlich EI-gesteuert mit dem Nutzer oder mit anderen Geräten des Nutzers kommunizieren, ohne dass die anderen Geräte dazu neu konfiguriert werden müssen.

Mit dem EI-Konzept erhält jedes steuerungsfähige Gerät, d.h. die Kommunikation ist zwangsverschlüsselt und nicht mehr offen, wie dies heute meist der Fall ist. Im Unternehmensbereich kann das Konzept hierarchisch eingesetzt werden. Sind beispielsweise zwei Unternehmensbereich A und B separat als PKI aufzubauen, wird eine EI genutzt, zwei Zwischen-EI zu signieren, die wiederum die Geräte der Bereiche signieren. Kommunikationen zwischen den Bereichen sind dann nur über Instanzen, die über die Zwischen-EI verfügen, möglich. Diese wiederum sind gegen die weitere Öffentlichkeit durch die Primär-EI abgesichert.

4 Authentifizierung

Da keine Signatur durch eine übergeordnete Instanz vorhanden ist, kann der Nutzer nicht sicher sein, dass die Daten auf der EI tatsächlich den gewünschten Kommunikationspartner kennzeichnen. Der Nutzer ist selbst dafür verantwortlich, dies fallweise festzustellen. Er wird durch implizite Verifizierung durch das System unterstützt.

4.1 Keine Authentifizierung

Bei gelegentlich genutzten Kontakten, insbesondere bei Beschränkung des EI-Austausches auf die EI des Servers, ist eine echte Authentifizierung in der Regel nicht notwendig. Die meisten Anwendungen, die nach dem X.509-Schema arbeiten, fordern bei selbst signierten EI eine Bestätigung des Nutzers an, die EI anzunehmen. Gelegentliche Kontakte können daher nach dem Schema für implizite Authentifizierung bearbeitet werden, wobei zur Beschränkung der Datenbankgröße die EI wieder entfernt werden kann, wenn eine erneute Nutzung in einem vorgegebenen Zeitraum nicht erfolgt.

4.2 Implizite Authentifizierung

Für die meisten Kontakte ist eine vollständige Authentifizierung nicht oder erst zu einem späteren Zeitpunkt notwendig, wenn wichtige Transaktionen abgewickelt werden. Die Authentifizierung kann implizit mit steigendem Sicherheitsniveau erfolgen.

Dazu werden die EI in einer Nutzerdatenbank gespeichert, in der auch jeder weitere Kontakt notiert wird. „Kontakt“ in diesem Sinn ist anwendungsabhängig eine URL, eine IPv4/6-Adresse, eine E-Mail-Adresse, eine Telefonnummer, ein Instant-Messaging Alias usw.

Da eine EI nicht verifiziert wird, sind verschiedene Angriffsmöglichkeiten vorhanden. Beispielsweise kann ein Angreifer versuchen, einen Kontakt umzuleiten und als Man-in-the-Middle die Kommunikation abzugreifen. Dazu muss er allerdings eine eigene EI verwenden, da die originalen privaten Daten nicht zur Verfügung stehen. Da die EI notiert werden, ist der Angreifer gezwungen, jeden Kontakt ohne Ausnahme in dieser Weise zu infiltrieren. Gelingt ihm das nicht, bemerkt das EI-Management das Vorlegen einer anderen EI für den gleichen Kontakt und gibt eine entsprechende Warnmeldung, dass möglicherweise ein Angriff vorliegt, an die Anwendung heraus. Wenn die Kommunikation global und vollständig verschlüsselt wird, ist es selbst mit extrem hohem Ressourcenaufwand kaum möglich, solche Angriffe erfolgreich durchzuhalten.

Werden über einen längeren Zeitraum wiederholt Verbindungen mit einem Kontakt hergestellt, ohne dass es zu Auffälligkeiten kommt, steigt die Vertrauenswürdigkeit der EI.

> Beispiel Shopkontakt: kommt es nach vielen Besuchen einer Seite zu einem Kauf, so kann man die Summe, um die es geht, in Relation zum Aufwand eines möglichen Angriffs stellen. Kauft man nach 10 Besuchen Waren im Wert von 100 €, steht der mögliche Gewinn für einen Angreifer in keinem Verhältnis zum Aufwand. Werden nach 5 Besuchen Waren im Wert von 50.000 € gekauft, empfiehlt es sich für den Nutzer, den Kontakt zusätzlich zu verifizieren (den Aufwand sollte er aber auch im X.509-Schema bei diesen Größenordnungen treiben).

Die Speicherung jeder fremden EI in einer Datenbank auf dem Nutzersystem gespeichert ist möglich, da selbst die von einem Nutzer beim Surfen im Internet angesprochenen EI im IT-Maßstab relativ begrenzt sind. Anwendungsabhängig kann die Speicherung auf bestimmte Umstände begrenzt werden (siehe 4.1).

> Für viel besuchte Webserver ist eine Speicherung jedes Kontakts unwichtig, es sei denn, der Serverbetreiber wünscht dies aus anderen Gründen. Die Speicherung für Identifizierungszwecke kann auf das Einrichten eines Kundenkontos beim Betreiber beschränkt werden.

4.3 Vollständige Authentifizierung

Die Anzahl der Kontakte, die bereits bei den ersten Datenaustauschen eine definitive Authentifizierung benötigen, ist relativ begrenzt. Die ausgetauschten EI können auf einem unabhängigen Weg verifiziert und in der Datenbank als „verifiziert“ markiert werden. In vielen Fällen beschränkt sich dieser Vorgang auf die Einrichtung eines Kontos

bzw. Inbetriebnahmen, d.h. die definitive Authentifizierung ist kein die Nutzerfreundlichkeit einschränkender Vorgang.

> Beispiel Internetbanking. Im Rahmen der Konteneinrichtung erhält der Kunde die notwendigen Daten, um sich in seinem Konto einzuloggen. Diese beinhalten die Verifizierung der Bank-EI. Im Gegenzug kann der Kunde zur Erleichtung des Einloggens seine EI der Bank in einem ähnlichen Verfahren bekannt machen, so dass die Name/Kennwort-Kombination entfällt.

4.4 Authentifizierung als Dienstleistung

Im Sinne des X.509-Schemas kann die Verifizierung einer EI auch als Dienstleistung durch eine Authority angeboten werden. Da die EI selbstsigniert sind, fällt eine Signatur durch die Authority aus. Die EI können von der Authority aber durch einen gesicherten LDAP-Dienst ausgeliefert werden (bei PGP-Zertifikaten üblich und eingeführt).

Die Dienstleistung kann Widerufslisten (CRL) und OCSP-Dienste (Online Certificate Status Protocol) umfassen. Diese Dienstleistungen können mit den Standardprozeduren für X.509 angeboten werden.

Datenfelder für die notwendigen Informationen über die Authority sind in den X.509-Extensions vorhanden bzw. anlegbar, ohne die Kompatibilität einzuschränken. Es ist jedoch anzumerken, dass hierdurch die Problematik der X.509-Zertifikate (welches Vertrauen verdient eine Authority) wieder eingeführt wird.

In einigen Anwendungsbereichen sind auch Web-of-Trust-Dienste denkbar, d.h. eine EI wird durch andere Nutzer signiert und ein Nutzer vertraut einer EI, weil der einem der Signierer vertraut. Das Schema ist im PGP-Bereich eingeführt und kann ggf. dort entlehnt werden.

Solche zusätzlichen Mechanismen sind nur mit einigem Softwareaufwand zu realisieren, der Nutzen scheint uns darüber hinaus gering bis kontraproduktiv, da der Nutzer wieder teilweise aus seiner Verantwortung entlassen wird. Wir gehen daher (vorläufig) nicht weiter auf diese Optionen ein.

5 Lifecycle von EI

5.1 Standard

Die Inhalte einer EI können sich bei konstantem **UniqueIdentifier** aus unterschiedlichen Gründen ändern:

- Änderung wesentlicher Daten der Inhaber (z.B. Name, Anschrift, E-Mail-Adressen usw.),
- Kompatibilitätsanpassungen an das X.509-Schema,
- Erweiterungen/Änderung des EI-Schemas (z.B. lokale ↔ globale EI, andere Zuweisung einer lokalen EI),
- Berücksichtigung neuer technischer Erkenntnisse (z.B. Änderung der Algorithmen oder Schlüssel).

Die EI erhält mit Ausstellung die Seriennummer 1. Bei jeder Änderung wird die Seriennummer inkrementiert. Das Feld **validFrom** enthält das Datum der Änderung. Die alte EI-Version bleibt gültig, da sie kein Ablaufdatum enthält. Der Inhaber behält alle EI-Versionen in seiner Datenbank. Für den Umgang mit den EI-Versionen gelten folgende Regeln:

- Der Empfänger identifiziert eine übertragene EI anhand des Fingerprints (Hashwert; Verhindern von Fälschungen).

- Wird eine Abweichung festgestellt, muss die übertragene EI eine höhere Seriennummer und ein späteres Gültigkeitsdatum aufweisen. Ältere EI werden nicht akzeptiert, eine geänderte EI mit gleicher Seriennummer wie die vorhandene wird als Täuschungsversuch gewertet.
- Ist die erhaltene EI neuer, sendet der Empfänger die Seriennummer der bei ihm gespeicherten EI an den Absender. Der Absender sendet im Gegenzug alle kompletten EI ab der vom Empfänger vorgelegten Seriennummer bis zur neuen. Der Besitz des privaten Schlüssels für jede EI wird Challenge-Response-Verfahren nachgewiesen.
- Bei Fehlerfreiheit des Nachweises wird die alte EI beim Empfänger durch die neue ersetzt.

Das Verfahren verifiziert automatisch die geänderte EI. Da die Kenntnis der privaten Schlüssel für sämtliche EI ab der bekannten notwendig ist, kann ein Angreifer weder eine fremde EI übernehmen noch eine DoS-Attacke zum Ungültigwerden einer EI durchführen.

5.2 Fehlerfälle

Fehlerfälle im Lifecycle einer EI sind

- Verlust des privaten Schlüssels,
- Kompromittierung des privaten Schlüssels.

Bei **Verlust** des privaten Schlüssels kann ein erfolgreiches Upgrade auf eine neue EI-Version nach 5.1 nicht durchgeführt werden. Die EI wird durch eine neue EI (einschließlich neuem **UniqueIdentifier**) ersetzt, die alte EI wird nicht mehr verwendet. Bestehende Vertrauensverhältnisse bei Kontakten sind neu aufzubauen.

Bei **Kompromittierung** des privaten Schlüssels kann der EI-Inhaber nach 5.1 eine neue EI-Version mit geändertem privaten Schlüssel generieren und durch seine Kontakte durch Verbinden mit der neuen Version upgraden.

- Wird die neue EI-Version vom Kommunikationspartner akzeptiert, ist die Sicherheit aufgrund der Regeln nach 5.1 wiederhergestellt. Der Angreifer, der im Besitz des alten Schlüssels ist, kann damit zumindest bei diesen Kontakten nichts mehr anfangen.
- Wird die neue EI-Version vom Kommunikationspartner nicht akzeptiert, hat der Angreifer bereits selbst eine neue Version dort verifiziert. Die EI ist folglich erfolgreich gestohlen worden, und der Kommunikationspartner ist auf anderem Weg davon in Kenntnis zu setzen.

Die Vorgehensweise und die Konsequenzen entsprechen denen kompromittierter Name/Kennwort-Kombinationen.

Im Sinne des X.509-Schemas kann die Ausgabe einer neuen EI auch als Dienstleistung durch eine Authority angeboten werden. Zur reinen Kommunikation mit einer kleinen Gruppe möglicher Gesprächspartner ist dies jedoch nicht sinnvoll.

6 Verwaltung

Ein Nutzer verwendet in der Regel verschiedene Geräte: PC, Notebook, Tablet, Handy. Fallweise können sogar Fremdgeräte zum Einsatz kommen: Internet-Cafe, Betriebs-Arbeitsplatz, Freunde. Zu verwalten sind

- die EI des Nutzers,

- die Kontaktdaten und EI der Kommunikationspartner.

Welche Daten gespeichert werden, um die gewünschten Funktionen zu erfüllen, wird im Rahmen der Diskussion der einzelnen Anwendungsbereiche geklärt.

Als Speicherorte der Daten sind zu betrachten:

a) die Speicherung der Daten auf jedem Gerät,

b) die Speicherung der Daten auf nur einem Gerät,

c) die Speicherung auf einem besonderen Token,

d) die Speicherung im Netzwerk

6.1 Speicherung auf jedem Gerät

Die Speicherung von Daten auf jedem Gerät bietet sich für den Fall an, dass der Nutzer verschiedene Sicherheitsbereiche auch hardwaremäßig trennen will. Sollen Bankgeschäfte aus Sicherheitsgründen beispielsweise auf den PC beschränkt bleiben, befinden sich nur dort die dazu notwendigen Daten.

Da jedes Gerät zunächst eigene Daten aufzeichnet, kommt es zu unterschiedlichen Bewertungen der Vertrauenswürdigkeit einzelner EI. Synchronisierungen zwischen den Geräten (beispielsweise abgesichert durch lokale EI) sind möglich, widersprechen jedoch der Aufteilung in verschieden Sicherheitsbereiche.

6.2 Speicherung auf einem Gerät

Werden die Daten nur auf einem Gerät gespeichert, kann die Kommunikation beliebig von jedem Gerät in gleicher Weise geführt werden. Das speichernde Gerät muss mit dem anderen Gerät verbunden sein, wenn von einem anderen kommuniziert werden soll. Eine Ankopplung kann per USB, Bluetooth, WLAN, LAN, WAN oder NFC erfolgen und über lokale EI abgesichert werden.

Sicherheitstechnisch ist zu bemerken, dass bei Verlust eines Gerätes die EI-Management-Daten durch Upgrade der lokalen EI abgesichert werden können (Ausschluss des verlorenen Gerätes vom weiteren Datenzugriff). Bis zu diesem Upgrade hätte ein Dieb die Möglichkeit, auf EI-gesicherte Funktionen zuzugreifen, muss dazu aber die lokalen Sicherungen des gestohlenen Gerätes umgehen und herausbekommen, welche EI-gesicherten Funktionen zur Verfügung stehen. Relevante Daten (private Schlüssel, Datenbankinhalte) verlassen allerdings nie das speichernde Gerät, so dass dies Option eine hohe Sicherheit besitzt.

6.3 Speicherung auf einem Token

Die Speicherung auf einem Gerät besitzt den Nachteil, dass das speichernde Gerät jederzeit erreichbar sein muss. Die Speicherung der Daten auf einem mobilen Token, das im Bedarfsfall an ein Gerät angeschlossen wird, beseitigt diesen Nachteil. Ein Token für die Speicherung der N-EI sollte über folgende Eigenschaften verfügen:

- Rechenkapazität zur Durchführung aller Rechnungen mit den Geheimdaten,
- mechanische Sicherheit vor dem gewaltsamen Auslesen der Geheimdaten,
- Schnittstellen für die problemlose Kopplung an die anderen Geräte,

- mechanisch klein, um es immer bei sich zu haben,
- genügen Speicherkapazität.

Eine optimale Möglichkeit wäre eine Konfektionierung als SIM, die in einen zweiten Schacht am Mobiltelefon passt. Alternativ sind USB- oder SmartCard-Anschlüsse denkbar, d.h. eine chipkartenähnliche Lösung, die nicht oder nur unwesentlich größer ist als gebräuchliche Chipkarten. Ungeeignet scheinen berührungslose Techniken aufgrund bekannter Sicherheitsprobleme durch Fremdansprache/Lauschen und der geringeren Rechenleistung.

Wenn der Token selbstsichernd ist, genügt die gebräuchliche PIN/Super-PIN-Absicherung für den Zugriff. Zwischen Token und Gerät wird keine Verschlüsselung eingerichtet. Das erlaubt selbst auf Fremdgeräten (Rechner von Freunden, in Unternehmen, in Internetcafes) eine hohe Sicherheit:

- ✗ Der Fremdrechner erhält keinen Zugriff auf die Geheimdaten, da alle kritischen Operationen auf dem Token ausgeführt werden.
- ✗ Ein entsprechend präparierter Fremdrechner kann zwar die über ihn laufende Kommunikation und die PIN des Tokens ausspähen, aber der Token muss gestohlen werden, um das auszunutzen zu können.
- ✗ Der Nutzer kann nach einem Einsatz auf einem Fremdrechner die PIN ändern.
- ✗ Es können einmal-PIN für die Nutzung auf Fremdrechnern vorgesehen werden, die nach Gebrauch nicht mehr gültig sind, so dass auch ein Diebstahl nichts nützt.

Die Token-Lösung wird aus sicherheitstechnischen Gründen als optimale Lösung betrachtet, da bei Speicherung auf Geräten in jedem Fall hinreichend sichere und damit unhandliche Kennworte notwendig sind. Die Verwendung guter Kennworte liegt aber oft außerhalb der Selbstdisziplin der Nutzer.

6.4 Cloud-Verwaltung

Auf einem Token steht möglicherweise nicht genügend Speicherplatz für die Verwaltung der Kontaktdaten zur Verfügung, da EI und Buchführung einigen Platz erfordern. Ein Token erlaubt jedoch eine Ergänzung durch eine Cloud-Lösung:

- Die Kommunikation zwischen Cloud-Server und Token ist zwischen diesen verschlüsselt; das vermittelnde Gerät kann nicht mitlesen (insbesondere kein Fremdrechner).
- Das Token meldet sich via benutztem Gerät beim Cloudserver mit starker Authentifizierung an (via spezieller EI, denkbar sind aber auch anonyme Name/Kennwortkombinationen, so dass der Cloudserver nicht weiß, welche Daten zu welcher EI gehören) und erhält damit Zugriff auf seine Daten.
- Das Token berechnet aufgrund des Kontaktwunsches einen Zugriffsschlüssel für die betreffende EI/Buchführung. Der Zugriffsschlüssel ist nicht mit den echten Kontaktdaten verknüpfbar, d.h. der Cloud-Server weiß nicht, was er verwaltet.
- Die vom oder zum Cloud-Server gesandten Daten sind ebenfalls hart verschlüsselt.