5. Signaturen und Zertifikate

Folgende Sicherheitsfunktionen sind möglich:

- Benutzerauthentikation: Nachweis der Identität des Benutzers
- Datenauthentikation: Nachweis, von wem die Daten stammen
- Datenintegrität: Schutz der Daten gegen Manipulation
- Nachweisbarkeit: Berechtigte Dritte können Authentizität & Integrität von Daten überprüfen.
- Digitale Unterschrift Verträge; 'Behördengänge' über das Internet
- Zahlungsverkehr Beglaubigung von Transaktionen

Grundlage sind kryptographische Algorithmen, wobei der Schlüssel bzw. das Zertifikat z.B. in einer Chipkarte gespeichert ist.

1. Digitale Signatur, elektronische Unterschrift (digital signature)

Digitale Signaturen (Unterschriften) sollen wichtige Eigenschaften herkömmlicher (handschriftlicher) Unterschriften nachbilden.

Wir fordern

Echtheitseigenschaft: Das Dokument kann nur vom Unterzeichner stammen. Die Unterschrift muss auf dem Dokument stehen.

Identitätseigenschaft: Die Unterschrift ist ein persönliches Merkmal einer **einzigen** Person.

Abschlusseigenschaft: Die Unterschrift signalisiert die Verbindlichkeit des Dokuments.

Warneigenschaft: Unterschrift wurde nicht übereilt (automatisch), sondern durch eine persönliche Aktivität des Unterzeichners erteilt (z.B. erst nach Eingabe einer PIN).

Verifikationseigenschaft: Dritte Personen (z.B. ein Gericht) müssen die Unterschrift verifizieren können.

Bei der digitalen Signatur kommt es nicht auf die Geheimhaltung der Nachricht an. Diese wird unverschlüsselt übertragen.

Die Signatur schützt die Daten (die Nachricht) vor unbemerkter Veränderung, indem sie die Daten an den Eigentümer eines privaten Schlüssels bindet. Der Schlüssel entspricht der (hoffentlich) einzigartigen Möglichkeit der persönlichen handschriftlichen Unterschrift.

Ein **asymmetrisches Signatursystem** nach dem RSA-Verfahren

Eine Schlüsselvergabestelle berechnet einen **geheimen** und einen **öffentlichen** Schlüssel für die Person A.

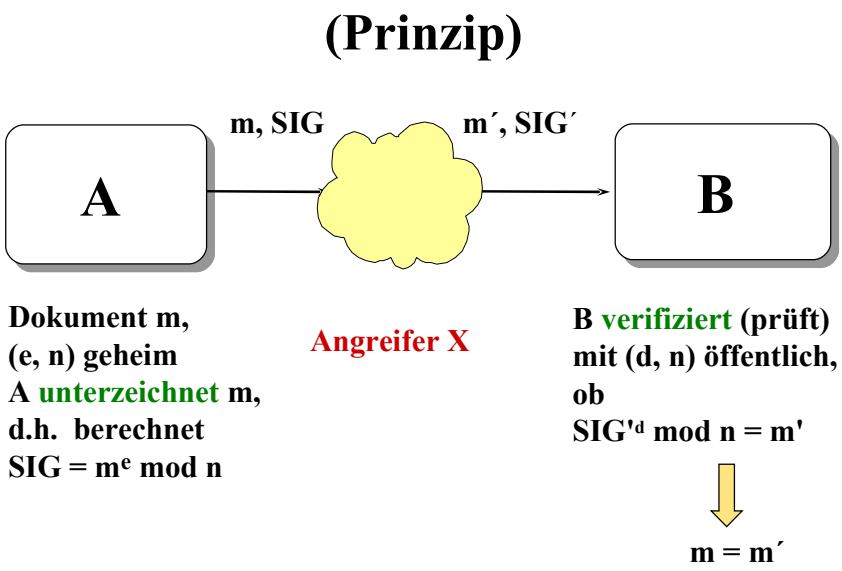

Abbildung 11. Elektronische Unterschrift

Die Unterschrift

Person A setzt zum Verschlüsseln seine **geheime Schlüsselzahl e** (encrypt) ein, der Empfänger verfügt über den **öffentlichen** Schlüssel d.

Die Nachricht m wird chiffriert: $\mathbf{SIG = m^e \bmod n}$

SIG heißt **Signatur** oder **Unterschrift**.

Die Verifikation

Der Empfänger vergleicht, ob $SIG'^{d} \bmod n = m'$.

Bei Gleichheit kann nur der Absender den geheimen Schlüssel besessen haben.

Eigenschaften

- A kann als Einziger die gültige Signatur erzeugen (Identitätseigenschaft).
- B kann die Signatur verifizieren, aber nicht fälschen (Echtheitseigenschaft).
- Jeder Dritte, der d kennt, kann verifizieren, ob die Nachricht von A stammt (Verifikationseigenschaft).

Bemerkungen

1. Falls es nur auf die Unterschrift ankommt, braucht man m überhaupt nicht zu übermitteln, da sich m aus SIG berechnen lässt.

2. Die Unterschrift hängt von jedem Bit der Nachricht m ab. Bei der handschriftlichen Art ist die Unterschrift lediglich auf dem Papier **hinzugefügt**.

Aufgaben

1. Ist eine digitale Signatur unter einem Dokument m1 auf ein anderes Dokument m2 übertragbar?
   Ist eine elektronische "Blanko"-Unterschrift auf einem leeren Blatt Papier möglich?

2. Gegeben sind p = 11, q = 13 und das geheime e = 7.
   Berechne schriftlich zur Nachricht m = 25 (26, 6) die Signatur SIG.

3. Ein Dokument m wurde mit einem RSA-Schlüssel unterschrieben (p = 7, q = 13, d = 5).
   Mit welchem Schlüssel wurde m = 12 unterschrieben, wenn SIG = 38 ist?

4. Prüfe schriftlich die folgenden Signaturen auf Korrektheit (p = 7, q = 13, d = 5):
   m = 10, SIG = 81 m = 11, SIG = 72

Signatur und Fingerprint

Da asymmetrische Verfahren rechenintensiv und damit relativ langsam sind, wird statt der gesamten Nachricht lediglich ein sog. **Fingerprint** signiert.

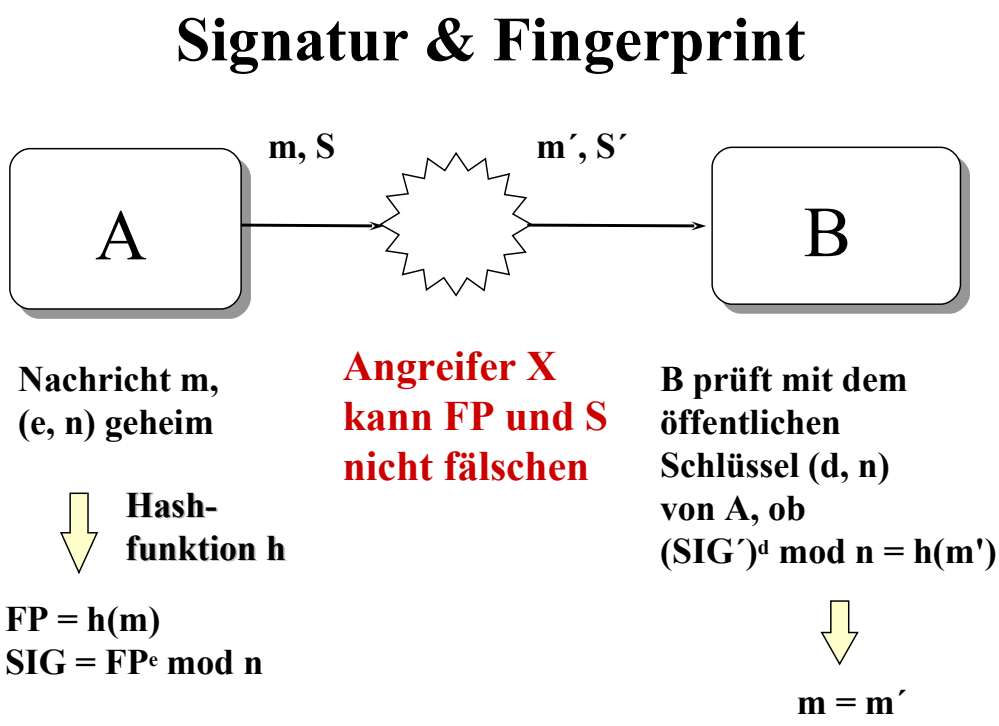

Abbildung 12. Signatur und Fingerprint

Der Absender bildet aus der Nachricht m ein Extrakt (Hash) mittels einer **Hashfunktion** h. Ein typischer Algorithmus hierfür ist der MD5-Algorithmus.[1]

Mittels einer Hash-Funktion wird der **Fingerprint** berechnet:

$$FP = h(m) \quad .$$

[1] MD5-Algorithmus: Der MD5-Algorithmus (Message-Digest 5)
Der MD5-Algorithmus dient zur Erzeugung einer fälschungssicheren Textprüfsumme (auch MAC genannt). Aus einer Nachricht wird anhand der Hash-Funktion ein Hash-Code mit der Länge von 128 Bits bestimmt. Diese Einwegfunktion bildet eine große Menge von Funktionsargumenten möglichst gleichmäßig auf eine vergleichsweise kleine Menge von Funktionswerten ab. Der Hash-Code ist eine Zahl, die bei der kleinsten Veränderung des Ursprungstextes nicht mehr übereinstimmen würde, d.h. wenn der Empfänger den Hash-Code des Urpsrungstextes hat, kann er damit die Unversehrtheit des Inhaltes der Nachricht überprüfen.

Der Wert FP ist relativ kurz (z.B. 128 Bit) und wird als Fingerabdruck (Fingerprint, Kryptoprüfsumme) bezeichnet. Es darf nicht möglich sein, aus dem Hashwert auf die Nachricht m zu schließen (Einweg-Hashfunktion).

Die Signatur S wird durch Verschlüsselung des Fingerprints mit dem geheimen RSA-Schlüssel erzeugt: $\mathbf{S = FP^e \bmod n}$.

Vorteil: Die Signatur des Fingerprints erfordert weniger Rechenzeit als die Signatur der gesamten Nachricht.

B und Mister X können die Signatur prüfen, aber nicht fälschen, da sie nicht im Besitz von (e, n) sind.

Verschlüsseln und Signieren

Kombiniert man den Einsatz des geheimen Schlüssels (e_A, n_A) des Absenders mit dem öffentlichen Schlüssel (d_E, n_E) des Empfängers, kann man sowohl verschlüsseln als auch signieren.

Asymmetrische Verfahren können außer zum Signieren auch gleichzeitig zum Verschlüsseln benutzt werden, indem vor dem Verschlüsseln die Signatur an die Nachricht angehängt wird.

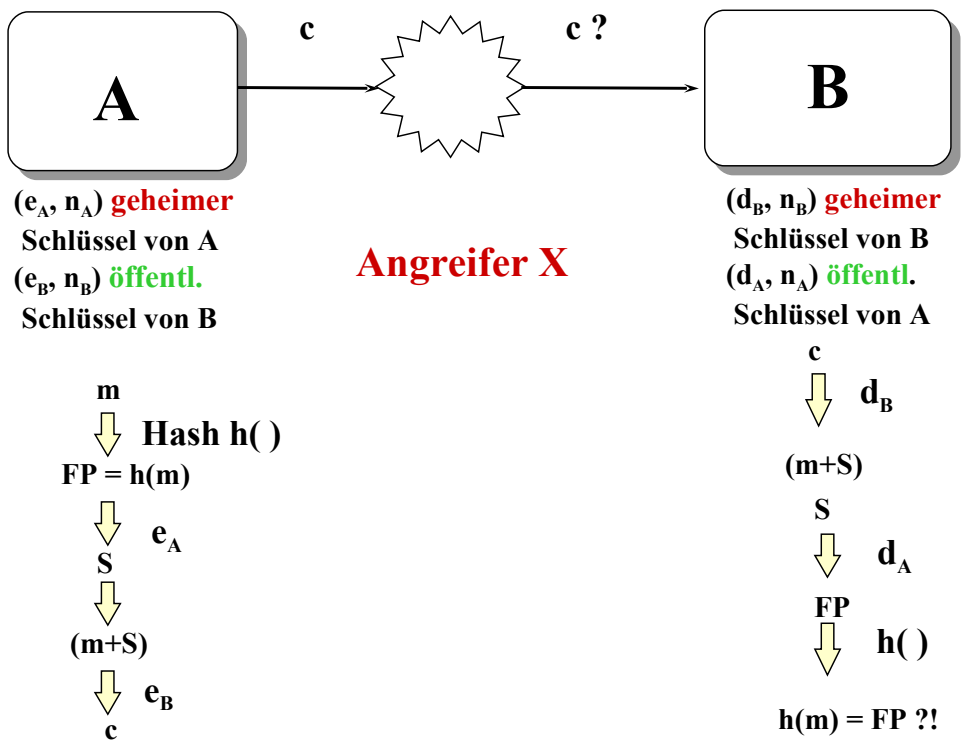

Abbildung 13. Signatur und Verschlüsselung

Absender A (möchte m signieren & senden)	**Empfänger B besitzt**
(e_A, n_A) **geheimer** Schlüssel von A	(d_A, n_A) öffentl. Schlüssel von A
(e_B, n_B) öffentl. Schlüssel von B	(d_B, n_B) **geheimer** Schlüssel von B

- A bildet den Fingerprint aus der Nachricht m: FP = h(m)
- A erzeugt aus FP und e_A die Signatur S: $S = FP^{e_A} \bmod n_A$
- A hängt an m die Signatur S an: (m + S)
- A verschlüsselt (m + S) mit dem öffentlichen Schlüssel von B und übermittelt c: $c = (m+S)^{e_B} \bmod n_B$

- B entschlüsselt c mit seinem geheimen d_B wieder zu (m + S) $c^{d_B} \bmod n_B = m + S$.
  Die Nachricht m liegt im Klartext vor und die Signatur S ist „frei“.
- E entschlüsselt die Signatur S mit dem öffentlichen Schlüssel von A zu $S^{d_A} \bmod n_A = FP$
- E generiert mit der Hashfunktion h den Fingerprint von m neu und vergleicht, ob h(m) = FP ist.

Eigenschaften

- Bei Gleichheit der Fingerprints kann B davon ausgehen, dass die Nachricht m nicht modifiziert wurde.
- Die Nachricht m kann nur von A stammen, da sie mit dem geheimen Schlüssel e_A von A verschlüsselt wurde.
- Die Nachricht kann nur B lesen, da sie mit dem öffentlichen Schlüssel von B verschlüsselt wurde und **nur** mit dem privatem Schlüssel von B in den Klartext überführt werden kann.

Was kann der Mister X?

Mister X kann mithören und die Signatur prüfen.

Er kann aber einen sog. **Man-in-the-middle-Angriff** starten. Dazu klinkt sich X in die Kommunikation zwischen A und B ein: Dies passiert z.B. bei der Übertragung des öffentlichen Schlüssels.
Man könnte glauben, dass die Übertragung des öffentlichen Schlüssels keine Gefahren birgt.

Aber genau hier setzt der Angriff an. Da A den potenziellen Empfängern seinen öffentlichen Schlüssel (d_A, n_A) übermitteln muss, fängt X diesen ab.

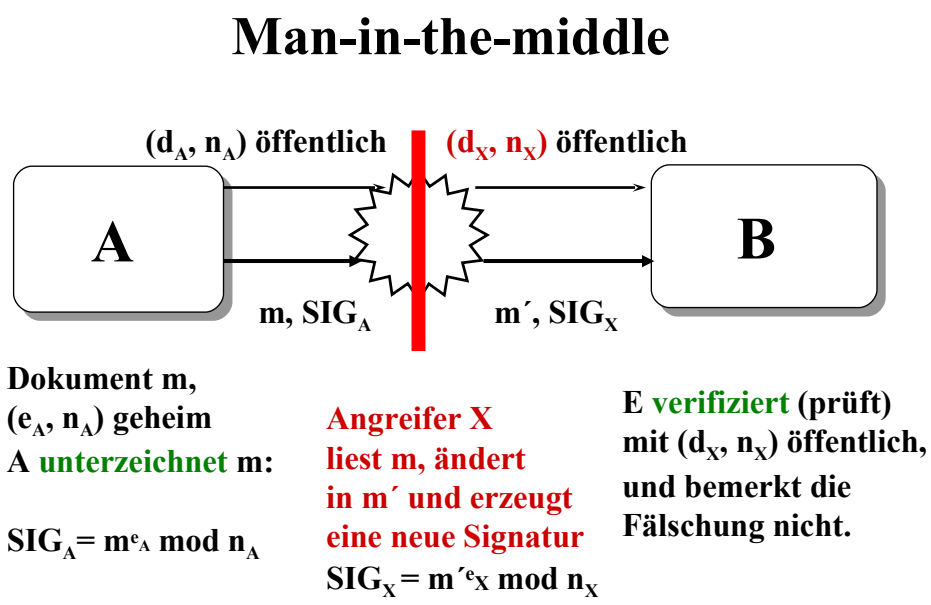

Abbildung 14. Man-in-the-Middle

Wenn der Empfänger B den Schlüssel (d_A, n_A) anfordert, wird dieser von X abgefangen und durch seinen eigenen öffentlichen Schlüssel (d_X, n_X) ersetzt.

B glaubt, dass er den öffentlichen Schlüssel von A besitzt:
A unterschreibt und sendet eine Nachricht an E. X liest sie mit, ändert sie nach Belieben und verschlüsselt sie mit seinem Schlüssel (e_X, n_X). B kann die Signatur überprüfen und ahnt nicht, dass er gefälschte Nachrichten erhält.

Um diesen Angriff zu verhindern, muss der Absender A **persönliche Informationen** in seine Nachricht bzw. die Signatur einfügen. Dies wird durch **Zertifikate** erreicht.

2. *Zertifikate (Public-Key-Zertifikate)*

http://www.rz.uni-passau.de/rank/ca/doc/certificate.html
www.inf.fu-berlin.de/lehre/SS04/SySi/.../KryptografieTeil2.pdf

Zertifikate gewährleisten die Authentizität eines öffentlichen Schlüssels, indem sie den öffentlichen Schlüssel an die Identität des Inhabers (Person oder dessen Computer) binden.
Digitale Zertifikate sind mit amtlichen Ausweisdokumenten vergleichbar: Eine vertrauenswürdige Instanz (CA, Certification Authority) bestätigt die Identität einer Person oder eines Computers.

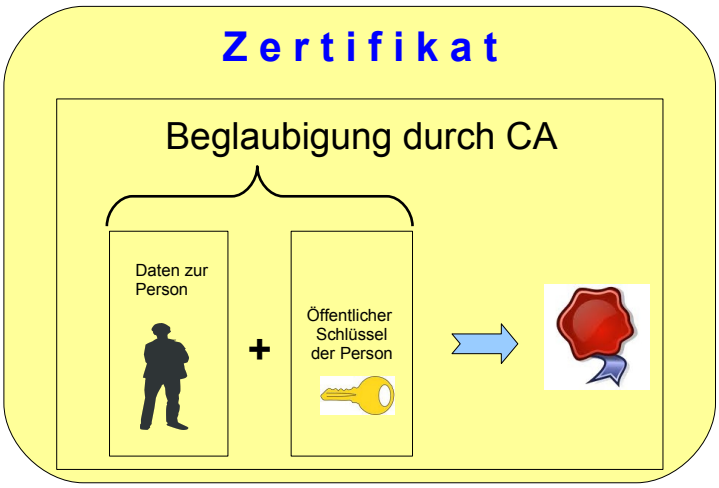

Abbildung 15. Zertifikat

Ziele

- Der öffentliche Schlüssel muss eindeutig an eine bestimmte Person gebunden sein.
- Die Clientsoftware muss gültige Zertifikate erkennen und gefälschte zurückweisen.

Nach dem Signaturgesetz muss ein Zertifikat folgende Angaben enthalten:

- Daten des Inhabers und sein öffentlicher Schlüssel
- **Signatur über die Daten und den Schlüssel**
- Daten der Zertifizierungsstelle, öffentlicher Schlüssel der CA und Signatur darüber
- Angaben über eine eventuelle Begrenzung auf bestimmte Anwendungen

- Angaben zu den Algorithmen, mit denen die öffentlichen Schlüssel benutzt werden können
- Beginn und Ende der **Gültigkeit des Zertifikats,**
- fortlaufende Nummer des Zertifikats

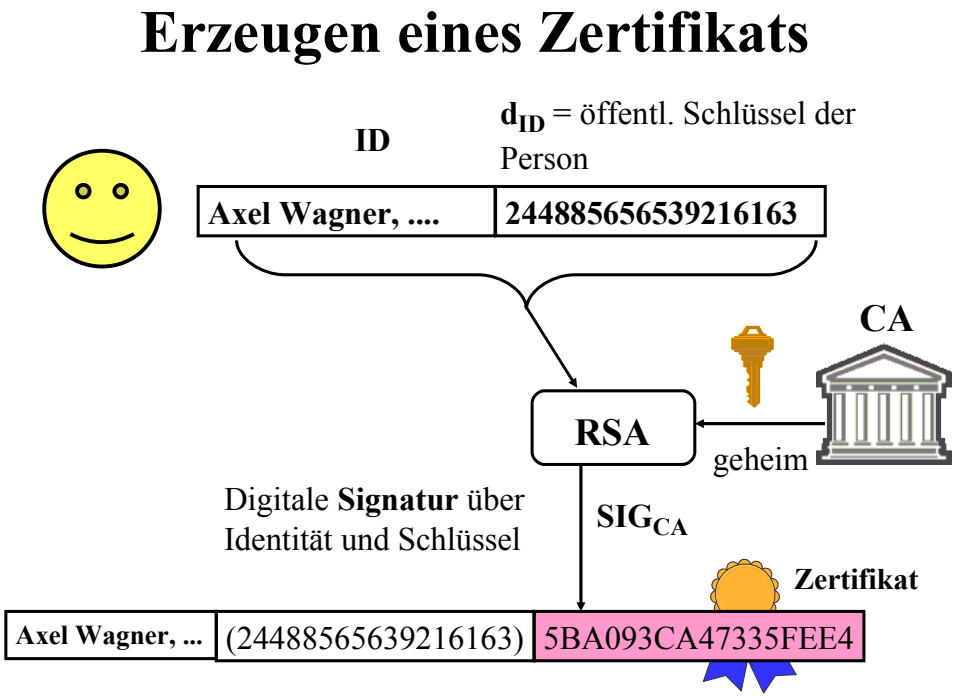

Abbildung 16. Zertifikat

Im Beispiel ist **ID = Axel Wagner** und der öffentliche Schlüssel der Person ist d_{ID} = **244885656539216163** .

Die Signatur wird von einer vertrauenswürdigen dritten Partei (Zertifizierungsstelle, Certification Authority CA) mit dem geheimen Schlüssel der CA erzeugt. Die Zertifikats-Signatur kann mit Hilfe des öffentlichen Schlüssels der CA überprüft werden.

Die CA signiert die Datenfolge ID | d_{ID} (an ID wird z.B. der Schlüssel angehängt).

Vereinfacht ausgedrückt:

Zertifikat = Name der Person + sein öffentlicher Schlüssel + SIG_{CA}

Anmerkungen

- Die Signatur wird von einer Zertifizierungsstelle erzeugt und ist z.B. auf einer Smart-card gespeichert.
- Die Gültigkeit eines Zertifikats ist auf eine festgelegte Zeit beschränkt.
- Das Trustcenter muss alle ausgestellten Zertifikate in einem zugänglichen Verzeichnis veröffentlichen.
- Die Zertifikate müssen online nachprüfbar sein.

Das Zertifikat wird über einen anderen Kanal (etwa Postident-Verfahren) von der CA an die zugehörige Person übermittelt.

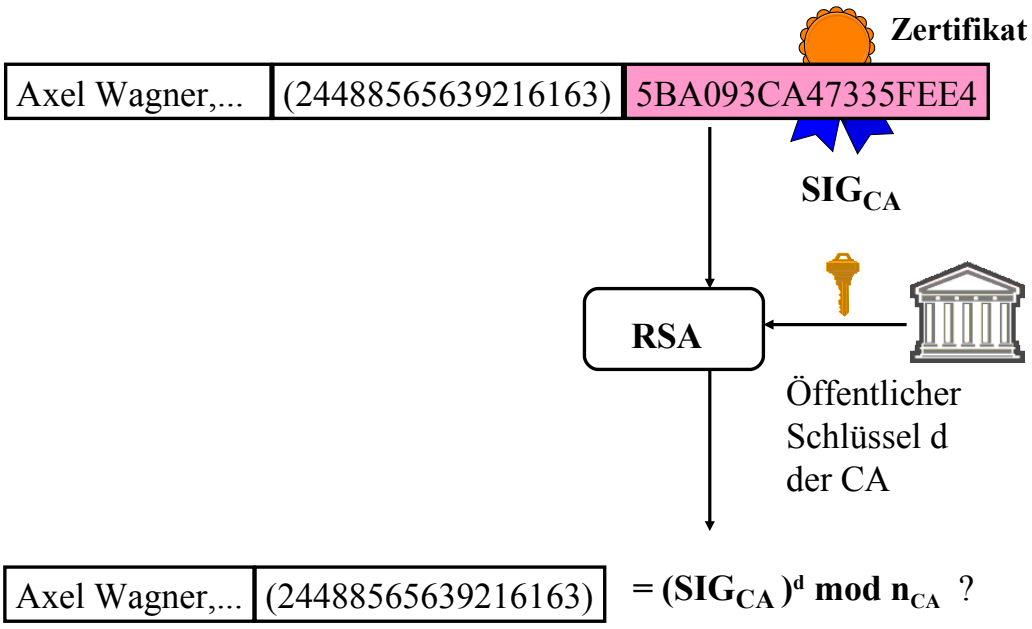

Abbildung 17. Prüfen des Zertifikats

Die Clientsoftware eines Empfängers weist solche Zertifikate zurück, bei denen die CA-Signatur bei Überprüfung mit dem öffentlichen Schlüssel der CA fehlschlägt.

3. Widerruf von Zertifikaten

Der geheime Schlüssel der Person A wurde gestohlen oder versehentlich offen gelegt. Beim Trustcenter ist das Zertifikat aber immer noch abrufbar und damit offiziell gültig! Und Sie können zudem nicht alle Rechner benachrichtigen, auf denen eine Kopie Ihres Zertifikats gespeichert ist.

Abhilfe

In sog. **Zertifikatssperrlisten** werden alle ungültig gewordenen Zertifikate aufgeführt. Wenn A nun eine Nachricht an eine Person mit gesperrtem Zertifikat sendet, sollte seine Clientsoftware prüfen, ob sich das Zertifikat des Empfängers in der Sperrliste befindet.

Webbrowser sind so eingestellt, dass Zertifikate automatisch online überprüft werden.

4. Überprüfen von *Zertifikaten*

Um ein ausgestelltes Zertifikat überprüfen zu können, benötigt man den öffentlichen Schlüssel der Zertifizierungsstelle. Falls dieses Zertifikat nicht von der höchsten Zertifizierungsinstanz stammt, ist es von einer höheren CA unterschrieben.

Am Ende dieser Hierarchie steht eine Root-CA, die ihr Zertifikat selbst signiert.

Da das Internet zunächst keine Sicherheit gegen die Verfälschung übertragener Daten bietet, muss die Korrektheit des Stammzertifikats überprüfbar sein.

Dies geschieht dadurch, dass der zum Zertifikat gehörende Fingerabdruck über ein anderes Medium zur Verfügung gestellt wird.