Elektronische Signaturen

Oliver Gasser

3. Juni 2009

1 Motivation

Vorweg: Die Begriffe „elektronische Signaturen“ und „digitale Signaturen“ werden meist synonym verwendet. Das ist aber nicht ganz korrekt, da „elektronische Signaturen“ explizit von den Juristen gewählt wurde, um das Signieren nicht auf eine Klasse von kryptographischen Verfahren zu beschränken, der „digitalen Signatur“, sondern allgemeine Anforderungen zu stellen.

Wozu braucht man aber elektronische Signaturen? Elektronische Signaturen dienen dazu, digitale Dokumente einer Person zuzuordnen. Um den Rechtsverkehr zu modernisieren, ist es beispielsweise nötig, Verträge übers Internet abschließen zu können. Konkret heißt das, dass die elektronische Signatur der handschriftlichen Unterschrift gleichgestellt wird.

2 Eigenschaften und Anforderungen

Damit die elektronische Signatur einer handschriftlichen Unterschrift gleichgesetzt werden kann, muss sie dieselben Eigenschaften wie diese haben. Unterschriften dienen der Identifikation der Unterzeichners, der Echtheit eines Dokuments, dem Abschluss eines Dokuments und der Warnung an den Unterzeichner. Um diese Eigenschaften zu erfüllen, müssen folgende Anforderungen an die elektronische Signatur gestellt werden:

- Sie muss die **Identität** des Signierenden zweifelsfrei bestimmen.
- Sie darf **nicht wiederverwendbar** sein, da ansonsten jemand anderes die Signatur an ein selbst erstelltes Dokument anhängen kann.
- Das signierte Dokument darf **nicht veränderbar** sein oder eine Veränderung muss erkennbar sein, indem beispielsweise die Signatur ungültig wird.
- Der Unterzeichner kann die Signatur **nicht zurückweisen**, sofern der private Schlüssel ausschließlich in dessen Besitz ist.

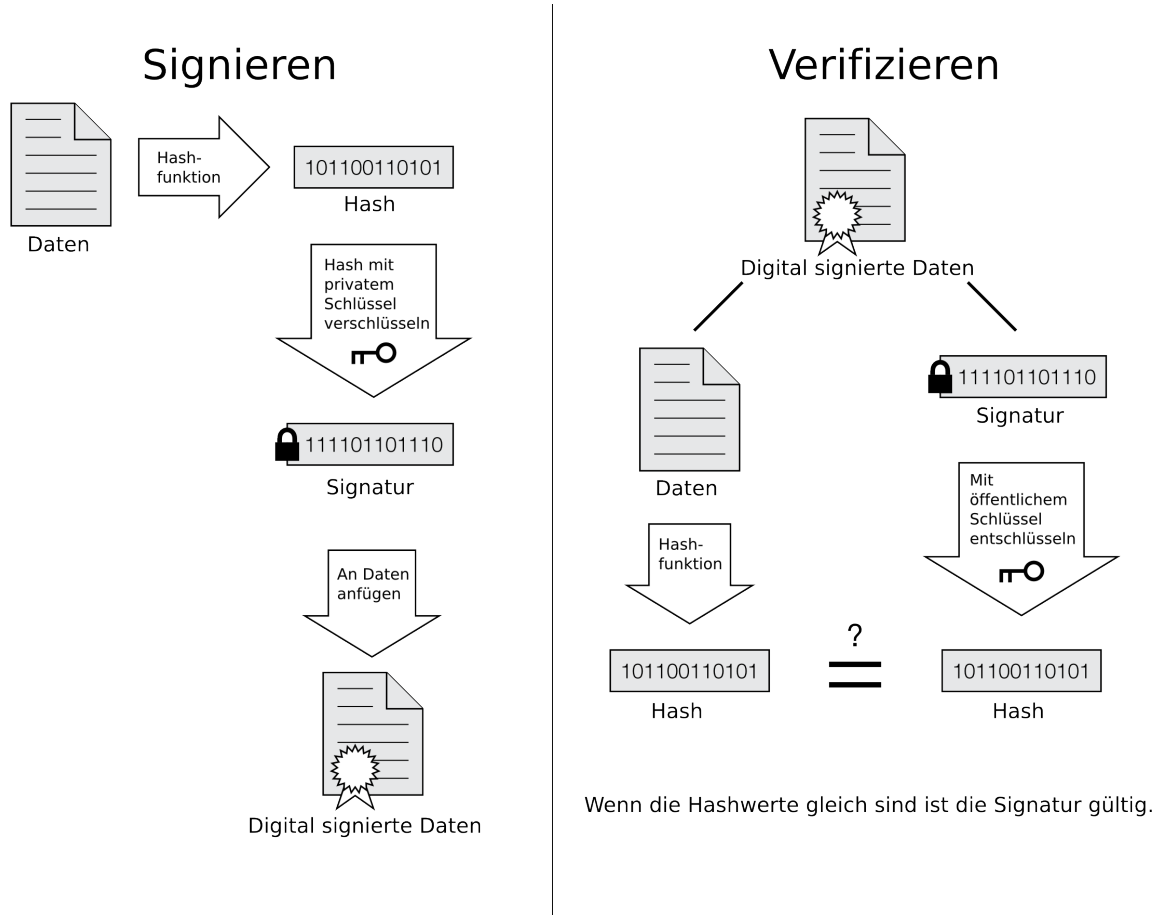
Eine weitere Anforderung ist die eindeutige Zuordbarkeit des öffentlichen Schlüssels zum privaten Schlüssel des Signierenden. Ist dies nicht möglich, so kann der Signierende die Signatur erfolgreich zurückweisen.

3 Geschichte

Da wir nun ein Verständnis dafür haben, was eine elektronische Signatur leisten muss, ist jetzt ein kurzer Überblick über die geschichtliche Entwicklung derselben angesagt. 1977 entwickeln Rivest, Shamir und Adleman am MIT das RSA-Kryptosystem. Mit ihm ist es auch möglich digitale Dokumente zu signieren. 1984 wurde von Goldwasser, Micali und wiederum Rivest das GRM-Signaturverfahren vorgestellt, das es noch schwieriger machte, eine Signatur zu fälschen. Im selben Jahr wurde von Taher Elgamal das ElGamal-Signaturschema eingeführt. In der Praxis findet dieses sehr wenig Verwendung. Es ist aber insofern wichtig, als dass der erstmals im Jahre 1991 vorgestellte vom NSA entwickelte Digital Signature Algorithm auf ElGamal basiert.

4 Signatur und Verifizierung

Wie geht eigentlich der Vorgang des Signierens und Verifizierens vor sich? Das lässt sich am einfachsten Anhand einer Grafik erklären:



Wenn die Hashwerte gleich sind ist die Signatur gültig.

Um eine Signatur für ein Dokument zu erstellen, wendet man auf das Dokument eine Hash-Funktion an. Auf den Hash-Wert wird anschließend der private Schlüssel angewendet und die daraus resultierende Signatur wird dem Dokument angehängt.

Um die Signatur eines Dokuments zu verifizieren trennt man zuerst die Dokumentdaten von der Signatur. Nun wendet man auf die Signatur den öffentlichen Schlüssel an und auf die Dokumentdaten die Hash-Funktion. Sind beide Ergebnisse identisch, so ist die Signatur gültig.

5 Digital Signature Algorithm

Um dieses theoretische Konzept greifbar zu machen, wollen wir das Signieren und Verifizieren anhand des DSA durchprobieren. Der DSA ist im Digital Signature Standard des NIST enthalten und basiert auf dem Problem des diskreten Logarithmus in endlichen Körpern.

5.1 Parameterwahl

Vor dem eigentlichen Signieren müssen folgende Parameter gewählt bzw. berechnet werden:

p Wähle eine Primzahl p der Länge 512-1024 Bit.

q Wähle eine Primzahl q der Länge 160 Bit, wobei q ein Primfaktor von $p-1$ sein muss.

h Wähle eine Zahl h, sodass $1 < h < p-1$ und $h^{\frac{p-1}{q}} \bmod p > 1$.

g Berechne $g = h^{\frac{p-1}{q}} \bmod p$.

x Generiere eine Zufallszahl x, wobei $0 < x < q$ gilt.

y Berechne $y = g^x \bmod p$.

Der **öffentliche Schlüssel** zur Verifikation ist (p, q, g, y), der **private Schlüssel** zur Signaturerstellung ist x.

5.2 Signieren

Nun wollen wir eine Nachricht M signieren. Dazu wenden wir die Hashfunktion SHA-1 auf M an.

k Generiere eine Zufallszahl k, wobei $0 < k < q$ gilt.

r Berechne $r = (g^k \bmod p) \bmod q$.

s Berechne $s = (k^{-1} \cdot (\text{SHA-1}(M) + x \cdot r)) \bmod q$.

Die Signatur für die Nachricht M ist nun das Paar (r, s).

Hinweis: k^{-1} bezeichnet das negative Inverse zu k modulo q. Das heißt: $(k^{-1} \cdot k) \bmod q = 1$ und $0 < k^{-1} < q$.

5.3 Verifizieren

Nun wollen wir die Signatur (r, s), die mit der Nachricht M verschickt wurde, mit dem öffentlichen Schlüssel verifizieren: Zuerst überprüfen wir, ob $0 < r < q$ und $0 < s < q$. Falls eine der Bedingungen verletzt ist, wird die Signatur zurückgewiesen.

w Berechne $w = s^{-1} \bmod q$.

u_1 Berechne $u_1 = ((\text{SHA-1}(M)) \cdot w) \bmod q$.

u_2 Berechne $u_2 = (r \cdot w) \bmod q$.

v Berechne $v = ((g^{u_1} \cdot y^{u_2}) \bmod p) \bmod q$.

Falls v gleich r ist, dann ist die Signatur gültig, ansonsten nicht.

6 Sicherheit

Die Berechnung des privaten Schlüssels kann in zwei diskrete Logarithmus Probleme aufteilen. Das erste kann mit dem sogenannten Zahlkörpersieb-Verfahren, mit dem auch die 663-Bit lange RSA-200-Zahl faktorisiert werden konnte, in der asymptotischen Laufzeit von $O(e^{((C+o(1))\cdot(\log p)^{\frac{1}{3}}\cdot(\log\log p)^{\frac{2}{3}}})$ gelöst werden, das zweite in $O(\frac{\pi * q}{2})$. Falls p beziehungsweise q 1024 respektive 160 Bit lang sind, so benötigt man ungefähr 10^{30} und 10^{24} Schritte. Damit gilt DSA bei genügend großen Primzahlen p und q als sicher.

Vorsicht ist jedoch geboten, wenn man eine Nachricht signiert und auch verschlüsselt und zwar mit demselben Schlüsselpaar. Sei (K_E^A, K_D^A) das Schlüsselpaar von Alice, wobei K_E^A der öffentliche Verifikations- und gleichzeitig Verschlüsselungsschlüssel und K_D^A der private Signatur- und gleichzeitig Entschlüsselungsschlüssel ist. Das Analoge gilt für Bob und Eve. Die Idee hinter diesem Angriff ist, dass die Angreiferin Eve den Empfänger Bob dazu bringt, die Verschlüsselung zu entfernen. Der Ablauf ist wie folgt:

1. Alice (A) sendet eine Nachricht M signiert $D(M, K_D^A) = sig$ und verschlüsselt $E(sig, KBE) = C$ an Bob (B).

2. Die Angreiferin Eve (E) fängt die Nachricht ab und sendet sie anschließend an Bob weiter.

3. Bob entschlüsselt das von Eve geschickte C mit seinem privaten Schlüssel, $D(C, K_D^B) = D(M, K_D^A) = sig$, und verfiziert die Signatur sig mit dem Verifikationsschlüssel von Eve, da er diese ja für den ursprünglichen Absender der Nachricht hält, $E(sig, K_E^E) = M'$. Bob erhält so die in der Regel unsinnige Nachricht M'.

4. Bob sendet nun M' signiert und verschlüsselt (als Empfangsbestätigung) zurück an Eve: $E(D(M', K_D^B), K_E^E) = C'$.

5. Eve entschlüsselt nun die Nachricht mit ihrem privaten Schlüssel, $D(C', K_D^E) = sig'$, verifiziert die Signatur sig' mit dem öffentlichen Schlüssel von Bob, $E(sig', K_E^B) = M' = E(sig, K_E^E)$, wendet erneut ihren eigenen privaten Schlüssel an, $D(M', K_D^E) = sig$ und verifiziert die Signatur sig anschließend mit dem Verifikationsschlüssel von Alice, $E(sig, K_E^A) = M$. Eve erhält so die ursprüngliche Nachricht M.

Dieser Angriff kann leicht dadurch wirkungslos gemacht werden, indem nicht die Nachricht, sondern der Hash-Wert derselben signiert wird. Damit erhält der Angreifer nur den Hash-Wert, der für ihn wegen der Einwegeigenschaft nutzlos ist.

Die „Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen“ veröffentlicht jährlich Empfehlungen für Algorithmen und deren Parameter zur Erstellung von Signaturen und für Hashfunktionen. Außerdem wird ein Ausblick gewagt, wie lange die jeweiligen Verfahren als sicher gelten können. In der Veröffentlichung vom Jahr 2009 wurden die Signaturverfahren RSA, DSA und ECDSA unter Beachtung der richtigen Parameterwahl bis Ende 2015 als sicher eingestuft. Dabei wird bei DSA beispielsweise für die Primzahl p eine Länge von 2048 Bit und für q eine Bitlänge von mindestens 224 gefordert.

7 Rechtlicher Rahmen

Die Signaturrichtlinie der Europäischen Union aus dem Jahre 1999 ist der Ausgangspunkt für die aktuelle Signaturgesetzgebung. Sie zielt auf die Gleichstellung der elektronischen Signatur mit der handschriftlichen Unterschrift. Diese Richtlinie wurde im Rahmen des Signaturgesetzes von 2001 und einer Reihe weiterer Vorschriften in Deutsches Recht gewandelt. Im Signaturgesetz wird zwischen drei verschiedenen Formen der elektronischen Signatur unterschieden:

Allgemeine elektronische Signatur An die einfache elektronische Signatur werden keine besonderen Anforderungen gestellt. So gilt beispielsweise das Anfügen des eigenen Namens unter eine E-Mail als „einfache“ Signatur. Einfache elektronische Signaturen haben deshalb eine geringe Aussagekraft.

Fortgeschrittene elektronische Signatur Eine fortgeschrittene elektronische Signatur hingegen muss mit einem einmaligen Signaturschlüssel erstellt worden sein. Der Signaturersteller muss, etwa durch einen Prüfschlüssel, identifizierbar sein. Außerdem muss ein nachträgliches Verändern der signierten Daten erkennbar sein. Rechtlich gesehen, haben fortgeschrittene elektronische Signaturen eine ähnlich geringe Beweiskraft wie einfache elektronische Signaturen.

Qualifizierte elektronische Signatur Dokumente, die mit einer qualifizierte elektronischen Signatur unterzeichnet wurden, können als einzige der drei die per Gesetz geforderte Schriftform auf Papier ersetzen. Zusätzlich zu den Anforderungen der fortgeschrittenen elektronischen Signatur muss die qualifizierte elektronische Signatur auf einem zum Signaturerstellungszeitpunkt gültigen qualifizierten Zertifikat beruhen und mit einer sicheren Signaturerstellungseinheit (SSEE) erstellt worden sein. Um die Schriftform

durch die elektronische Form zu ersetzen, muss dem elektronischen Dokument der Name des Signierenden und die qualifizierte elektronische Signatur angehängt werden.

8 Anwendung in der Praxis

In der Praxis werden elektronische Signaturen abgesehen von Beispielen wie PGP/GnuPG oder X.509 eher selten angewendet. Pretty Good Privacy (PGP) basiert dabei auf dem sogenannten „Web of trust“. Das heißt, im Gegensatz zu den X.509-Zertifikaten, die meist in Browsern zur Identifikation der Webseiten eingesetzt werden, gibt es keine zentralisierte Zertifizierungsstelle, die Zertifikate ausgibt. Digitale Zertifikate sind dabei strukturierte Daten, die einen öffentlichen Schlüssel einer Person oder einer Organisation zuordnen. Bei PGP können Zertifikate im Gegensatz zu X.509 von mehreren Personen signiert werden und dienen zum sicheren Austausch des öffentlichen Schlüssels (Verifizerungsschlüssel). Es geht sogar soweit, dass man sagen kann: Je mehr Signaturen ein PGP-Zertifikat enthält, desto glaubwürdiger ist es.

9 Fazit und Ausblick

Elektronische Signaturen werden in Form von digitalen Signaturen vor allem als Identifikationsbestätigung beim E-Mail-Versand eingesetzt. Im Leben von „technikaversen“ Menschen haben sie aber eine sehr geringe bis gar keine Bedeutung. In Zukunft sollen mehrere Projekte (beispielsweise elektronischer Personalausweis, De-Mail, elektronische Steuererklärung), die die Verwendung einer digitalen Signatur vorsehen, eingeführt werden. Dadurch soll die Verbreitung von elektronischen Signaturen gefördert, der Zugang für den Bürger erleichtert und Geld in der Verwaltung eingespart werden. Da sich die Einführung dieser Techniken aber schon über Jahre wegen sicherheitstechnischen und finanziellen Problemen verzögert, ist nicht mit einer baldigen hohen Verfügbarkeit von qualifizierten Signaturen und Diensten dafür zu rechnen. Bis diese Technologien von einem Großteil der Bevölkerung akzeptiert und eingesetzt werden, dürfte wohl noch das eine oder andere Jahr vergehen.

10 Weiterführende Literatur

- IT-Sicherheit: Konzepte – Verfahren – Protokolle
  Prof. Dr. Claudia Eckert Oldenbourg Verlag, 5. Ausgabe, 2008
- Applied Cryptography
  Bruce Schneier John Wiley & Sons Verlag, 2. Ausgabe, 1996
- Digital Signature Standard (DSS)
  National Institute of Standards and Technology (NIST) Federal Information Processing Standards Publication 186-2, 27. Januar 2000