Elektronische Signaturen

Oliver Gasser

TUM

3. Juni 2009

Gliederung

Warum digital signieren?

Motivation:

- Digitale Dokumente einer Person zuordnen
- Rechtsverkehr modernisieren
- Gleichstellung mit handschriftlicher Unterschrift

Eigenschaften und Anforderungen

Eigenschaften

- Identifikation
- Echtheit
- Abschluss
- Warnung

Eigenschaften und Anforderungen

Eigenschaften

- Identifikation
- Echtheit
- Abschluss
- Warnung

Anforderungen und Schwierigkeiten

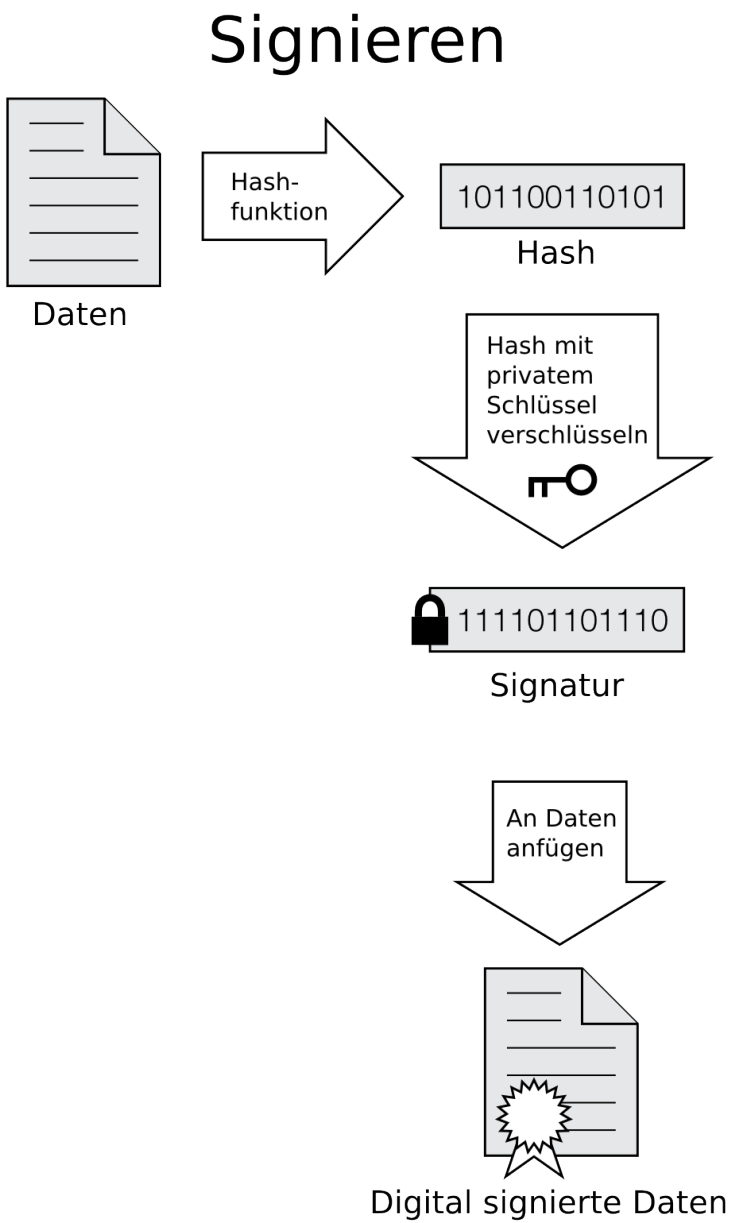
- Zweifelsfreie Bestätigung der Identität
- Nicht wiederverwendbar
- Gültigkeit nur mit Originaldokument
- Dokument nicht veränderbar

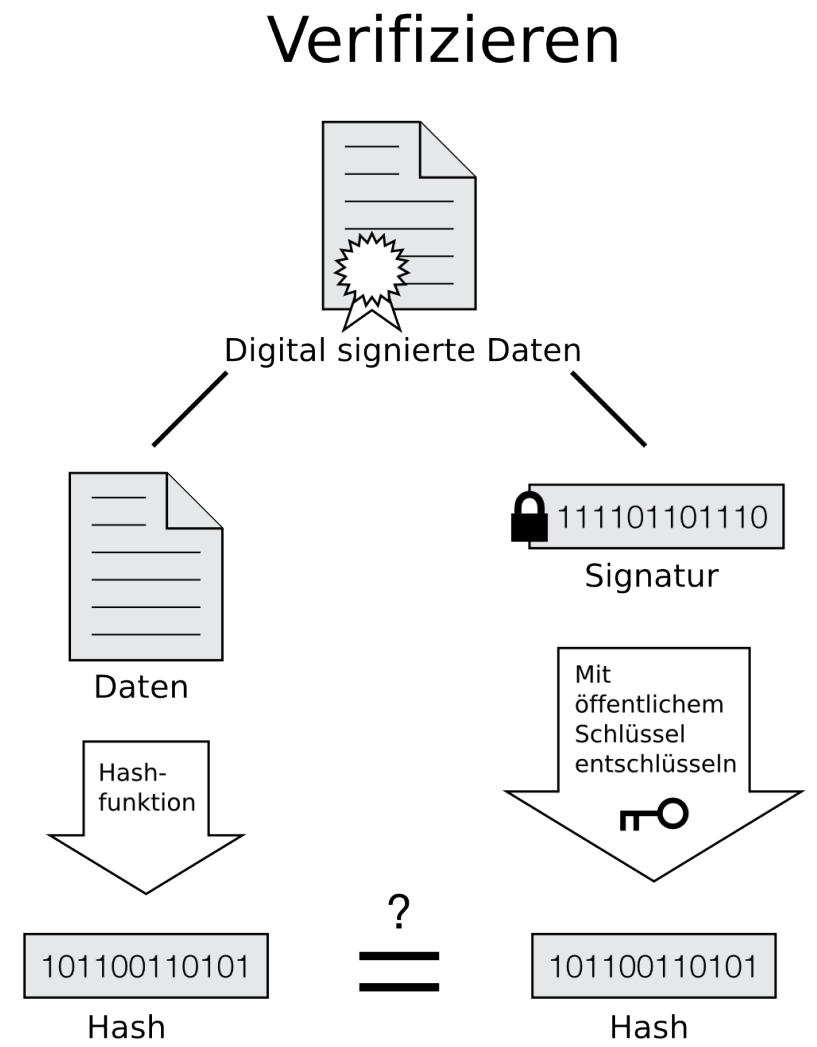
Geschichte der Digitalen Signaturen

- 1977: RSA
- 1984: GMR
- 1984: ElGamal Signaturschema
- 1991: DSA

Gliederung

Signieren und Verifizieren





Wenn die Hashwerte gleich sind ist die Signatur gültig.

Gliederung

Fakten

Kurzübersicht über DSA

- Entwicklung durch die NSA.
- 1991 vom NIST als FIPS vorgeschlagen.
- 1993 als Standard akzeptiert.
- 2000 letzte überarbeitete Version.
- Basiert auf dem Problem des diskreten Logarithmus.

DSA-Parameter

Parameter

- **p**: Primzahl, 512-1024 Bit

DSA-Parameter

Parameter

- **p**: Primzahl, 512-1024 Bit
- **q**: Primzahl, 160 Bit, q Primfaktor von $p-1$

DSA-Parameter

Parameter

- **p**: Primzahl, 512-1024 Bit
- **q**: Primzahl, 160 Bit, q Primfaktor von $p-1$
- **h**: $1 < h < p-1$ und $h^{\frac{p-1}{q}} \mod p > 1$

DSA-Parameter

Parameter

- **p**: Primzahl, 512-1024 Bit
- **q**: Primzahl, 160 Bit, q Primfaktor von $p-1$
- **h**: $1 < h < p-1$ und $h^{\frac{p-1}{q}} \mod p > 1$
- **g**: $g = h^{\frac{p-1}{q}} \mod p$

DSA-Parameter

Parameter

- **p**: Primzahl, 512-1024 Bit
- **q**: Primzahl, 160 Bit, q Primfaktor von $p-1$
- **h**: $1 < h < p-1$ und $h^{\frac{p-1}{q}} \mod p > 1$
- **g**: $g = h^{\frac{p-1}{q}} \mod p$
- **x**: Zufallszahl, $0 < x < q$

DSA-Parameter

Parameter

- **p**: Primzahl, 512-1024 Bit
- **q**: Primzahl, 160 Bit, q Primfaktor von $p-1$
- **h**: $1 < h < p-1$ und $h^{\frac{p-1}{q}} \mod p > 1$
- **g**: $g = h^{\frac{p-1}{q}} \mod p$
- **x**: Zufallszahl, $0 < x < q$
- **y**: $y = g^x \mod p$

DSA-Parameter

Parameter

- **p**: Primzahl, 512-1024 Bit
- **q**: Primzahl, 160 Bit, q Primfaktor von $p-1$
- **h**: $1 < h < p-1$ und $h^{\frac{p-1}{q}} \mod p > 1$
- **g**: $g = h^{\frac{p-1}{q}} \mod p$
- **x**: Zufallszahl, $0 < x < q$
- **y**: $y = g^x \mod p$

Schlüssel

- (p, q, g, y) ist Verifizierungsschlüssel.
- x ist Signaturschlüssel.

DSA-Signaturerstellung

Signieren

- **M**: Nachricht

DSA-Signaturerstellung

Signieren

- **M**: Nachricht
- **SHA-1**: Hashfunktion für Nachricht

DSA-Signaturerstellung

Signieren

- **M**: Nachricht
- **SHA-1**: Hashfunktion für Nachricht
- **k**: Zufallszahl, $0 < k < q$

DSA-Signaturerstellung

Signieren

- **M**: Nachricht
- **SHA-1**: Hashfunktion für Nachricht
- **k**: Zufallszahl, $0 < k < q$
- **r**: $r = (g^k \mod p) \mod q$

DSA-Signaturerstellung

Signieren

- **M**: Nachricht
- **SHA-1**: Hashfunktion für Nachricht
- **k**: Zufallszahl, $0 < k < q$
- **r**: $r = (g^k \mod p) \mod q$
- **s**: $s = (k^{-1} \cdot (\text{SHA-1}(\text{M}) + x \cdot r)) \mod q$

DSA-Signaturerstellung

Signieren

- **M**: Nachricht
- **SHA-1**: Hashfunktion für Nachricht
- **k**: Zufallszahl, $0 < k < q$
- **r**: $r = (g^k \mod p) \mod q$
- **s**: $s = (k^{-1} \cdot (\text{SHA-1}(\text{M}) + x \cdot r)) \mod q$

Signatur

- (r, s) ist Signatur.

DSA-Signaturverifikation

Verifizieren

- Überprüfe, ob $0 < r < q$ und $0 < s < q$

DSA-Signaturverifikation

Verifizieren

- Überprüfe, ob $0 < r < q$ und $0 < s < q$
- **w**: $w = s^{-1} \mod q$

DSA-Signaturverifikation

Verifizieren

- Überprüfe, ob $0 < r < q$ und $0 < s < q$
- **w**: $w = s^{-1} \mod q$
- $\mathbf{u}_1$: $u_1 = ((\text{SHA-1}(\text{M})) \cdot w) \mod q$

DSA-Signaturverifikation

Verifizieren

- Überprüfe, ob $0 < r < q$ und $0 < s < q$
- **w**: $w = s^{-1} \mod q$
- $\mathbf{u}_1$: $u_1 = ((\text{SHA-1}(\text{M})) \cdot w) \mod q$
- $\mathbf{u}_2$: $u_2 = (r \cdot w) \mod q$

DSA-Signaturverifikation

Verifizieren

- Überprüfe, ob $0 < r < q$ und $0 < s < q$
- **w**: $w = s^{-1} \mod q$
- $\mathbf{u}_1$: $u_1 = ((\text{SHA-1}(\text{M})) \cdot w) \mod q$
- $\mathbf{u}_2$: $u_2 = (r \cdot w) \mod q$
- **v**: $v = ((g^{u_1} \cdot y^{u_2}) \mod p) \mod q$

DSA-Signaturverifikation

Verifizieren

- Überprüfe, ob $0 < r < q$ und $0 < s < q$
- **w**: $w = s^{-1} \mod q$
- $\mathbf{u}_1$: $u_1 = ((\text{SHA-1}(M)) \cdot w) \mod q$
- $\mathbf{u}_2$: $u_2 = (r \cdot w) \mod q$
- **v**: $v = ((g^{u_1} \cdot y^{u_2}) \mod p) \mod q$

Gültigkeit

- Wenn $v = r$ gilt, dann ist die Signatur gültig.

Gliederung

Angriffsmöglichkeiten bei Signaturen

DSA-Angriffe

- Berechnung des privaten Schlüssels: $O(e^{(C+o(1))(\log p)^{\frac{1}{3}}(\log\log p)^{\frac{2}{3}}})$ oder $O(\frac{\pi * q}{2})$.
- Erraten der Zufallszahl k

Angriffsmöglichkeiten bei Signaturen

DSA-Angriffe

- Berechnung des privaten Schlüssels: $O(e^{(C+o(1))(\log p)^{\frac{1}{3}}(\log\log p)^{\frac{2}{3}}})$ oder $O(\frac{\pi * q}{2})$.
- Erraten der Zufallszahl k

RSA-Angriff

- Signieren und Verschlüsseln

Angriff auf RSA-Signaturschema

Idee

Der Angreifer Eve bringt den Empfänger Bob dazu, die Verschlüsselung zu entfernen.

Szenariobeschreibung

- Alice schickt an Bob eine signierte und verschlüsselte Nachricht.
- (K_E^A, K_D^A) ist das Schlüsselpaar von Alice:
  - K_E^A ist der öffentliche Verifikations- und gleichzeitig Verschlüsselungsschlüssel.
  - K_D^A ist der private Signatur- und gleichzeitig Entschlüsselungsschlüssel.
- Analoges gilt für Bob und Eve.

Ablauf des Angriffs

1. Alice signiert M: $D(M, K_D^A) = sig$.

Ablauf des Angriffs

1. Alice signiert M: $D(M, K_D^A) = sig$.
2. Alice verschlüsselt sig: $E(sig, K_E^B) = C$.

Ablauf des Angriffs

1. Alice signiert M: $D(M, K_D^A) = sig$.
2. Alice verschlüsselt sig: $E(sig, K_E^B) = C$.
3. Alice verschickt C an Bob.

Ablauf des Angriffs

1. Alice signiert M: $D(M, K_D^A) = sig$.
2. Alice verschlüsselt sig: $E(sig, K_E^B) = C$.
3. Alice verschickt C an Bob.
4. Eve fängt C ab und schickt die Nachricht an Bob weiter.

Ablauf des Angriffs

1. Alice signiert M: $D(M, K_D^A) = sig$.
2. Alice verschlüsselt sig: $E(sig, K_E^B) = C$.
3. Alice verschickt C an Bob.
4. Eve fängt C ab und schickt die Nachricht an Bob weiter.
5. Bob entschlüsselt das von Eve geschickte C:
   $D(C, K_D^B) = D(M, K_D^A) = sig$.

Ablauf des Angriffs

1. Alice signiert M: $D(M, K_D^A) = sig$.
2. Alice verschlüsselt sig: $E(sig, K_E^B) = C$.
3. Alice verschickt C an Bob.
4. Eve fängt C ab und schickt die Nachricht an Bob weiter.
5. Bob entschlüsselt das von Eve geschickte C:
   $D(C, K_D^B) = D(M, K_D^A) = sig$.
6. Bob verifiziert die Signatur sig mit dem Verifikationsschlüssel von Eve: $E(sig, K_E^E) = M'$.

Ablauf des Angriffs

1. Alice signiert M: $D(M, K_D^A) = sig$.
2. Alice verschlüsselt sig: $E(sig, K_E^B) = C$.
3. Alice verschickt C an Bob.
4. Eve fängt C ab und schickt die Nachricht an Bob weiter.
5. Bob entschlüsselt das von Eve geschickte C:
   $D(C, K_D^B) = D(M, K_D^A) = sig$.
6. Bob verifiziert die Signatur sig mit dem Verifikationsschlüssel von Eve: $E(sig, K_E^E) = M'$.
7. Bob sendet M' signiert und verschlüsselt an Eve zurück:
   $E(D(M', K_D^B), K_E^E) = C'$.

Ablauf des Angriffs (2)

8 Eve entschlüsselt die Nachricht C': $D(C', K_D^E) = sig'$.

Ablauf des Angriffs (2)

8. Eve entschlüsselt die Nachricht C': $D(C', K_D^E) = sig'$.
9. Eve verifiziert die Signatur sig': $E(sig', K_E^B) = M' = E(sig, K_E^E)$.

Ablauf des Angriffs (2)

8 Eve entschlüsselt die Nachricht C': $D(C', K_D^E) = sig'$.

9 Eve verifiziert die Signatur sig': $E(sig', K_E^B) = M' = E(sig, K_E^E)$.

10 Eve entschlüsselt M': $D(M', KED) = sig$.

Ablauf des Angriffs (2)

8 Eve entschlüsselt die Nachricht C': $D(C', K_D^E) = sig'$.

9 Eve verifiziert die Signatur sig': $E(sig', K_E^B) = M' = E(sig, K_E^E)$.

10 Eve entschlüsselt M': $D(M', KED) = sig$.

11 Eve verifiziert sig mit dem Verifikationsschlüssel von Alice: $E(sig, K_E^A) = M$.

Ablauf des Angriffs (2)

8 Eve entschlüsselt die Nachricht C': $D(C', K_D^E) = sig'$.

9 Eve verifiziert die Signatur sig': $E(sig', K_E^B) = M' = E(sig, K_E^E)$.

10 Eve entschlüsselt M': $D(M', KED) = sig$.

11 Eve verifiziert sig mit dem Verifikationsschlüssel von Alice: $E(sig, K_E^A) = M$.

Gegenmaßnahmen

- Hashwert statt Nachricht signieren.
- Zwei verschiedene Schlüsselpaare für Signatur und Verschlüsselung.

Gliederung

Richtlinien und Gesetze

- 1999: Signaturrichtlinie der EU.
- 2001: Signaturgesetz in Deutschland.
- 2007: Letzte Änderung des Signaturgesetzes.

Signaturgesetz

Einfache elektronische Signatur

- Keine besonderen Anforderungen.
- Beispielsweise Namen unter E-Mail.
- Geringe Aussagekraft.

Signaturgesetz

Einfache elektronische Signatur

- Keine besonderen Anforderungen.
- Beispielsweise Namen unter E-Mail.
- Geringe Aussagekraft.

Fortgeschrittene elektronische Signatur

- Einmaliger Signaturschlüssel.
- Identifizierbarkeit des Signaturerstellers.
- Nachträgliches Verändern erkennbar.
- Geringe Beweiskraft.

Signaturgesetz

Einfache elektronische Signatur

- Keine besonderen Anforderungen.
- Beispielsweise Namen unter E-Mail.
- Geringe Aussagekraft.

Fortgeschrittene elektronische Signatur

- Einmaliger Signaturschlüssel.
- Identifizierbarkeit des Signaturerstellers.
- Nachträgliches Verändern erkennbar.
- Geringe Beweiskraft.

Qualifizierte elektronische Signatur

- Beruht auf gültigem qualifizierten Zertifikat.
- Mit sicherer Signaturerstellungseinheit erstellt.
- Ersetzt gesetzlich geforderte Schriftform.

Bundesnetzagentur

- Veröffentlicht jährlich Empfehlungen für Signaturverfahren.
- Ausblick über Sicherheit für Signaturverfahren.

Bundesnetzagentur

- Veröffentlicht jährlich Empfehlungen für Signaturverfahren.
- Ausblick über Sicherheit für Signaturverfahren.

Empfehlungen 2009

- RSA, DSA und ECDSA bis 2015 sicher, falls richtige Parameter.

Bundesnetzagentur

- Veröffentlicht jährlich Empfehlungen für Signaturverfahren.
- Ausblick über Sicherheit für Signaturverfahren.

Empfehlungen 2009

- RSA, DSA und ECDSA bis 2015 sicher, falls richtige Parameter.
- Richtige Parameter für DSA:

Bundesnetzagentur

- Veröffentlicht jährlich Empfehlungen für Signaturverfahren.
- Ausblick über Sicherheit für Signaturverfahren.

Empfehlungen 2009

- RSA, DSA und ECDSA bis 2015 sicher, falls richtige Parameter.
- Richtige Parameter für DSA:
  1. Länge von p mindestens 2048 Bit.

Bundesnetzagentur

- Veröffentlicht jährlich Empfehlungen für Signaturverfahren.
- Ausblick über Sicherheit für Signaturverfahren.

Empfehlungen 2009

- RSA, DSA und ECDSA bis 2015 sicher, falls richtige Parameter.
- Richtige Parameter für DSA:
  1. Länge von p mindestens 2048 Bit.
  2. Länge von q mindestens 224 Bit.

Gliederung

Fazit und Ausblick

Praxisbeispiele

- PGP
- X.509

Fazit und Ausblick

Praxisbeispiele

- PGP
- X.509

Ausblick

- Geringen Bedeutung abseits von technikaffinen Menschen.

Fazit und Ausblick

Praxisbeispiele

- PGP
- X.509

Ausblick

- Geringen Bedeutung abseits von technikaffinen Menschen.
- Diverse Projekte in Entwicklung: ePerso, De-Mail, ElStEr.

Fazit und Ausblick

Praxisbeispiele

- PGP
- X.509

Ausblick

- Geringen Bedeutung abseits von technikaffinen Menschen.
- Diverse Projekte in Entwicklung: ePerso, De-Mail, ElStEr.
- Verzögerung bei Einführung wegen Problemen.

Fazit und Ausblick

Praxisbeispiele

- PGP
- X.509

Ausblick

- Geringen Bedeutung abseits von technikaffinen Menschen.
- Diverse Projekte in Entwicklung: ePerso, De-Mail, ElStEr.
- Verzögerung bei Einführung wegen Problemen.
- Einige Jahre bis elektronische Signaturen von größerem Teil der Bevölkerung eingesetzt werden.

Gliederung

Weiterführende Literatur


Prof. Dr. Claudia Eckert
IT-Sicherheit: Konzepte - Verfahren - Protokolle.
Oldenbourg Verlag, 5. Ausgabe, 2008.


Bruce Schneier
Applied Cryptography.
John Wiley & Sons Verlag, 2. Ausgabe, 1996.


National Institute of Standards and Technology (NIST)
Digital Signature Standard (DSS).
Federal Information Processing Standards Publication 186-2, 27. Januar 2000.