



openHPI – Sicherheit im Internet
Digitale Signaturen

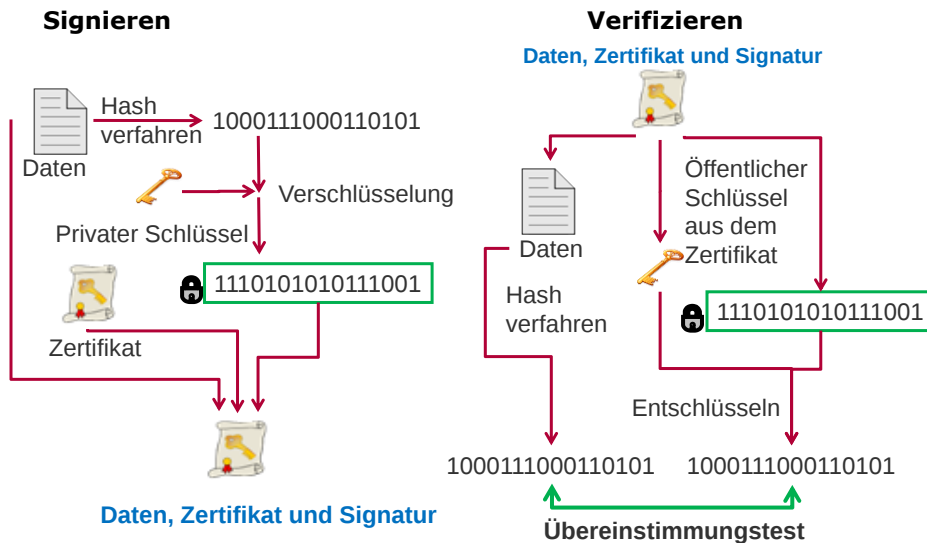
Prof. Dr. Christoph Meinel
Hasso-Plattner-Institut, Potsdam

Digitale Signaturen

Digitale Signaturen sind Sicherheitsprotokolle, mit denen die **Authentizität des Senders** und die **Integrität der Nachricht** sichergestellt werden kann

- Basieren auf 2-Schlüssel Verschlüsselung und nutzen Hash-Verfahren
- (Hash der) Nachricht wird mit privatem Schlüssel des Senders verschlüsselt „**signiert**“
- Kann die Signatur mit öffentlichen Schlüssel des Senders entschlüsselt werden, ist die Urheberschaft der Nachricht (**Authentizität des Senders**) und ihre Unversehrtheit (**Integrität der Nachricht**) bewiesen

Achtung: Beweiskraft hängt von korrekter Zuordnung des öffentlichen Schlüssels zu seinem Besitzer ab



Digitale Signaturen: Verfahren

Bekannte Verfahren zur **Signaturberechnung**:

- El-Gamal, DSA, Schnorr, ...
 - Beruhen auf 2-Schlüssel Verschlüsselung mit diskretem Logarithmus (über endlichen Körpern) oder RSA
- ECDSA, ECGDSA, ...
 - Beruhen auf 2-Schlüssel Verschlüsselung mit Kryptoverfahren über elliptischen Kurven

Verwendete **Hash-Verfahren** beliebig wählbar, solange sie mit der angewandten 2-Schlüssel Verschlüsselung harmonisieren

- **Beispiel:** RSA arbeitet auf Zahlen modulo einer großen zusammengesetzten Zahl → eingesetztes Hash-Verfahren muss Ergebnisse innerhalb dieses Intervalls produzieren

Sicherheit Digitaler Signaturen hängt ab von Sicherheit der verwendeten

- **2-Schlüssel Verschlüsselung**
- **Hashfunktion**
- Schlüssellänge des privaten Schlüssels muss hinreichend groß sein, um gegen Brute-Force Angriffe geschützt zu sein
- Hashfunktion muss kollisionsresistent sein, d.h. es ist praktisch unmöglich, unterschiedliche Eingaben mit dem gleichen Hash zu finden
- ansonsten bekämen unterschiedliche Nachrichten die gleiche Signatur ...

Authentizität/Identifikation des Senders

- Mögliche Entschlüsselung mit öffentlichem Schlüssel des Senders beweist, dass nur dieser die Signatur erstellen konnte, denn kein anderer hat Zugriff auf seinen privaten Schlüssel

Integrität

- Wurde Nachricht auf Transportweg verändert, unterscheidet sich der für die Verifikation auf Empfängerseite neu berechnete Hash vom entschlüsselten Hash
- Empfänger kann Nachricht erneut vom Sender anfordern

Nichtabstreitbarkeit

- Bei erfolgreicher Verifikation der Signatur kann Sender Urheberschaft der Nachricht nicht abstreiten

Zu signierende Nachricht: „Internetsicherheit ist wichtig“

Erzeugen Digitale Signatur

- Z.B. mit RSA mit
 - öffentlichem Schlüssel (6.355.561, 582.044.791)
 - privatem Schlüssel (641, 582.044.791)
- Hashverfahren CRC32 (liefert überschaubare Werte)

Signaturberechnung:

- 1) Hashwert berechnen von „Internetsicherheit ist wichtig“:
$$\text{CRC32}(\text{Internetsicherheit ist wichtig})$$
$$= 375.235.237 = (10110010111011010001010100101)_2$$
- 2) Signatur berechnen vom Hash von „Internetsicherheit ist wichtig“:
$$\text{Signatur}(375.235.237) = 375.235.237^{641} \bmod 582.044.791$$
$$= \mathbf{56.347.449}$$

RSA: öffentlichem Schlüssel (6.355.561, 582.044.791)
privatem Schlüssel (641, 582.044.791)

Verifikation der Signatur

- Empfänger empfängt
 - Nachricht „Internetsicherheit ist wichtig“ und
 - Signatur **56.347.449**
- Empfänger berechnet den Hash der Nachricht und dekodiert Signatur mit öffentlichen Schlüssel des Senders
 - 1) $\text{CRC32}(\text{Internetsicherheit ist wichtig}) = \mathbf{375.235.237}$
 - 2) Entschlüsselung der Signatur 56.347.449 mit öffentlichem Schlüssel des Senders 6.355.561:
$$56.347.449^{6.355.561} \bmod 582.044.791 = \mathbf{375.235.237}$$
 - 3) Übereinstimmungstest beweist Senderauthentizität und Unversehrtheit der empfangenen Nachricht