



openHPI – Internet Security Digital Signatures

Prof. Dr. Christoph Meinel
Hasso Plattner Institute, Potsdam, Germany

Digital Signatures

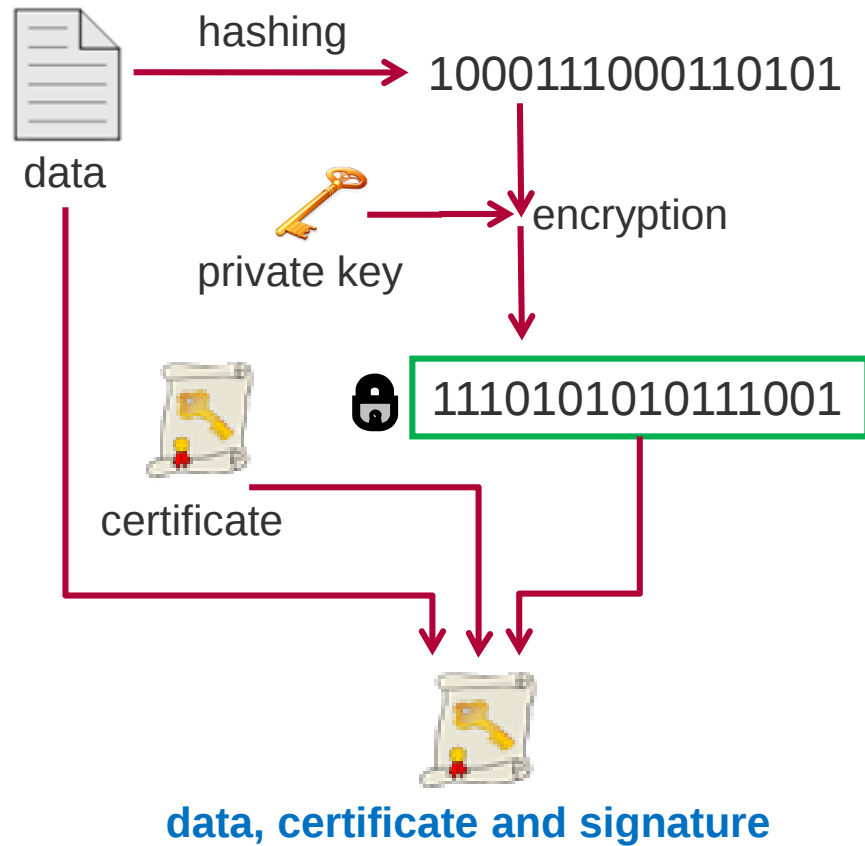
Digital signatures are security protocols that ensure the **authenticity of the sender** and the **integrity of the message**

- Based on 2-key encryption and use hash method
- (Hash of) message is encrypted and "**signed**" with the sender's private key
- If the signature can be decrypted with the public key of the sender, the origin of the message (**authenticity of the sender**) and its integrity (**integrity of the message**) is proved

Caution: The proof depends on the correct allocation of the public key to its owner

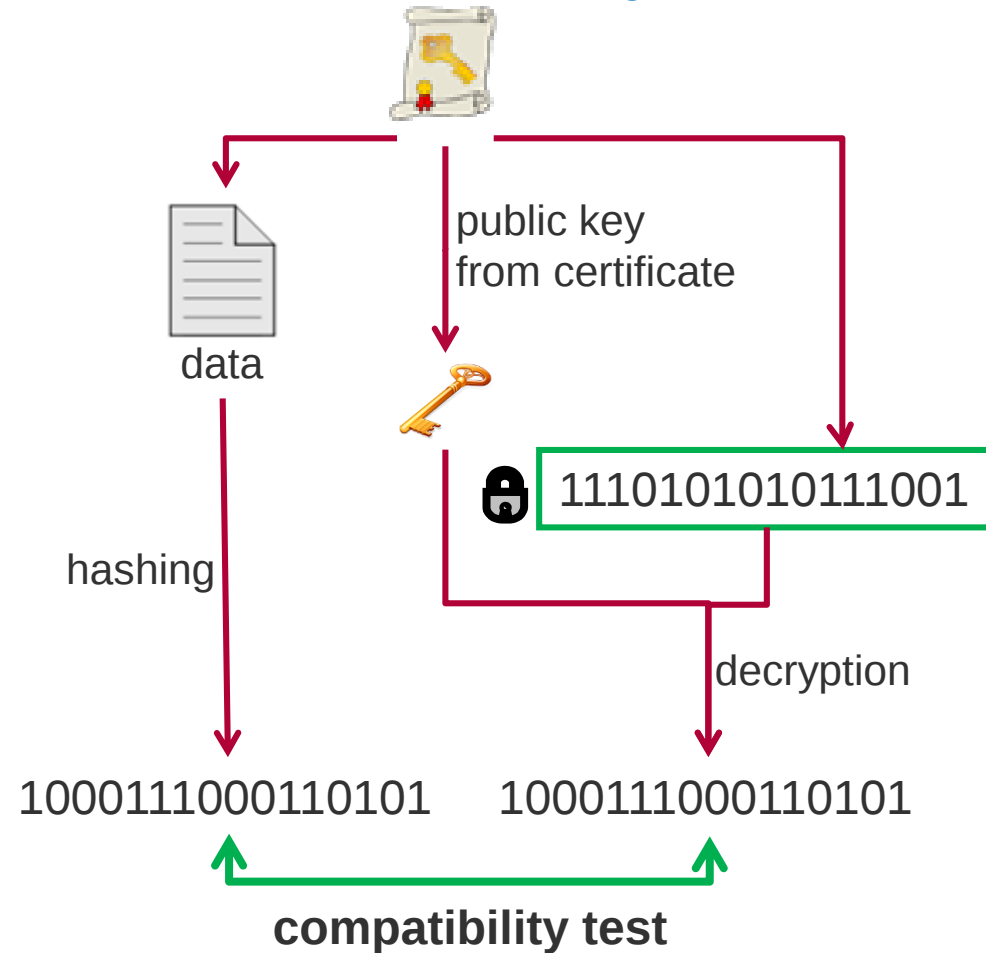
Digital Signatures: Overview

Sign



Verify

data, certificate and signature



Digital Signatures: Methods

Well-known methods of **signature calculation**:

- El-Gamal, DSA, Schnorr, ...
 - Based on 2-key encryption with discrete logarithm (finite fields) or RSA
- ECDSA, ECGDSA, ...
 - Based on 2-key encryption with elliptical curves encryption method

Hash method used can be arbitrarily selected as long as it is compatible with the 2-key encryption used

- **Example:** RSA computes the modulo numbers with a large composite number → hash method used must produce results within this interval

Digital Signatures: Security

Security of digital signatures depends on security of the following used:

- **2-key encryption**
- **Hash function**
- Key length of the private key must be large enough to protect against brute force attacks
- Hash function must be collision resistant, i.e. it must be virtually impossible to find different inputs with the same hash
 - otherwise different messages would have the same signature ...

Authenticity / identification of the sender

- Decryption possible with the sender's public key proves that only he/she could create the signature, as no one else has access to his/her private key

Integrity

- If the message has been modified on the transport path, the hash newly calculated for the verification on the recipient side differs from the decrypted hash
- Recipient can request message again from sender

Non-repudiation

- If the signature is successfully verified, sender cannot deny the authority of the message

Digital Signatures: An Example (1/2)

Message to be signed: *"Internet security is important"*

Generation of digital signature

- For example, with RSA with
 - public key (6.355.561, 582.044.791)
 - private key (641, 582.044.791)
- Hash method CRC16 (provides manageable values)

Signature calculation

- 1) Calculate the hash value of *"Internet security is important"*:

$$\begin{aligned} &\text{CRC16}(\text{Internet security is important}) \\ &= 39.331 = (1001100110100011)_2 \end{aligned}$$

- 2) Calculate hash signature of *„Internet security is important“*:

$$\begin{aligned} \text{Signature}(39.331) &= 39.331^{641} \bmod 582.044.791 \\ &= \mathbf{325.185.697} \end{aligned}$$

Digital Signatures: An Example (2/2)

RSA: public key (6.355.561, 582.044.791)

private key (641, 582.044.791)

Signature verification

- receiver gets:
 - Message „*Internet security is important*“ and
 - Signature **325.185.697**
- Receiver calculates the hash of the message and decodes the signature with the sender's public key
 - 1) $\text{CRC16}(\text{Internet security is important}) = \mathbf{39.331}$
 - 2) Decryption of the signature **325.185.697** with sender's public key 6.355.561:
$$325.185.697^{6.355.561} \bmod 582.044.791 = \mathbf{39.331}$$
 - 3) Consistency test verifies sender's authenticity and integrity of the received message