Elektronische Siegel

Arturo Bertoglia & Marc Schlicker

Versiegelung durch QR-Code

- Als Beispiel: Studienbescheinigung der HU Berlin
- Möglichkeit per QR-Code auf die Verifizierungsseite zu kommen
- Die Verifizierungsseite wird auf dem Dokument selbst angezeigt


Verifikationsnummer 2IQG 0VIK

Gültigkeitsprüfung unter:
https://agnes.hu-berlin.de/verify

Stärken

- Einfach einzuführen
- Überall einsetzbar

Schwachstellen

- Sybil-Angriff, Überprüfungsseite imitieren
- Verifizierungsprozess ist umständlich und erfordert externe Ressourcen
- Verifizierung erfolgt online

QR-Code Hijacking

Wir führen einen Sybil-Attacke in dem wir einen fiktiven Dokument in Namens der Humboldt Universität erstellen die uns unwahre Leistungen akkreditiert.

Dafür erstellen wir zuerst einen Dokument, das eine Bescheinigung der Humboldt Universität entspricht, mit einen QR-Code die zur unsere Webseite hinführt.

Unsere Webseite ähnelt das Aussehen der Humboldt Verifizierungsseite und ist durch ausgeklügelte URL-Gestaltung schwer von einer echte Webseite der Humboldt zu unterscheiden.

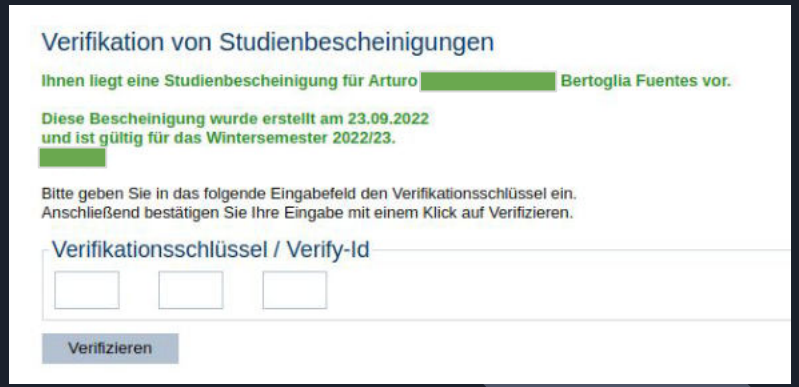

HUMBOLDT-UNIVERSITÄT ZU BERLIN

HU | Studienabteilung | Referat Studierendenservice | 10099 Berlin

Arturo [redacted] Bertoglia Fuentes

[redacted]

Studienbescheinigung


Verifikation von Studienbescheinigungen

Ihnen liegt eine Studienbescheinigung für Arturo [redacted] Bertoglia Fuentes vor.

Diese Bescheinigung wurde erstellt am 23.09.2022
und ist gültig für das Wintersemester 2022/23.

Bitte geben Sie in das folgende Eingabefeld den Verifikationsschlüssel ein.
Anschließend bestätigen Sie Ihre Eingabe mit einem Klick auf Verifizieren.

Verifikationsschlüssel / Verify-Id

Verifizieren

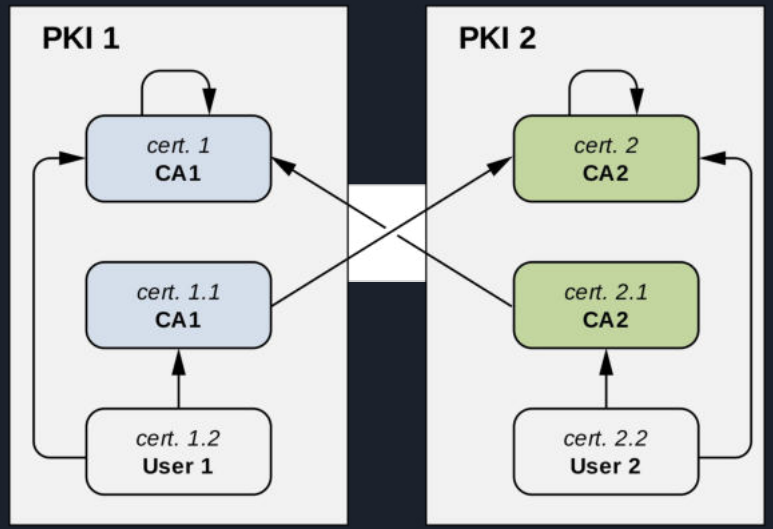
Versiegelung durch elektronische Signatur

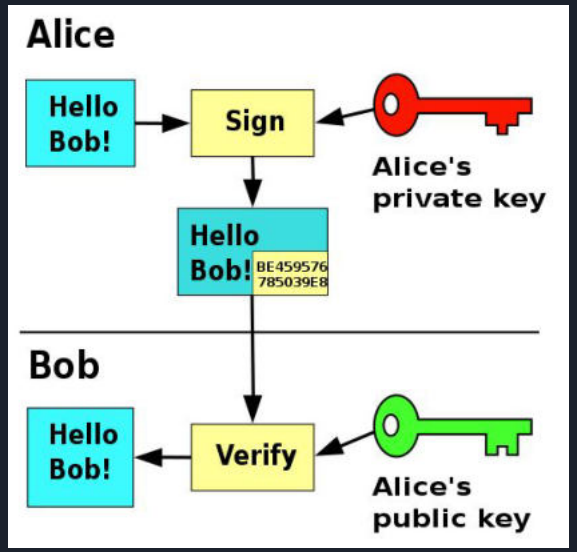
Elektronische Signaturen basieren auf die X.509 Standard und werden sowohl in online Protokolle wie TLS/SSl als auch in offline Bereich für elektronische Signaturen benutzt.

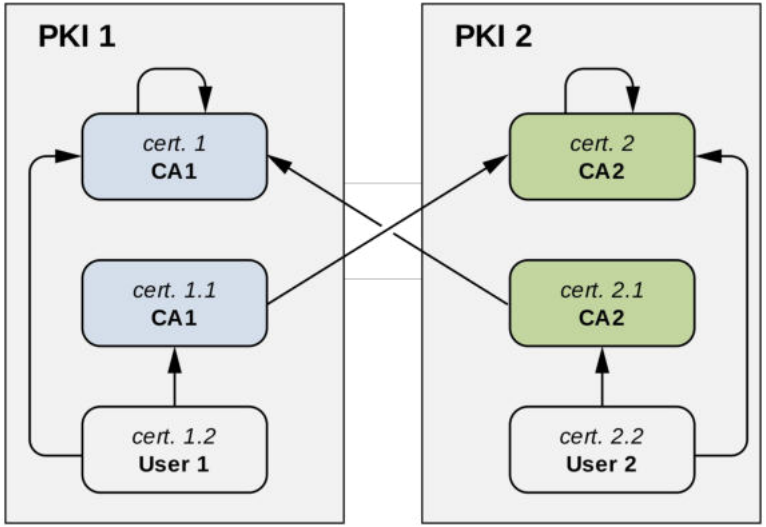
Darüber hinaus existiert der Advanced Electronic Signature (AES) Standard basieren auf die EU Verordnung Nr. 910/2014 (eIDAS) mit folgenden Anforderungen:

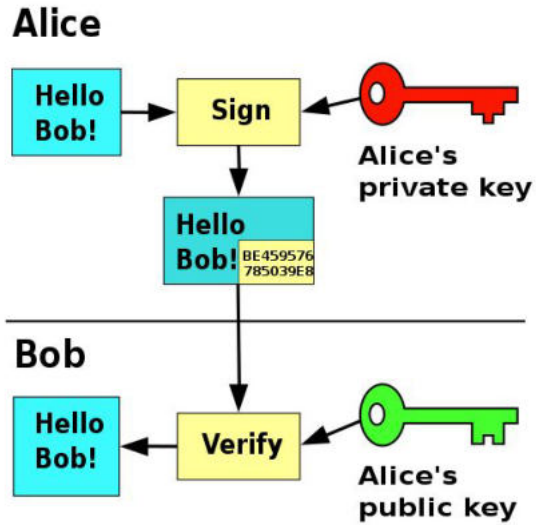
1. Der Unterzeichner kann eindeutig identifiziert und mit der Unterschrift verknüpft werden.
2. Der Unterzeichner muss die alleinige Kontrolle über die zur Erstellung der elektronischen Unterschrift verwendeten Daten haben (in der Regel ein privater Schlüssel).
3. Die Unterschrift muss in der Lage sein, festzustellen, ob die begleitenden Daten nach der Unterzeichnung des Dokuments manipuliert wurden.
4. Falls die begleitenden Daten geändert wurden, muss die Unterschrift ungültig werden.

Versiegelung durch elektronische Signatur Umsetzung PKI






PKI 1
cert. 1
CA1
cert. 1.1
CA1
cert. 1.2
User 1
PKI 2
cert. 2
CA2
cert. 2.1
CA2
cert. 2.2
User 2

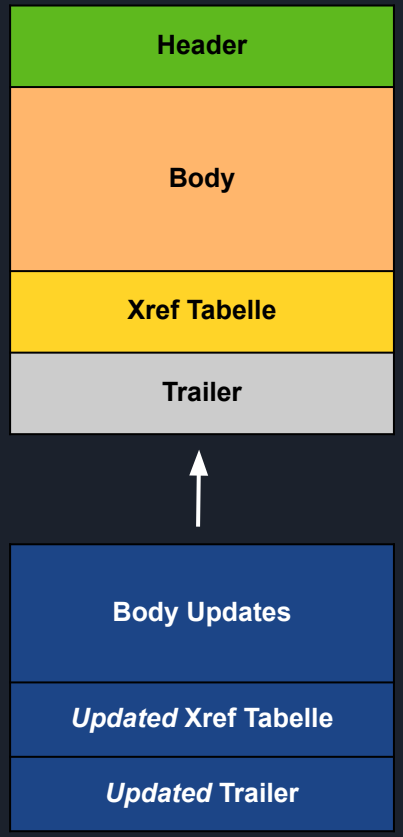

Alice
Hello Bob!
Sign
Alice's private key
Hello Bob!
BE459576 785039E8
Bob
Hello Bob!
Verify
Alice's public key

Die Struktur einer PDF Datei

PDF Dateien besitzen grundsätzlich vier Sektionen: Header, Body, XRef Tabelle und Trailer.

Das Lesen einer PDF Datei erfolgt von unten nach oben.

Weitere Elemente können am Ende der Datei angehängt werden.




```
%PDF-1.5
```

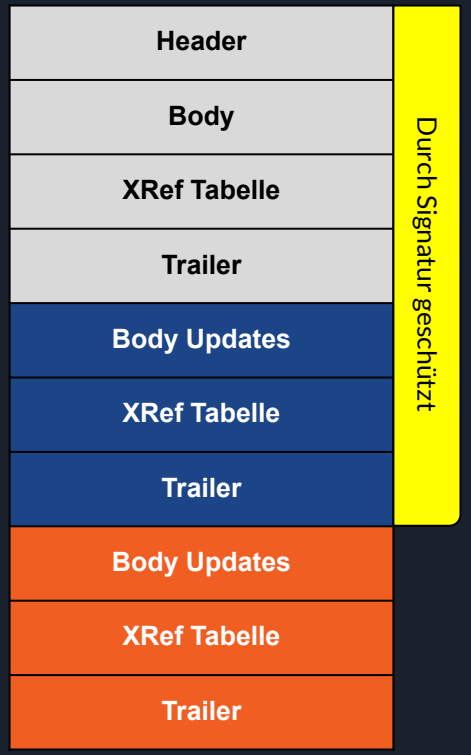
```
4 0 obj
<< /Length 5 0 R
  /Filter /FlateDecode>>
stream
x\9C3T0\00B]C aab\A8gaainn\...
endstream
endobj
5 0 obj
  117
endobj
3 0 obj
…
```

```
xref
0 14
0000000000 65535 f
0000008246 00000 n
0000000340 00000 n
…
```

```
trailer
<< /Size 14
  /Root 13 0 R
  /Info 12 0 R >>
```

Incremental Saving Angriff (ISA)

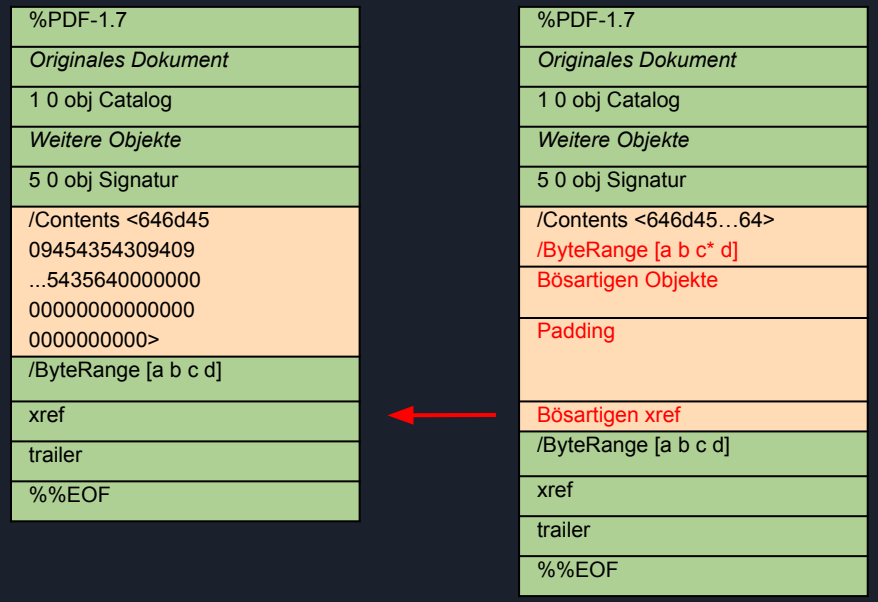
Einen Update auf der PDF Datei ausführen mit dem bösartigen Inhalte eingefügt werden ohne dabei die Signaturintegrität zu gefährden.



Unser Versuch

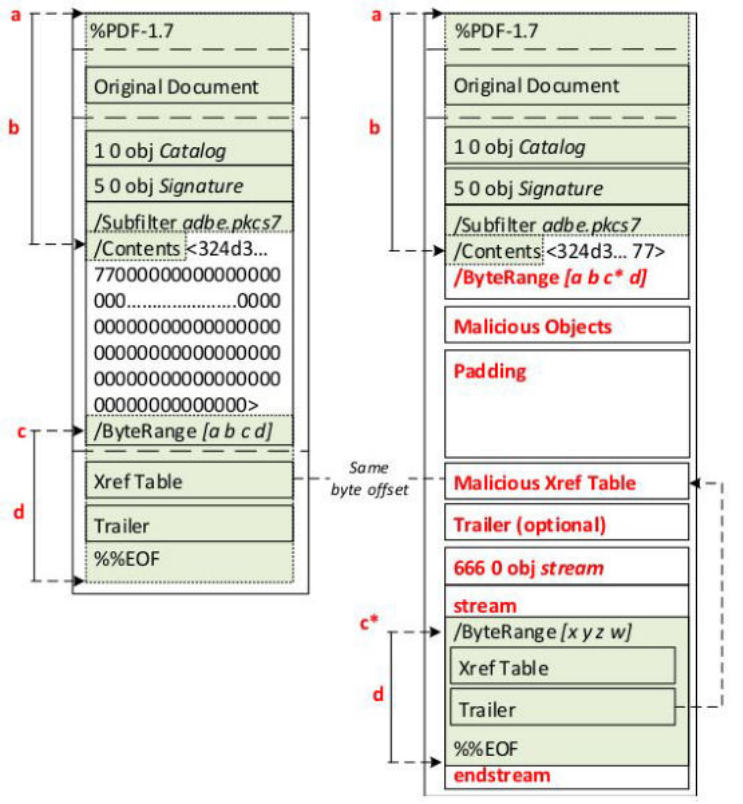
Signature Wrapping Angriff

- Zieldokument wird in signiertes Dokument unsichtbar eingefügt
- Kommt aus der XML-Verschlüsselung
- Bei einer Signatur wird nicht das ganze Dokument geprüft, ob es signiert ist, sondern nur der signierte Teil



Signature Wrapping Angriff (SWA)

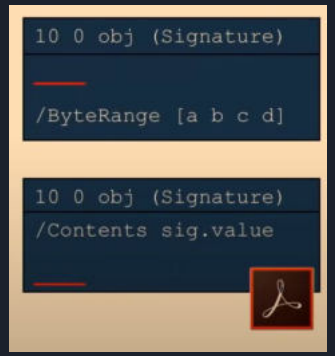
- Zieldokument wird in signiertes Dokument unsichtbar eingefügt
- Kommt aus der XML-Verschlüsselung
- Bei einer Signatur wird nicht das ganze Dokument geprüft, ob es signiert ist, sondern nur der signierte Teil

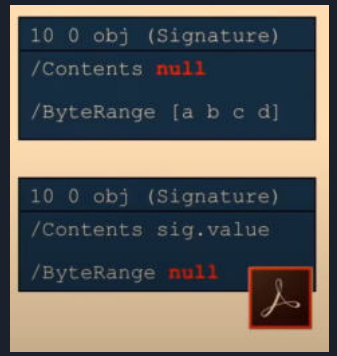


https://pdf-insecurity.org/signature/swa.html

Universal Signature Forging Angriff

- Direkter Angriff auf der Signatur, sein Inhalt und sein Byte Range.
- Unterschiedlichen PDF Reader sind anfällig gegenüber unterschiedliche Varianten dieser Angriff.





Welche PDF-Reader funktionieren

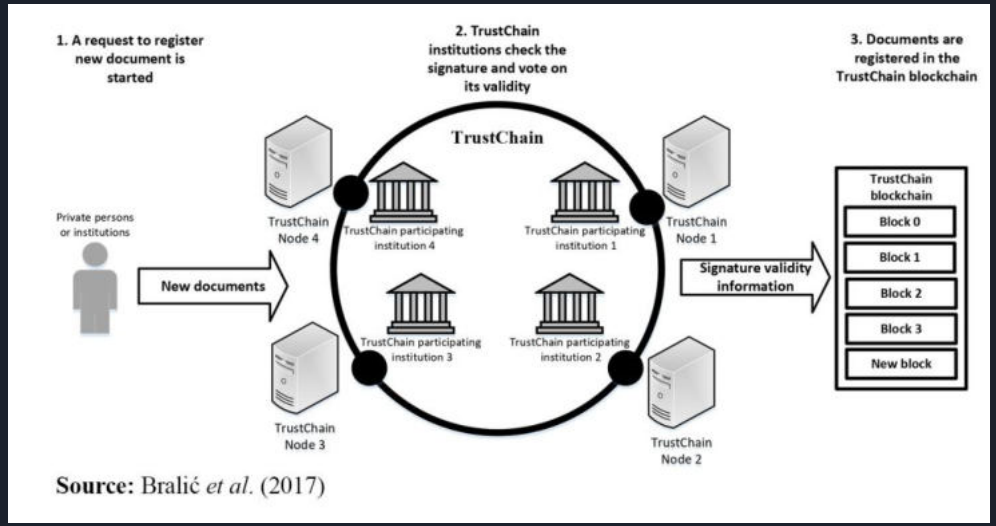
PDF Viewer	Version	OS	PDF Signature			Comments
			USF	ISA	SWA	
Adobe Acrobat Reader DC	2018.011	Win10, MacOS	●	✓	✓	Error when a visible signature is clicked, for invisible signatures this is not a problem
Adobe Reader 9	9.5.5	Linux	✓	✓	✓	
Adobe Reader XI	11.0.10	Win10, MacOS	●	✓	✓	Error when a visible signature is clicked, for invisible signatures this is not a problem
eXpert PDF 12 Ultimate	12.0.20	Win10	✓	✓	●	
Expert PDF Reader	9.0.180	Win10	✓	✓	●	
Foxit Reader	9.1.0; 9.2.0	Win10, Linux, MacOS	✓	●	●	No signature verification on Linux and MacOS available (latest version 2.4.1)
LibreOffice (Draw)	6.0.6.2; 6.0.3.2, 6.1.0.3	Win10, Linux, MacOS	✓	◑	✓	Detects ISA when certificate is trusted
Master PDF Editor	5.1.12/24	Linux, Win10, MacOS	✓	●	✓	Attack only on Linux and Windows successful. On MacOS the original, not manipulated signature was already invalid.
Nitro Pro	11.0.3.173	Win10	✓	◑	●	Detects ISA when certificate is trusted
Nitro Reader	5.5.9.2	Win10	✓	◑	●	Detects ISA when certificate is trusted

Archivierung durch Blockchain

Blanchette (2006) erörtert drei Möglichen Wege für digitale Archive:

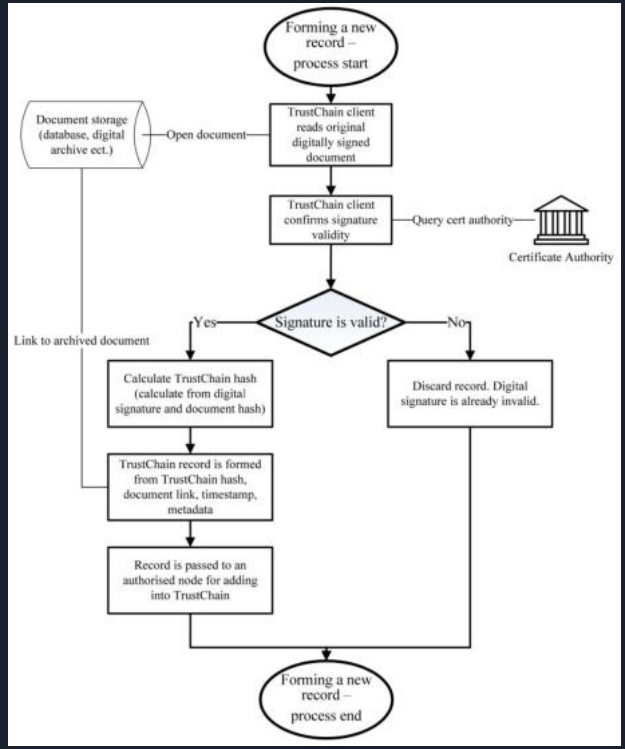
1. Digitale Signaturen/Siegel aufbewahren
2. Digitale Signaturen/Siegel zu entfernen
3. Die Spuren der digitale Signaturen/Siegel als Metadaten aufbewahren.

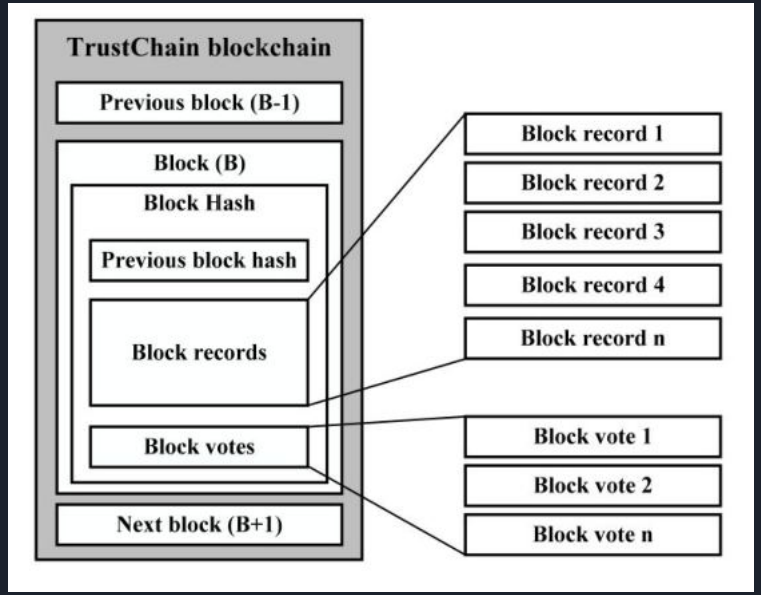
Blockchain Technologie erlaubt uns eine vierte Möglichkeit für digitale Archive mit Trustchain.



Source: Bralić *et al.* (2017)

Ablauf der Archivierung durch TrustChain





Bralić, Kuleš, Stančić (2017)

Forward-Secure Seals & Signatures

Forward-Security erfasst die Idee, dass es selbst im Falle einer Offenlegung des aktuellen geheimen Schlüssels für jeden Angreifer rechnerisch unmöglich sein sollte, eine Signatur für einen vergangenen Zeitraum zu fälschen.

Die erste formelle Forward-Secure Signatur Verfahren wurde von Bellare and Miner in 1999 präsentiert.

```
algorithm GQ.key(k, l)
  Generate random ⌈k/2⌉-bit
    primes p_1, p_2
  n ← p_1 p_2
  s ←R Z*_n
  e ←R [2^l, 2^(l+1))
    s.t. gcd(e, φ(n)) = 1
  v ← 1/s^e mod n
  SK ← (n, s, e)
  PK ← (n, v, e)
  return (SK, PK)

algorithm GQ.sign(M, (n, s, e))
  r ←R Z*_n
  y ← r^e mod n
  σ ← H(y, M)
  z ← r s^σ mod n
  return (z, σ)

algorithm GQ.ver(M, (n, v, e), (z, σ))
  if z ≡ 0 (mod n) then return 0
  y' ← z^e v^σ mod n
  if σ = H(y', M) then return 1
                  else return 0
```

Forward-Secure signature scheme

Itkins und Reyzin (2001) stellen ein Verfahren vor die Forward-Security auf digitalen Signaturen garantiert.

algorithm IR.key(k, l, T)
- Generate random $(\lceil k/2 \rceil - 1)$-bit primes q_1, q_2 s.t. $p_i = 2q_i + 1$ are both prime
- $n \leftarrow p_1 p_2$
- $t_1 \overset{R}{\leftarrow} Z_n^*$
- Generate primes e_i s.t. $2^l(1 + (i-1)/T) \leq e_i < 2^l(1 + i/T)$ for $i = 1, 2, \ldots, T$.
  (This generation is done either deterministically or using a small seed *seed* and H as a pseudorandom function.)
- $f_2 \leftarrow e_2 \cdot \ldots \cdot e_T \bmod \phi(n)$, where $\phi(n) = 4q_1 q_2$
- $s_1 \leftarrow t_1^{f_2} \bmod n$
- $v \leftarrow 1/s_1^{e_1} \bmod n$
- $t_2 \leftarrow t_1^{e_1} \bmod n$
- $SK_1 \leftarrow (1, T, n, s_1, t_2, e_1, seed)$
- $PK \leftarrow (n, v, T)$
- **return** (SK_1, PK)

---

algorithm IR.update(SK_j)
- Let $SK_j = (j, T, n, s_j, t_{j+1}, e_j, seed)$
- **if** $j = T$ **then return** ϵ
- Regenerate $e_{j+1}, \ldots, e_T$ using *seed*
- $s_{j+1} \leftarrow t_{j+1}^{e_{j+2} \cdot \ldots \cdot e_T} \bmod n$; $t_{j+2} \leftarrow t_{j+1}^{e_{j+1}} \bmod n$
- **return** $SK_{j+1} = (j+1, T, n, s_{j+1}, t_{j+2}, e_{j+1}, seed)$

---

algorithm IR.sign(SK_j, M)
- Let $SK_j = (j, T, n, s_j, t_{j+1}, e_j, seed)$
- $r \overset{R}{\leftarrow} Z_n^*$
- $y \leftarrow r^{e_j} \bmod n$
- $\sigma \leftarrow H(j, e_j, y, M)$
- $z \leftarrow r s^{\sigma} \bmod n$
- **return** (z, σ, j, e_j)

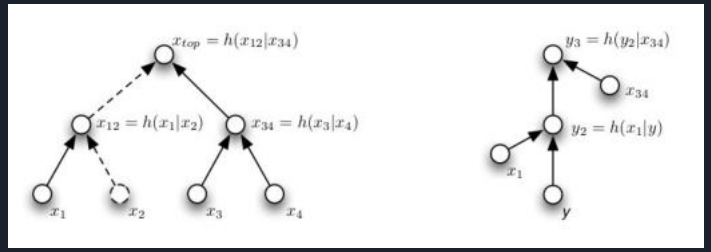
---

algorithm IR.ver$(PK, M, (z, \sigma, j, e))$
- Let $PK = (n, v)$
- **if** $e \geq 2^l(1 + j/T)$ **or** $e < 2^l$ **or** e is even **then return** 0
- **if** $z \equiv 0 \pmod n$ **then return** 0
- $y' \leftarrow z^e v^{\sigma} \bmod n$
- **if** $\sigma = H(j, e, y', M)$ **then return** 1 **else return** 0

Keyless Digital Seals & Signatures

Schlüssellose Signaturen sind eine alternative Lösung zu traditionellen PKI-Signaturen. Das Wort 'schlüssellos' bedeutet nicht, dass bei der Erstellung der Signatur keine kryptografischen Schlüssel verwendet werden.

Schlüssel sind nach wie vor für die Authentifizierung erforderlich, aber die Signaturen können zuverlässig überprüft werden, ohne die fortgesetzte Geheimhaltung der Schlüssel vorauszusetzen.



Keyless Digital Seals & Signatures System Architektur

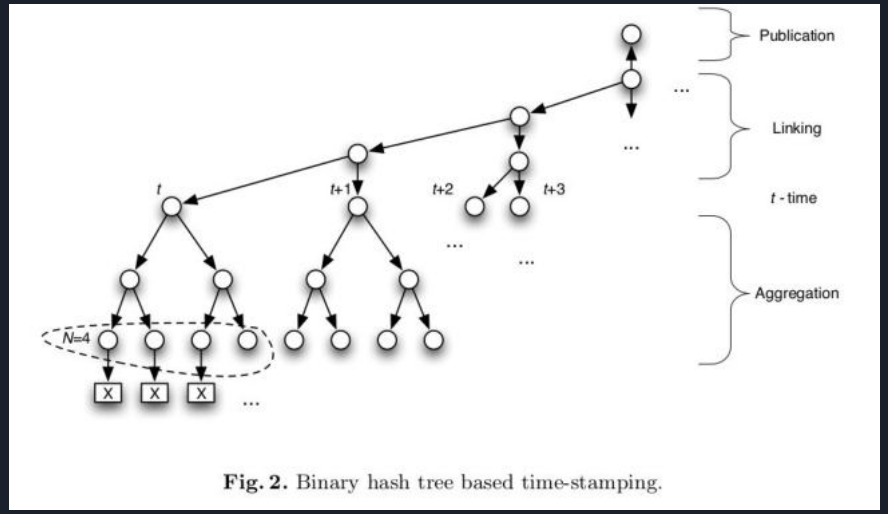
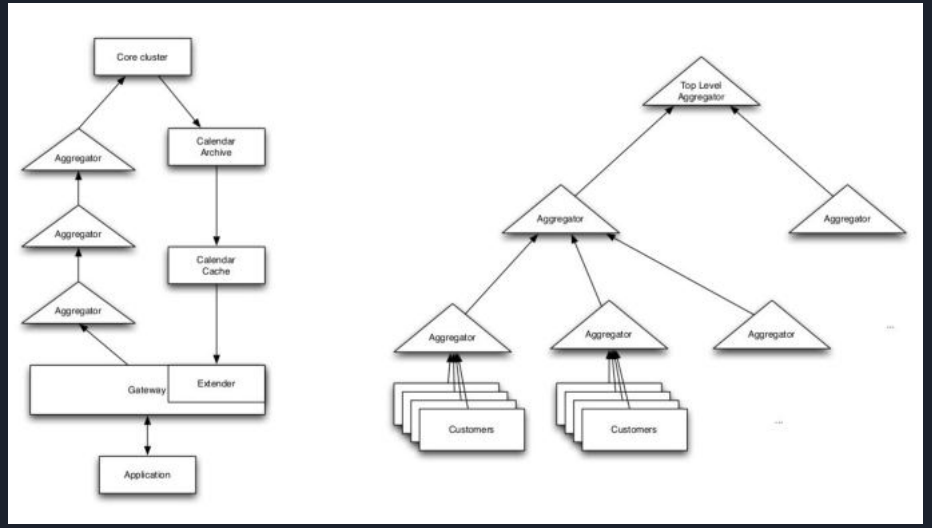


Fig. 2. Binary hash tree based time-stamping.



Buldas, Kroonmaa, Laanoja (2013)

Wünsche für ein digitales Siegel

01 **Automatische Verifizierung**

02 **Schutz vor Sybil Angriffe**

03 **Unabhängige Verifizierung wie TLS/SSL Zertifikate**

Unser Siegel

Unser Siegel besteht aus einer Datenbank, in welcher die Namen der Institutionen mit deren Verifikationswebseiten verknüpft sind, und einer elektronischen Signatur, welche den Namen der Institution, des Empfängers, der Dokumentart und einen Verfikationscode beinhaltet.

META:

- Empfänger
- Institution
- Dokumentenart

Verifizierungscode:

XXXX-XXXX-XXXX-XXXX

Vielen Dank!

Au revoir!