Hashfunktionen in der Kryptographie

Prof. Dr.-Ing. Damian Weber

Hochschule für Technik und Wirtschaft des Saarlandes

Inhalt

- Hashfunktionen

  was ist das

  wozu braucht man das

  wie macht man es kaputt ;-)

Was tut eine Hashfunktion?

Hashfunktionen (Definition)

$h: U \longrightarrow V$

U = Universum, V = Hashwerte, $|V|<\infty$

Urbilder

Bilder

Hashfunktionen (Beispiel)

EAN (ISBN-13) 978047111709-4

- $U = \{ n \in \mathbf{N} \mid n<10^{12} \}$
- $V=\{0,1,2,3,4,5,6,7,8,9\}$ .... 10 Kisten

$9+3\cdot7+8+3\cdot0+4+3\cdot7+1+3\cdot1+1+3\cdot7+0+3\cdot9 = 116$

$10-6=4$ 978047111709 → 4

Hashfunktionen (Anwendung)

- Integritätschecks
- Digitale Signaturen

Integritätscheck

Gleiche Kiste, also Dokument gleich
(ungleiche mit 75% Chance in anderer Kiste)

Datenstrukturen: 10^8 Kisten

Krypto: 10^{80} Kisten

Digitale Signaturen

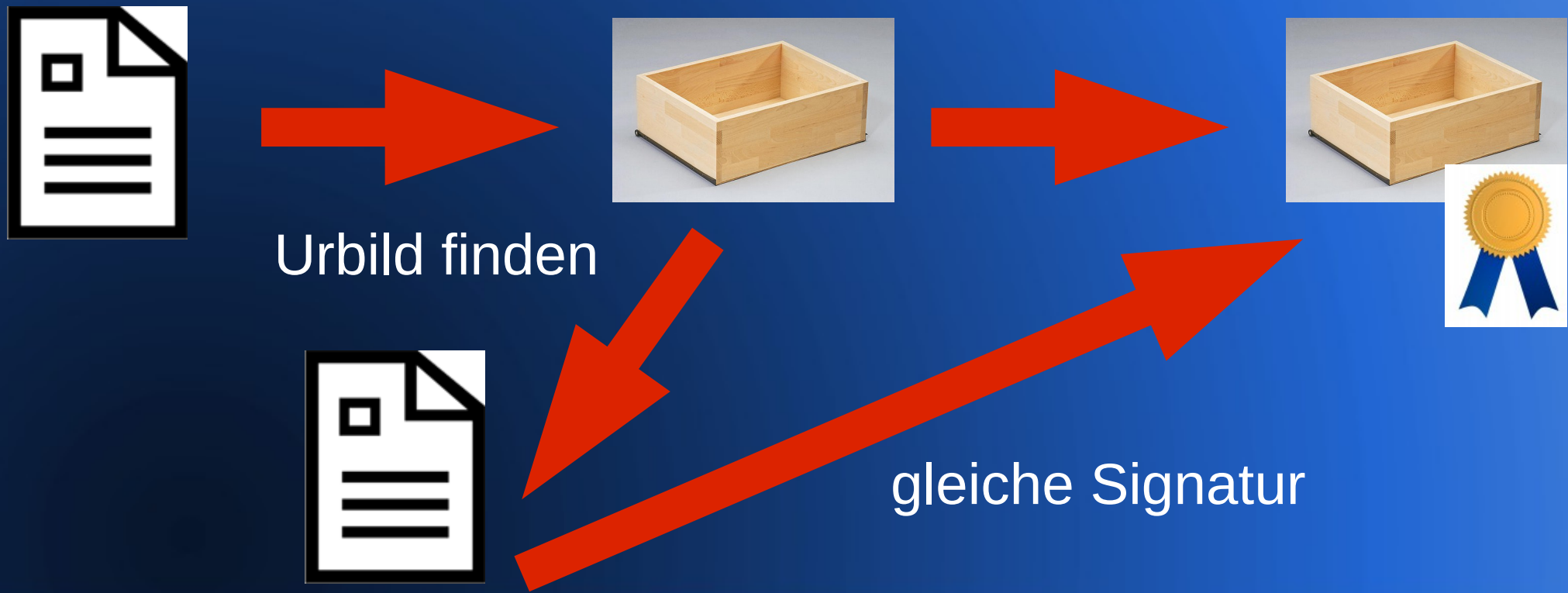


Digitale Signaturen
Gefahr: 2 Dokumente, gleiche Kiste

Kollision

Digitale Signaturen

Gefahr: 2. Dokument für diese Kiste künstlich erzeugen



Hashfunktionen (Anforderungen)

Sicher, wenn „unmöglich“:

- Urbild
  (Anwendung: finde gültiges Passwort)
- 2. Urbild
  (Anwendung: finde 2. Dokument für gegebene Signatur)
- Kollision
  (Anwendung: finde 2 Dokumente mit gleicher Signatur)

Hashfunktionen (Urbild)

Wir knacken die Urbildeigenschaft der EAN:

finde eine EAN mit Prüfziffer 7

$a+3b+c+3d+e+3f+g+3h+i+3j+k+3l$

muss als Endziffer 7 haben

Hashfunktionen (2. Urbild)

Wir knacken die 2. Urbildeigenschaft der EAN:

Kenne EAN mit Prüfziffer 4:

978047111709-4

Finde noch eine mit Prüfziffer 4
a+3b+c+3d+e+3f+g+3h+i+3j+k+3l

a und c tauschen ändert nichts am Ergebnis:
879047111709-4

Hashfunktionen (Kollision)

Kollision finden leicht, wenn man 2. Urbild kann

Wichtige Attacke: viele Hashes zufällig erzeugen

prüfen, ob Hashwert schon einmal gesehen

```
978-0312979478 And then there were none
978-0062073563 Murder of Roger Ackroyd
978-0062073495 Murder on the Orient Express
978-0573619236 The Mousetrap
978-0002315968 Miss Marple Final Cases
```

Kollision mit Hashwert 8 gefunden

Hashfunktionen (Anforderungen)

Effizienz

- Anwendungen mit vielen Signaturen
- Signieren von großen Dokumenten
- Signieren von Software

Hashfunktionen (aktuell)

- Momentan eingesetzte Hashes schwach
  MD5, SHA-1

  Interimslösung SHA-256 (eine SHA-2-Funktion)

- SHA-3 Wettbewerb von NIST

Hashwettbewerb NIST

SHA-3 Hashstandard

Runde 1: Nov 2008
Runde 2: Juli 2009
Runde 3: Dez 2010

5 Finalisten Blake, Grøstl, JH, Keccak, Skein
JH: preimage (http://ehash.iaik.tugraz.at/wiki/The_SHA-3_Zoo)

Auswahl Okt. 2012 **Keccak**

Hashfunktionen (Kollisionssuche)

Gesucht: x,y mit h(x)=h(y)

Generische Attacke:

- x1=h(x0)
- x2=h(x1)
- x3=h(x2)
- ...
- bis zwei x-Werte übereinstimmen, z.B.

x27 und x12

dann ist h(x26)=h(x11)

Hashfunktionen (Kollisionssuche)

Algorithmische Fragen:

- Wie lange dauert das?
- Wie erkennt man die Kollision?

Hashfunktionen (Kollision - wie lange dauert das?)

Geburtstagsparadox

bei wievielen Leuten im Raum
darf man davon ausgehen
dass zwei am selben Tag Geburtstag haben?

Hashfunktionen (Kollision - wie lange dauert das?)

Geburtstagsparadox

23 Leute > 50 %

30 Leute > 70 %

50 Leute > 97 %

Hashfunktionen (Kollision - wie lange dauert das?)

$$P(k) = \left(\frac{n-1}{n}\right)\left(\frac{n-2}{n}\right)\cdots\left(\frac{n-k+1}{n}\right) = \prod_{j=1}^{k-1}\left(1-\frac{j}{n}\right)$$

$$k \geq \frac{1}{2} + \frac{1}{2}\sqrt{1+8\log(2)n}$$

Hashfunktionen (Kollision - wie lange dauert das?)

n = Größe Bildraum der Hashfunktion

MD5: 128 Bits, $n=2^{128}$, Kollision in 2^{64} Schritten

SHA-1: 160 Bits, $n=2^{160}$, Kollision in 2^{80} Schritten

Analyse von SHA-1: Verbesserung auf 2^{70} Schritte

Hashfunktionen (wie die Kollision erkennen)

Naive Methode:

alles abspeichern,

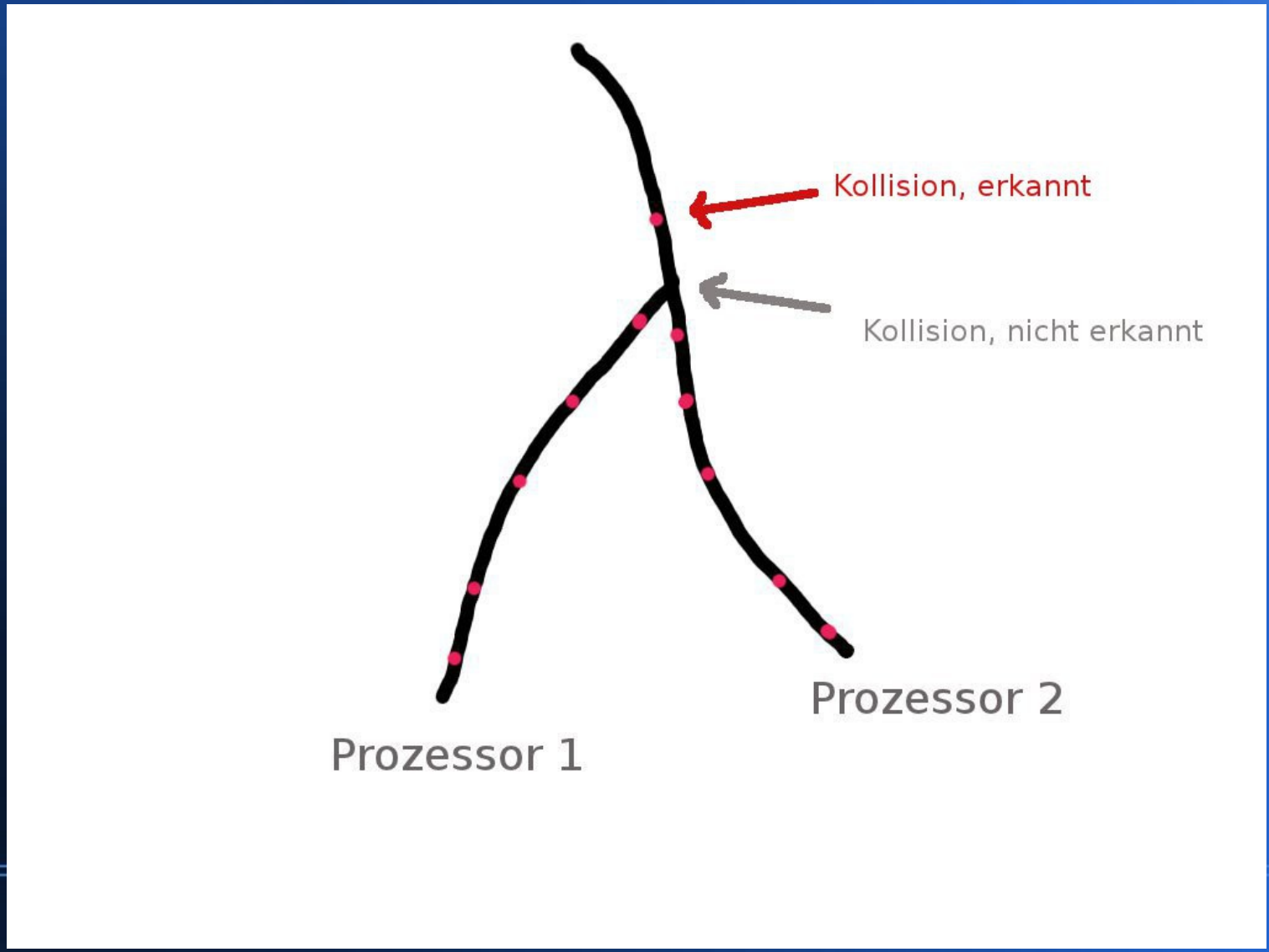
jedesmal testen,
ob bereits gesehen

Hashfunktionen (Kollision hier?)

```
356a192b7913b04c54574d18c28d46e6395428ab
da4b9237bacccdf19c0760cab7aec4a8359010b0
77de68daecd823babbb58edb1c8e14d7106e83bb
1b6453892473a467d07372d45eb05abc2031647a
ac3478d69a3c81fa62e60f5c3696165a4e5e6ac4
c1dfd96eea8cc2b62785275bca38ac261256e278
902ba3cda1883801594b6e1b452790cc53948fda
fe5dbbcea5ce7e2988b8c69bcfdfde8904aabc1f
0ade7c2cf97f75d009975f4d720d1fa6c19f4897
b1d5781111d84f7b3fe45a0852e59758cd7a87e5
17ba0791499db908433b80f37c5fbc89b870084b
7b52009b64fd0a2a49e6d8a939753077792b0554
bd307a3ec329e10a2cff8fb87480823da114f8f4
fa35e192121eabf3dabf9f5ea6abdbcbc107ac3b
f1abd670358e036c31296e66b3b66c382ac00812
1574bddb75c78a6fd2251d61e2993b5146201319
0716d9708d321ffb6a00818614779e779925365c
9e6a55b6b4563e652a23be9d623ca5055c356940
b3f0c7f6bb763af1be91d9e74eabfeb199dc1f1f
91032ad7bbcb6cf72875e8e8207dcfba80173f7c
```

Problem: Speicherplatz (2^{80} Hashwerte)
2417851639229258 3 GB

Pollard-λ


Kollision, erkannt
Kollision, nicht erkannt
Prozessor 2
Prozessor 1

Pollard-λ

Cuda-GPU: Tesla C2050, 448 Kerne

ca 10^{14} SHA's pro Monat,
d.h. Kollisionen auf 80 der 160 Bit möglich^{448}

Master Thesis Manuel Schaeidt 2011

Zusammenfassung

- Hashfunktionen

  „praktisch“ nicht invertierbare Abbildungen

  Attacken brute-force (mit indiv. Beschleunigung)

  SHA-3 Wettbewerb (NIST, siehe SHA-3-Zoo)