Kryptographische Hashfunktionen

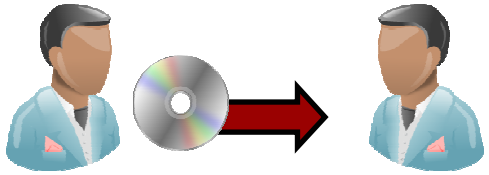
AiS Autonomous Intelligent Systems

Cyrill Stachniss

Basierend auf [Buchmann'08, Daum'05, Lucks & Daum'05, Wang & Yu'05, Sridharan'06, Stevens'07, Sotirov et al.'08, Lucks'07, Gebhard et al.'06, Selinger'06, Wang'04, Dobbertin'96-'98, Menezes et al.'96,…]

Ein typisches Problem

- Sie geben einem Kollegen eine DVD mit Daten



- Zu Hause erhält der Kollege einen Lesefehler beim Kopieren der Daten

- Wie können Sie feststellen, ob die Daten **korrekt kopiert** wurden?

Hashfunktion als Lösung

- Hash steht für zerstückeln/zerhacken
- Hashfunktionen berechnen eine Art **Prüfsumme (Hashwert)** für Dateien
- **Vergleich des Hashwertes** zeigt, ob die Daten korrekt kopiert wurden

Anforderungen

- Unterschiedliche Dokumente sollten zu unterschiedlichen Hashwerten führen
- Kleine Änderungen sollten detektierbar sein
- Hashwerte sollten kurze Zeichenfolgen sein

Potentielle Anwendungen

- Einsatzbereiche von Hashfunktionen:
  - **Änderungen** in Dateien **detektieren** (potentiell über eine langsame Netzverbindung)
  - Prüfsummen, Backups, Datei-Synchronisation
  - Erstellung **rechtsverbindlicher elektronische Unterschriften**
  - Effiziente Speicherung in Hashtabellen
  - …
- **Analogie: Fingerabdruck für Daten**

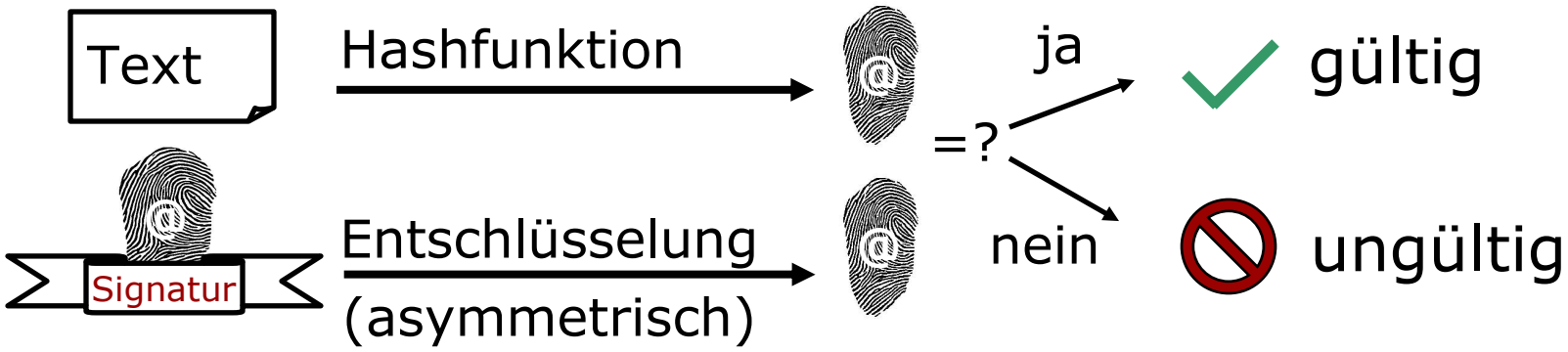
Elektronische Unterschriften (1)

- Es ist lästig, rechtsverbindliche Dokumente ausdrucken, unterschreiben und per Post versenden zu müssen
- Es wäre hilfreich, wenn man Dokumente direkt am Rechner digital unterschreiben und versenden könnte
- Problem dabei:
  - Unterschrift gilt nur für das signierte Dokument, d.h. Unterschrift darf nicht übertragbar sein
  - Eine gültige Unterschrift darf nur vom Unterzeichner erstellt werden können

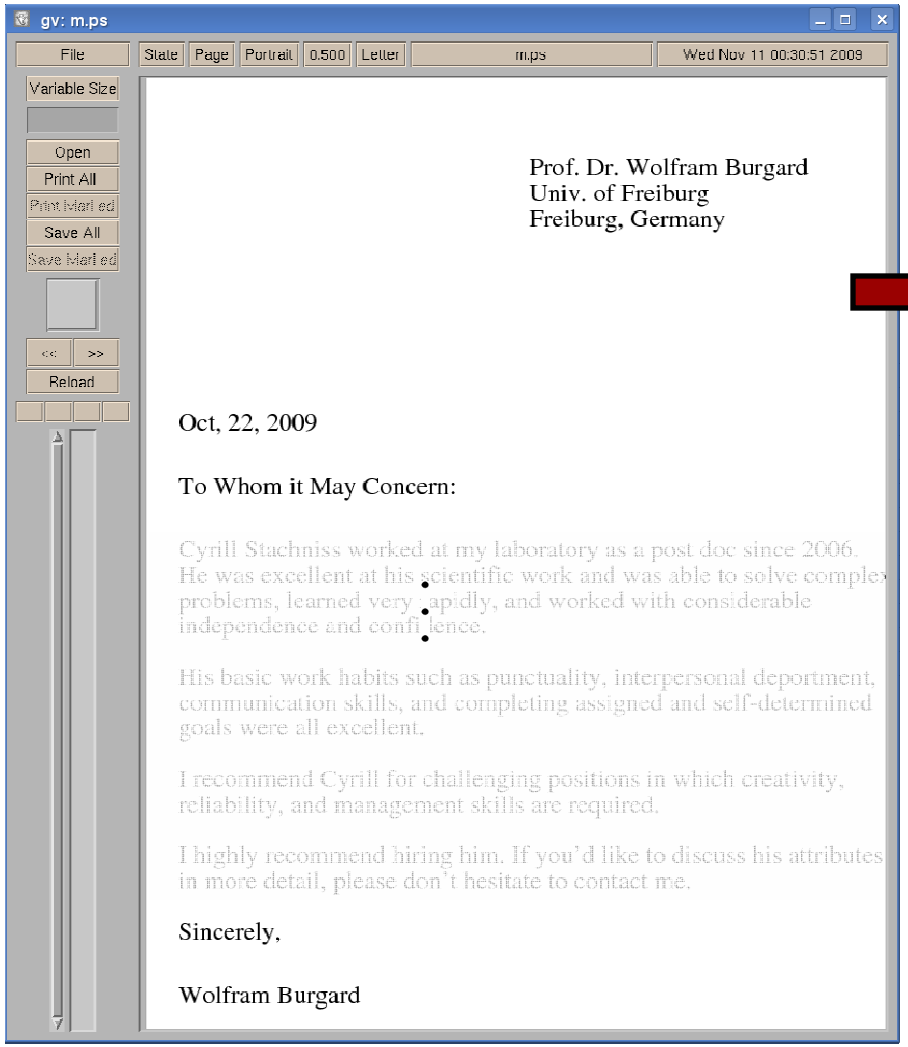
Elektronische Unterschriften (2)

- Signaturverfahren sind meist langsam
- Daher **unterschreibt** man in der Praxis nur **den kurzen Fingerabdruck** der Datei

- Verifizieren der Unterschrift



Beispiel



MD5-Fingerabdruck:
b06ad29719868c1
f15ff86a0bbf72c1a

Fingerabdruck soll nur für dieses Dokument gelten

Kryptographische Hashfunktionen

- Fingerabdruck für elektronische Dokumente
- Funktionen, die beliebig lange Nachrichten auf Nachrichten fester Länge komprimieren

$$h : \{0,1\}^* \mapsto \{0,1\}^n$$
$$X \mapsto h(X)$$

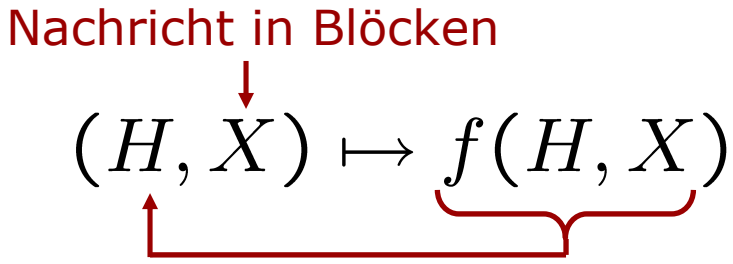
- Im Idealfall eine **Einwegfunktion**, d.h. nicht umkehrbar
- **Effizient** berechenbar
- Unterschiedliche Dokumente sollten zu **unterschiedlichen Hashwerten** führen

Kompressionsfunktionen

- Oft ist es einfacher mit Funktionen zu arbeiten, deren Ein- und Ausgabe Strings feste Länge haben
- Man verwendet Kompressionsfunktionen zur Erstellung einer Hashfunktion

$$f : \{0,1\}^n \times \{0,1\}^l \mapsto \{0,1\}^n$$

- Durch wiederholte Anwendung kann man beliebig lange Strings auf Strings der Länge n abbilden



Kollisionen

- Der Definitionsbereich ist größer als der Wertebereich
- Es gibt daher verschiedene Nachrichten, die den gleichen Hashwert erzeugen
- Dies definiert eine **Kollision** von h:

$$X \neq X' \quad \text{mit} \quad h(X) = h(X')$$

- **Kollisionen sind problematisch**
  - Unterschiedliche Dokumente haben einen identischen Hashwert
  - Gültigkeit elektronischer Unterschriften wird dadurch fragwürdig

Kollisionsresistenz

- **Kollision** einer Hashfunktion

$$h(X) = h(X') \quad \text{mit} \quad X \neq X'$$

- **Pseudo-Kollision** einer Kompressionsfunktion

$$f(H, X) = f(H', X') \quad \text{mit} \quad (H, X) \neq (H', X')$$

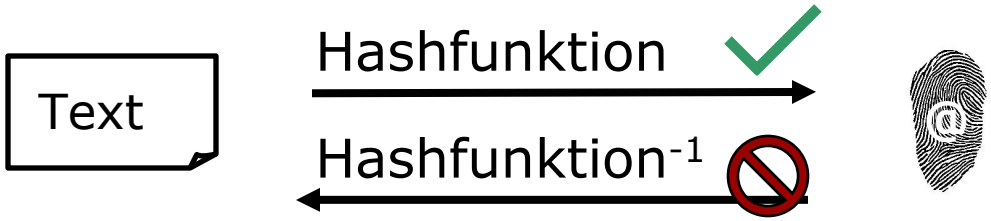
- **Kollisionsresistenz**
  h heißt (stark) kollisionsresistent wenn es keine effiziente Methode gibt um ein Paar (X, X') mit $X \neq X'$ zu finden, so dass $h(X) = h(X')$

- **Pseudo-Kollisionsresistenz**
  definiert sich analog

Ideale kryptographische Hashfunktion

- **Effizient berechenbar**

- **Einwegfunktion**



- **Kollisionsresistent**

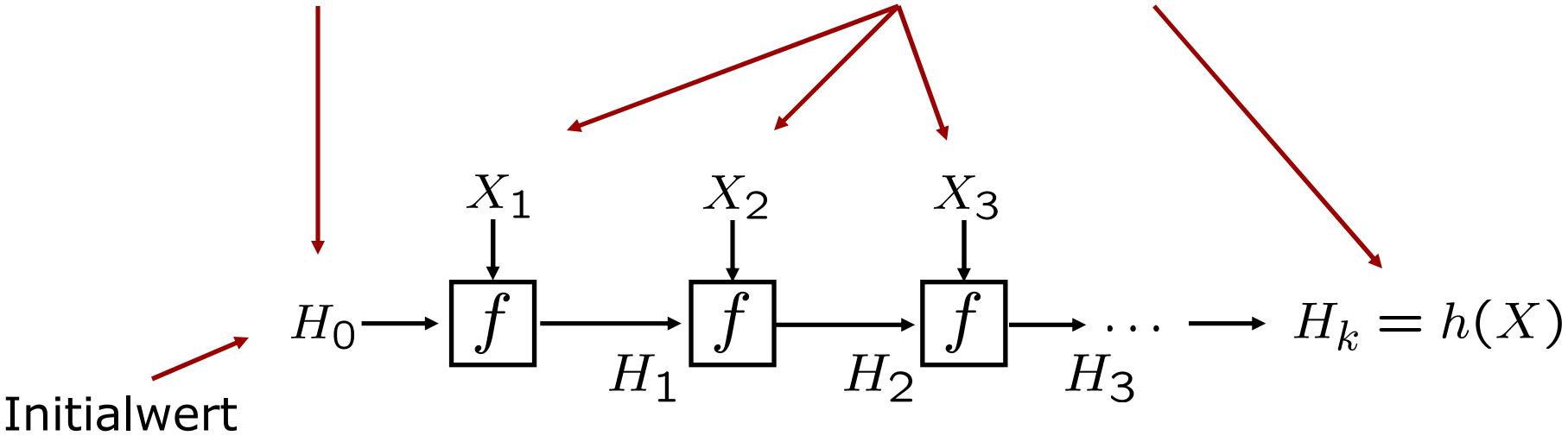
Merkle-Damgård Prinzip (1)

- Prozedur zur wiederholten Anwendung der Kompressionsfunktion
- Technik zum Erweitern der Nachricht, damit diese in Blöcke der Länge l zerlegt werden kann (MD strengthening)
- (Pseudo-) **Kollisionsresistenz** der Kompressionsfunktion **überträgt sich** direkt auf die Hashfunktion
- Das MD Prinzip findet sich in fast allen praktisch eingesetzten Hashfunktionen

Merkle-Damgård Prinzip (2)

1. Erweitere $X \leftarrow X \,||\, 1 \,||\, 0 \ldots 0 \,||\, length(X)$
   so dass $length(X) \bmod l = 0$
2. Zerlege X in Blöcke $X = X_1 \,||\, \ldots \,||\, X_k$
3. Berechne den Hashwert mittels

$$H_0 = \text{const.}; \quad H_i = f(H_{i-1}, X_i); \quad h(X) = H_k$$



Merkle-Damgård Theorem

Sei f eine Kompressionsfunktion, mit der nach dem MD Prinzip eine Hashfunktion h konstruiert wurde. Dann gilt

f pseudo-kollsionsresistent $\Rightarrow$ h kollsionsresistent.

Beweis: Angenommen wir haben ein Verfahren, um für h Kollisionstexte X und X' zu finden.

1. Fall: X und X' haben unterschiedliche Länge. Somit unterscheiden sich die letzten Blöcke X_k und $X'_{k'}$, d.h. $X_k \neq X'_{k'}$. Dadurch ergibt sich eine Pseudokollision von f

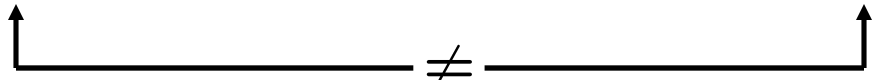
$$f(H_{k-1}, X_k) = h(X) = h(X') = f(H'_{k'-1}, X'_{k'}).$$

$$\neq$$

Merkle-Damgård Theorem

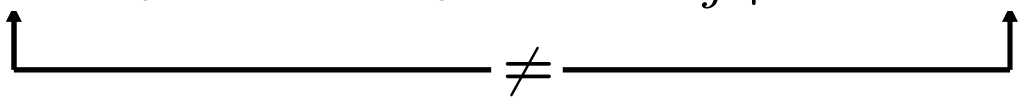
2. Fall: X und X' haben gleiche Länge und für alle Indizes i gilt $H_i = H'_i$.
Es existiert ein Index j, so dass $X_j \neq X'_j$.
Dadurch ergibt sich eine Pseudokollision von f

$$f(H_{j-1}, X_j) = H_j = H'_j = f(H'_{j-1}, X'_j).$$



3. Fall: X und X' haben gleiche Länge und es existiert ein Index i mit $H_i \neq H'_i$.
Sei j der maximale solche Index. Dadurch ergibt sich eine Pseudokollision von f

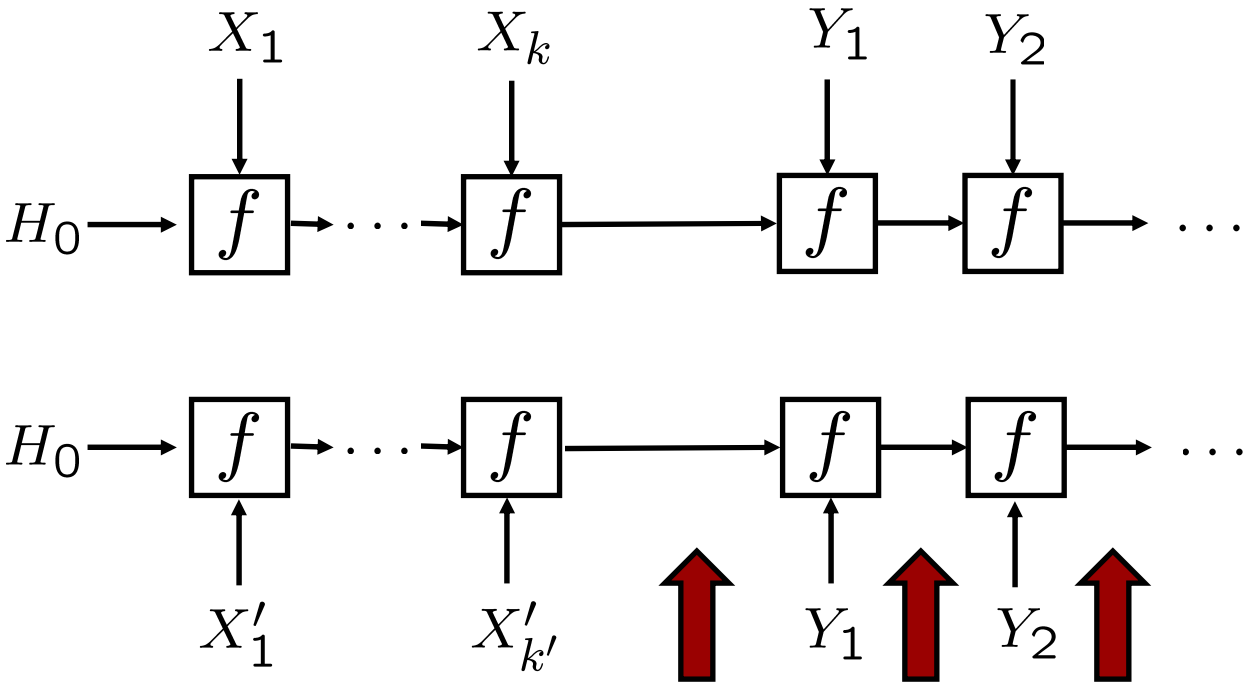
$$f(H_j, X_{j+1}) = H_{j+1} = H'_{j+1} = f(H'_j, X'_{j+1}) \quad \square$$



Strukturelle Schwäche

- Kollisionen können direkt erweitert werden

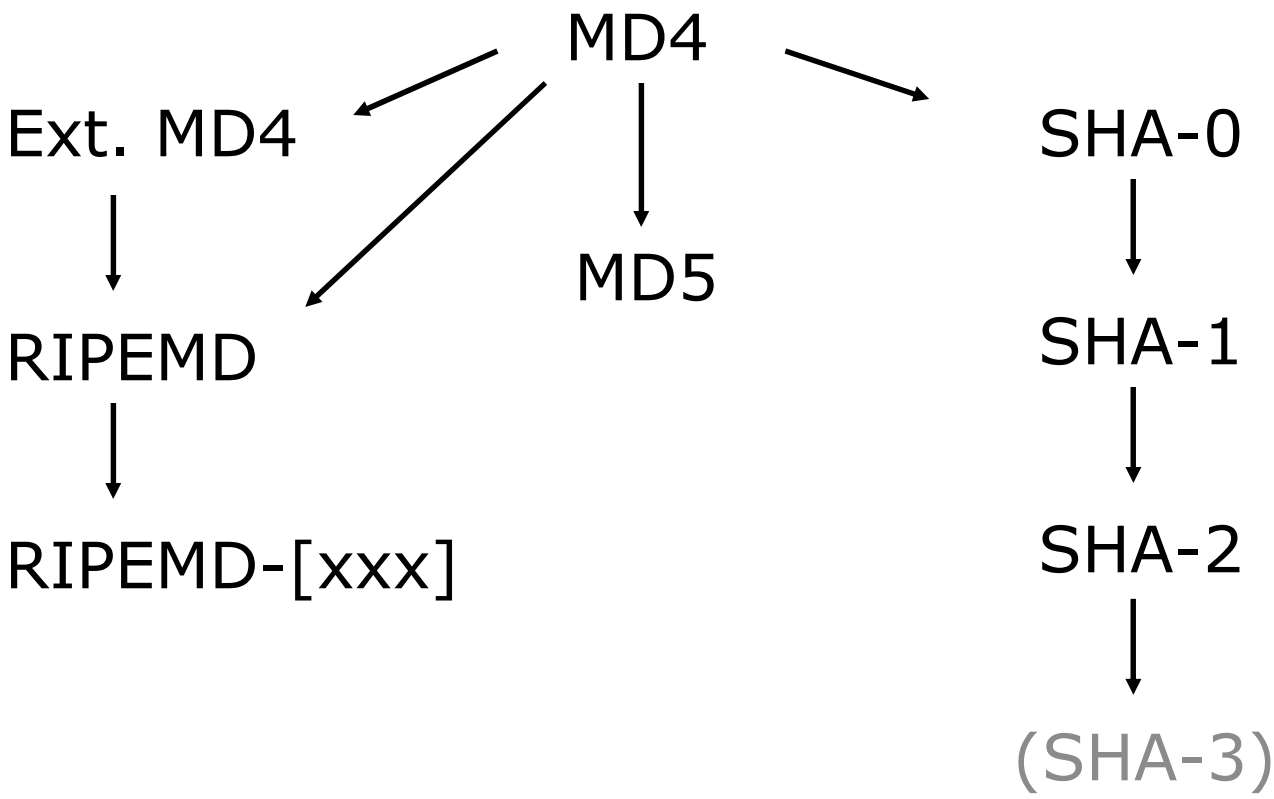
$$f(X) = f(X') \Rightarrow f(X||Y) = f(X'||Y)$$



Kollisionen pflanzen sich fort

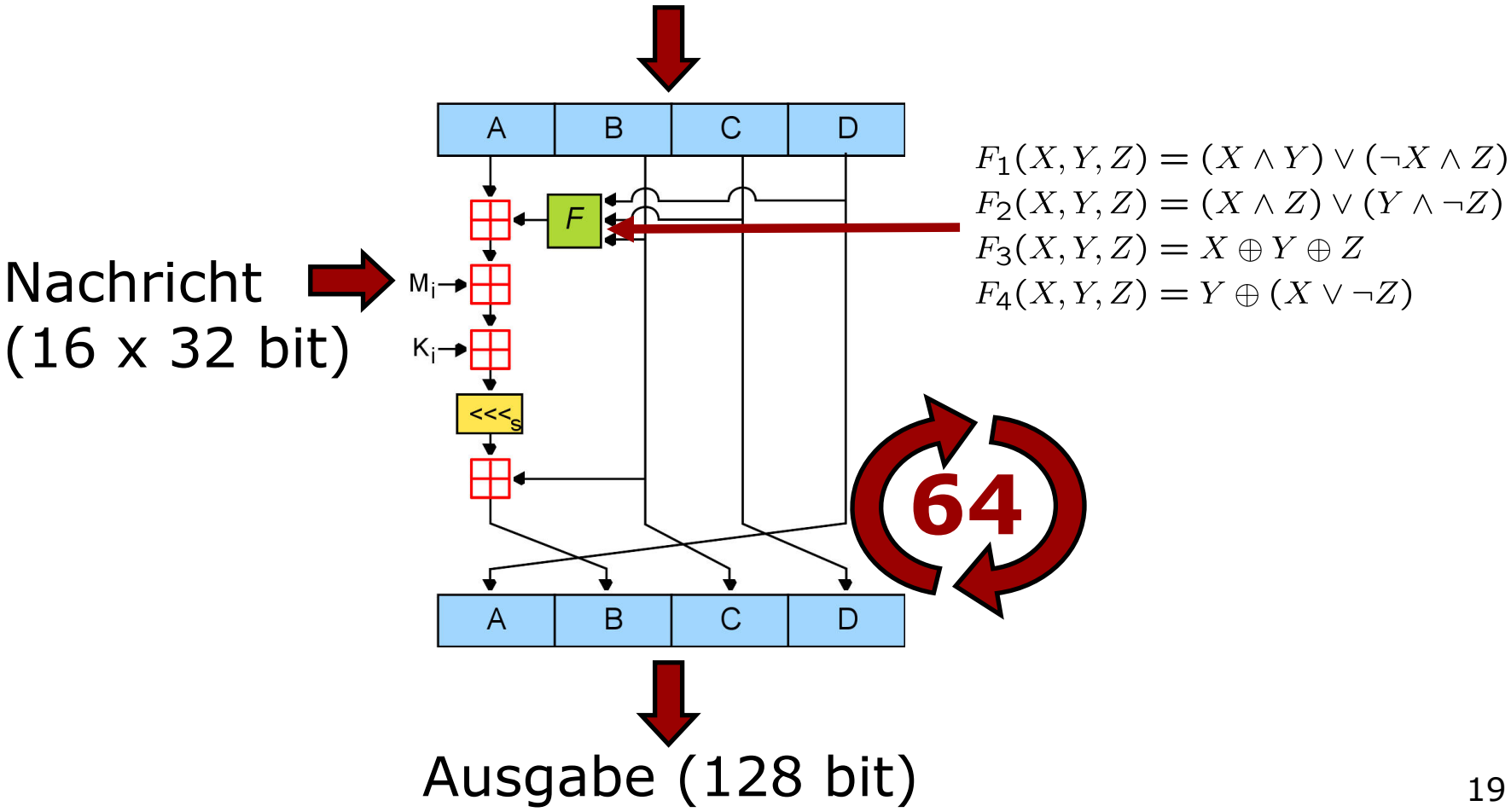
Merkle-Damgård Design

- Fast alle praktischen Hashfunktionen folgen diesem Prinzip
- Die Familie der MD(4) Hashfunktionen



MD5 Überblick

$$f : \{0,1\}^{128} \times \{0,1\}^{512} \mapsto \{0,1\}^{128}$$



Geburtstagsattacke

- Generelle Angriffsmöglichkeit
- Orientiert sich an dem Geburtstagsproblem: „Wie viele Personen benötigt man, damit mit p>0.5 zwei Personen am gleichen Tag Geburtstag haben?“ ➡ Antwort: 23

Umsetzung der Attacke:

- Erstelle zufällige Nachricht X, berechne $h(X)$
- Teste, ob bereits ein identischer Hashwert generiert wurde
  - Nein: Speichere das Tupel und wiederhole
  - Ja: Kollision gefunden

Komplexität der Attacke

- Hashwert mit n-bit Länge
- Man benötigt ungefähr $2^{n/2}$ Versuche um mit p>0.5 eine Kollision zu finden
- Für typische Hashfunktionen heißt dies:

Verfahren	MD4 (128 bit)	MD5 (128 bit)	SHA-1 (160 bit)	SHA-256 (256 bit)
Aufwand	2^{64}	2^{64}	2^{80}	2^{128}

- Der Aufwand bezieht sich auf **Rechenzeit und Speicherplatz**
- Klassische Geburtstagsattacke scheitert oft am fehlenden Speicherplatz

Effizientere Angriffe

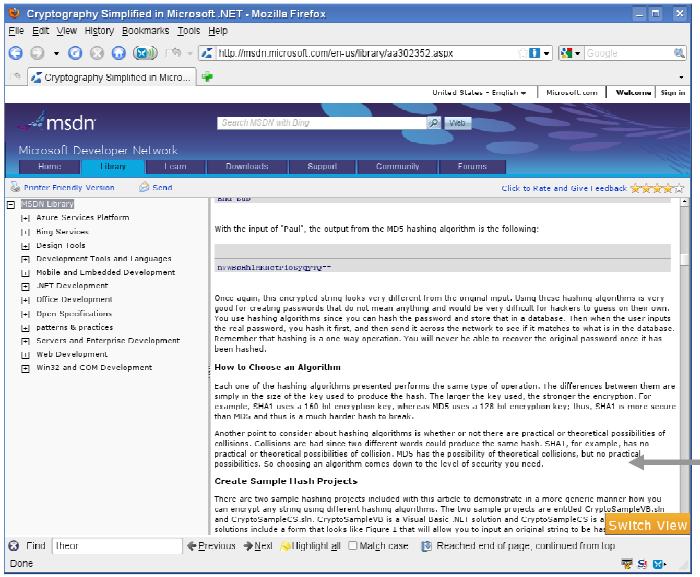
- Es gibt aber für fast alle Hashfunktionen effizientere Methoden zum Finden von Kollisionen

Verfahren	MD4 (128 bit)	**MD5** **(128 bit)**	SHA-1 (160 bit)
Versuche	per hand	**2^{39} / 2^{33} / 2^{30}**	2^{69} / 2^{63} / 2^{55} (?)
Rechenzeit (grob)	(per hand)	**~8-12 Std. (2^{39})** **~5 Min. (2^{33})** **< 1 Min. (2^{30})**	> 30.000 Jahre (2^{69})
Arbeit von	[Wang'04/05]	[Wang & Yu'05] [Klima'05] [Xie et al'08]	[Wang et al.'05] [McDonald'09] (zurückgezogen)

- Diese Art der Attacke liefert eine Kollision für zwei **zufällige** Texte
- Sind **zufällige** Kollisionstexte gefährlich?

Sind Kollisionen gefährlich?

- Microsoft Developer Network, 29.10.2009



How to chose an algorithm

[…] SHA1, for example, has no practical or theoretical possibilities of collisions. **MD5 has the possibility of theoretical collisions, but no practical possibilities**.

[http://msdn.microsoft.com/en-us/library/aa302352.aspx]

Ist dies korrekt?

Szenario

- Cyrill (C.) arbeitet für Wolfram (W.) und fragt nach einem Arbeitszeugnis
- W. schreibt ein Arbeitszeugnis für C.
- C. sendet W. das Arbeitszeugnis als Postscript Dokument zurück und bittet um eine digitale Signatur um seinen Bewerbungsprozess zu vereinfachen
- W. signiert die Postscript Datei für C.

Gefahr durch zufällige Kollisionen

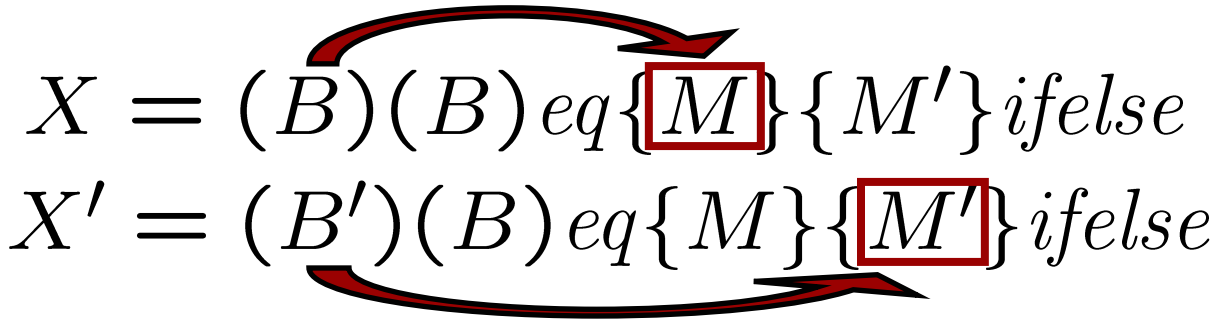
- Selbst **zufällige Kollisionen** stellen ein **praktisches Sicherheitsrisiko** dar
- Dies liegt in der Natur der Nachrichten
- Beispiel: Postscript Dateien erlauben bedingte Anweisungen (if-then-else)

$$(B)(B')eq\{M\}\{M'\}ifelse$$

- Bedingte Anweisungen lassen sich zur Signaturfälschung ausnutzen

Idee zur Signaturfälschung

- Erstelle zwei Postscript Dokumente mit der folgenden Struktur



$$X = (B)(B)eq\{M\}\{M'\}ifelse$$
$$X' = (B')(B)eq\{M\}\{M'\}ifelse$$

- Lasse Dokument X von W. signieren (M ist das Arbeitszeugnis)
- Falls X und X' den gleichen Hashwert erzeugen, gilt die Unterschrift für beide Postscript Dateien und somit für ein Dokument, das den Text M' anzeigt

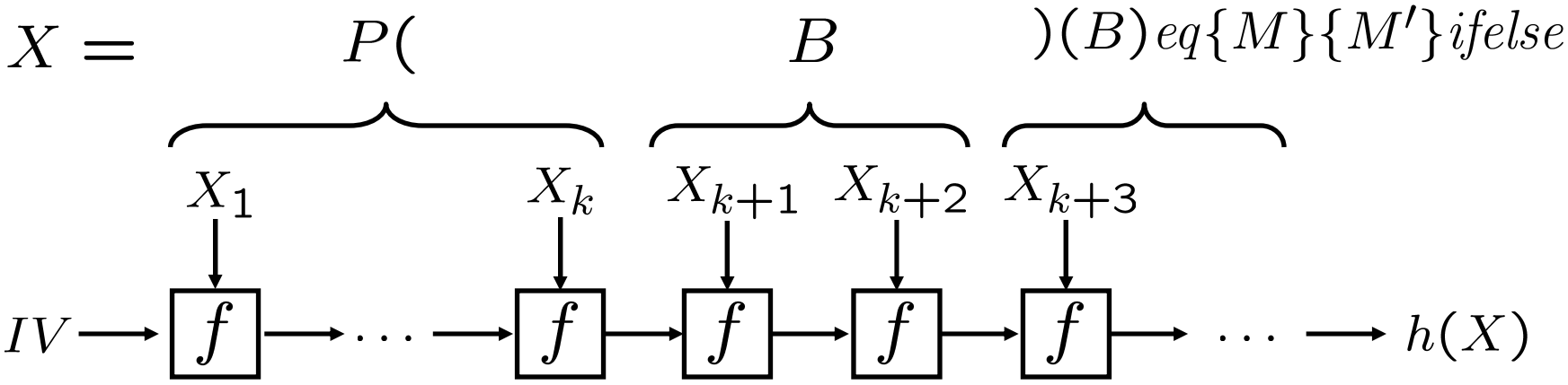
Eine konkrete Fälschung (1)

- Erstelle Postscript Dokument wie folgt

$$X = P(B)(B)eq\{M\}\{M'\}ifelse$$

wobei P der Postscript Header gefolgt von Kommentarzeichen ist, so dass

$length(\text{"}\,P(\text{"}\,)\bmod blocksize = 0$

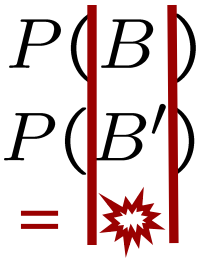


Methode von Wang & Yu [2005]

- Erstelle die Kollisionstexte B, B' für MD5 mittels der Methode von [Wang & Yu'05]
- Methode findet für beliebigen gegebenen Zwischenhashwert H zwei (zufällige) 1024-bit Strings B, B', die zu einer Kollision führen
- D.h. $f(f(H, \underbrace{B_1}_{B}), B_2) = f(f(H, \underbrace{B'_1}_{B'}), B'_2)$
- Aufwand 2^{39} Operationen (~10 Stunden)

Eine konkrete Fälschung (2)

- Setze $X = P(B)(B)eq\{M\}\{M'\}ifelse$
  $X' = P(B')(B)eq\{M\}\{M'\}ifelse$
  $= \quad =$

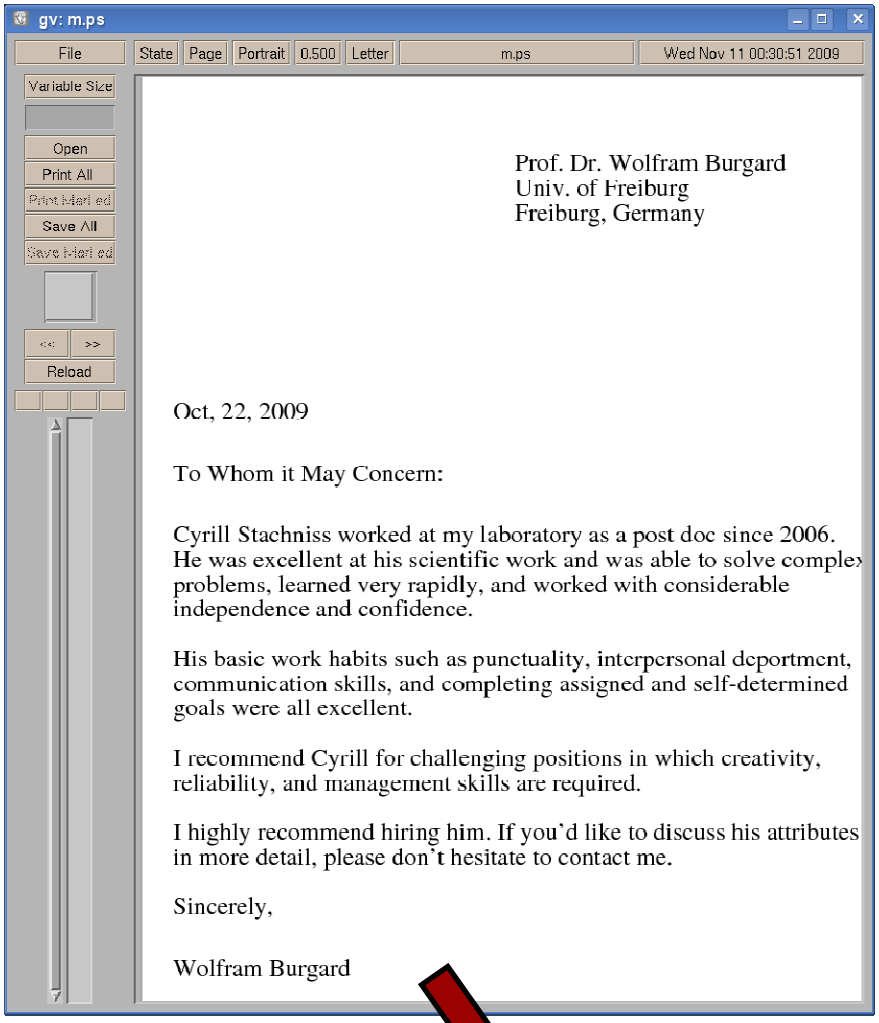


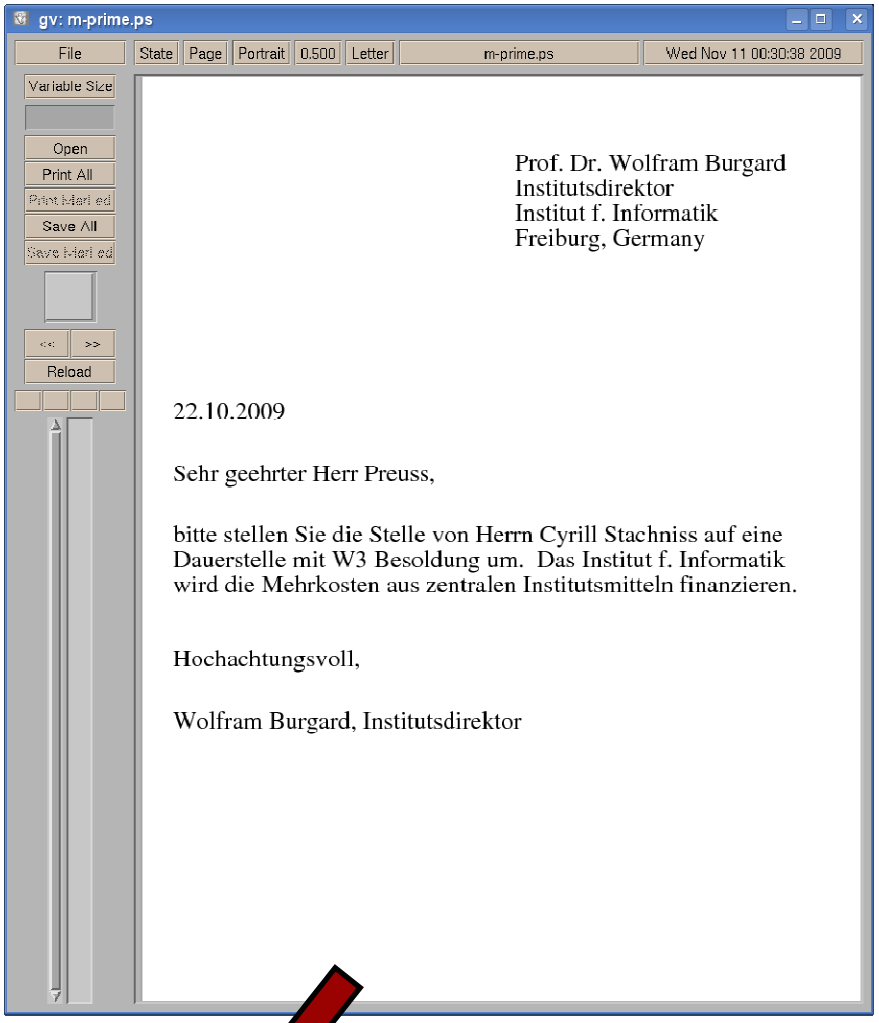
- Da sich aufgrund des MD Designs Kollisionen fortpflanzen, folgt

$$h(X) = h(X')$$

- Wir haben also aus zwei **zufälligen** Kollisionstexten B, B' zwei beliebige (sinnvolle) Nachrichten mit gleichem MD5-Hashwert erstellt

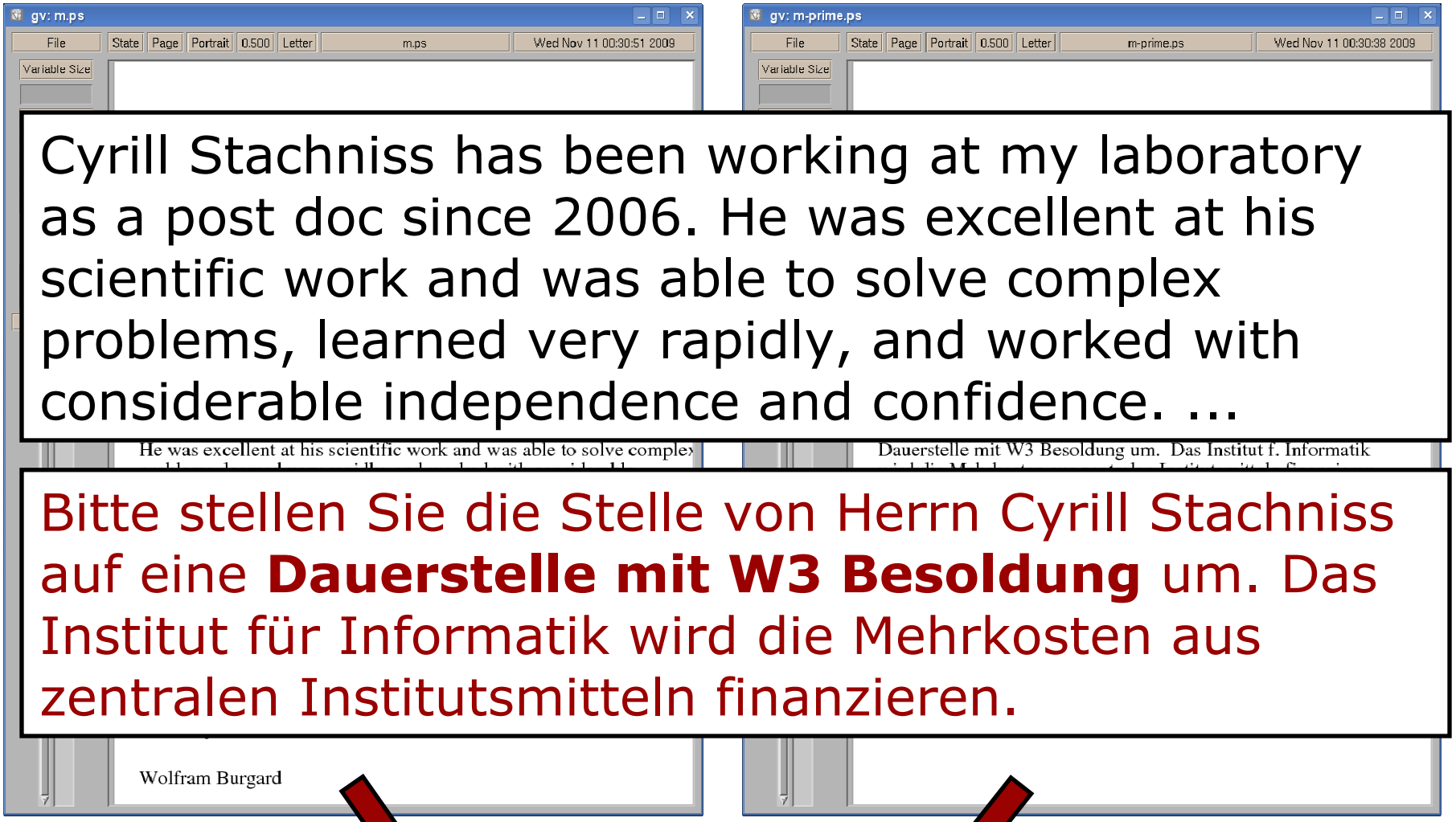
Eine konkrete Fälschung (3)





MD5: b06ad29719868c1f15ff86a0bbf72c1a

Eine konkrete Fälschung (3)



Cyrill Stachniss has been working at my laboratory as a post doc since 2006. He was excellent at his scientific work and was able to solve complex problems, learned very rapidly, and worked with considerable independence and confidence. ...

Bitte stellen Sie die Stelle von Herrn Cyrill Stachniss auf eine **Dauerstelle mit W3 Besoldung** um. Das Institut für Informatik wird die Mehrkosten aus zentralen Institutsmitteln finanzieren.

MD5: b06ad29719868c1f15ff86a0bbf72c1a

Signatur

Andere Dokumente/Formate

- **PDF**: Kein if-then-else aber Farbtabellen, in denen man die Kollisionstexte verstecken sowie unterschiedliche Textfarben setzen kann (schwarzer bzw. weißer Text)
- **MS Word**: Via Makros und ungenutzten/reservierten Bereichen
- **TIF**: Offset von Bildpartien verschieben
- **…**

Zufällige Kollisionen sind ein Sicherheitsrisiko, MD5 sollte nicht mehr verwendet werden

Alternativen zur MD5 Hashfunktion

- SHA-1 strukturell ähnlich zu MD5
- SHA-2 bisher ungebrochen (strukturell ähnlich zu SHA-1 aber mit deutlich längerem Hashwert (224-512 Bit)
- SHA-2 folgt dem Merkle-Damgård Prinzip
- SHA-3 Wettbewerb hat im November 2007 begonnen
- SHA-3 Finalisten im 3. Quartal 2010
- Neuer SHA-3 Standard Ende 2012
- Aktuell sind nur noch 14 von 64 Hashfunktionen im Rennen

Zusammenfassung

- Einführung in kryptographische Hashfunktionen
- Merkle-Damgård Design
- Attacken auf Hashfunktionen
- Praktische und effiziente Methode zur Signaturfälschung unter Verwendung zufälliger Kollisionen
- Kritische Anwendungen sollten keinesfalls MD5 verwenden
- Zur Zeit gilt SHA-2 noch als sicher