Willkommen zu

IT Security - Tutorium 5

:)

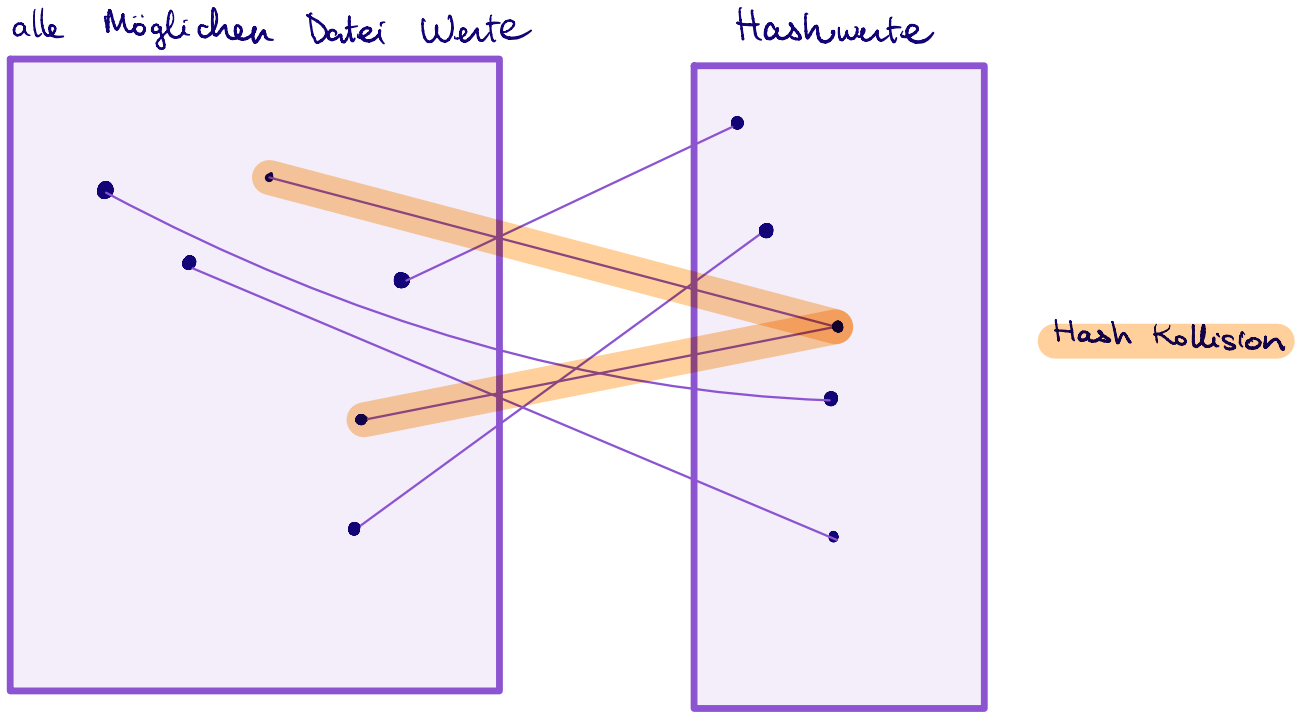
Nguyet Ha (Hannah) Nguyenová
Montag, 20. November 2023

Kryptographie III

1 Eigenschaften von kryptographischen Hashfunktionen

Hash Funktionen

- Input variabler Länge
- Output fixer Länge (digest, hash)
- surjektiv, nicht injektiv



1 Eigenschaften von kryptographischen Hashfunktionen

a) Erklären Sie kurz die vier Kriterien für sichere Hashfunktionen in eigenen Worten

1 Eigenschaften von kryptographischen Hashfunktionen

a) Erklären Sie kurz die vier Kriterien für sichere Hashfunktionen in eigenen Worten

für eine Hashfunktion H

1) Hash effizient berechenbar

2) Urbildresistenz (pre-image resistance)

- H^{-1} nicht effizient berechenbar
- aus Kenntniss des Hashes Informationen über die ursprüngliche Nachricht nicht erhaltbar

3) schwache Kollisionsresistenz (second pre-image resistance)

- gegeben Nachricht m mit Hash $H(m)$
- schwierig $m' \neq m$ ein $H'(m) = H(m)$ zu finden

4) starke Kollisionsresistenz

- $\forall m$ mit Hash $H(m)$ schwierig ein $m' \neq m$ ein $H'(m) = H(m)$ zu finden

1 Eigenschaften von kryptographischen Hashfunktionen

fiktive Hashfunktion *MH5* (Manuels Hashfunktion)

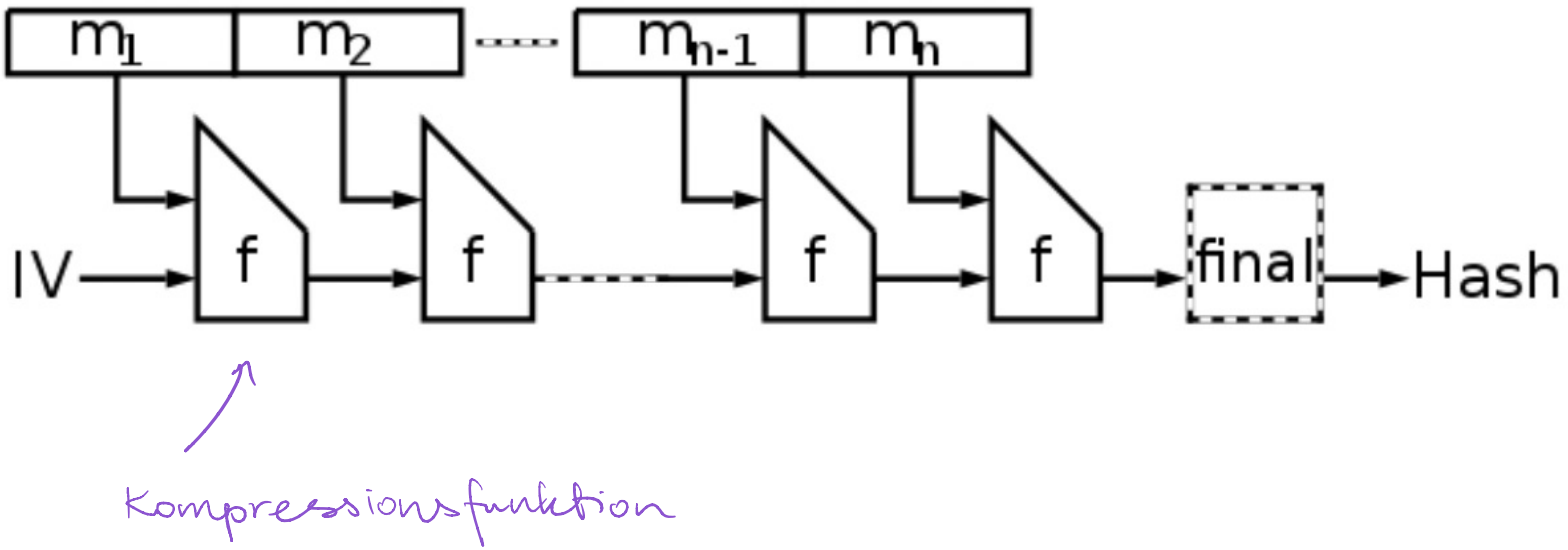
Gegeben sei die *fiktive* Hashfunktion ihrer Übungsleitung: *MH5* (Manuels Hashfunktion No.5). MH5 ist nach der Merkle-Damgård-Konstruktion aufgebaut. Die Kompressionsfunktion f, die Blöcke zu jeweils 32-Bit verarbeitet, sei wie folgt definiert:

$$f(x, y) = x + y \mod 2^{32}$$

Der Startzustand (und damit gleichzeitig der leere Hash) sei 0. Die Eingaben für f, x und y, werden für die Addition im Big-Endian Format verarbeitet.

1 Eigenschaften von kryptographischen Hashfunktionen

Merkle-Damgård-Konstruktion



Message in Blöcke unterteilen
Kompressionsfunktion hintereinander geschaltet

1 Eigenschaften von kryptographischen Hashfunktionen

b) Welche Länge hat die Ausgabe (Digest) von MH5 in Bit?

1 Eigenschaften von kryptographischen Hashfunktionen

b) Welche Länge hat die Ausgabe (Digest) von MH5 in Bit?

32-bit :) (Hintereinenderschalten der Kompressionsfunktion f)

1 Eigenschaften von kryptographischen Hashfunktionen

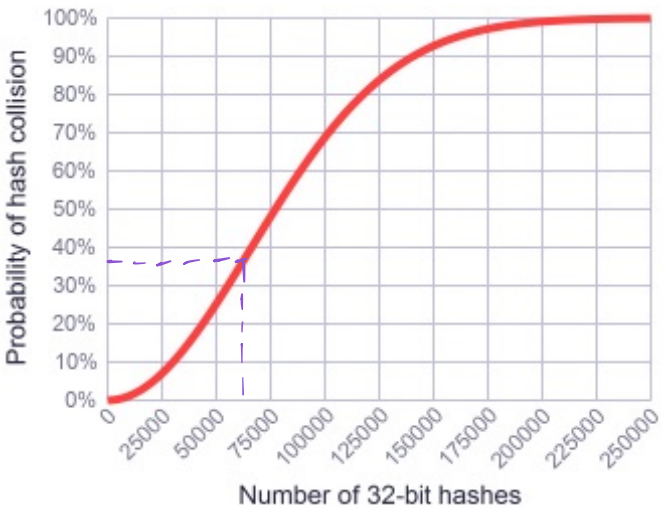
c) Beurteilen Sie, ob die Länge des Digests von MH5 ausreichend groß für sicherheitstechniche Anwendungen ist

1 Eigenschaften von kryptographischen Hashfunktionen

c) Beurteilen Sie, ob die Länge des Digests von MH5 ausreichend groß für sicherheitstechniche Anwendungen ist

Nicht ausreichend - sehr hohe Kollisionswahrscheinlichkeit schon bei 2^{16} Hashes

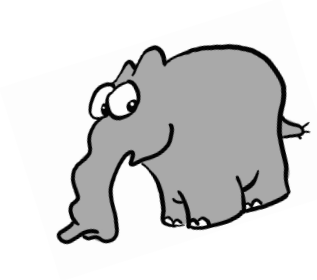
$2^{16} = 65\,536$

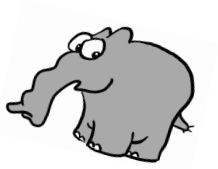


Number of **32-bit** hash values	Number of **64-bit** hash values	Number of **160-bit** hash values	Odds of a hash collision	
77163	5.06 billion	1.42×10^{24}	1 in 2	
30084	1.97 billion	5.55×10^{23}	1 in 10	
9292	609 million	1.71×10^{23}	1 in 100	Odds of a full house in poker 1 in 693
2932	192 million	5.41×10^{22}	1 in 1000	Odds of four-of-a-kind in poker 1 in 4164
927	60.7 million	1.71×10^{22}	1 in 10000	
294	19.2 million	5.41×10^{21}	1 in 100000	Odds of being struck by lightning 1 in 576000
93	6.07 million	1.71×10^{21}	1 in a million	Odds of winning a 6/49 lottery 1 in 13.9 million
30	1.92 million	5.41×10^{20}	1 in 10 million	
10	607401	1.71×10^{20}	1 in 100 million	Odds of dying in a shark attack 1 in 300 million
	192077	5.41×10^{19}	1 in a billion	
	60740	1.71×10^{19}	1 in 10 billion	
	19208	5.41×10^{18}	1 in 100 billion	
	6074	1.71×10^{18}	1 in a trillion	
	1921	5.41×10^{17}	1 in 10 trillion	Odds of a meteor landing on your house 1 in 182 trillion
	608	1.71×10^{17}	1 in 100 trillion	
	193	5.41×10^{16}	1 in 10^{15}	
	61	1.71×10^{16}	1 in 10^{16}	
	20	5.41×10^{15}	1 in 10^{17}	
	7	1.71×10^{15}	1 in 10^{18}	

https://preshing.com/20110504/hash-collision-probabilities/

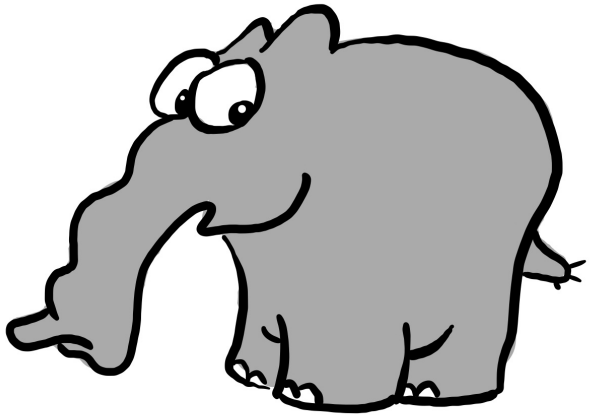
1 Eigenschaften von kryptographischen Hashfunktionen





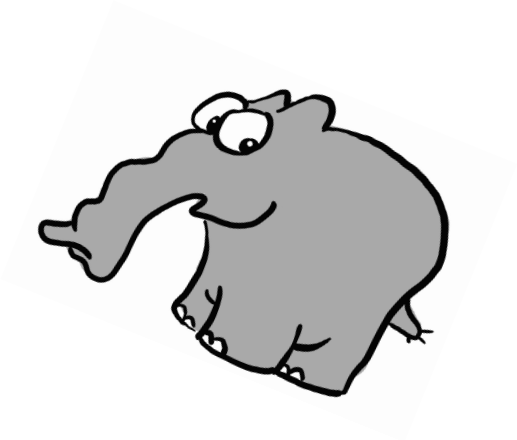
d) Berrechnen Sie den Hashwert für die Eingabe "Ottifant". Gehen Sie davon aus, dass die Zeichenkette im ASCII Encoding kodiert ist.

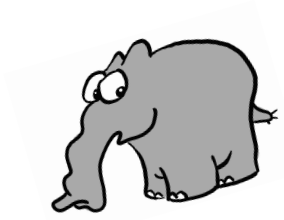
O	t	t	i	f	a	n	t



1 Eigenschaften von kryptographischen Hashfunktionen

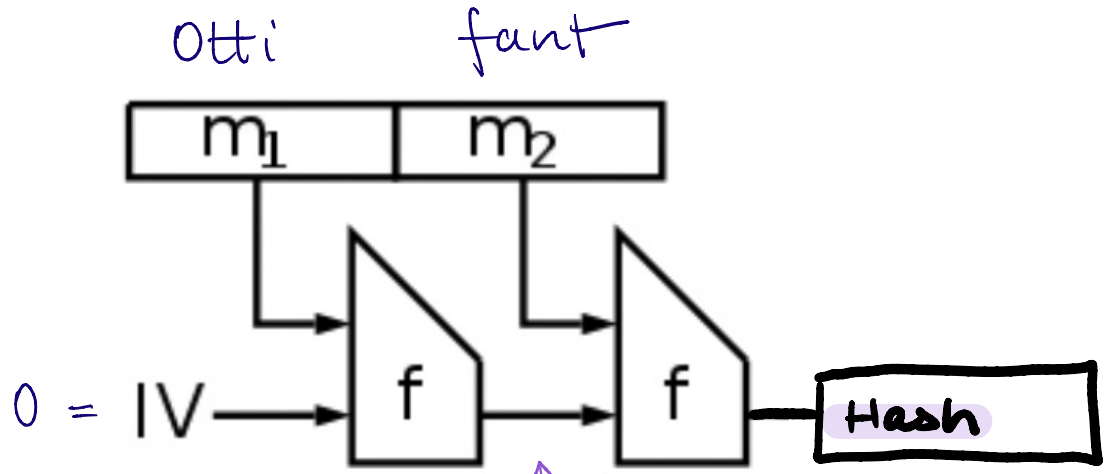
d) Berrechnen Sie den Hashwert für die Eingabe "Ottifant". Gehen Sie davon aus, dass die Zeichenkette im ASCII Encoding kodiert ist.





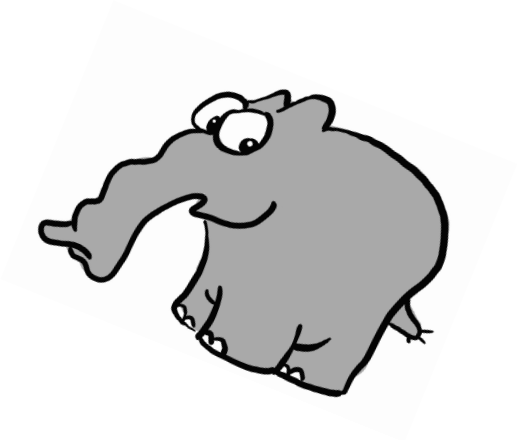
1 Eigenschaften von kryptographischen Hashfunktionen

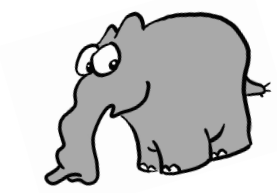
d) Berrechnen Sie den Hashwert für die Eingabe "Ottifant". Gehen Sie davon aus, dass die Zeichenkette im ASCII Encoding kodiert ist.



0 + Otti
= 0 + 0x4f747469 = 0x4f747469

Otti + fant
= 0x4f747469 + 0x66616e74 = 0xb5d5e2dd





1 Eigenschaften von kryptographischen Hashfunktionen

e) Welchen der Kriterien für sichere Hashfunktionen entspricht MH5? Zeigen Sie ggf. anhand geeigneter Gegenbeispiele welchen Kriterien für sichere Hashfunktionen MH5 nicht entspricht

1 Eigenschaften von kryptographischen Hashfunktionen

e) Welchen der Kriterien für sichere Hashfunktionen entspricht MH5? Zeigen Sie ggf. anhand geeigneter Gegenbeispiele welchen Kriterien für sichere Hashfunktionen MH5 nicht entspricht

1) Hash effizient berechenbar

2) Urbildresistenz

3) schwache Kollisionsresistenz

4) starke Kollisionsresistenz

1 Eigenschaften von kryptographischen Hashfunktionen

e) Welchen der Kriterien für sichere Hashfunktionen entspricht MH5? Zeigen Sie ggf. anhand geeigneter Gegenbeispiele welchen Kriterien für sichere Hashfunktionen MH5 nicht entspricht

1) Hash effizient berechenbar

JA Addition und Modulo rechnen

2) Urbildresistenz

NEIN für Fall der Nachrichten der Länge ≤ 32 bit

4) starke Kollisionsresistenz

NEIN Digest Länge ist zu kurz

$\forall m$: findet man m' mit $\mathrm{MH5}(m) = \mathrm{MH5}(m')$: $m' = m \,||\, 00000000_{16}$

3) schwache Kollisionsresistenz

NEIN 4) $\Longrightarrow$ 3)

1 Eigenschaften von kryptographischen Hashfunktionen

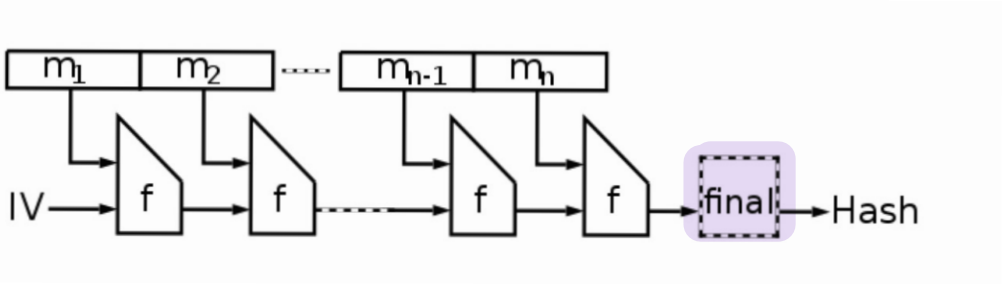
f) Welches Problem tritt aktuell auf, wenn Sie die Zeichenkette „Fabian“ hashen wollen?
Schlagen Sie ein Verfahren für die Berechnung eines MH5-Hashes von Nachrichten mit beliebiger Länge vor

1 Eigenschaften von kryptographischen Hashfunktionen

f) Welches Problem tritt aktuell auf, wenn Sie die Zeichenkette „Fabian“ hashen wollen?
Schlagen Sie ein Verfahren für die Berechnung eines MH5-Hashes von Nachrichten mit beliebiger Länge vor

Padding um die Anforderungen der Kompressionsfunktion aufzufüllen (z.B. mit Nullen)

Bsp. für Finalisierungsschritt (bekannt und umkehrbar):
Padding für Sicherheitsgarantien, damit Nachrichten nicht Präfixe voneinander sind
Länge der gehashten Eingabe im letzten Block



2 Kollisionen in Hash-Funktionen

a) Welche Gefahr geht davon aus, wenn eine Hashfunktion nicht mehr stark kollisionsresistent ist?

2 Kollisionen in Hash-Funktionen

a) Welche Gefahr geht davon aus, wenn eine Hashfunktion nicht mehr stark kollisionsresistent ist?

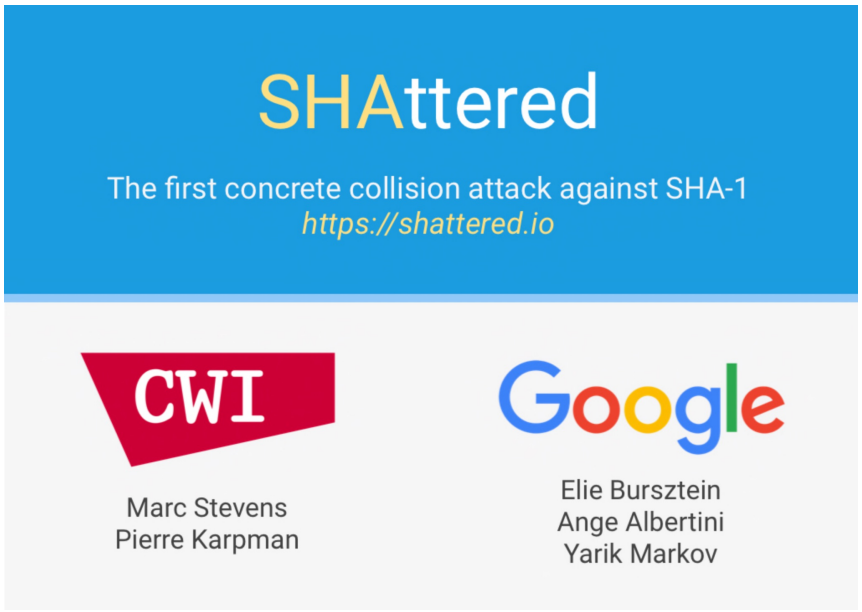
- $\exists\ x,\ y$ mit h(x) = h(y)
- Hash in digitale Signaturen: bei Kollision Daten nachträglich verändern

2 Kollisionen in Hash-Funktionen

Bsp: Scoreboard git-repo :)

```
commit 3b0ed001ad97e335713b8ef038dfea17955bf501
Author: Fabian Franzen <franzen@sec.in.tum.de>
Date:   Thu Oct 26 11:26:19 2023 +0200

    Adjusted port to gunicorn in README

commit 48b41ac48d28fb8f381f33af55eec943524ae170
Author: Fabian Franzen <franzen@sec.in.tum.de>
Date:   Thu Oct 26 11:25:20 2023 +0200

    Fixing Dockerfile

commit fe7b451f99f15326ec0203e6cf746a0f6c871609
Author: Carl <carl.koenig@tum.de>
Date:   Thu Oct 26 04:43:58 2023 -0400

    required presentations can be changed in config file
```

2 Kollisionen in Hash-Funktionen



shattered-1.pdf

shattered-2.pdf

```
$ sha1sum shattered-*.pdf

38762cf7f55934b34d179ae6a4c80cadccbb7f0a  shattered-1.pdf
38762cf7f55934b34d179ae6a4c80cadccbb7f0a  shattered-2.pdf

$ sha256sum shattered-*.pdf

2bb787a73e37352f92383abe7e2902936d1059ad9f1ba6daaa9c1e58ee6970d0  shattered-1.pdf
d4488775d29bdef7993367d541064dbdda50d383f89f0aa13a6ff2e0894ba5ff  shattered-2.pdf
```

2 Kollisionen in Hash-Funktionen

b) Begründen Sie warum für MD5 folgende Eigenschaft gilt: Wenn zwei 64-Byte-Blöcke x und y denselben Hashwert $MD5(x) = MD5(y)$ haben, dann gilt für jeden beliebigen 64-Byte-Block k:

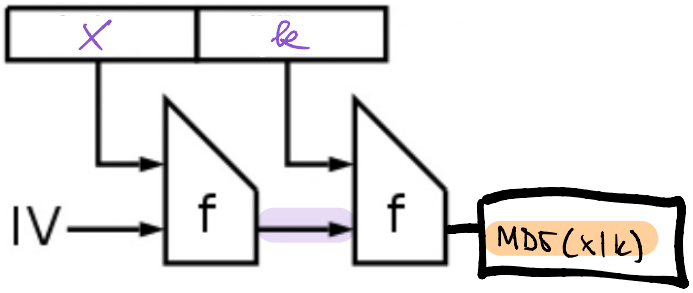
$$MD5(x|k) = MD5(y|k)$$

2 Kollisionen in Hash-Funktionen

b) Begründen Sie warum für MD5 folgende Eigenschaft gilt: Wenn zwei 64-Byte-Blöcke x und y denselben Hashwert $MD5(x) = MD5(y)$ haben, dann gilt für jeden beliebigen 64-Byte-Block k:

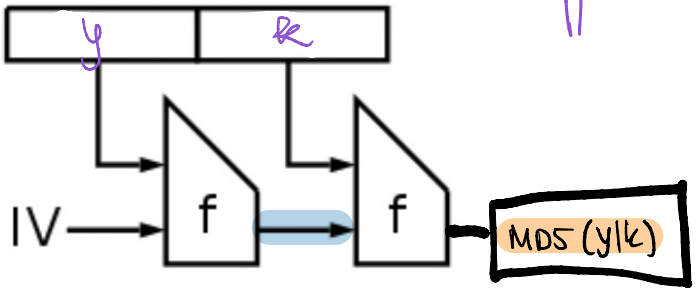
$$MD5(x|k) = MD5(y|k)$$

für x



=

für y



2 Kollisionen in Hash-Funktionen

c) Bei Kollisionsangriffen unterscheiden wir zwischen einem klassichen Kollisionsangriff und einem Chosen-prefix Kollisionsangriff.

Bei einem klassischen Kollisionsangriff finden wir zwei Nachrichten m_1, m_2 , sodass $H(m_1) = H(m_2)$.
Bei einem Chosen-prefix Kollisionsangriff, findet ein Angreifer für zwei gegebene, unterschiedliche Prefixe p_1, p_2, zwei Anhänge, sodass gilt $H(p_1||m_1) = H(p_2||m_2)$.

Seit 2007 gibt es auf die Hashfunktion MD5 auch einen Chosen-Prefix Kollisionsangriff.
Skizzieren Sie mit Ihrem Wissen nun jeweils einen Angriff für jede Art des Kollisionsangriffes!

2 Kollisionen in Hash-Funktionen

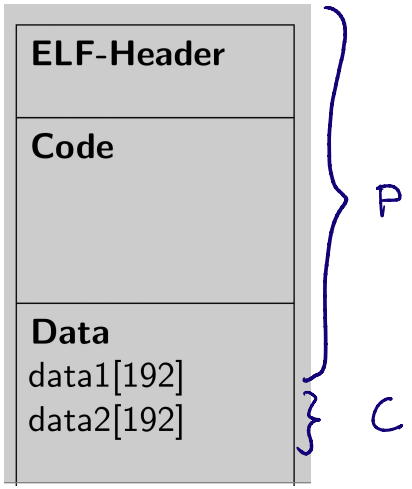
c) Bei Kollisionsangriffen unterscheiden wir zwischen einem klassichen Kollisionsangriff und einem Chosen-prefix Kollisionsangriff.
Bei einem klassischen Kollisionsangriff finden wir zwei Nachrichten m_1, m_2 , sodass $H(m_1) = H(m_2)$.

Szenario: Wir erstellen ein Programm, das scheinbar harmlosen Code ausführt (korrekt funktioniert), aber bauen zusätzlich eine verborgene "böse" Funktionalität

$\Longrightarrow$ finde Kollisionsblöcke $C \neq C'$, sodass bei Prefix P: $H(P|C) = H(P|C')$

Gute Version : data1 == data2
Böse Version : data1 ≠ data2

```
1 char data1[192] = {...}
2 char data2[192] = {...}
3
4 if (memcmp(data1, data2, 192) == 0) {
5   /* good_program */
6 }
7 else {
8   /* evil_program */
9 }
```



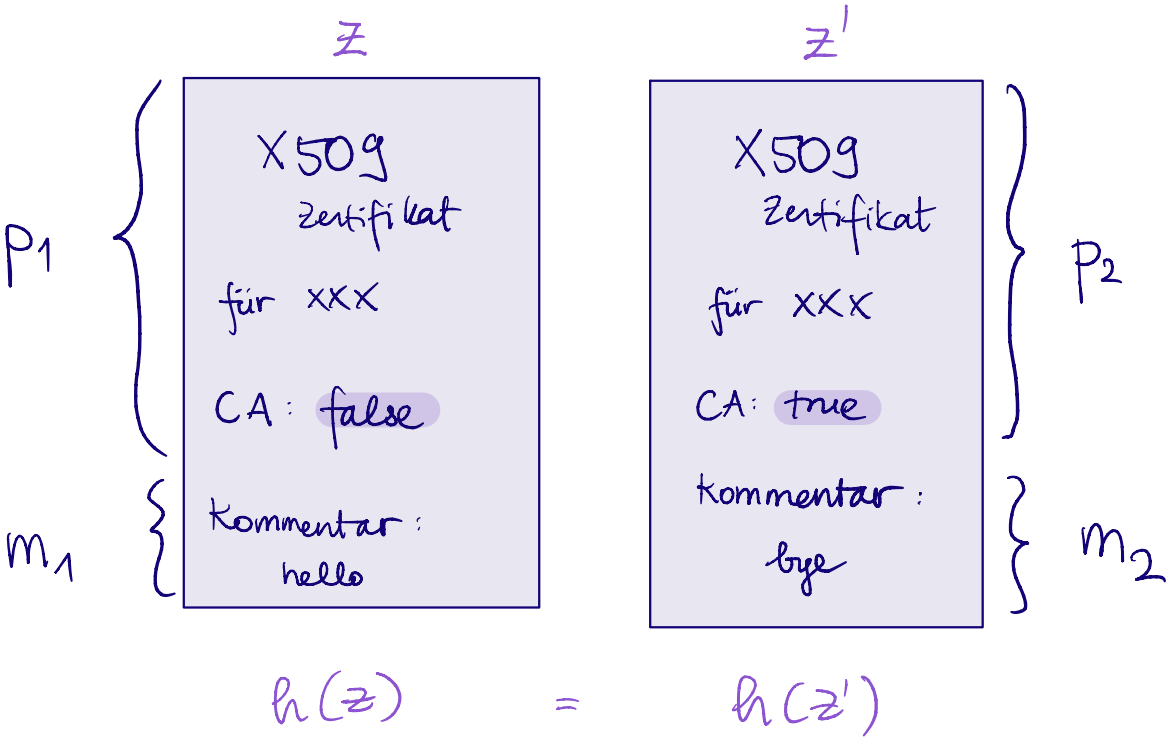
2 Kollisionen in Hash-Funktionen

c) Bei Kollisionsangriffen unterscheiden wir zwischen einem klassichen Kollisionsangriff und einem Chosen-prefix Kollisionsangriff.

Bei einem Chosen-prefix Kollisionsangriff, findet ein Angreifer für zwei gegebene, unterschiedliche Prefixe p_1, p_2, zwei Anhänge, sodass gilt $H(p_1||m_1) = H(p_2||m_2)$.

m_1, m_2

Szenario: Angriff auf PKI (public key infrastructure) mit X509 Zertifikaten

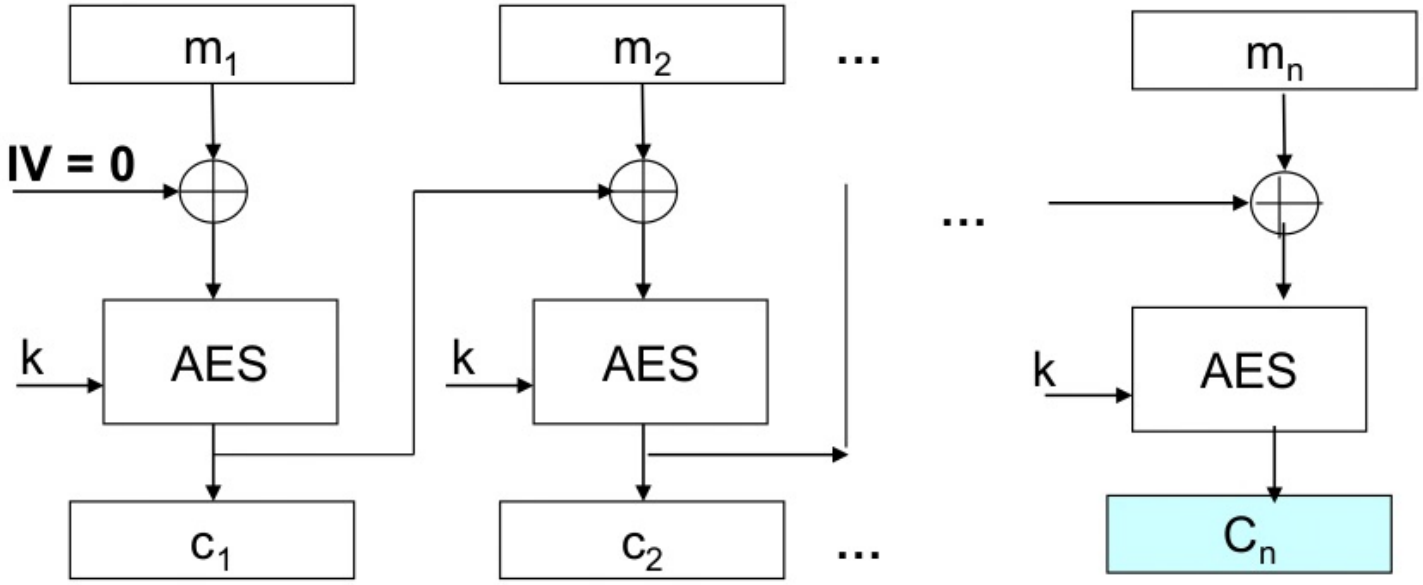


Hash von z wird Signiert

⇓

z' mit Signiertem Hash von z (h(z)) verwenden

3 Message-Authentication-Codes



Integrität und Authentizität

- bekannt: AES-CBC, AES-GCM

3 Message-Authentication-Codes

a) Warum ist diese Art von Message-Authentication Code als kritisch einzustufen?

$$h = MAC(m, k_{AB}) = H(m|k_{AB})$$

3 Message-Authentication-Codes

a) Warum ist diese Art von Message-Authentication Code als kritisch einzustufen?

$$h = MAC(m, k_{AB}) = H(m|k_{AB})$$

MAC soll sicher sein, egal welches Hashverfahren verwendet wird (inkl. SHA-1, MD5)
Es reicht eine Kollision mit m, um MAC für jeden beliebigen Schlüssel zu brechen

Bsp: MD5 - nur schwach kollisionsresistent

$$\exists m \neq m' \text{ mit } H(m) = H(m') \implies H(m|k_{AB}) = H(m'|k_{AB})$$

egal welches k_{AB} verwendet wird

3 Message-Authentication-Codes

b) Ein selbsternannter Sicherheitsexperte behauptet nun, es wäre sicherer die Formel auf

$$h = MAC'(m, k_{AB}) = H(k_{AB}|m)$$

umzustellen, da so der geheime Schlüssel k_{AB} durch Diffusion und Konfusion in der Hash-Funktion besser mit den Daten „verwoben“ wird und so bei bekanntem m die Rückrechnung des geheimen Schlüssels erschwert wird.

Wie ist dieser Vorschlag zu bewerten?

3 Message-Authentication-Codes

b) unsicher für Hashfunktionen basierend auf Merkle-Damgård Konstruktion

$$h = MAC'(m, k_{AB}) = H(k_{AB}|m)$$

Length extension Angriff

gegeben: abgefangene Nachricht m mit zugehörigem MAC mac
Ziel: bilde $m' = m \mid p \mid x$ mit gültigem MAC mac', ohne Kentniss von k_{AB}

p ... Padding
x ... (fast) beliebige erweiterung von m

$$mac = H_{IV_{original}}(k_{AB}|m)$$

"weiterhashen" ermöglicht durch Merkle-Damgård Konstruktion:

$$mac' = H_{IV=mac}(x)$$

Empfänger überprüft MAC mac':

$$MAC(m', k_{AB}) = MAC((m \mid p \mid x),\ k_{AB}) = H(k_{AB} \mid m \mid p \mid x) = H(k_{AB}|m')$$

$\Longrightarrow$ mac' ist ein gültiger MAC für m'

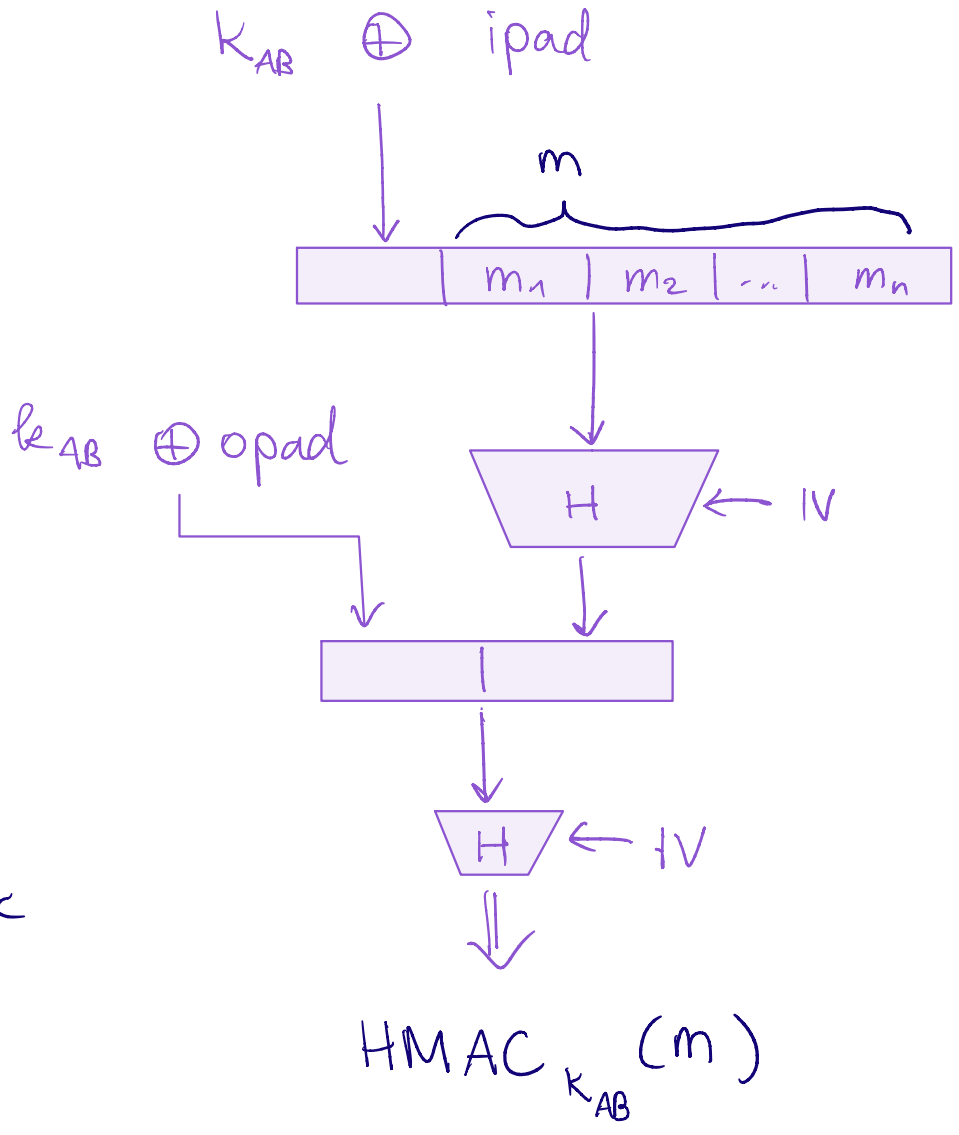
3 Message-Authentication-Codes

c) Erläutern Sie nun Vorteile des HMAC-Verfahrens gegenüber den gerade eben genannten, unsicheren Varianten

3 Message-Authentication-Codes

c) Erläutern Sie nun Vorteile des HMAC-Verfahrens gegenüber den gerade eben genannten, unsicheren Varianten

$$HMAC(m, k_{AB}) = H((k_{AB} \oplus opad)|H((k_{AB} \oplus ipad)|m))$$



Definiert in RFC 2104

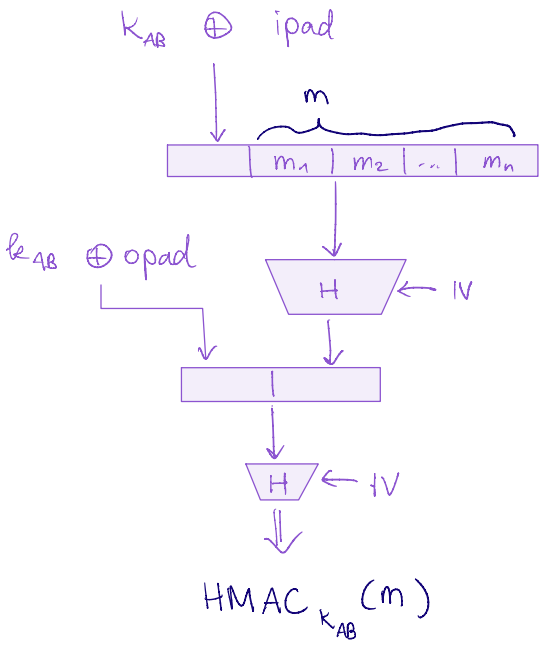
ipad = 0x36 36 ... 36

opad = 0x5C5C ... 5c

3 Message-Authentication-Codes

c) Erläutern Sie nun Vorteile des HMAC-Verfahrens gegenüber den gerade eben genannten, unsicheren Varianten

$$HMAC(m, k_{AB}) = H((k_{AB} \oplus opad)|H((k_{AB} \oplus ipad)|m))$$



im Vergleich zu 3a):
schwierig Kollisionsnachrichten auszurechnen (interner Zustand beeinflusst von unbekanntem k_{AB})

im Vergleich zu 3b):
kein length extension durch "weiterhashen" mehr möglich

Wer die Folien (+ Mitschrift) noch heute haben will - gerne über Zulip anfragen
Handout aus den Folien wird sonst am Donnerstag auf Moodle hochgeladen

Bei Unklarheiten gerne jetzt oder später über Zulip fragen