

\int Skripte

Mo. 4. März 24

Kryptologie ICS.KRYPTO

Folien zur Präsenz 3

«Einweg- und Hashfunktionen sowie MAC's», FS 24, V2.2



Bild aus Klaus Schmeh „Kryptografie. Verfahren–Protokolle–Infrastrukturen“, dpunkt Verlag.

©Josef Schuler, dipl. math., dipl. Ing. NDS ETHZ, MSc Applied IT-Security, Feldhof 25, 6300 Zug, j.schuler@bluewin.ch resp. josef.schuler@hslu.ch

Inhaltsübersicht

- Einweg- und Hashfunktionen sowie MAC's
 - In Kap. 5.4 & 6.5 im JS Skript „Einführung in die Kryptologie“ wurden Hashfunktionen schon angesprochen, resp. ad hoc eingeführt. Dies im Zusammenhang mit der „Stellvertreterfunktion“ des Hashs bei der Signatur.
 - Nun besprechen wir die Kap. 14, 8.2, 27 & 8.4 im JS Skript „Einf. i. d. Kryptologie.“.
 - In Kap. 14 „Hashfunktionen“ gehen wir vom allg. Begriff „Einwegfunktion“ über zum Begriff „Hashfunktion“.
 - Kap. 8.2 „Integritätsschutzmechanismen“ beschreibt, wie man mit Blockchiffren (CBC-MAC) resp. mit Hashfunktionen (HMAC) sogenannte MAC's (Message Authentication Code) erzeugen kann, so dass die Unversehrtheit (= Integrität = Integrity) einer Meldung nachgewiesen werden kann.
 - In Kap. 27 „Einwegfunktionen“ beschreiben diesen Funktionstyp im Detail.
 - Kap. 8.4 „Randomgenerator“ beschreibt, wie man mit Blockchiffren spezifische Einwegfunktionen erzeugt.
 - Kap. 24 „Prüfzifferberechnung“ ist eine Ergänzung und **nicht prüfungsrelevant!!**
 - **Bemerkung:** Mit „CBC-MAC“ sind im Folg. immer CBC-MAC ähnliche Algorithmen gemeint, also die Berechnung eines MAC's mit einem Blockchiffrierer.

Der Slogan zu dieser Präsenz

Eine kryptograph. sichere Hashfunktion \Rightarrow ist eine Einwegfunktion.
 \nLeftarrow

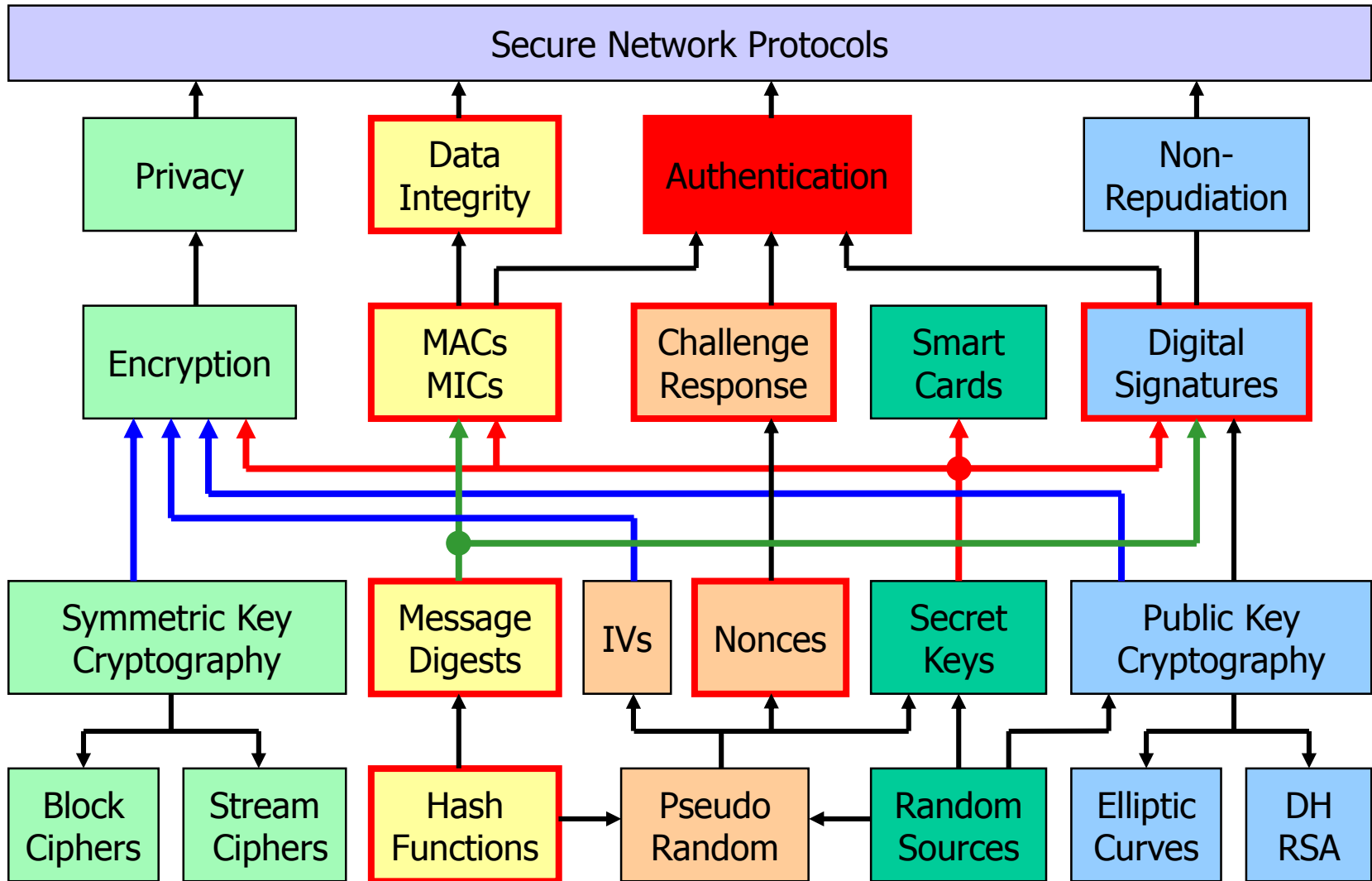
Lernziele

- Ich kenne die allgemeine Definition des Begriffs Einwegfunktion.
- Ich kenne die allgemeine Definition des Begriffs Hashfunktion.
- Ich weiss, was im Normalfall unter dem Begriff Hashfunktion (= MDC) verstanden wird.
- Ich kann die Begriffe Einwegfunktion, Hashfunktion, MDC & Prüfziffer richtig einordnen.
- Ich kenne die 3 Sicherheitseigenschaften einer Hashfunktion.
- Ich kenne den Pre-Image- und den Kollisionsangriff und deren Unterschiede.
- Ich kenne aktuelle Hashfunktionen und die weitere zentrale Eigenschaft.
- Ich kann den Unterschied zwischen einer keyed-Hashfunktion und einer keyless-Hashfunktion erklären.
- Ich kann den CBC-MAC (Grundfunktion und Variante nach ISO 9797-1) und HMAC aufzeichnen.
- Ich weiss, dass es viele Varianten von CBC-MAC ähnlichen Typen gibt. Ich kenne davon die Grundfunktion (ANSI X9.9) und die Variante nach ANSI X9.19, welche dem Algorithmus 3 (von 6) in ISO 9797-1 entspricht.
- Ich kann die Einsatzgebiete von MDC einerseits und CBC-MAC & HMAC andererseits benennen.
- Ich kenne die verschiedenen Typen von Einwegfunktionen.
- **Wichtig:** Die (#) bezeichn. Folien, Teile & Aufgaben sind **nicht** prüfungsrelevant.

Verweise zur Literatur

- JS Skripte „Einführung in die Kryptologie“, Kap. 8.2, 8.4, 14, 24 (Kap. 24 „**Prüfzifferberechnung**“ ist nicht prüfungsrelevant) & 27.
- In [CP-D] die Kap. 6.1.2, 11 & 12.
- Die Kapitelnummerierung in den folgenden Folien entspricht derjenigen im oben erwähnten JS Skript „Einführung in die Kryptologie“. D.h. die Details zu den Folien können im Skript nachgelesen werden. Zudem hat es im Skript weitere Übungen und Beispiele. **Die Aufgaben- und Beispielnummerierung im JS Skript «Einführung in die Kryptologie» und in den vorliegenden Folien stimmen nicht überein!**
- **Wichtig:** Es ist unbedingt zu beachten, dass nur das Bearbeiten und Lernen der Folien nicht genügt. Das Durcharbeiten der oben erwähnten Kapitel in JS Skripte „Einführung in die Kryptologie“ sind absolut zentral zum Bestehen der Modulendprüfung.

Cryptographical Building Blocks



Allgemeine Einführung

- Die Übersichtszeichnung...
 - ... soll zeigen, wie komplex die Kryptologie ist.
 - ... zeigt einmal mehr, dass man Kryptologie nicht – wie z.B. Mathematik – „straight forward“ unterrichten kann. Man kommt nicht darum herum hie und da in Zyklen zu unterrichten, resp. zu lernen. D.h. gewisse Themen zu verschiedenen Zeitpunkten des Semesters und aus verschiedenen Sichtweisen zu betrachten. Hashfunktionen sind genau so ein Beispiel.
 - ... ist in diesem Sinne nicht vollständig, es fehlen z.B. die „Einwegfunktionen“.
- Hashfunktionen:
 - Beim Diskutieren mit Security-Spezialisten verschiedener Couleur merke ich immer wieder, dass die Begriffe Hash, MDC, Signatur, HMAC usw. nicht immer exakt auseinander dividiert werden.
- Nicht alles ist Prüfungsstoff...
 - ... die mit (#) bezeichneten Folien sind **nicht** Prüfungsstoff. Sie werden auch nur kurz angesprochen und nicht wirklich durchgenommen.
- **Bemerkung**
 - Da wir schon in früheren Präsenzen Hashfunktionen, MAC, HMAC usw. angesprochen haben, sind im Folgenden gewisse Folien telquel wiedergegeben.

Kap. 14

HASHFUNKTION

Kap. 14.1

EINLEITUNG

Eine übergeordnete Betrachtung

Betrachten wir unser Slogan zur Präsenz:

Eine kryptograph. sichere Hashfunktion \Rightarrow ist eine Einwegfunktion.
 \nLeftarrow

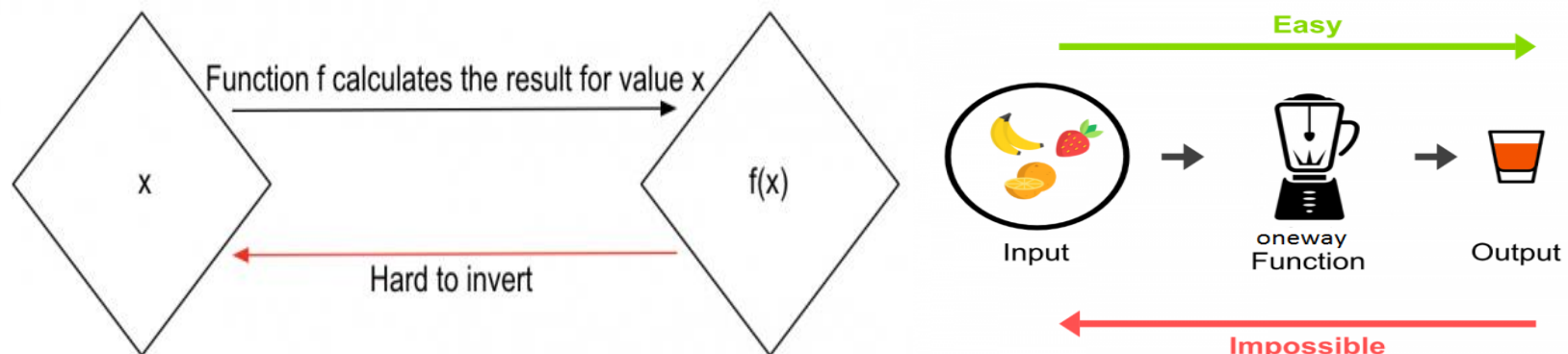
Somit ist eine Einwegfunktion etwas Allgemeineres als eine Hashfunktion. Etwas salopp gesagt, eine Einwegfunktion lässt sich in eine Richtung „einfach“ berechnen, in die andere Richtung „unmöglich“ berechnen. Sogar in [CP-D], S. 177 wird das nicht wesentlich anders definiert.

Definition 6.1 (Einwegfunktion)

Eine Funktion $f(\cdot)$ ist eine Einwegfunktion, wenn:

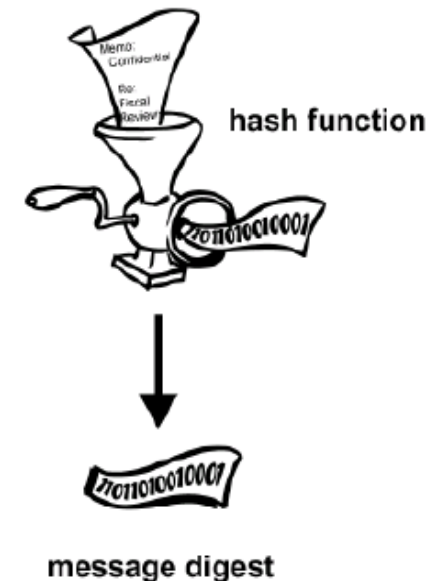
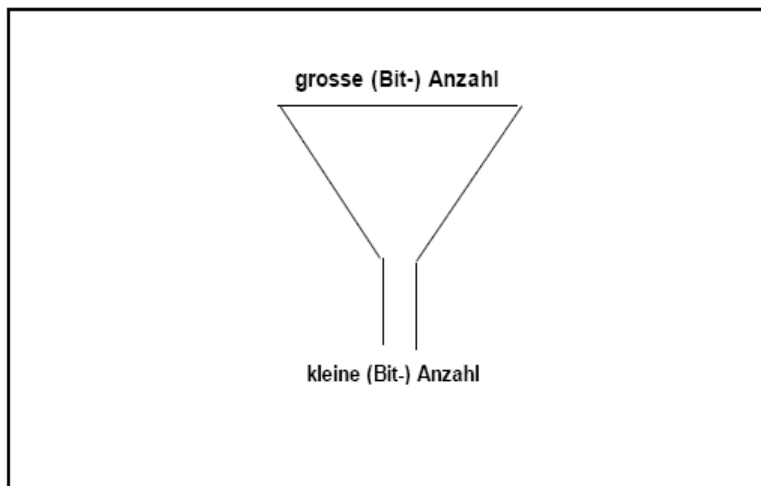
1. $y = f(x)$ rechentechnisch einfach und
2. $x = f^{-1}(y)$ technisch unmöglich zu berechnen ist.

Und so kommen wir zu zwei typischen Zeichnungen zu den Einwegfunktionen:



Definition

Unter einer **Hashfunktion** verstehen wir eine Funktion, die die Elemente von einer „grossen“ Menge in eine „kleine“ abbildet. Wir können uns diese Funktion als Trichter vorstellen. Die kleine Bitzahl kann z.B. sein: 1 16, 32, 64, 96, 128, 160, 224, 256, 384, 512.

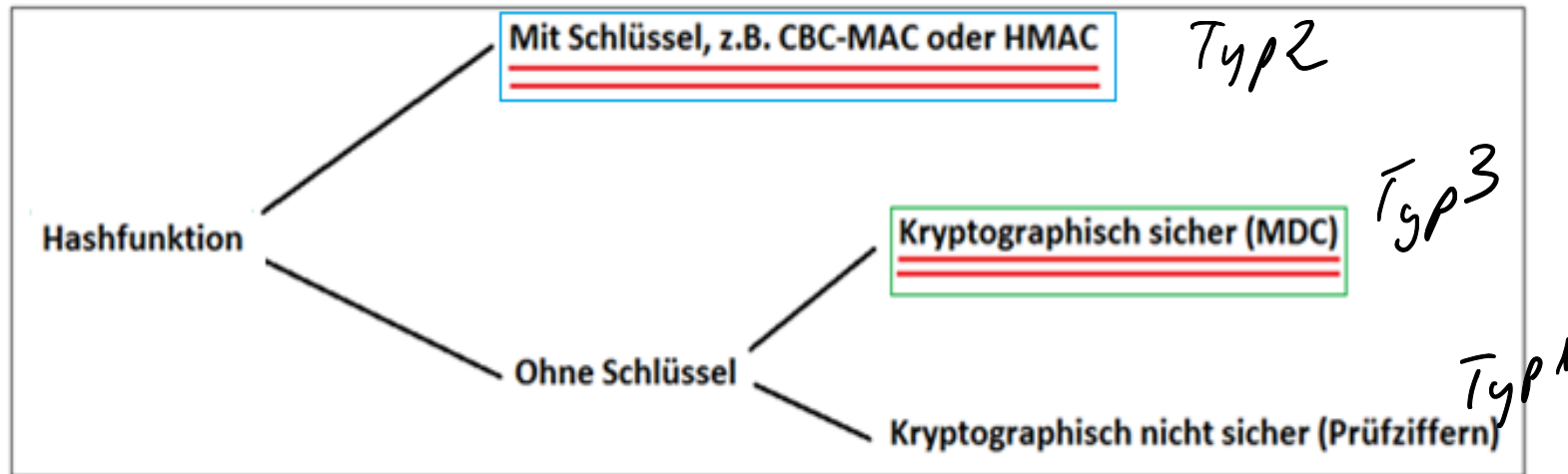


Wir kennen:

- Kryptographisch nicht sichere, schlüssellose Hashfunktionen = Berechnen von Prüfziffern. *Typ 1*
- Kryptographisch sichere Hashfunktionen mit Schlüssel (= MAC) *Typ 2*
- Kryptographisch sichere, schlüssellose Hashfunktionen (= MDC) *Typ 3*

Typen von Hashfunktionen, Übersicht

Die unterstrichenen Funktionen sind kryptographisch sichere Hashfunktionen.



MAC = Message Authentication Code (er sollte besser MIC = Message Integrity Code heißen, da er Integrität und nicht die ganze Palette der Authentizität gewährt).

MDC = Manipulation Detection Code = „Hashfunktion“, wie sie üblicherweise verstanden werden. Eine Hauptanwendung ist die Stellvertreterfunktion bei der Signaturerstellung, resp. –verifikation.

Typ 1 von Hashfunkt. = Prüfziffern, Hausaufgabe

- Kryptographisch nicht sichere (schlüssellose) Hashfunktionen

- Prüfzifferberechnung, letzte Ziffer der ISBN-Nummer = eine EAN Nummer (**EAN** = **E**uropäische **A**rtikel **N**ummerierung).

Die Prüfziffer der EAN-Nummern (13. Ziffer) berechnet sich, indem man die ersten zwölf Ziffern abwechselnd mit 1 und 3 multipliziert (links mit 1 anfangen) und diese Produkte summiert. Die Prüfziffer ist die Differenz der Summe zum nächsten Vielfachen von 10. Falls die Summe durch 10 teilbar ist, ist die Prüfziffer die 0.

- **Beispiel: 978381582086[?]**

$$9 \cdot 1 + 7 \cdot 3 + 8 \cdot 1 + 3 \cdot 3 + 8 \cdot 1 + 1 \cdot 3 + 5 \cdot 1 + 8 \cdot 3 + 2 \cdot 1 + 0 \cdot 3 + 8 \cdot 1 + 6 \cdot 3$$

$$= 9 + 21 + 8 + 9 + 8 + 3 + 5 + 24 + 2 + 0 + 8 + 18 = 115$$

$$115 + 5 = 120 \Rightarrow \text{Prüfziffer: } 5$$

- Eine einstellige Prüfziffer $\{0; 1; \dots; 9\}$. Die Prüfziffer ermöglicht das Erkennen von Eingabe- und Lesefehlern: Erkannt werden ein Einzelfehler (genau eine Ziffer falsch) und die meisten Vertauschungen von zwei Nachbarziffern (ISBN-13: außer $0 \leftrightarrow 5$, $1 \leftrightarrow 6$, $2 \leftrightarrow 7$, $3 \leftrightarrow 8$, $4 \leftrightarrow 9$), siehe

https://de.wikipedia.org/wiki/Internationale_Standardbuchnummer#ISBN-13

<http://www.arndt-bruenner.de/mathe/scripts/pruefziffern.htm>

Typ 1 von Hashfunktionen, Fort. Hausaufgabe

- Beispiel einer EAN Nummer mit Strichcode



- **Aufgabe 1**

- a) Überprüfen Sie die ISBN-Prüfziffer von [CP-D] 978-3-662-49296-3
- b) Überprüfen Sie die EAN vom obigen Bild.
- c) Überlegen Sie sich, die oben erwähnten Eigenschaften.

- **Bemerkung:** Um einen reibungslosen Verlauf von der alten 10-stelligen ISBN Nummer zur neuen 13-stelligen EAN zu machen, hat man einfach ein neues «Land» erfunden. 978 = das Land «Buch».

- Für CH & Li gelten die Nummern 760 – 769

Siehe <https://de.wikipedia.org/wiki/GS1-L%C3%A4nderpr%C3%A4fix>

- Weiterer Typus von Prüfziffer

- CRC (Cyclic Redundancy Check) bei der physikalischen Übertragung.

- Wichtige Eigenschaft der Prüfziffer:

- Ist ein Schutzmechanismus um unbewusste Veränderungen (Vertippen oder Bitfehler bei der Übertragung) einer Meldung zu entdecken.

Typ 2 von Hashfunktionen = MAC, HMAC

- Kryptographisch sichere Hashfunktionen mit Schlüssel (MAC)
 - CBC-MAC ähnliche Algorithmen, also MAC Berechnung mit einem Blockchiffrierer (cf. Kap. 8.2.1 & Kap. 8.2.2)
 - HMAC = MAC-Berechnung mit Hashfunktionen (cf. Kap. 8.2.3); das „H“ HMAC steht für „Hash“, also HMAC = Hash-MAC.
 - Ist ein Schutzmechanismus um bewusste Veränderungen einer Meldung zu entdecken. Dient zur Gewährung der Integrity und bietet automatisch auch Schutz vor Insertion.
- Die Hashfunktionen mit Schlüssel heissen MAC (Message Authentication Code) genannt. Eigentlich sollten sie MIC (Message Integrity Code) benannt werden. Letztlich ist das ein historisches Versäumnis, denn zu Beginn der 70'er Jahre hat man (leider) Integrität mit Authentizität gleichgesetzt. Wir wissen aber, dass Integrität (Unversehrtheit) nur ein Teilaspekt der Authentizität einer Meldung ist.

Typ 3 von Hashfunktionen = MDC "Hash"

- Kryptographisch sichere (schlüssellose) Hashfunktionen.
 - Auch MDC (Manipulation Detection Code) genannt.
 - In PGP auch (MD-5) Fingerprint genannt.
 - Das Resultat eines MDC wird oft kurz „Hash“ bezeichnet.
 - Ein MDC alleine gewährt keine Integrität.
- Im Folgenden geht es um diese MDC.
 - Deren (vielfältigen) Einsatz, z.B. Stellvertreterfunktion bei Signaturen.
 - Empfehlungen zur Verwendung
 - Usw.
- Das Berechnen eines Hashwertes einer Meldung bewirkt noch keine Integrität der Meldung. Denn ein Angreifer kann die Meldung verändern und dazu den zugehörigen Hashwert berechnen.

Aufgabe 2

Kreuzen Sie den richtigen Typus an.

Hashfunktion	Kryptographisch sicher		Kryptographisch nicht sicher
	Mit Schlüssel	Ohne Schlüssel	
1) Alph. Registratur			
2) Ablage nach Datum			
3) Parity-Bit (*)			
4) Byte Addition (mod 256) (*)			
5) CRC (*)			
6) CBC-MAC			
7) HMAC			
8) Konv. Prüfsumme (*)			
9) MDC = „Hashfunktion“			

(*) Kann bei Interesse in Kap. 24 «Anhang 2: Prüfzifferberechnung» nachgelesen werden. Diese Prüfzifferberechnungen sind aber nicht Prüfungsstoff.

WS 15/16

Kap. 14.2

MDC = Schlüssellose, kryptograph. sichere Hashfunktion

Aufgabe 3

Typ 2

In der Einleitung (cf. Kap. 14.1) haben wir 3 Typen von Hashfunktionen definiert. Um welchen Typus handelt es sich hier?

Abmachung

- Nachdem wir nun einen Einblick in den ganz allgemeinen Begriff „Einwegfunktion“ und dem ebenfalls allgemeinen Begriff „Hashfunktion“ erhalten haben, ist es nun Zeit eine Abmachung zu treffen.

Unter dem Begriff „Hashfunktion“ wollen wir nun eine kryptographisch sichere, schlüssellose Hashfunktion wie z.B. SHA-2 oder SHA-3 verstehen. Der – nicht sehr vertraute – Begriff „MDC“ werden wir in diesem Sinne nur noch selten verwenden.

- Im Weiteren gilt:

Die primäre Grundmotivation, resp. –anwendung einer Hashfunktion ist die Stellverteterfunktion. D.h. ein (grosses) Dokument wird gehasht, damit danach der Hash – und damit das Dokument – signiert werden kann.

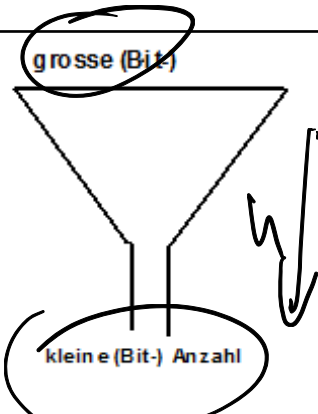
Sicherheitseigenschaften von Hashfunktionen

Siehe Aufteilung der Kryptologie gemäss Ueli Maurer → er betrachtet die „schlüssellose“ Kryptographie. Die Hashfunktion ist eine solche Ausprägung.

Aus den 2 Stichworten „schlüssellos“ und „Stellvertreterfunktion“ ergeben sich sofort verschiedene Fragen:

- A) Was für Sicherheitseigenschaften soll so eine schlüssellose Stellvertreterfunktion haben?
- B) Gibt es überhaupt Sicherheitseigenschaften?

Bis jetzt haben wir ja festgestellt, dass es hierzu sehr diffizile Betrachtungen und Angriffe gibt, auf die man nicht so ohne weiteres kommt.

	<p>Die „grosse“, besser „beliebig grosse“ Bitzahl kann mit $\{0, 1\}^*$ beschrieben werden. Dabei charakterisiert „*“ das „beliebig“.</p> <p>Die kleine, fixe Bitzahl kann mit $\{0, 1\}^n$ für $n \in \mathbb{N}$ beschrieben werden. Dabei charakterisiert „n“ die feste Anzahl Bit im Output, z.B. $n = 256$.</p>
<p>Somit ist also eine Hashfunktion eine Funktion</p> <p>$h: \{0, 1\}^* \rightarrow \{0, 1\}^n$</p>	

Handwritten notes:

- Below the first cell: $n = 512 \text{ bit}$, $2^{512} \approx 10^{160}$
- Below the second cell: 2^{256} Mögli. d. Werte, $2^{256} \approx 10^{77} \approx \# \text{ Atome im Weltall}$

Die 3 zentralen Sicherheitseigenschaften

1) Urbildresistenz (oder Einwegeigenschaft)

Für ein gegebenes $y \in \{0, 1\}^n$ ist es praktisch nicht möglich einen Wert $x \in \{0, 1\}^*$ zu finden mit $h(x) = y$.

2) Schwache Kollisionsresistenz (oder zweite Urbildresistenz oder 2nd-Preimage Eigenschaft)

Für ein gegebenes $x \in \{0, 1\}^*$ ist es praktisch nicht möglich, einen Wert $x' \in \{0, 1\}^*$ und $x' \neq x$ zu finden mit $h(x) = h(x')$.

3) Starke Kollisionsresistenz (oder Kollisionsresistenz)

Es ist praktisch nicht möglich zwei Werte $x, x' \in \{0, 1\}^*$ und $x' \neq x$ zu finden, dass $h(x) = h(x')$.

Aufgabe 4

- Ordnen Sie die 3 Eigenschaften den Diagrammen richtig zu.
- Erklären Sie nun, dass eine Prüfziffer keine kryptographisch sichere Hashfunktion ist.

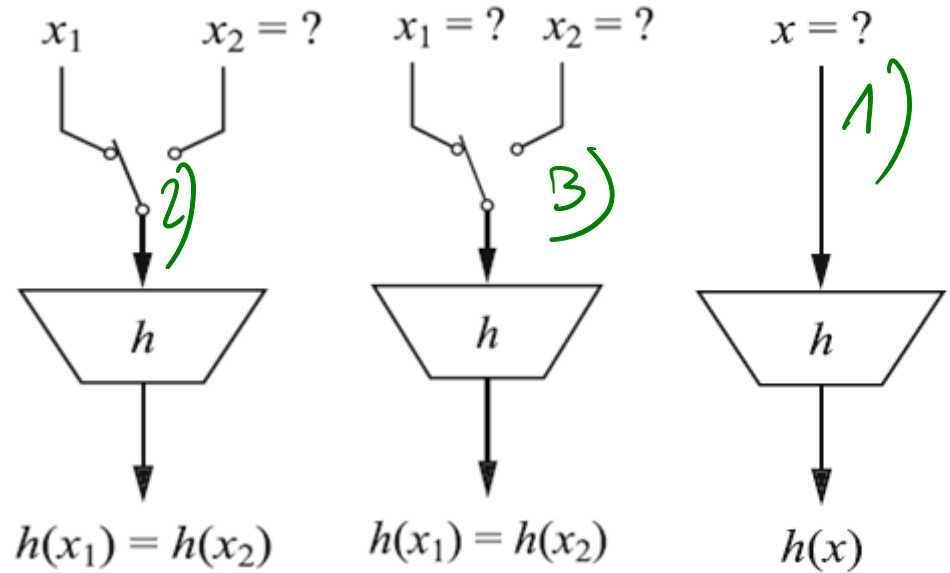


Fig 1 alle 3 eig. erfüllt
sind nicht bis 164 10

Die wichtigsten Hashfunktionen (MDC)

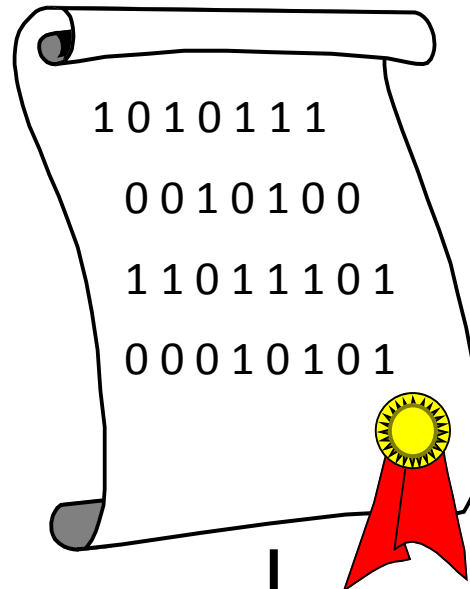
Namen	Output in Bit	Information zur Verwendung
MD-4	128	Darf nicht mehr verwendet werden!
MD-5	128	Darf nicht mehr verwendet werden! Wird in PRF noch verwendet.
SHA-0	128	Darf nicht mehr verwendet werden!
RIPEMD	160	Darf nicht mehr verwendet werden!
SHA-1	160	Darf für Signaturen nicht mehr verwendet werden! Als Grundfunktion für HMAC grundsätzlich OK. [BSI1] empfiehlt aber sie auch dort nicht mehr einzusetzen. In Pseudo Random Function PRF wird sie vermutlich noch verwendet.
SHA2-224	224	Gemäss [BSI1] wird in der tech. Richtlinie ab 2023 mind. 240 Bit (d.h. für Kollisionsangriffe mind. 120 Bit) gefordert. Daher ab 2023 für Signaturen nicht mehr zugelassen.
SHA3-224	224	Wie bei SHA2-224.
Ab 2023 sind nur noch die folg. Hashfunktionen für Signaturen zugelassen!		
SHA2	256, 384 oder 512	Alle Einsätze OK; für Signaturen müssen ab 2023 mindestens 240 Bit genommen werden.
SHA3	256, 384 oder 512	Wie bei SHA2

Wie sieht die Zukunft aus?

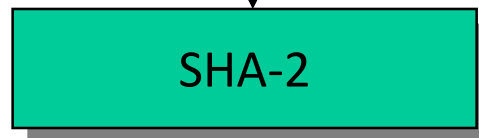
- Die SHA-2 Familie (genauer SHA-256, SHA-384 & SHA-512), wird noch eine Weile eine erlaubte Alternative beim Einsatz in den Signaturen sein.
- Da sie aber im Wesentlichen auf SHA-1 beruht, erwartet man früher oder später entsprechende Attacken.
- Ein Wettbewerb – ähnlich wie bei der Ablösung des DES – wurde 2013 beendet.
- Der Sieger des Auswahlverfahrens ist der Hash mit Namen Keccak und heisst nun SHA-3. Er kann ebenfalls mit 256 bis 512 Bit eingesetzt werden, siehe vorherige Folie.
- D.h. zukünftige Entwicklungen wird es in absehbarer Zeit in diesem Bereich nicht mehr geben.

Schlüssellose kryptographisch sichere Hashfunktionen sind notwendige Hilfsmittel als Stellvertreterfunktion bei Signaturen

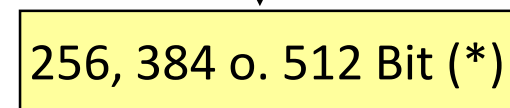
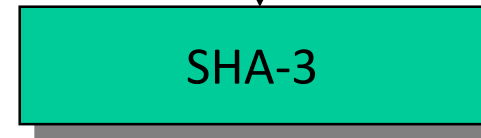
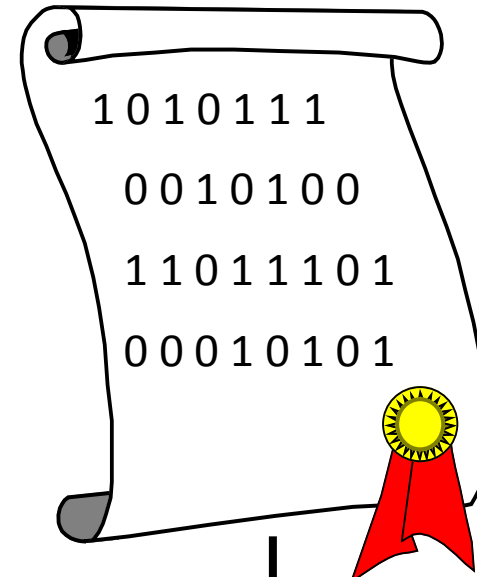
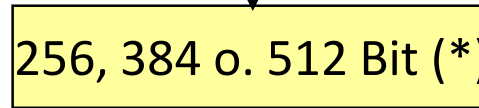
Dokument
oder
Meldung



Hash Funktion

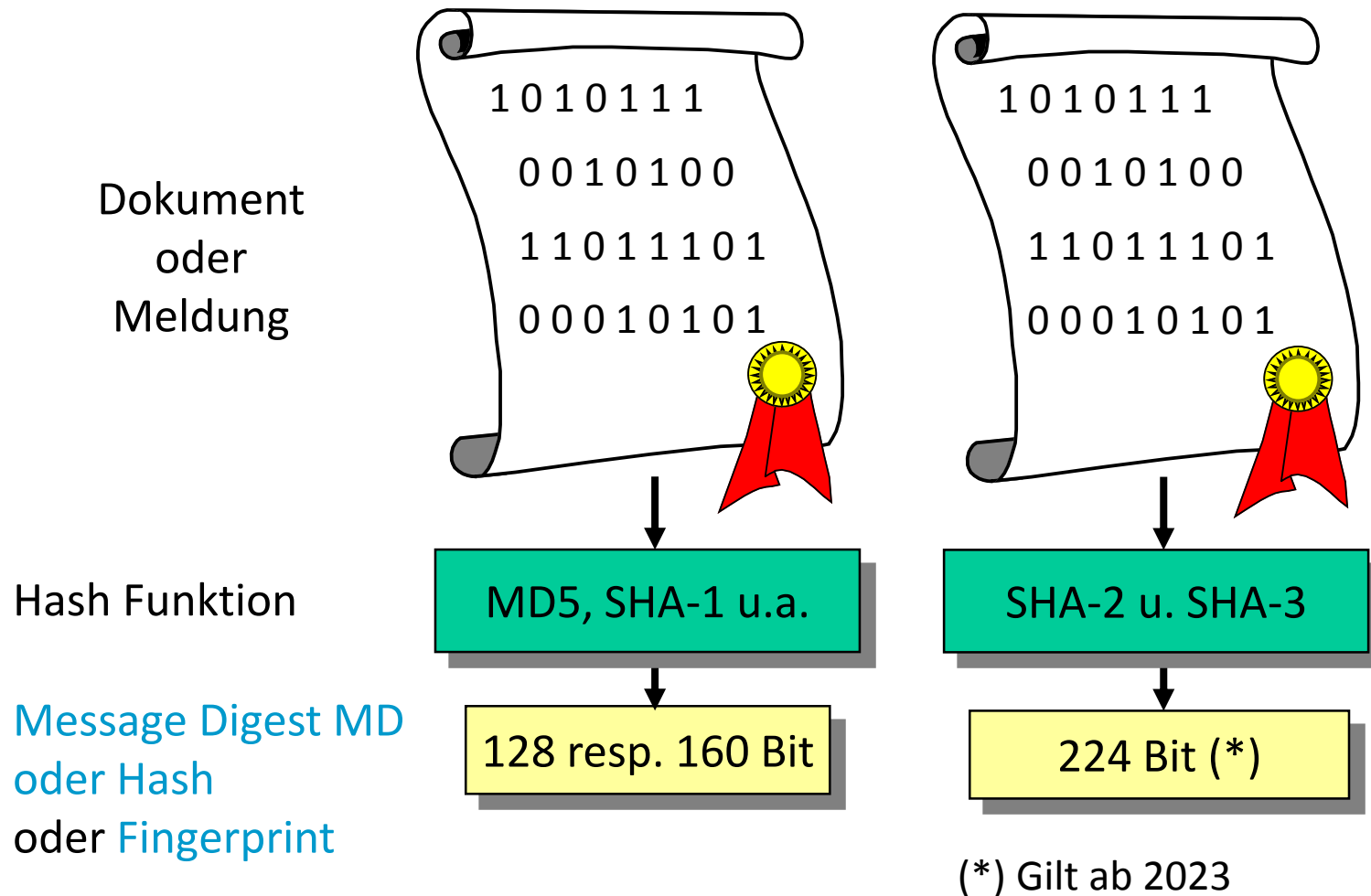


Message Digest MD
oder Hash
oder **Fingerprint**

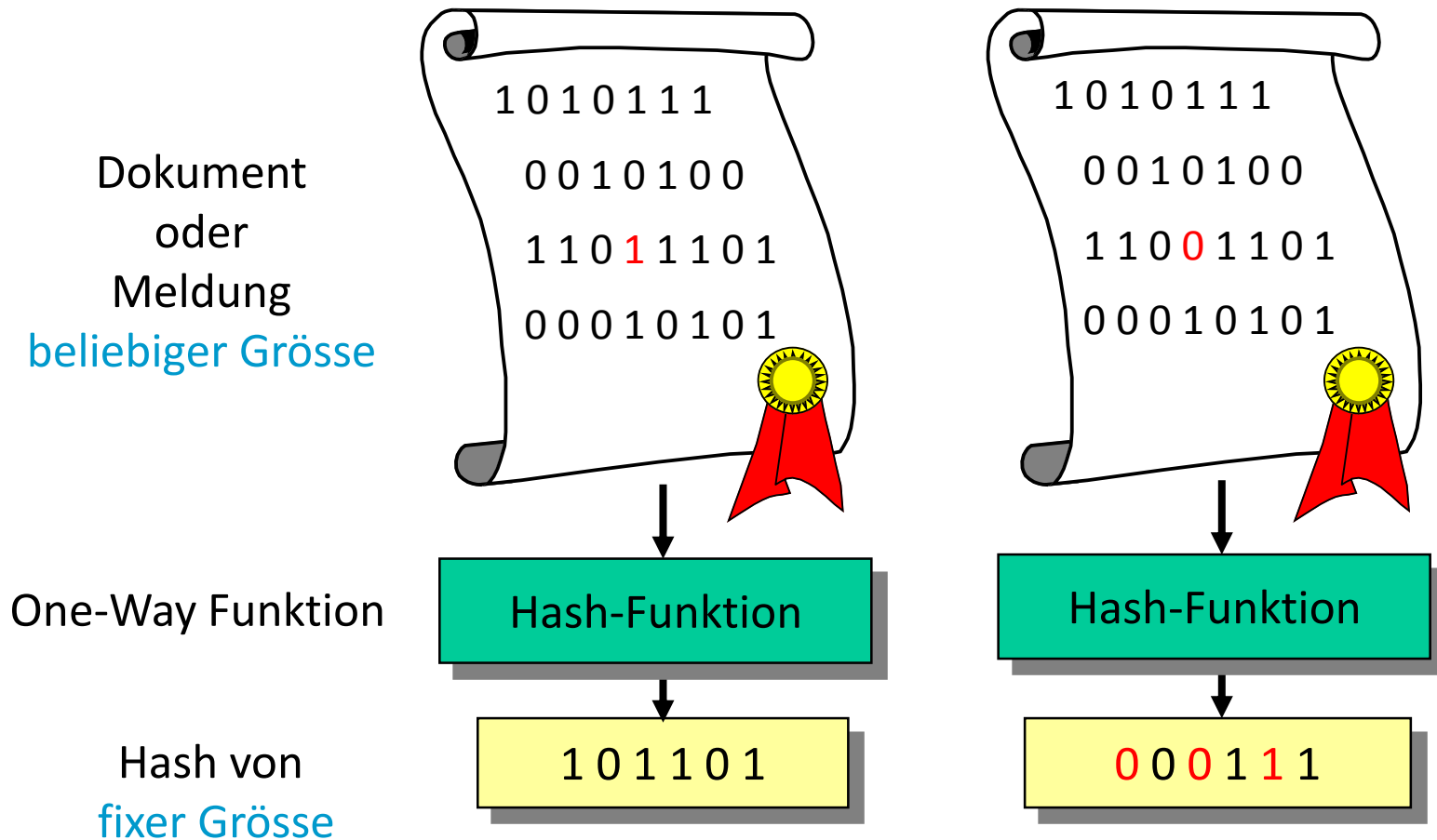


(*) Ab 2023 gilt mind. 240 Bit, also faktisch 256 Bit.

Diese schlüssellosen Hashfunktionen dürfen für Signaturen nicht mehr verwendet werden



Weitere zentrale Eig. solcher Hashfunktionen



- Das Ändern eines einzigen Bits im Dokument hat zur Folge, dass im statistischen Mittel ca. 50% im Hash geändert werden.

Einschub: Binomialkoeffizient $\binom{n}{m}$

1) Es gilt: $\binom{n}{m} = \frac{n!}{m! \cdot (n-m)!} = \binom{n}{n-m}$ *symmetrie* **Beispiel:** $\binom{11}{4} = \frac{11!}{4! \cdot 7!} = \binom{11}{7} = 330$

2) Bei $m = n/2$ ist der Binomialkoeffizient maximal, d.h. $\binom{n}{n/2}$ ist maximal.

3) $\binom{n}{m}$ beschreibt wie viele Möglichkeiten gibt es m Elemente auf k Plätze zu verteilen. Siehe dazu Lotto 6 aus 42 (Reihenfolge der Zahlen nicht berücksichtigt)

- 42 Möglichkeiten für 1 Zahl.
- $42 \cdot 41$ für 2 Zahlen mit Reihenfolge, $\frac{42 \cdot 41}{1 \cdot 2} = \binom{42}{2} = 861$ für 2 Zahlen o. RF.
- ...
- $42 \cdot 41 \cdot \dots \cdot 37$ für 6 Zahlen mit Reihenfolge, $\frac{42 \cdot 41 \cdot \dots \cdot 37}{1 \cdot 2 \cdot \dots \cdot 6} = \binom{42}{6} = 5'245'786$ für 6 Zahlen ohne Reihenfolge. → Beim CH-Lotto werden die Möglichkeiten wegen der Zusatzzahl noch mit 6 vergrößert, also $31'474'716$.

4) Analog nun für Ändern der Bits bei Hash der Grösse 256 Bit:

- $\binom{256}{1} = 256$ Möglichkeiten für 1 Bit.
- $\binom{256}{128} = 5,8 \cdot 10^{75}$ Möglichkeiten für die Hälfte, also 128 Bit.

Allg. Eig. v. krypt. sicheren Hashfunktionen

1. Beim Ändern eines Inputbits, wird im statistischen Mittel die Hälfte des Hashs geändert.
2. Mit Hilfe von sogenannten Binomialkoeffizienten kann man zeigen, dass es am idealsten ist, wenn die Hälfte des Hashs geändert wird. Denn die Anzahl der Möglichkeiten sind maximal, resp. die Wahrscheinlichkeit zum Erraten minimal.
3. Allgemein ist $\binom{n}{m} = \frac{n!}{m! \cdot (n-m)!}$ bei $m = n/2$ maximal, d.h. $\binom{n}{n/2}$ ist maximal.
 - Siehe Blockchiffren: bei $n = 128 \rightarrow 2,4 \cdot 10^{37} \rightarrow$ heute ungenügend!
 - Bei $n = 160$: $\binom{160}{80} = \frac{160!}{80! \cdot (160-80)!} = \frac{160!}{80! \cdot 80!} = 9,2 \cdot 10^{46} \rightarrow$ ungenügend!
 - Bei $n = 224$: $\binom{224}{112} = \frac{224!}{112! \cdot (224-112)!} = \frac{224!}{112! \cdot 112!} = 1,4 \cdot 10^{60}$
 - Bei $n = 256$: $\binom{256}{128} = \frac{256!}{128! \cdot 128!} = 5,8 \cdot 10^{75} \rightarrow 10^{80} \approx$ Anz. Atome im Weltall
 - Bei $n = 384 \rightarrow 1,6 \cdot 10^{114}$; bei $n = 512 \rightarrow 4,7 \cdot 10^{152}$
 - Es gibt total ins 2^n mögliche Hashwerte:
 - $n = 384 \Rightarrow 2^n = 3,9 \cdot 10^{115}$
 - $n = 512 \Rightarrow 2^n = 1,34 \cdot 10^{154}$

Ein paar Zahlen für $n = 512$

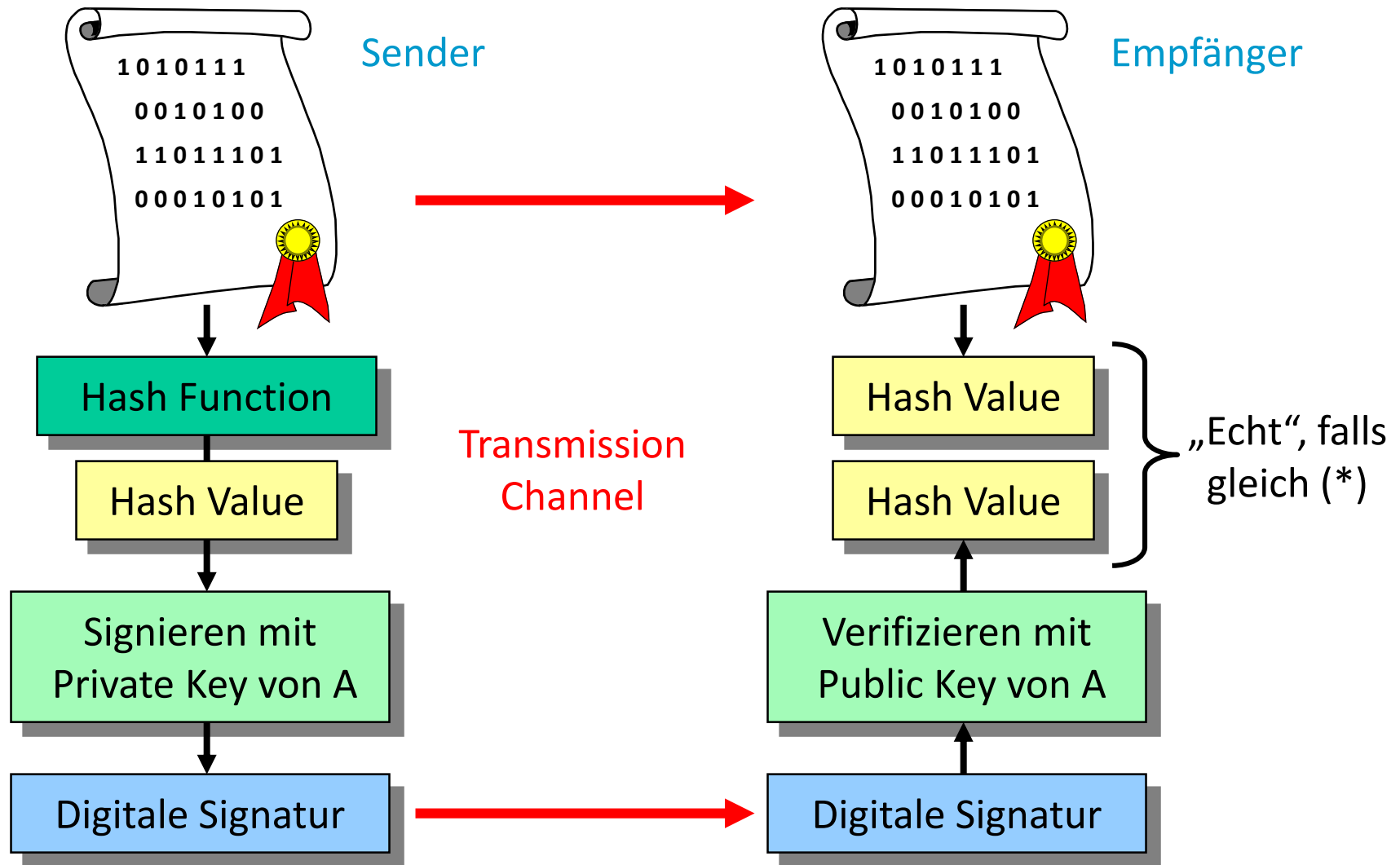
- Es gibt total ins 2^n mögliche Hashwerte:
 - $n = 512 \Rightarrow 2^n = 1,34 \cdot 10^{154}$
 - Anzahl genau die Hälfte geändert: $\binom{512}{256} = 4,7 \cdot 10^{152} \approx 3,6\%$ aller möglichen Hashwerten. D.h. in ca. 3,6% aller Fälle werden genau die Hälfte aller Bits geändert.
 - Anzahl der geänderten Bits zw. $250 - 262$: $\binom{512}{250} + \dots + \binom{512}{262} = 5,8 \cdot 10^{153} \approx 45\%$ aller möglichen Hashwerten. D.h. in ca. 45% aller Fälle werden zwischen 250 – 262 Bits geändert.
 - Anzahl der geänderten Bits zw. $240 - 272$: $\binom{512}{240} + \dots + \binom{512}{272} = 1,1 \cdot 10^{154} \approx 85\%$ aller möglichen Hashwerten. D.h. in ca. 85% aller Fälle werden zwischen 240 – 272 Bits geändert.
 - Anzahl der geänderten Bits zw. $230 - 282$: $\binom{512}{230} + \dots + \binom{512}{282} = 1,315 \cdot 10^{154} \approx 98\%$ aller möglichen Hashwerten. D.h. in ca. 98% aller Fälle werden zwischen 230 – 282 Bits geändert.
 - Anzahl der geänderten Bits zw. $220 - 292 \rightarrow$ ca. 99,85%.
 - Anzahl der geänderten Bits zw. $200 - 312 \rightarrow$ ca. 99,99993%.
 - Anzahl der geänderten Bits < 100 oder $> 412 \rightarrow$ ca. $5 \cdot 10^{-44}\%$.
 - Anzahl der geänderten Bits < 50 oder $> 462 \rightarrow$ ca. $10^{-82}\%$.

Ein paar Zahlen für $n = 256$

- Es gibt total ins 2^n mögliche Hashwerte:
 - $n = 256 \Rightarrow 2^n = 1,15 \cdot 10^{77}$
 - Anzahl genau die Hälfte geändert: $\binom{256}{128} = 5,8 \cdot 10^{75} \approx 5\%$ aller möglichen Hashwerten. D.h. in ca. 5% aller Fälle werden genau die Hälfte aller Bits geändert.
 - Anzahl der geänderten Bits zw. 118 – 138: $\binom{256}{118} + \dots + \binom{256}{138} = 9,4 \cdot 10^{76} \approx 81\%$ aller möglichen Hashwerten. D.h. in ca. 81% aller Fälle werden zwischen 118 – 138 Bits geändert.
 - Anzahl der geänderten Bits zw. 108 – 148 \rightarrow ca. 99%.
 - Anzahl der geänderten Bits zw. 98 – 158 \rightarrow ca. 99,99%.
 - Anzahl der geänderten Bits < 50 oder $> 206 \rightarrow$ ca. $10^{-21}\%$.

Die Rolle des MDC bei der digitalen Signatur

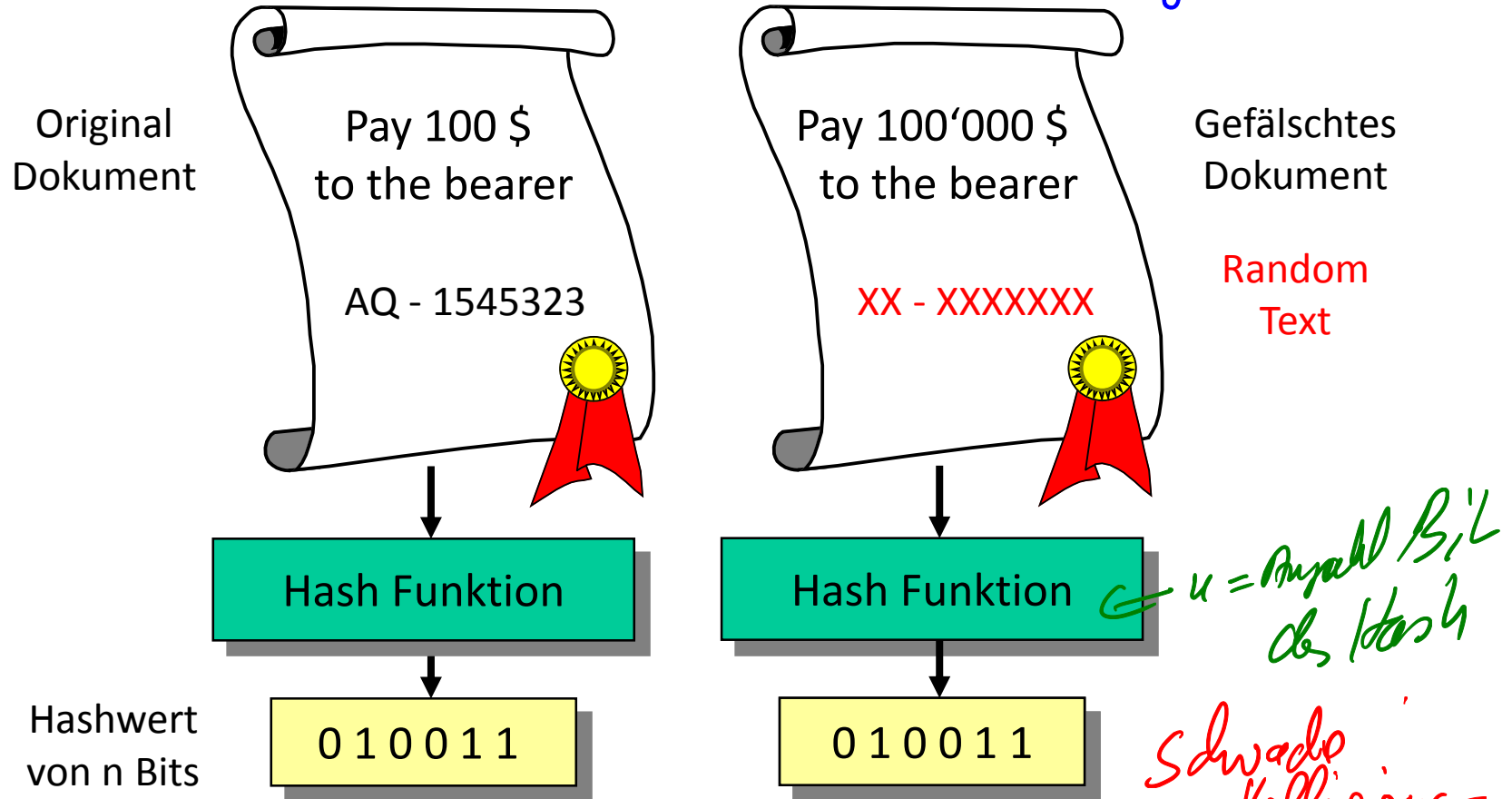
Public Key Krypto. 3: Digitale Signatur (z.B. RSA)



(**) zu 99,99... 9% echt falls gleich, aber zu 100% gefälscht, falls nicht gleich!

Die Gefahr 1: Pre-Image Angriff Fälschen von Dokumenten

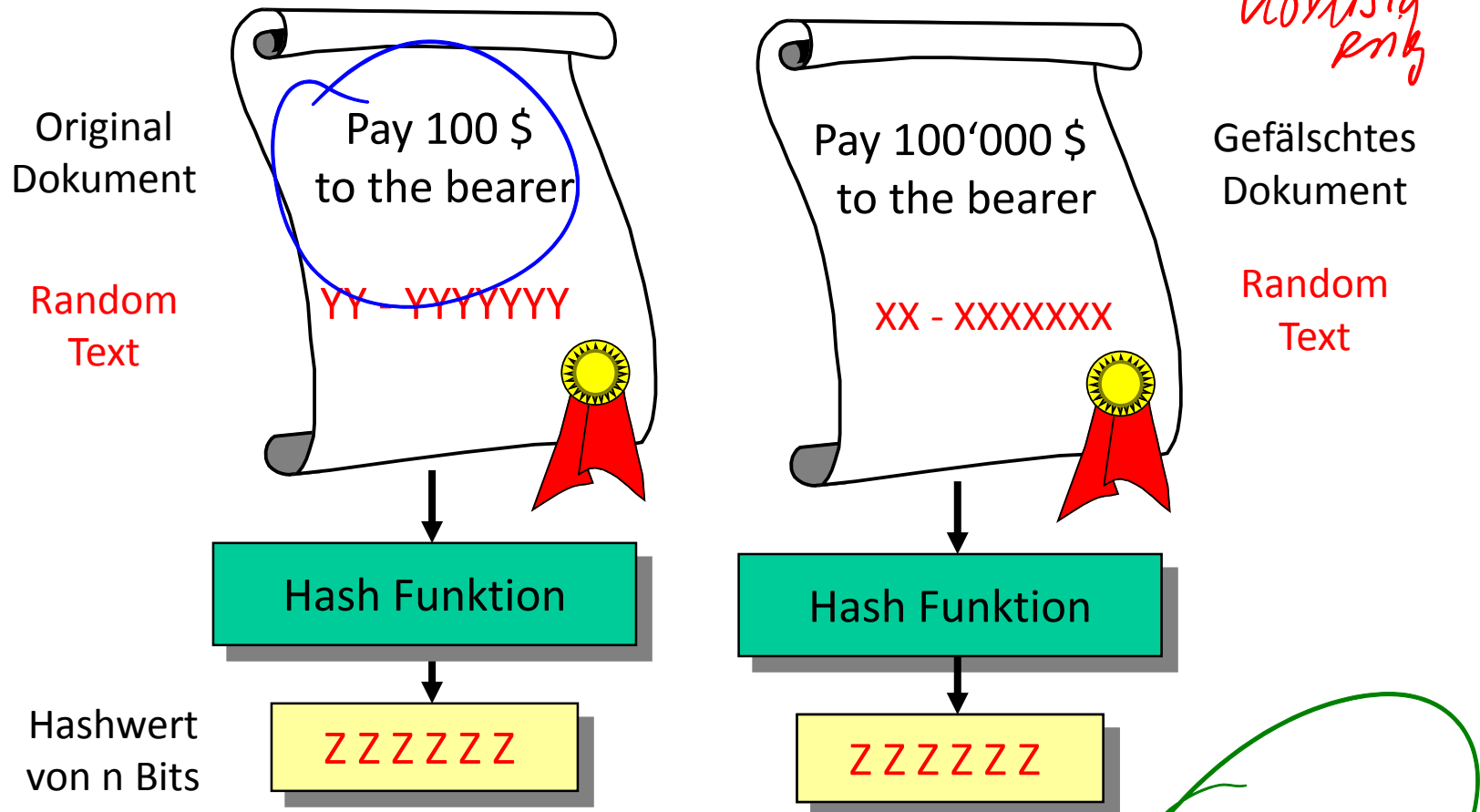
Realistischer Angriff



- Bei einem n-Bit Hash sind im stat. Mittel ca. 2^n Versuche nötig, um ein zweites Dokument mit gleichem Hash zu finden.

Die Gefahr 2: Kollisionsangriff (Geburtsstags-Attacke) Fälschen von Dokumenten

starke
Kollisionen



- Bei einem n-Bit Hash sind im stat. Mittel nur die Wurzel, also $2^{\frac{n}{2}} = \sqrt{2^n}$ Versuche nötig, um zwei Dokumente mit gleichem Hash zu finden.

Das Geburtstagsparadoxon, Teil 1

- Wir nehmen an, die Geburtstage sind über das ganze Jahr gleichverteilt, und die Geburtstage von 2 Personen seien unabhängig voneinander, also z.B. keine Zwillinge.
- Wie gross ist die Wahrscheinlichkeit, dass eine zufällig angetroffene Person P am gleichen Tag Geburtstag hat wie Sie?
 - Klar $p = \frac{1}{365}$, resp. $\bar{p} = 1 - \frac{1}{365} = \frac{364}{365}$, dass P nicht am gleichen Tag Geb. hat wie Sie.
- Gleiches Gedankenspiel mit zwei zufällig angetroffenen Personen P_1, P_2
 - $\bar{p} = \left(1 - \frac{1}{365}\right)^2 = \left(\frac{364}{365}\right)^2$, dass P_1 & P_2 nicht am gleichen Tag Geb. haben wie Sie. D.h.
 $p = 1 - \bar{p} = 1 - \left(\frac{364}{365}\right)^2 \approx 0,107$
- Gleiches Gedankenspiel mit n zufällig angetroffenen Personen
 - $\bar{p} = \left(1 - \frac{1}{365}\right)^n = \left(\frac{364}{365}\right)^n$, dass n Personen nicht am gleichen Tag Geb. wie Sie haben.
D.h. $p = 1 - \bar{p} = 1 - \left(\frac{364}{365}\right)^n$
- **HA: Aufgabe 5** Wie viele Leute müssten nun mit Ihnen in einem Raum sein, damit die Wsk. grösser als 50% ist, so dass jemand am gleichen Tag Geburtstag hat wie Sie?

Das Geburtstagsparadoxon, Teil 2

- Nun betrachten wir das Taubenschlagprinzip von Dirichlet (siehe D-MATH Präsenz 6). Wie viele Personen müssen in einem Raum sein, damit zu 100% zwei (beliebige) Personen am gleichen Tag Geburtstag haben?
 - Klar, wenn 366 (resp. 367, wenn wir Schaltjahre auch berücksichtigen) Personen im gleichen Raum sind, so muss es zwangsläufig zwei Personen geben, die am gleichen Tag Geburtstag haben → cf. Taubenschlagprinzip.
- Wie viele Leute müssten nun in einem Raum sein, damit die Wsk. grösser als 50% ist, so dass sich im Raum zwei Personen befinden, die am gleichen Tag Geburtstag haben?
- Diese Aufgabe (resp. die Herleitung der Lösung) ist nun viel schwieriger. In D-MATH haben wir die Lösung einmal angetönt, in [CP-D], Kap. 11.2.3 ist die Herleitung im Detail beschrieben (nicht Lehr- und Prüfungsstoff). Kurz zusammengefasst gilt:
 - $\bar{p} = p(\text{keine Kollision bei } n \text{ Personen}) = \left(1 - \frac{1}{365}\right) \cdot \left(1 - \frac{2}{365}\right) \cdot \dots \cdot \left(1 - \frac{n-1}{365}\right)$
 - Bei $n = 23$ ist diese Wahrscheinlichkeit bei ca. 50%, d.h. ab 23 Personen wird die Wahrscheinlichkeit über 50%, dass es zwei Personen im Raum gibt, die am gleichen Tag Geburtstag haben.
 - Bei $n = 40$ ist diese Wahrscheinlichkeit bei ca. 10%, d.h. zu 90% gibt es zwei Personen im Raum, die am gleichen Tag Geburtstag.
 - Verallgemeinerung: ab ca. \sqrt{n} Elementen ist die Wahrscheinlichkeit einer Kollision > 50%.

Mitte 2004: MD5 Kollision gefunden in 1 Std. (#)

d131dd02c5e6eec4693d9a0698aff95c2fcab5**8**712467eab4004583eb8fb7f89
55ad340609f4b30283e4888325**7**1415a085125e8f7cdc99fd91dbdf**f**280373c5b
960b1dd1dc417b9ce4d897f45a6555d535739a**c**7f0ebfd0c3029f166d109b18f
75277f7930d55ceb22e8adba79**c**155ced74cbdd5fc5d36db19b0a**d**835cca7e3

d131dd02c5e6eec4693d9a0698aff95c2fcab5**0**712467eab4004583eb8fb7f89
55ad340609f4b30283e4888325**f**1415a085125e8f7cdc99fd91dbd**7**280373c5b
960b1dd1dc417b9ce4d897f45a6555d535739a**4**7f0ebfd0c3029f166d109b18f
75277f7930d55ceb22e8adba79**4**c155ced74cbdd5fc5d36db19b0a**5**835cca7e3

These two 1024-bit values only differ in 6 bit positions

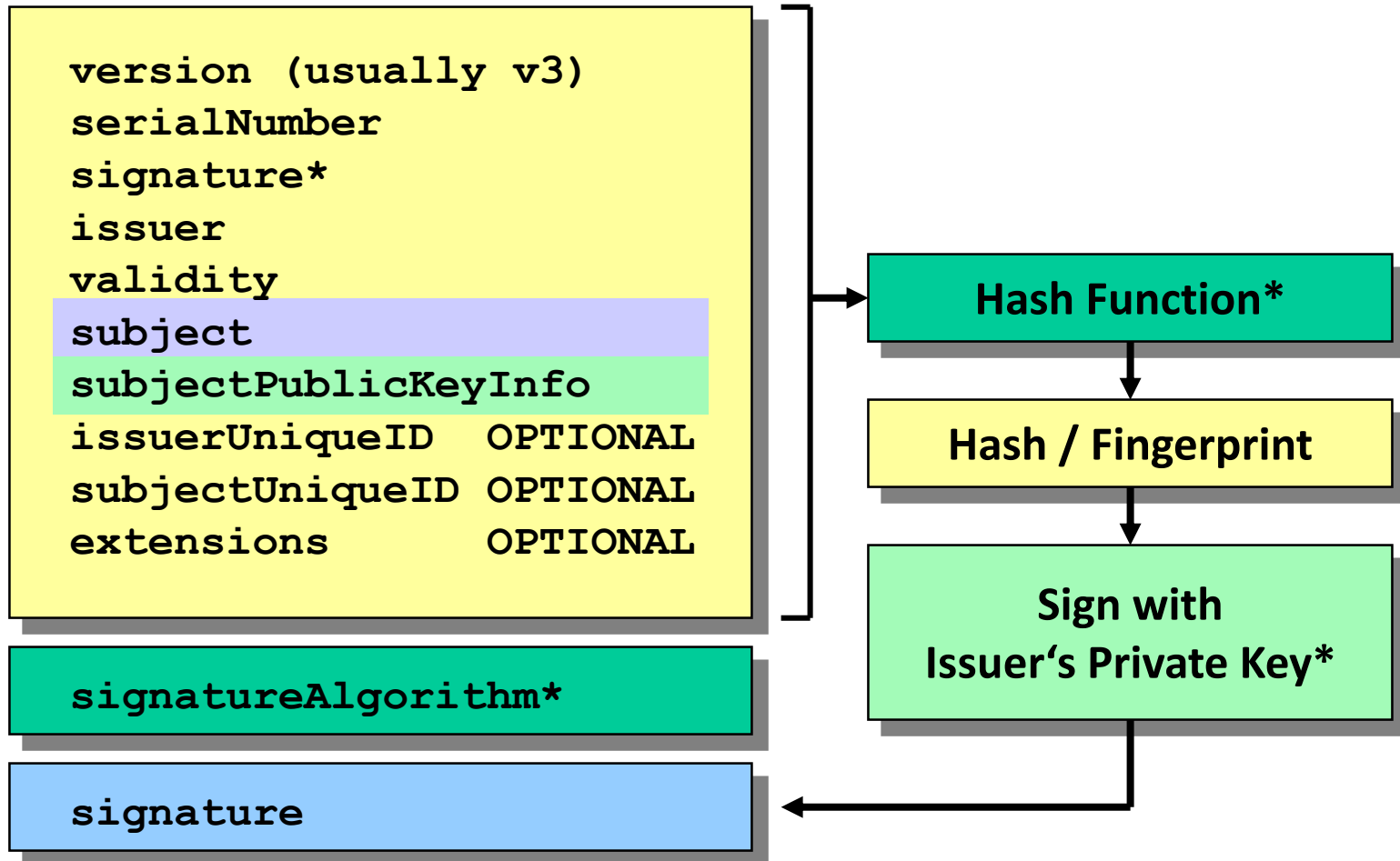
Common MD5 digest: **a4c0 d35c 95a6 3a80 5915 367d cfe6 b751**

- Xiaoyun Wang, Dengguo Feng, Xuejia Lai, and Hongbo Yu invented an algorithm that finds MD5 collisions on an IBM P690 computer within little more than one hour.
- Similar attacks exist for RIPEMD-128 and SHA-0.
- How large is the security margin of SHA-1?

Feb. 2005 & Juni 2009: Erfolgreiche Angriffe auf den SHA-1 (#)

- Pre-Image Angriff: Aufwand $2^{160} \rightarrow$ unmöglich
- Kollisionsangriff: Aufwand $2^{80} \rightarrow$ „möglich“
- Ist der Angriff mit deutlich weniger Versuchen als beim Brute-Force-Ansatz, gilt das Verfahren als geknackt.
- Feb. 2005 gelang das einer Chinesischen Forschergruppe: Sie haben ein Verfahren entwickelt, eine Kollision statt mit 2^{80} bereits mit 2^{69} Operationen zu ermitteln. Das reduziert die Zahl der notwendigen Operationen um den Faktor 2048 (2^{11}).
- Juni 2009 weitere Reduktion auf 2^{52} Operationen.
- Wichtig ist, dass das immer Kollisionsangriffe waren (Birthday attacks auf die Kollisionsresistenz).
- Aktuell findet man Kollisionen ca. mit einem Aufwand von € 10'000.- (siehe [BSI1], Kap. 4).
- Keine Panik: Um Signaturen zu knacken braucht es Preimage Attacken.
- **Kein Problem mehr \rightarrow Wir haben nun den SHA-3**

MDC in X.509 Zertifikat eingebettet, mehr dazu in Präsenz 11 bei Prof. Portmann in PKI



* specifies algorithm used to sign certificate, e.g. SHA2-256RSA

gewährt
Integrität
und verhindert
Insertionen
mit Schlüssel

Kap. 8.2, 14.3 & 14.4

CBC-MAC (*) und HMAC

(*) Mit CBC-MAC sind im Folgenden immer CBC-MAC ähnliche Algorithmen gemeint. Also die Berechnung eines MAC's mit einem Blockchiffrierer.

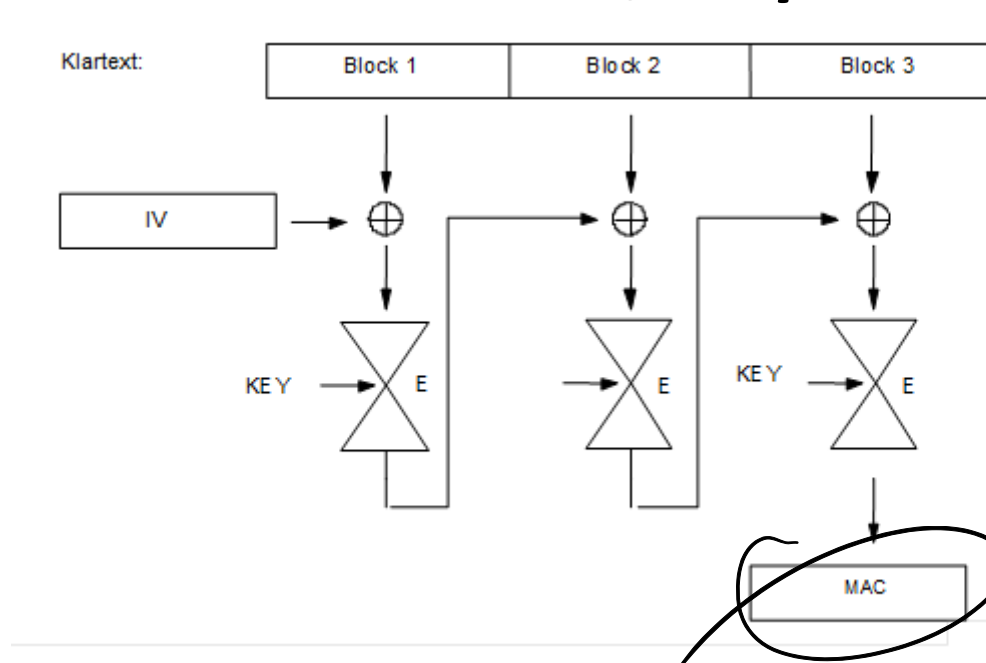
Bis 16450

Typ 2

Aufgabe 6

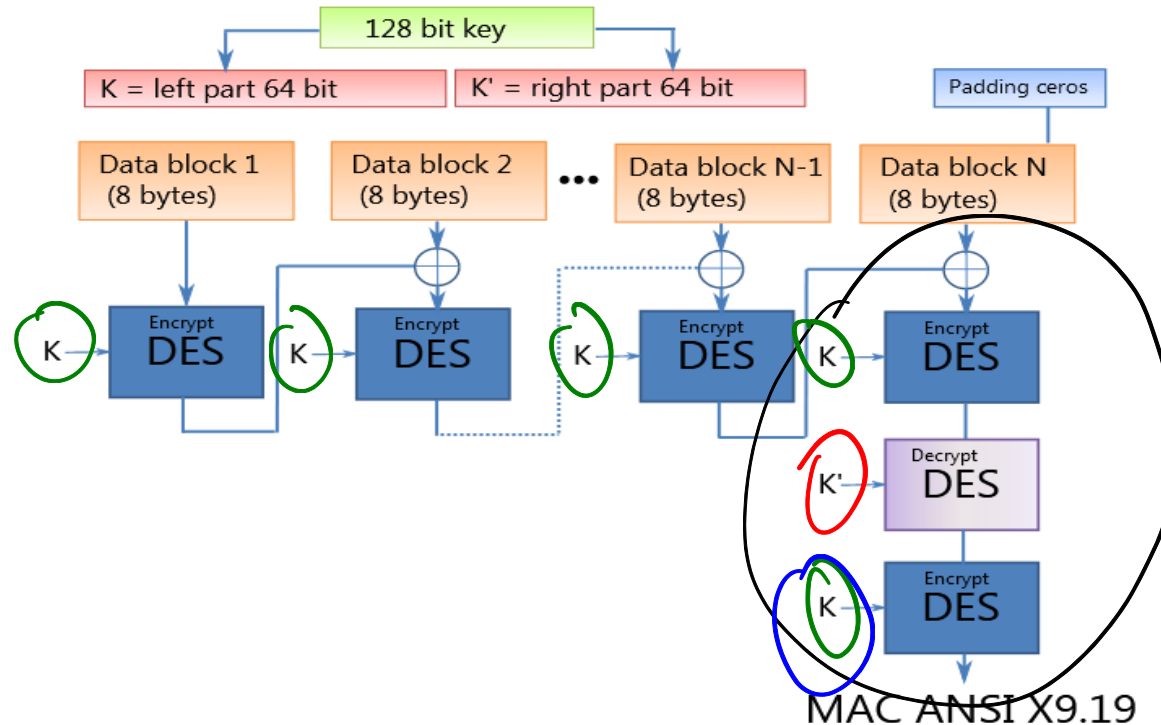
In der Einleitung (cf. Kap. 14.1) haben wir 3 Typen von Hashfunktionen definiert. Um welchen Typus handelt es sich hier?

Der CBC-MAC ANSI X9.9, resp. ISO 8730



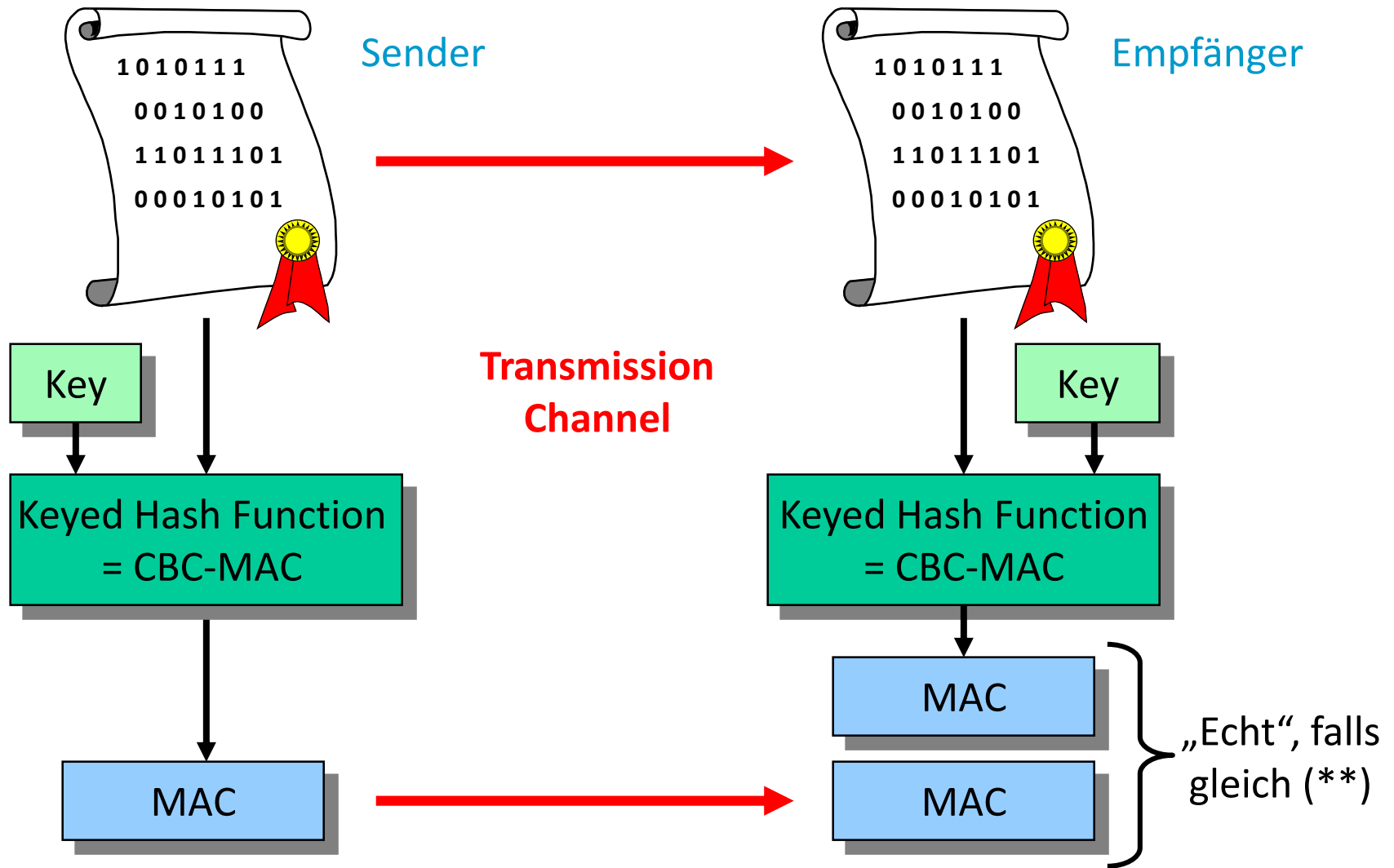
- Der CBC-MAC, wie oben dargestellt, ist die **Mutter** aller Integritätsschutzmechanismen mit Blockchiffren. Er wird in allen Lehrbüchern dargestellt.
- Er hat gewisse (theoretische) Schwächen und ist eigentlich als Standard schon länger zurückgezogen.
- Es gibt diverse Anpassungen (cf. nächste Folie), gemäss ISO 9797-1 → ist aber auch als Standard zurückgezogen.
- In der Industrie gibt es viele weitere, individuell leicht angepasste Versionen.
- All diese Formen werden aber nach wie vor konkret verwendet. Dies gemäss Aussagen von Securityspezialisten, die „nahe am Geschehen sind“, so im Sinne „never change a winning team“.

Der CBC-MAC nach ISO 9797-1, resp. ISO 9807



- In ISO 9797-1 wurden 6 Varianten definiert, die oben dargestellte ist eine dieser Varianten und wurde als ANSI X9.19 Standard aufgeführt.
- Es gibt auch neuere Varianten, eben vor allem auch im Zusammenhang von kombinierten Modi.
- **Bemerkung:** Anstatt DES könnte natürlich auch AES oder ein anderer Blockchiffrierer stehen. Die Schlüsselgrößen müssten dann entsprechend angepasst werden. Im **grünen** Block würde bei AES als Blockchiffre 256, 384 resp. 512 bit key stehen.

Verifikation mit CBC-MAC (*)



(**) zu 99,99... 9% echt falls gleich, aber zu 100% gefälscht, falls nicht gleich!

Spezielle Modi (#)

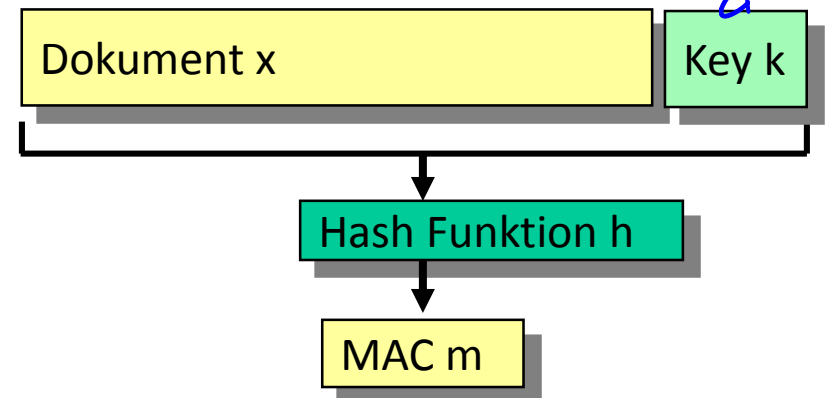
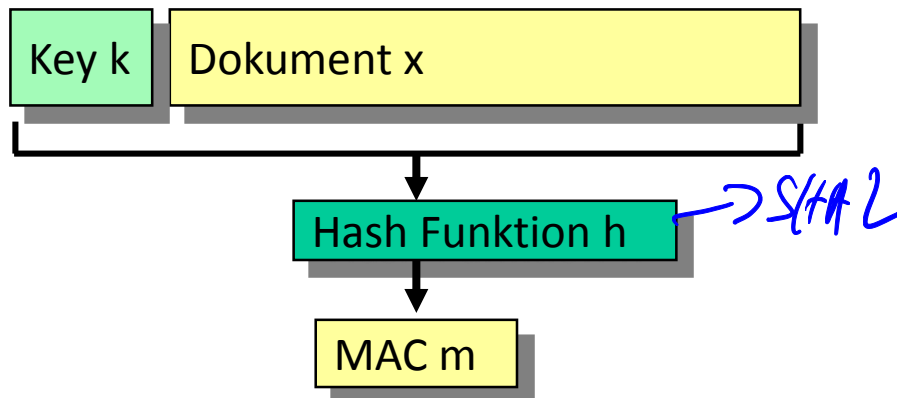
- Mit dem GCM (Galois Counter Mode) wurde ein Mode kreiert, der sowohl verschlüsseln wie MAC'en kann → sie Kap. 5.1.6 in [CP].
- Es gibt weitere solche Modi wie man z.B. im Buch von Bruce Schneier «Applied Cryptography» nachlesen kann.
- Mit dem GMAC (Galois Counter Message Authentication Code) hat man einen Mode entwickelt, so dass auch die MAC Berechnung parallelisiert werden kann. Dieser Mode wird z.B. in IPSec verwendet.
- Viele weitere sind in [CP] in Kap. 12.5 erwähnt.
- Spannend sind kombinierte Modi (Verschlüsseln und MAC'en) als Ersatz des Galois Counter Mode (GCM), da dieser Schwächen enthält. Die Sieger des CAESAR Wettbewerbs heissen ACRON & AEGIS.
- **CAESAR: Competition for Authenticated Encryption: Security, Applicability and Robustness**
- Das ist aber alles für uns im Rahmen dieses Moduls nicht von Bedeutung.

Idee resp. Geschichte des HMAC

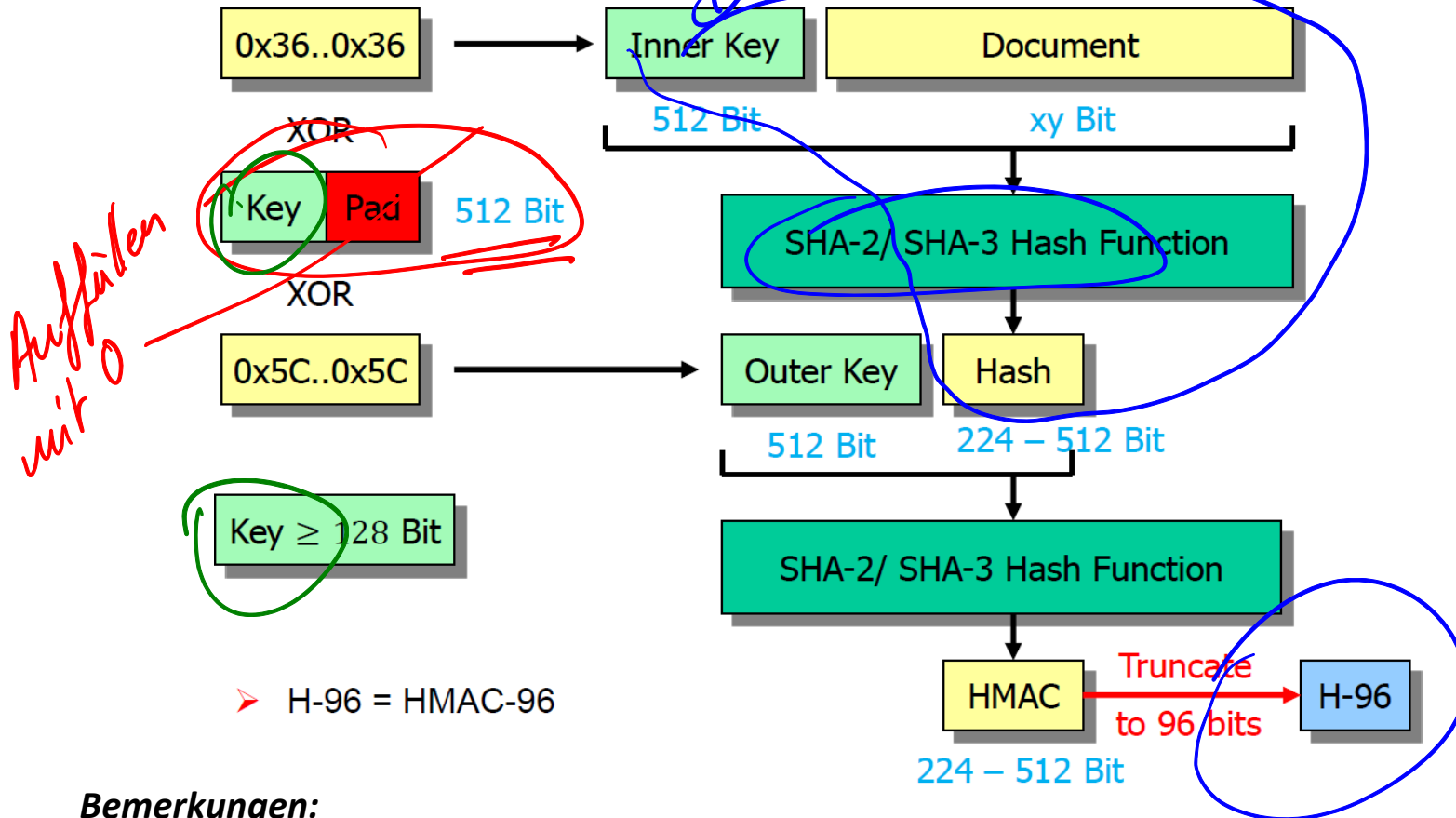
Schon früh kam der Wunsch auf, (grosse) Files gegen Abändern zu schützen. Oft enthielten diese Files keine Daten, die gegen Abhören geschützt werden mussten. D.h. eine Verschlüsselung – und damit z.B. Blockchiffren – waren nicht nötig. Dazu kommt, dass dieser Integritätsschutz nicht rechenintensiv sein sollte. Da kam die Idee schlüssellose Hashfunktionen zu nehmen und der Meldung M einen Schlüssel vor-, resp. anzuhängen und Ganze zu hashen.

MAC-Konstruktionen mit Hash-Funktionen, cf. Kap. 12.2 in [CP-D]. Dazu gibt es zwei naheliegende Ansätze:

- Der Secret-Prefix-MAC: $m = MAC_k(x) = h(k \parallel x)$ (linke Skizze)
- Der Secret-Suffix-MAC: $m = MAC_k(x) = h(x \parallel k)$ (rechte Skizze)
- Beide haben Schwächen, cf. [CP-D], Kap. 12.2.1 & 12.2.2
- Konsequenz: HMAC nach RFC 2104



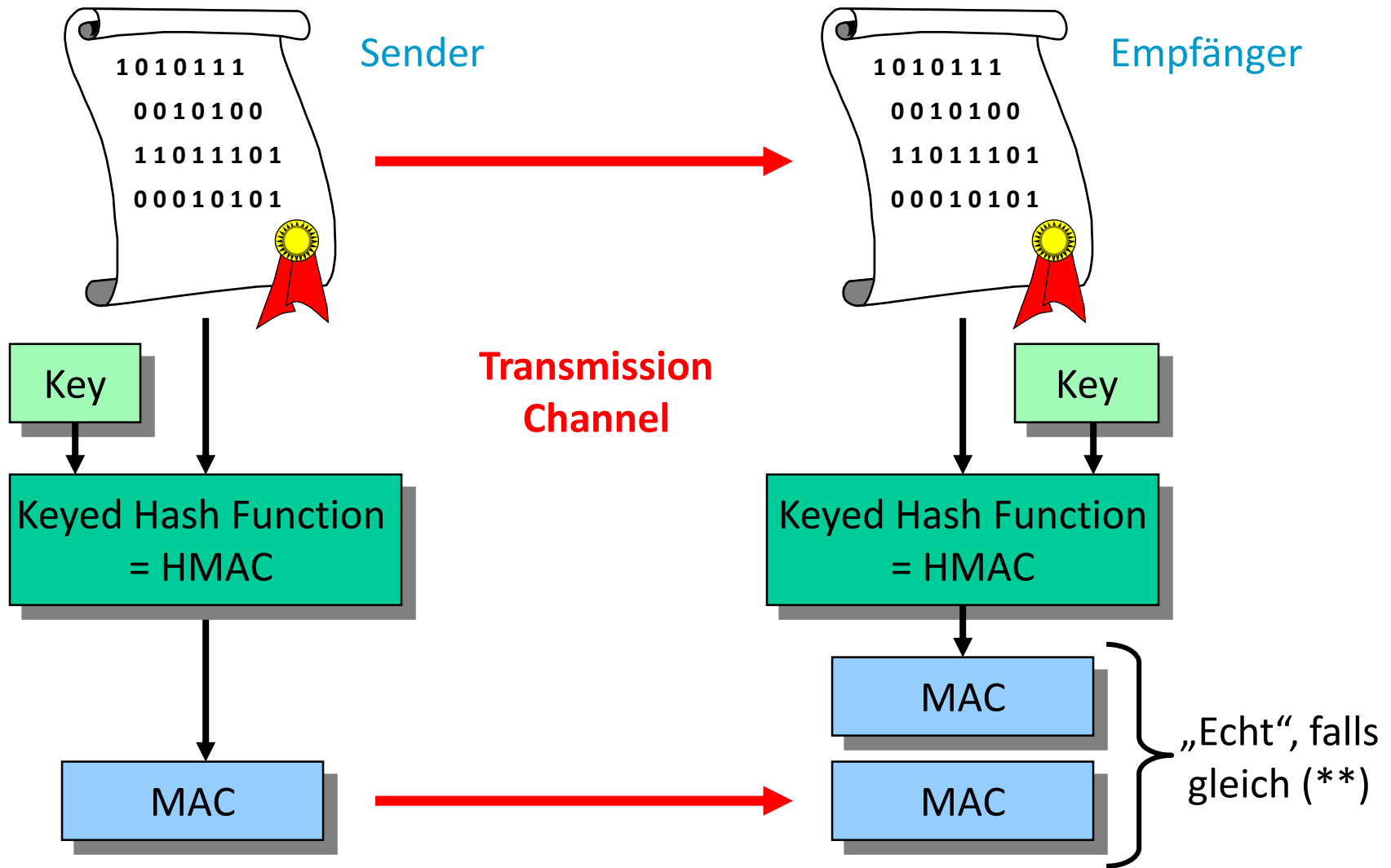
HMAC (HashMAC) nach RFC 2104



Bemerkungen:

- Grundsätzlich kann der Key bis 512 Bit gross sein.
- Warum mit den Konstanten 36 und 5C gepadded wird, ist im RFC nicht ersichtlich.
- Gemäss BSI sind als Hashfunktionen nur noch SHA2 & 3 erlaubt.
- Wird oft in den Internetprotokollen wie TLS, IPSEC usw. verwendet, wobei in den neueren Versionen nun auch Kombi Modi wie GCM verwendet werden, cf. JS Skript, Kap. 8.6.
- Das Abschneiden auf 96 Bit ist eigentlich eine Schwächung, cf. JS Skript, Kap. 8.2.4.

Verifikation mit HMAC



(*) zu 99,99... 9% echt falls gleich, aber zu 100% gefälscht, falls nicht gleich!

MAC (Typ 2) und MDC (Typ 3) im Vergleich

WAS	Charakteristika
MAC = Typ 2	<ol style="list-style-type: none"> 1. Muss (wegen dem Schlüssel) in einem Sicherheitsmodul implementiert sein. 2. Bietet (wegen dem Geheimnis) echte Integrität. 3. Für CBC-MAC gilt speziell: oft eher geringe Rechenrate. (Grund: sehr oft ist die Schnittstelle zum Sicherheitsmodul sehr langsam!!) 4. Es ist schwierig (resp. es muss schwierig sein), zwei Nachrichten mit dem gleichen MAC zu finden. 5. Ist ein <u>Nachrichtenintegritäts- & „Modifikationserkennungswert“ (*)</u>.
Hashfunktion = MDC = Typ 3	<ol style="list-style-type: none"> 1. Braucht kein Sicherheitsmodul. 2. Bietet alleine keine Integrität (in Kombination mit anderen Mechanismen kann aber Integrität erreicht werden, cf. Kap. 14.7.). 3. Ist so konzipiert, dass der Alg. auch in SW sehr, sehr schnell ist. 4. Es ist schwierig (resp. es muss schwierig sein), zwei Nachrichten mit dem gleichen MAC zu finden. 5. Hat beim Signieren die Rolle eines <u>„Meldungsrepräsentanten“</u>. 6. Ist nur ein <u>„Modifikationserkennungswert“</u> im Sinne von unbeabsichtigtem Verändern („unintelligente“ Angreifer). Ein „intelligenter“ kann die Nachricht verändern und den neuen Hash berechnen.

(*) Der Begriff „Modifikationserkennungswert“ ist kein offizieller Begriff.

Reihenfolge von Verschlüsseln und MAC'en

Bez. der Reihenfolge von Verschlüsseln und MAC'en, resp. Signieren ist schon länger eine relativ hitzige Debatte am Laufen, cf. JS Skript Kap. 8.2.5.

Wenn auf dem Application Layer die Kryptographie eingesetzt wird, lautet die klassische Variante ist:

Über den Klartext den MAC oder die Signatur berechnen und dann Verschlüsseln.

Das sieht dann in einer Übersichtszeichnung wie folgt aus:

siehe Präsenz 13

Header	Body oder Meldungsinhalt	Trailer
Mit MAC geschützt		MAC ist im Trailer
unverschlüsselt	verschlüsselt	unverschlüsselt
Enthält z.B. Routingdaten, Datum, Zeit & IV	Enthält die gesamten Nutzdaten	Enthält z.B. MAC und/oder verschlüsselte Schlüssel.
Unverschlüsselt und mit MAC geschützt.	Verschlüsselt und mit MAC geschützt.	In der Regel weder noch. (**)

(**) Es gibt aber Protokolle, z.B. der Encapsulating Security Payload (ESP) bei IPSec, wo ein Teil des Trailers verschlüsselt und mit einem MAC versehen ist. In SSL 2.0 wird der MAC ebenfalls verschlüsselt der Grund liegt hier darin, dass in SSL 2.0 eine schwache MAC-Konstruktion gewählt wurde.

CBC-MAC (HMAC) & digitale Signaturen in Authentication Schemes

Prüfung 12

User Authentication → Präsenz 12 „Protokolle“



“On the Internet, nobody knows you’re a dog.”

- Username / Password
Dictionary Attacks
- One-Time Passwords
Token: SecureID, etc.
- Symmetric Algorithms
C-R-Protocols
- Public Key Algorithms
Smartcards, Certificates,
Public Key Infrastructure,
C-R-Protocols
- Biometrical Methods
Fingerprint, Iris-Scan,
Voice, Face, Hand, etc.

MDC ✓

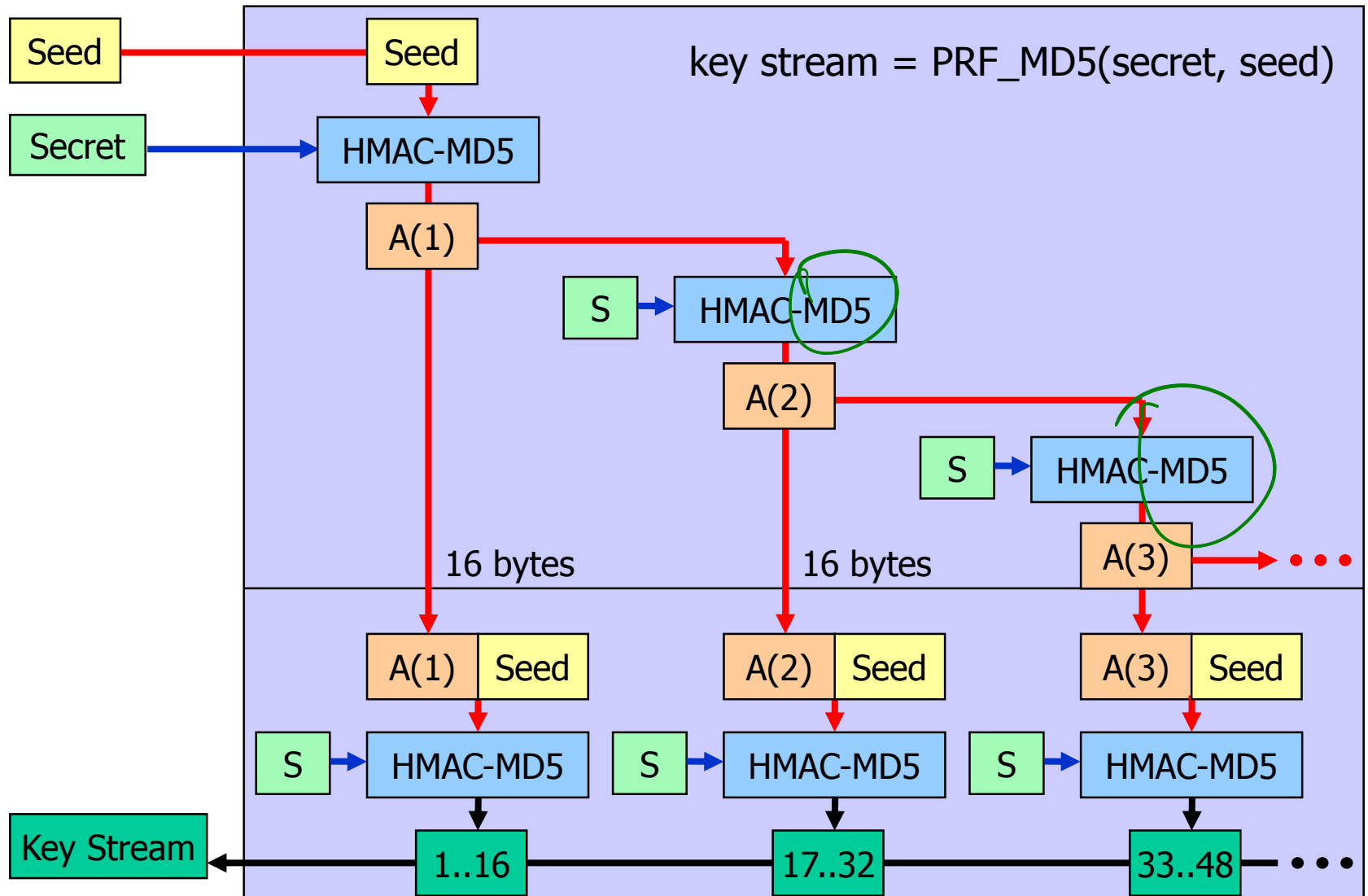
Typ 3

in der Schlüsselgenerierung (#)

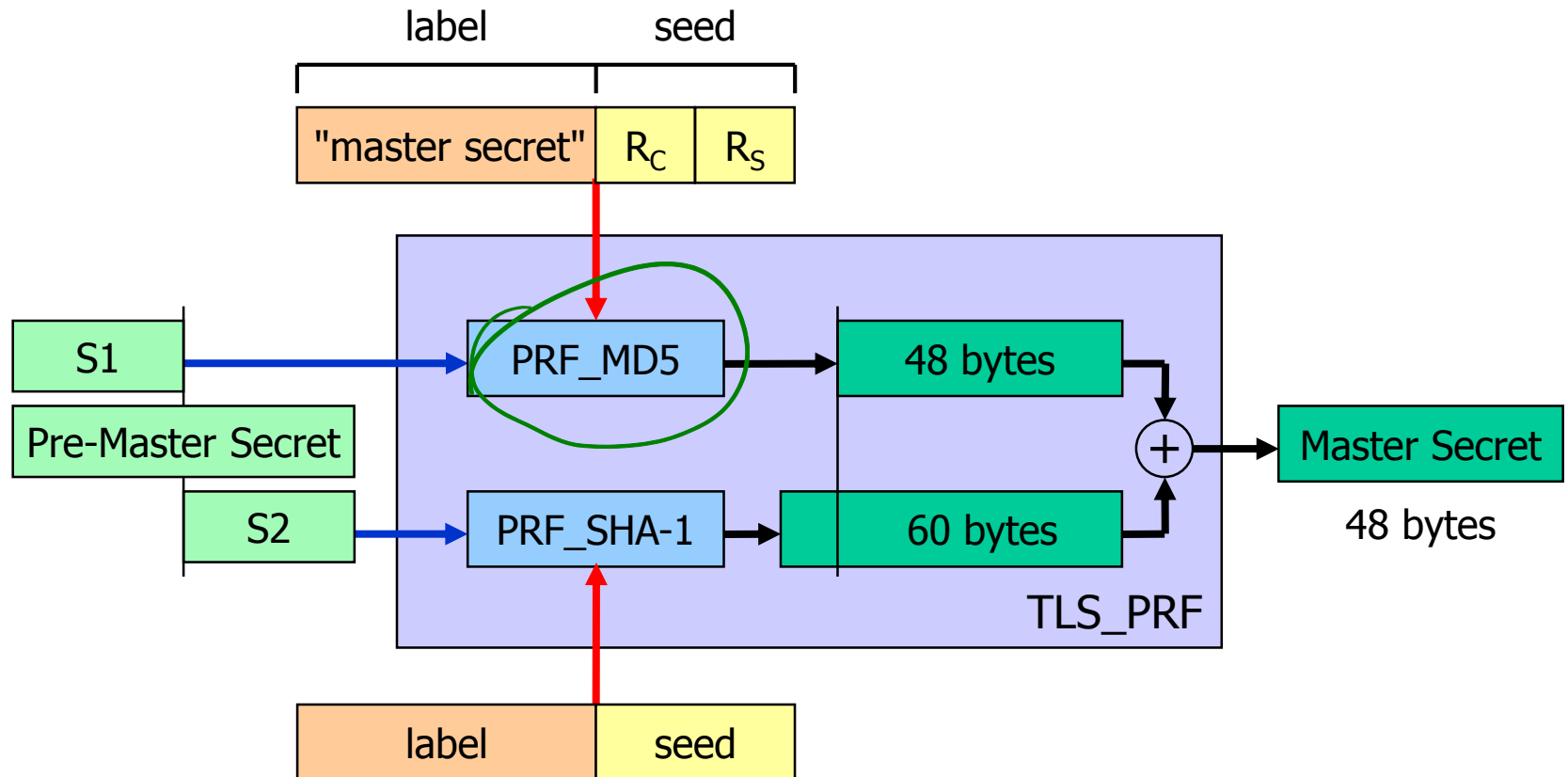
Hashfunktionen in der Schlüsselgenerierung (#)

- In der elektronischen Zahlungswelt wird die Schlüsselgenerierung oft mit Blockchiffren gemacht, cf. Präsenz 2, Kap. 8.3 & 8.4. Wir gehen aber nicht weiter darauf ein.
- In der Netzwerksicherheit wird die Schlüsselgenerierung in vielen Fällen mit Hashfunktionen gemacht.

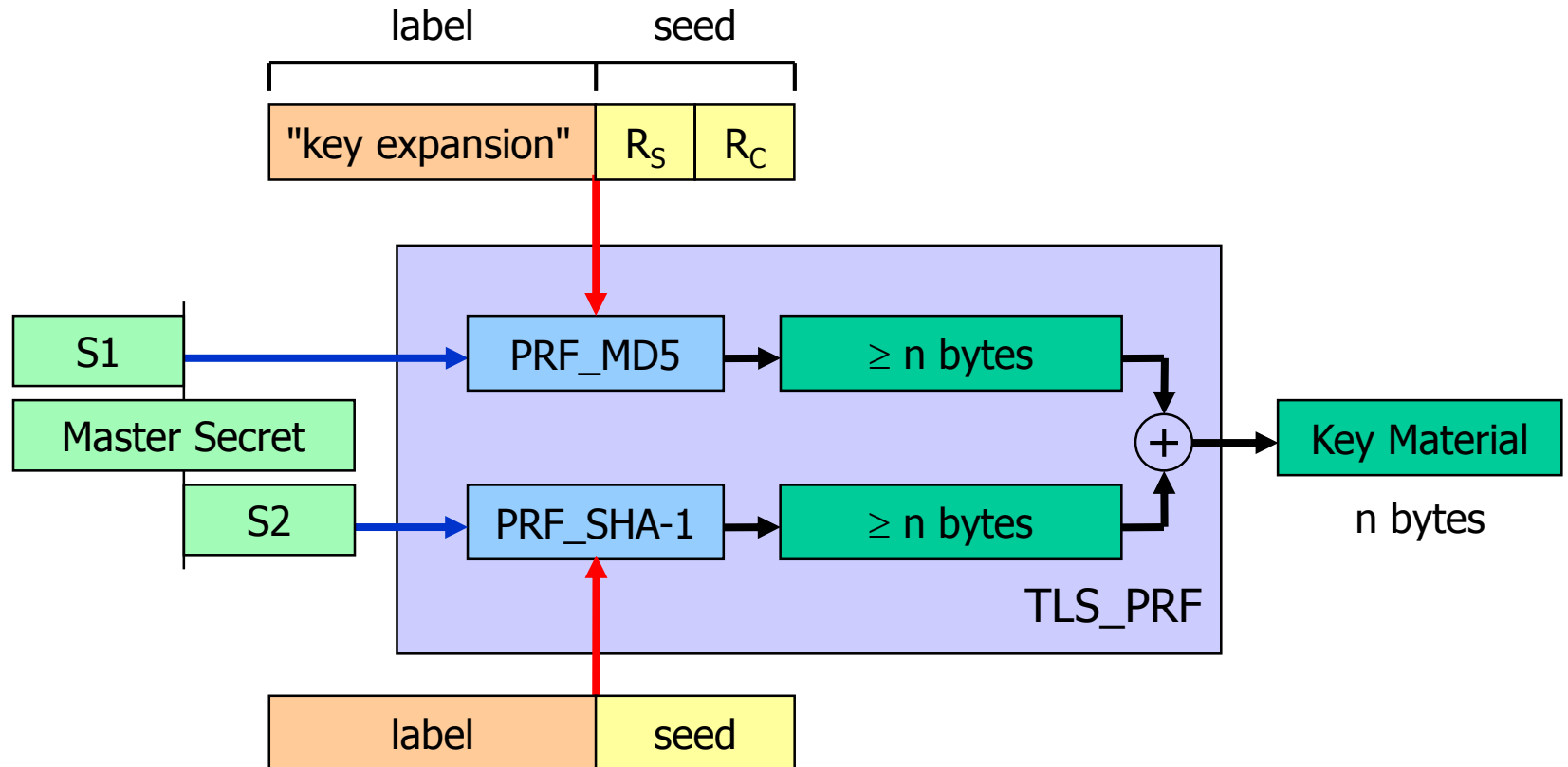
Pseudo Random Function (PRF) (#)



Computing the Master Secret (#)



Generating Key Material (#)



key stream = TLS_PRF(secret, label, seed)

Warum besteht ein Unterschied zw. Unterschrift und HMAC Generierung und Schlüsselerzeugung?

- Frage: Warum besteht ein Unterschied zw. Unterschrift und HMAC Generierung und Schlüsselerzeugung?
- Antwort:
 - Eine Unterschrift muss in 15 – 20 Jahren noch gültig, resp. integer sein.
 - Ein Integritätsschutz einer Meldung ist sehr viel kurzlebiger, in der Regel nach Erhalt der Meldung nicht mehr relevant.
 - Dito für einen Sessionschlüssel (man beachte auch, dass z.T. während einer Verbindung ein Schlüsselwechsel passiert).

Kap. 8.4

RANDOMFUNKTIONEN

MIT ...

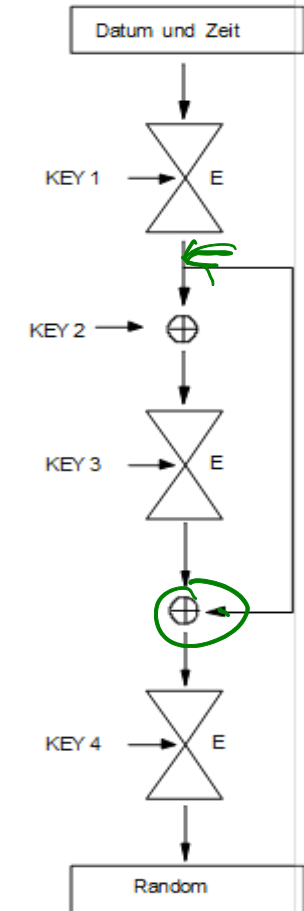
- ... BLOCKCHIFFREN
- ... HASHFUNKTIONEN

Beispiel einer Randomfkt. mit Blockchiffren

Einwegfunktion

Bemerkungen:

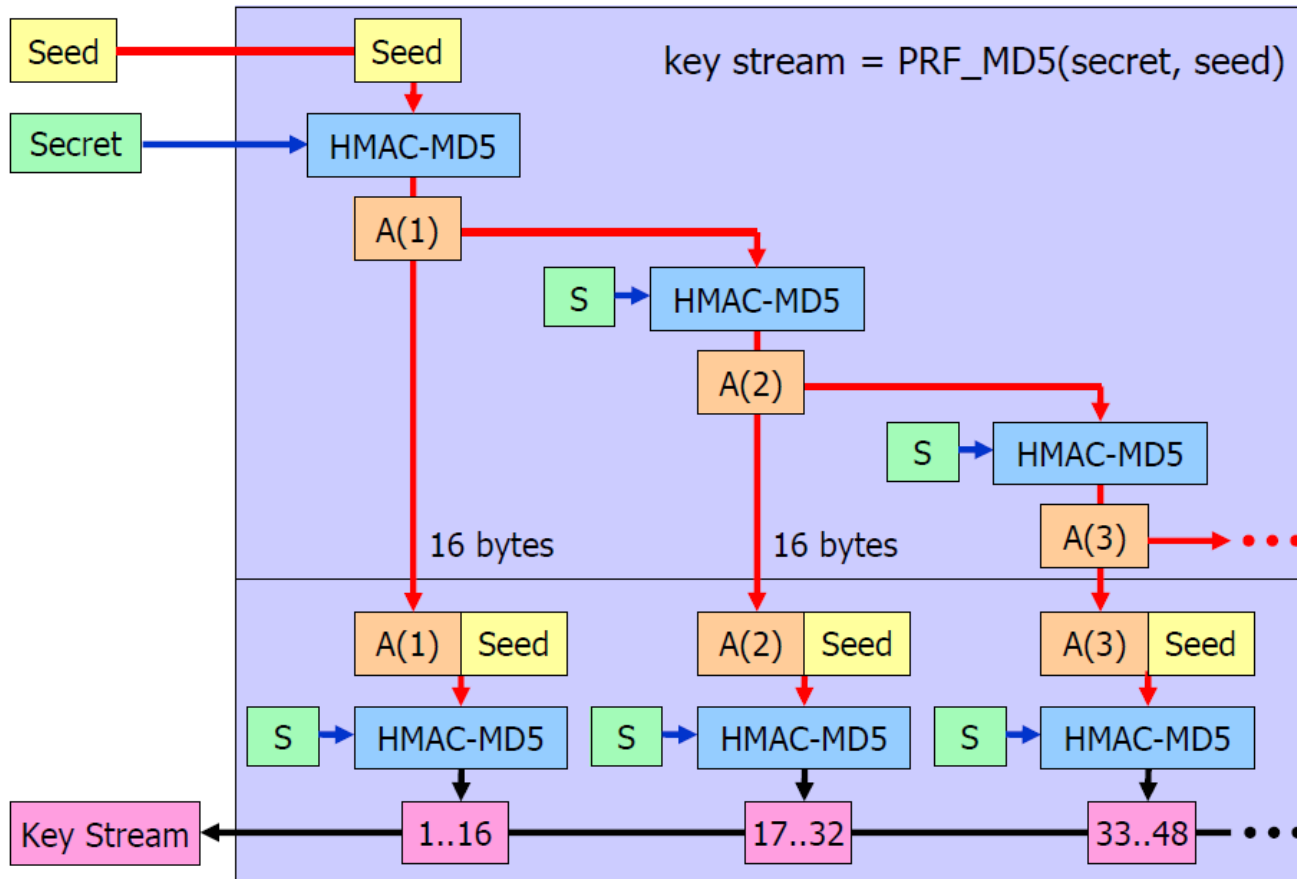
- Die Version mit nur 2 verschiedenen Schlüsseln (Key 3 = Key 4 = Key 1) war früher im Annex von ISO 8732 standardisiert. Der Standard – und damit der Annex – wurde zurückgezogen.
- **Nie** Hardware Zufallszahlengeneratoren (z.B. Rauschdioden) alleine verwenden.
- Wenn ein Hardware Zufallszahlengenerator (z.B. Rauschdiode) zur Verfügung steht, dann soll er nur benutzt werden, um den obigen Input zu verbessern.
- **Vorschlag:** Input = (Datum || Zeit) \oplus Output der „Rauschdiode“
- **Analyse des Vorschlags:** Sollte die Rauschdiode kleine Zyklen liefern, ist der Input immer noch so gut, wie (Datum || Zeit) alleine.
- Wird eine solche Funktion zur Erzeugung von Schlüsseln gebraucht, dann nennt man sie auch **KDF (Key Derivation Function)**.
- Im Rahmen des Key Managements (Präsenz 4) werden wir auf diese Funktion zurückkommen.



Beispiel einer Randomfkt. mit Hashfkt. (*)

(*) Nicht Prüfungstoff, nur Wissen, dass es solche Randomfunktionen gibt.

Pseudo Random Function (PRF)



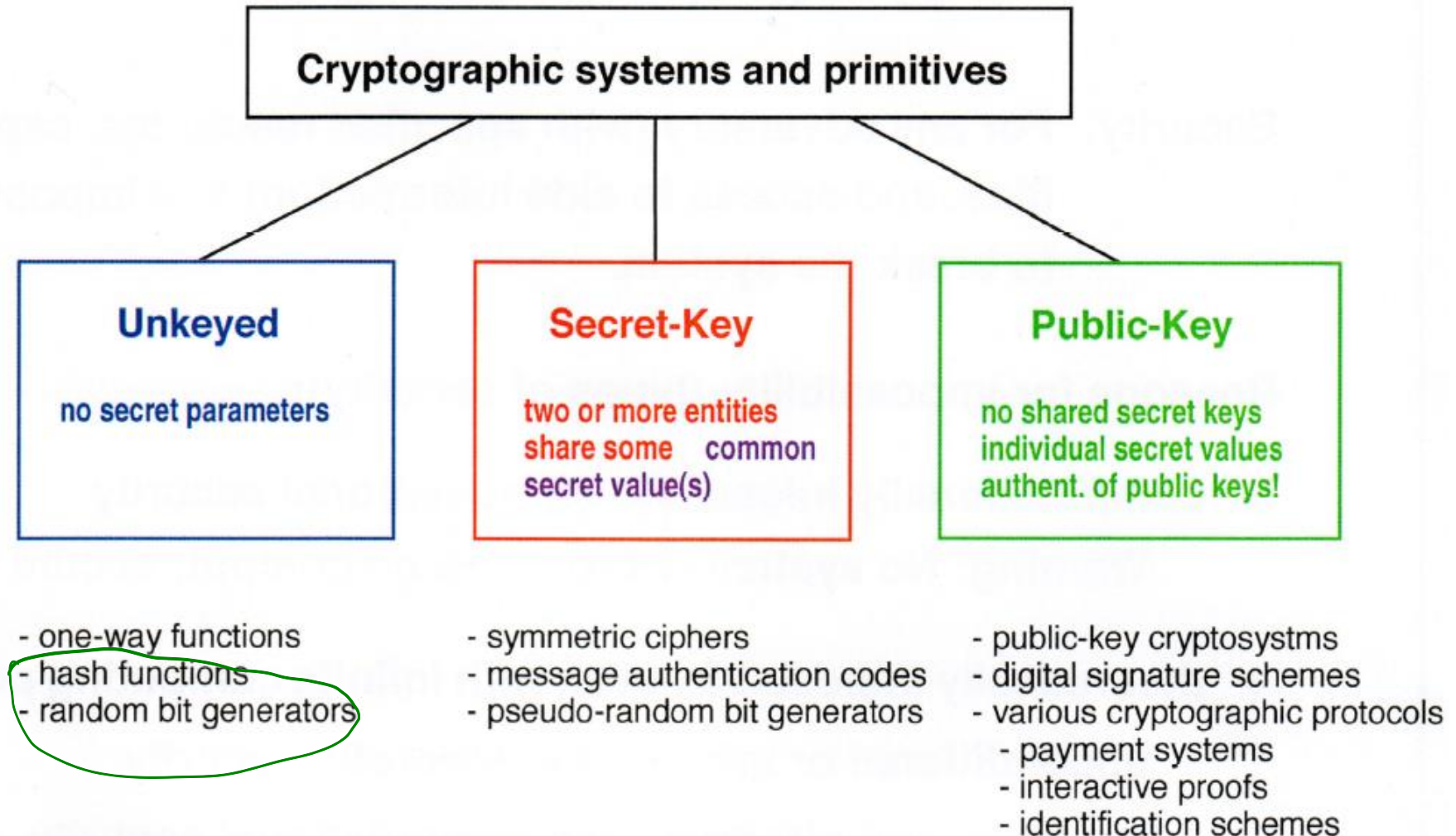
Kap. 27

EINWEGFUNKTIONEN

ÜBERSICHT & VORBEREITUNG AUF ASYMMETRISCHE VERFAHREN

Einleitung

Neben der **Secret und Public Key** Kryptographie betrachtet Prof. Dr. U. Maurer in seiner erweiterten Übersicht die **Unkeyed** (also schlüssellose) Kryptographie. Wobei wir im Verlaufe des Moduls gesehen haben, dass es sehr wohl auch **keyed** Einwegfunktionen gibt.



Schreibweise

Die (verschiedenen) Schreibweisen vom Begriff „Einwegfunktion“ in deutschsprachigen Büchern:

- **Einwegfunktion (wir benutzen diese Schreibweise, sie kommt am häufigsten vor).**
- Einweg-Funktion
- Ein-Weg-Funktion
- Es ist nicht auszuschliessen, dass es weitere gibt.

OWF

In der englischen Literatur wird mehrheitlich der Begriff „one-way function“ verwendet.

Bemerkungen:

- 1) Es ist mir kein (deutsches oder englisches) Buch bekannt, dass das Thema „Einwegfunktion“ oder „one-way function“ zentral in einem Kapitel zusammengefasst thematisiert.
- 2) Wir haben uns im Rahmen der asymmetrischen Kryptographie (Kap. 14 im Skript), Randomfunktionen, resp. Schlüsselableitungsfunktionen mit Blockchiffren (Kap. 8.4.1) und Hashfunktionen (Kap. 14) mit Einwegfunktionen beschäftigt.

Definition

Ich zitiere hier die Definition aus [CP-D], S. 177. CP spricht hier auch von einer informellen Definition.

Definition 6.1 (Einwegfunktion)

Eine Funktion $f(\cdot)$ ist eine Einwegfunktion, wenn:

1. $y = f(x)$ rechentechnisch einfach und
2. $x = f^{-1}(y)$ technisch unmöglich zu berechnen ist.

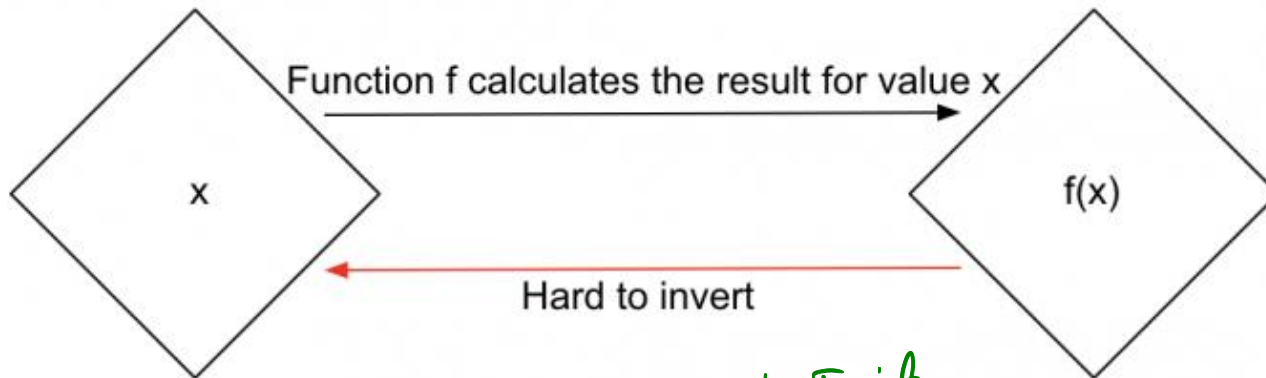
- Offensichtlich sind die Adjektive **einfach** und **unmöglich** nicht sehr exakt.
- Mathematische Sicht: **einfach** = **Berechnung in polynomialer Zeit möglich**.
- Also:
 - Berechnung von $y = f(x)$ so schnell sein, dass keine unverhältnismäßig langen Verzögerungen entstehen.
 - Berechnung der Umkehrfunktion $x = f^{-1}(y)$ muss derart aufwendig sein, dass selbst mit den besten Algorithmen und sehr vielen Rechenressourcen nicht in einer akzeptablen Zeit durchgeführt werden kann, z.B. 100.000 Jahre oder länger dauern würde (*).

(*) Anmerkung JS:

Dies erreicht man (als notwendige Bedingung), dass sich die Rechenzeit der Umkehrfunktion pro Bit exponentiell erhöht.

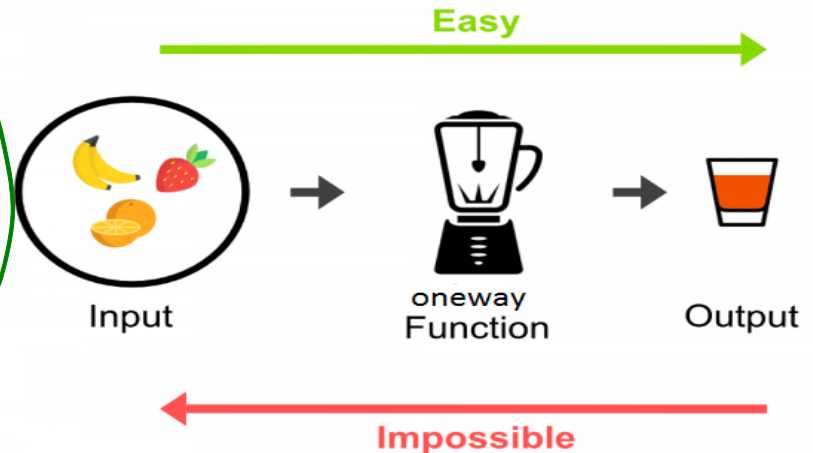
Bilder von Einwegfunktionen

Die typische mathematische Zeichnung sieht in etwa wie folgt aus:



Es gibt aber auch sehr anschauliche:

*evtl. mit Trick
besser!*



Typen von Einwegfunktionen

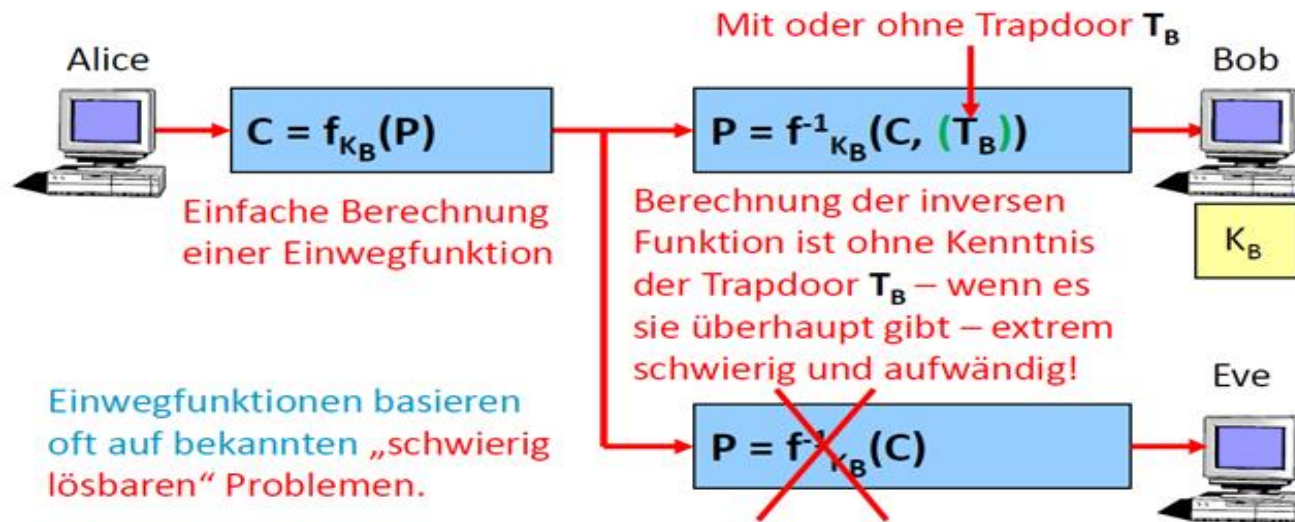
- Es gibt zunächst Einwegfunktionen ...
 - ... mit Trapdoor (d.h. mit einem Trick kann $x = f^{-1}(y)$ ebenfalls schnell berechnet werden).
 - ... ohne Trapdoor (d.h. niemand kennt einen Trick, um $x = f^{-1}(y)$ schnell berechnen zu können).
- Einwegfunktionen in der asymmetrischen Kryptographie, z.B.
 - Das Faktorisierungsproblem: für eine grosse Zahl N die Primfaktoren p und q zu finden, so dass $N = p \cdot q$
 - Das Berechnen von der diskreten e -ten Wurzel mod N : $x = \sqrt[e]{y} \bmod N$
 - Das Berechnen des diskreten Logarithmus mod p :
$$y = g^x \bmod p \Rightarrow x = \log_g y \bmod p$$
- Kryptographisch sichere Hashfunktionen ...
 - ... ohne Schlüssel wie SHA-3 usw.
 - ... mit Schlüssel wie (CBC-)MAC oder HMAC
- Schlüsselableitungsfunktionen (KDF = Key Derivation Function) oder Randomfunktionen mit Blockchiffren, wie z.B. in Folie 55, resp. Kap. 8.4.

Aufgabe 7 Warum ist eine Prüfziffer eine Hashfunktion, aber nicht eine Einwegfunktion ist.

Einwegfunktionen in der asym. Kryptographie

- Die Erfinder

- Whitfield Diffie und Martin Hellman 1976
- Ralph Merkle 1978



- Einwegfunktionen mit Trapdoor, d.h. es gibt einen «Trick», so dass die „unmögliche“ Berechnung von $x = f^{-1}(y)$ gleich einfach ist wie $y = f(x)$. Die Sicherheit liegt hier ja genau darin, dass dieser «Trick» nur einer Stelle bekannt ist.
- Einwegfunktionen ohne Trapdoor, d.h. es gibt keinen Trick. D.h. die Berechnung von $x = f^{-1}(y)$ ist für alle schwierig! Die Sicherheit liegt hier ja genau darin, dass es «keinen Trick» gibt.
- Eine solche Einwegfunktion kann eine Einwegpermutation sein.

Einwegfunktionen mit/ohne Trapdoor

Einwegfunktion mit Trapdoor

Beim RSA: Die e-te Wurzel mod N berechnen: mit Trapdoor

- Das Potenzieren $y = f(x) \equiv x^e \bmod N$ ist rechenintensiv, aber einfach.
- Das Inverse $x \equiv f^{-1}(y) \equiv y^{1/e} \equiv \sqrt[e]{y} \bmod N$ ist extrem schwierig.
- Dabei ist das Berechnen der „e-ten Wurzel mod N“ das eine Problem, das Faktorisieren von $N = p \cdot q$ das andere. Der RSA kann gebrochen werden, wenn nur eines der zwei Probleme gelöst ist.

Einwegfunktion ohne Trapdoor

Beim Diffie-Hellman: Der Logarithmus mod p berechnen: ohne TP

- Die Potenz $y = f(x) \equiv a^x \bmod p$, p prim ist rechenintensiv, aber einfach.
- Das Inverse (der diskrete Logarithmus) $x \equiv f^{-1}(y) \equiv \log_a y \bmod p$ ist für alle – also auch für Alice und Bob – extrem schwierig.

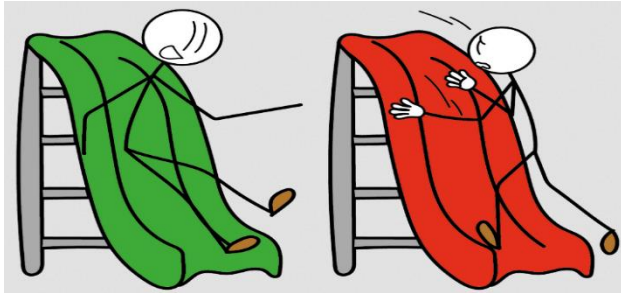
Bei Elliptischen Kurven: ein anderes disk. Log. Problem: ohne TP

- Die Multiplikation $Q = k \cdot P$, k eine unbekannte Zahl, P und Q bekannte Punkte einer Elliptischen Kurve.
- Das Bestimmen von k aus $Q = k \cdot P$ ist ebenfalls für alle extrem schwierig, dabei um handelt es sich um eine andere Art von diskretem Logarithmus.
- Die Punkt-Division von $k = \frac{Q}{P}$ gibt es in diesem Sinne nicht.

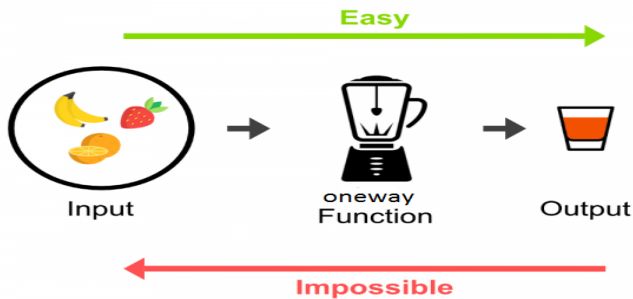
Bemerkung: Diese Einwegfunktionen sind genau genommen Einwegpermutationen.

Aufgabe 7 zu Einwegfunktionen

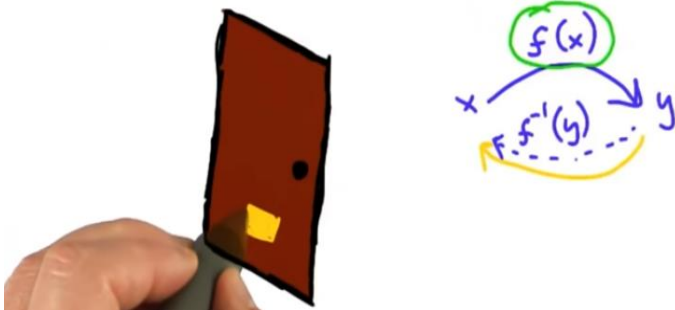
Geben Sie an bei den folgenden Einwegfunktion an, ob sie eher „mit“ oder eher „ohne“ Trapdoor sind. Geben Sie im Falle „mit“ den Trick an.



Einwegfunktion _____ Trapdoor



Einwegfunktion _____ Trapdoor



Einwegfunktion _____ Trapdoor

bei 12440 bis hier und ab
7073

Was zum Kuckuck sind Einwegpermutationen?

Permutation

Die Farben blau, grün, rot kann man auf $3! = 6$ Möglichkeiten anordnen:

(B, G, R), (B, R, G), (G, R, B), (G, B, R), (R, B, G), (R, G, B)

Wir betrachten das RSA-System mit

$$N = 3 \cdot 11 \text{ und } e = 13 \Rightarrow d = 13^{-1} \bmod \varphi(N) = 17 = 13^{-1} \bmod 20$$

Nun kann man alle Werte $x \in \mathbb{Z}_{33} = \{0, \dots, 32\}$ mit $y \equiv x^{13} \bmod 33$ verschlüsseln.
 $y \in \mathbb{Z}_{33}$

$x \in \mathbb{Z}_{33}$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$y = x^{13} \bmod 33$	0	1	8	27	31	26	18	13	17	3	10	11	12	19	5	9	4

$x \in \mathbb{Z}_{33}$	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
$x^{13} \bmod 33$	29	24	28	14	21	22	23	30	16	20	15	7	2	6	25	32

Frage: Wie viele solche Anordnungen gibt es?

Antwort: Theoretisch $33! \approx 8,7 \cdot 10^{36}$, aber die allermeisten sind nicht in der Form $x^y \bmod 33$ darstellbar. Also muss man sich fragen, wie viele solche y gibt es. Im Rahmen des RSA wird das noch genauer betrachtet. Sicher sind es weniger als 33, dann z.B. $y = 0$ muss ausgeschlossen werden. Die Grösse von mod N macht es aus, es trotzdem sehr viele solcher Werte y hat.

Weitere Beispiele von Einwegpermutationen

Wir betrachten $(\mathbb{Z}_{17}^*, \odot_p)$ und berechnen alle Potenzen $y = 3^x \bmod 17$.

$x \in \mathbb{Z}_{17}^*$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$3^x \bmod 17$	3	9	10	13	5	15	11	16	14	8	7	4	12	2	6	1

Nun sehen wir, dass die Elemente von \mathbb{Z}_{17}^* neu angeordnet werden. Theoretisch gäbe es $16! \approx 2,1 \cdot 10^{13}$, solcher Anordnungen. Mit der Formel $y = g^x \bmod 17$, g ist ein Generator $(\mathbb{Z}_p^*, \odot_p)$ gibt es nur $\varphi(\underbrace{\varphi(17)}_{=16}) = \varphi(2^4) = (2-1) \cdot (2^{4-1}) = 2^3 = 8$

Generatoren und somit auch nur 8 solcher Anordnungen.

Wir betrachten nun die Kurve $E: y^2 \equiv x^3 + x + 6 \bmod 11$ und den Punkt $P(2; 4)$.

Ohne auf Details einzugehen, gebe ich an, dass die Kurve genau 13 Punkte enthält und dass der Punkt $P(2; 4)$ mit $k \cdot P$ all diese Punkte erzeugt.


k	1	2	3	4	5	6	7	8
$k \cdot P$	(2; 4)	(5; 9)	(8; 8)	(10; 9)	(3; 5)	(7; 2)	(7; 9)	(3; 6)

k	9	10	11	12	13	14	15	16
$k \cdot P$	(10; 2)	(8; 3)	(5; 2)	(2; 7)	\mathcal{O}	(2; 4)	(5; 9)	(8; 8)


Wenn ich nun einen beliebigen anderen Punkt Q der Kurve – ausser der Nullpunkt \mathcal{O} – nehme, so gibt eine andere Anordnung der Punkt $k \cdot Q$. Es gibt somit 12 verschiedene Anordnungen. Theoretisch gäbe es aber $12! \approx 479'001'600$, denn der 13-te Punkt muss in diesem Falle immer \mathcal{O} sein.

$y = g^x \bmod p$ als Einwegpermutation

In der Praxis (siehe Kap. 13.3 Diffie-Hellman und 16.1 Einleitung zu den Kennzahlen) ist muss die Primzahl p eine Grösse von 2000 (bis 2022) und danach 3000 Bit haben. Wobei man in einer Untergruppe der Ordnung q arbeitet, dabei ist q eine Primzahl der Grösse von 250 Bit (man nimmt ja dann 256 Bit). Da die Gruppenordnung prim ist, sind alle $\varphi(q) = q - 1$ Elemente Generatoren. D.h. es gibt dann $2^{256} - 1 \approx 2^{256} \approx 10^{77}$ möglicher Anordnungen. Das entspricht in etwa der Anzahl Atome im Weltall!!



Date	Symmetric	Factoring Modulus	Discrete Key	Logarithm Group	Elliptic Curve	Hash	
2020 - 2022	128	2000	250	2000	250	SHA-256 SHA-512/256 SHA-384 SHA-512	SHA3-256 SHA3-384 SHA3-512
2023 - 2026	128	3000	250	3000	250	SHA-256 SHA-512/256 SHA-384 SHA-512	SHA3-256 SHA3-384 SHA3-512



Hashfunktionen als Einwegfunktionen

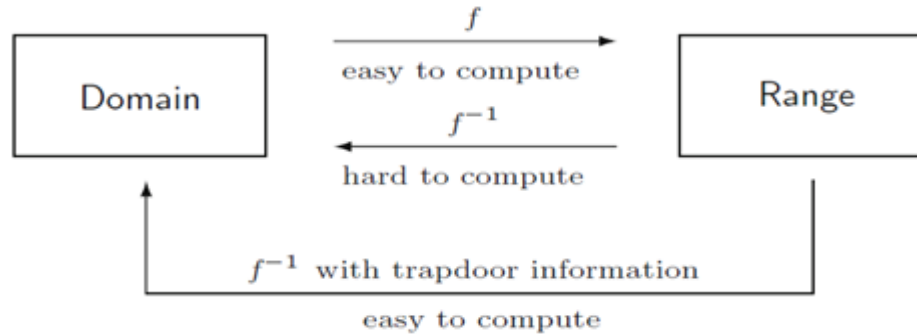
- Hashfunktionen ohne Schlüssel (MDC's) als Einwegfunktionen, cf. Kap. 27.5.1, sowie Kap. 14.2.
- Hashfunktionen mit Schlüssel (MAC, HMAC) als Einwegfunktionen, cf. Kap. 27.5.2, sowie Kap. 8.2 & 14.3 – 14.6.
- **Nachbearbeitung:** Bitte lesen Sie das Kap. 27.5 im JS Skript.

Blockchiffren als Einwegfunktionen

- Einführende Gedanken in Kap. 27.6.1.
- Blockchiffren als Schlüsselableitungsfunktionen (KDF = Key Derivation Function) in Kap. 27.6.2 & 8.4.1.
- **Nachbearbeitung:** Bitte lesen Sie das Kap. 27.5 im JS Skript.

Aufgabe 8

Figure: Illustration of an one-way trapdoor function



NR	Aufgabe	Auswahl
a)	Welche Berechnungen sind im Rahmen von Einwegfunktionen von der Art „ f easy to compute“?	<input type="checkbox"/> Die Ver- resp. Entschlüsselung mit XOR. <input type="checkbox"/> Die Ver- resp. Entschlüsselung mit einer Blockchiffre. <input type="checkbox"/> Die Berechnung der Werte, die beim DH Schlüsselaustausch über die Leitung geschickt werden. <input type="checkbox"/> Die Berechnung des sogenannten diskreten Logarithmus beim Diffie-Hellman Schlüsselaustausches. <input type="checkbox"/> Das Berechnen von $c = m^e \bmod N$ beim RSA. <input type="checkbox"/> Das Berechnen von $m = \sqrt[e]{c} \bmod N$ beim RSA. <input type="checkbox"/> Die Multiplikation $r \cdot P$ bei den Elliptischen Kurven. <input type="checkbox"/> Das Bestimmen vom unbekannten k bei Elliptischen Kurven, falls $Q = k \cdot P$. <input type="checkbox"/> Keine der Angaben ist zutreffend.
b)	Welche Berechnungen sind im Rahmen von Einwegfunktionen von der Art „ f^{-1} hard to compute“?	<input type="checkbox"/> Die Ver- resp. Entschlüsselung mit XOR. <input type="checkbox"/> Die Ver- resp. Entschlüsselung mit einer Blockchiffre. <input type="checkbox"/> Die Berechnung der Werte, die beim DH Schlüsselaustausch über die Leitung geschickt werden. <input type="checkbox"/> Die Berechnung des sogenannten diskreten Logarithmus beim Diffie-Hellman Schlüsselaustausches. <input type="checkbox"/> Das Berechnen von $c = m^e \bmod N$ beim RSA. <input type="checkbox"/> Das Berechnen von $m = \sqrt[e]{c} \bmod N$ beim RSA. <input type="checkbox"/> Die Multiplikation $r \cdot P$ bei den Elliptischen Kurven. <input type="checkbox"/> Das Bestimmen vom unbekannten k bei Elliptischen Kurven, falls $Q = k \cdot P$. <input type="checkbox"/> Keine der Angaben ist zutreffend.
c)	Welche Berechnungen sind im Rahmen von Einwegfunktionen von der Art „ f^{-1} with trapdoor information easy to compute“?	<input type="checkbox"/> Die Ver- resp. Entschlüsselung mit XOR. <input type="checkbox"/> Die Ver- resp. Entschlüsselung mit einer Blockchiffre. <input type="checkbox"/> Die Berechnung der Werte, die beim DH Schlüsselaustausch über die Leitung geschickt werden. <input type="checkbox"/> Die Berechnung des sogenannten diskreten Logarithmus beim Diffie-Hellman Schlüsselaustausches. <input type="checkbox"/> Das Berechnen der Potenzen beim RSA. <input type="checkbox"/> Das Berechnen des ursprünglichen Wertes beim RSA. <input type="checkbox"/> Die Multiplikation $r \cdot P$ bei den Elliptischen Kurven. <input type="checkbox"/> Das Bestimmen vom unbekannten k bei Elliptischen Kurven, falls $Q = k \cdot P$. <input type="checkbox"/> Keine der Angaben ist zutreffend.

Basis-Test Präsenz 3

Aussage	Richtig oder falsch?	Begründung
Eine kryptographisch sichere Hashfunktion ist auch eine Einwegfunktion.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch	
Eine Einwegfunktion ist auch eine kryptographisch sichere Hashfunktion.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch	
Eine Prüfziffer schützt z.B. vor versehentlichem Verschreiben.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch	
Für SHA-3 kann man vier zentrale Sicherheitseigenschaften aufzählen.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch	
Eine Hashfunktion (= MDC) muss ab 2023 mindestens 240 Bit Output haben.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch	
Den SHA-224 darf noch lange für Signaturen verwendet werden.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch	
Die wichtigste Funktion eines Hashes ist die Komprimierung einer Meldung.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch	
Der Kollisionsangriff ist weniger aufwendig, aber auch weniger realistisch als ein Pre-Image Angriff.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch	
Der Pre-Image Angriff ist weniger aufwendig, aber auch weniger realistisch als ein Kollisionsangriff.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch	
Den CBC-MAC gibt es in verschiedenen Ausprägungen.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch	
Ein HMAC und ein CBC-MAC haben unterschiedliche Anwendungszwecke.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch	

Basis-Test Präsenz 3, Fortsetzung

Aussage	Richtig oder falsch?	Begründung
Mit Blockchiffren kann man Zufallszahlen erzeugen.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch	
Mit Hashfunktionen kann man keine Zufallszahlen erzeugen.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch	
Die asymmetrische Kryptographie basiert vornehmlich auf Hashfunktionen.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch	
Die asymmetrische Kryptographie basiert vornehmlich auf Einwegfunktionen.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch	
Einwegfunktion mit Trapdoor ist ein Widerspruch in sich.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch	
Das Ziehen einer e-ten Wurzel mod N ($N = p \cdot q$ mit p, q grosse Primzahlen) ist eine Einwegfunktion ohne Trapdoor.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch	
Der diskrete Logarithmus ist eine Einwegfunktion ohne Trapdoor.	<input type="checkbox"/> richtig <input type="checkbox"/> falsch	
Sei g ein Generator in \mathbb{Z}_p^* und p eine Primzahl, dann ist $y \equiv g^x \text{ mod } p$ eine Einwegpermutation.	<input checked="" type="checkbox"/> richtig <input type="checkbox"/> falsch	



LÖSUNGEN DER AUFGABEN

Aufgabe 1

- Beim Vertippen bei einer Zahl mit Gewicht 1, ist klar, dass es eine andere Summe mod 10 gibt.
- Beim Vertippen bei einer Zahl mit Gewicht 3, gibt es ebenfalls es eine andere Summe mod 10 gibt. Das liegt daran, dass $3 \cdot x \bmod 10$ für $x = 0, \dots, 9$ immer einen anderen Wert gibt.

Berechnung $3x \bmod 10$

x	0	1	2	3	4	5	6	7	8	9
$3x \bmod 10$	0	3	6	9	2	5	8	1	4	7

Beim Vertauschen ist zu zeigen, wann gilt $x+3y \bmod 10 = y+3x \bmod 10$

x	0	1	2	3	4	5	6	7	8	9
y	0	1	2	3	4	5	6	7	8	9
$x+3y \bmod 10$	0	4	8	2	6	0	4	8	2	6
$3x+y \bmod 10$	0	4	8	2	6	0	4	8	2	6

0 ↔ 5 4 ↔ 6 2 ↔ 7 3 ↔ 8 1 ↔ 9

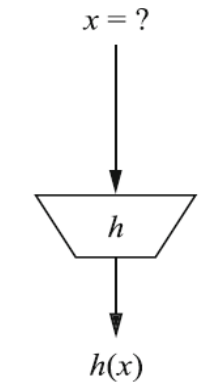
Aufgabe 2

Hashfunktion	Kryptographisch sicher		Kryptographisch nicht sicher
	Mit Schlüssel	Ohne Schlüssel	
1) Alph. Registratur			X
2) Ablage nach Datum			X
3) Parity-Bit			X
4) Byte Addition (mod 256)			X
5) CRC			X
6) CBC-MAC	X		
7) HMAC	X		
8) Konv. Prüfsumme			X
9) MDC = „Hashfunktion“		X	

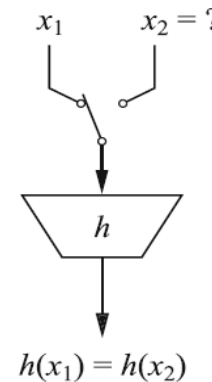
Aufgabe 3 In der Nomenklatur der Einleitung um den Typ 3 (krypt. sichere Hashfkt. ohne Schlüssel

Aufgabe 4

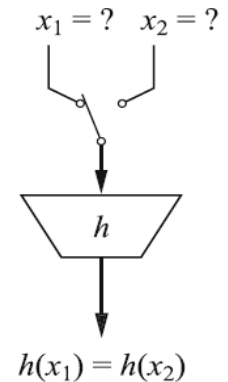
- Siehe rechts stehendes Diagramm.
- Eine Prüfziffer (Parity-Bit, CRC-16, Prüfziffer am Schluss der ISBN-Nummer usw.) sind kryptographisch nicht sichere Hashfunktionen, da keine der 3 Sicherheitseigenschaften erfüllt ist.



Urbildresistenz



schwache Kollisionsresistenz
(zweite Urbildresistenz)



Kollisionsresistenz

Aufgabe 5

Wir gehen vom Gedankenspiel mit n zufällig angetroffenen Personen aus:

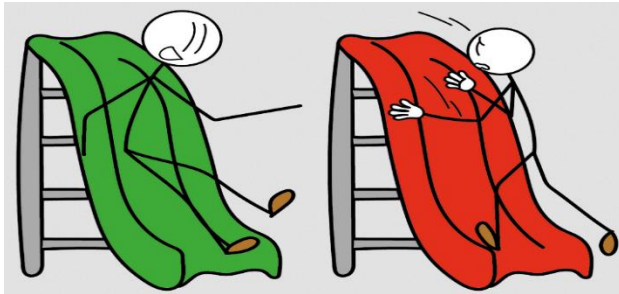
$$p = 1 - \bar{p} = 1 - \left(\frac{364}{365}\right)^n > \frac{1}{2} \Rightarrow -\left(\frac{364}{365}\right)^n > -\frac{1}{2} \Rightarrow \left(\frac{364}{365}\right)^n < \frac{1}{2} \Rightarrow n \cdot \log \frac{364}{365} < \log \left(\frac{1}{2}\right)$$

$$\Rightarrow n \cdot \log \left(\frac{364}{365}\right) < \log \left(\frac{1}{2}\right) \Rightarrow n \underset{(*)}{\geq} \frac{\log \left(\frac{1}{2}\right)}{\log \left(\frac{364}{365}\right)} = 252,6$$

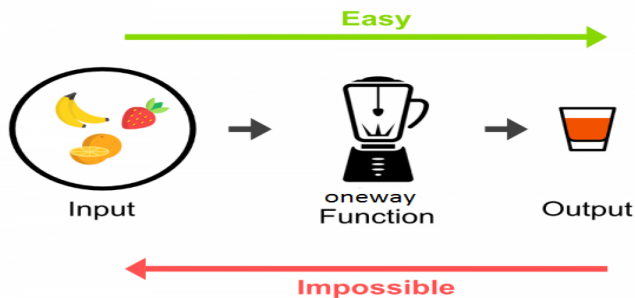
(*) Da $\log \left(\frac{364}{365}\right) < 0$, ändert das Ungleichheitszeichen.

Aufgabe 6 In der Nomenklatur der Einleitung ist ein CBC-MAC vom Typ 2.

Aufgabe 7



Einwegfunktion mit Trapdoor (z.B. Klebemittel an den Schuhen)



Einwegfunktion ohne Trapdoor



Einwegfunktion mit Trapdoor, z.B. beim RSA
ist es die Kenntnis der Faktorisierung von $N = p \cdot q$, damit kann $\varphi(N)$ und damit $d = e^{-1} \bmod \varphi(N)$ berechnet werden.

Aufgabe 8

NR	Aufgabe	Auswahl
a)	Welche Berechnungen sind im Rahmen von Einwegfunktionen von der Art „ f easy to compute“?	<input type="checkbox"/> Die Ver- resp. Entschlüsselung mit XOR. <input type="checkbox"/> Die Ver- resp. Entschlüsselung mit einer Blockchiffre. <input checked="" type="checkbox"/> Die Berechnung der Werte, die beim DH Schlüsselaustausch über die Leitung geschickt werden. <input type="checkbox"/> Die Berechnung des sogenannten diskreten Logarithmus beim Diffie-Hellman Schlüsselaustausches. <input checked="" type="checkbox"/> Das Berechnen von $c = m^e \bmod N$ beim RSA. <input type="checkbox"/> Das Berechnen von $m = \sqrt[e]{c} \bmod N$ beim RSA. <input checked="" type="checkbox"/> Die Multiplikation $r \cdot P$ bei den Elliptischen Kurven. <input type="checkbox"/> Das Bestimmen vom unbekannten k bei Elliptischen Kurven, falls $Q = k \cdot P$. <input type="checkbox"/> Keine der Angaben ist zutreffend.
b)	Welche Berechnungen sind im Rahmen von Einwegfunktionen von der Art „ f^{-1} hard to compute“?	<input type="checkbox"/> Die Ver- resp. Entschlüsselung mit XOR. <input type="checkbox"/> Die Ver- resp. Entschlüsselung mit einer Blockchiffre. <input type="checkbox"/> Die Berechnung der Werte, die beim DH Schlüsselaustausch über die Leitung geschickt werden. <input checked="" type="checkbox"/> Die Berechnung des sogenannten diskreten Logarithmus beim Diffie-Hellman Schlüsselaustausches. <input type="checkbox"/> Das Berechnen von $c = m^e \bmod N$ beim RSA. <input checked="" type="checkbox"/> Das Berechnen von $m = \sqrt[e]{c} \bmod N$ beim RSA. <input type="checkbox"/> Die Multiplikation $r \cdot P$ bei den Elliptischen Kurven. <input checked="" type="checkbox"/> Das Bestimmen vom unbekannten k bei Elliptischen Kurven, falls $Q = k \cdot P$. <input type="checkbox"/> Keine der Angaben ist zutreffend.
c)	Welche Berechnungen sind im Rahmen von Einwegfunktionen von der Art „ f^{-1} with trapdoor information easy to compute“?	<input type="checkbox"/> Die Ver- resp. Entschlüsselung mit XOR. <input type="checkbox"/> Die Ver- resp. Entschlüsselung mit einer Blockchiffre. <input type="checkbox"/> Die Berechnung der Werte, die beim DH Schlüsselaustausch über die Leitung geschickt werden. <input type="checkbox"/> Die Berechnung des sogenannten diskreten Logarithmus beim Diffie-Hellman Schlüsselaustausches. <input type="checkbox"/> Das Berechnen der Potenzen beim RSA. <input checked="" type="checkbox"/> Das Berechnen des ursprünglichen Wertes beim RSA. <input type="checkbox"/> Die Multiplikation $r \cdot P$ bei den Elliptischen Kurven. <input type="checkbox"/> Das Bestimmen vom unbekannten k bei Elliptischen Kurven, falls $Q = k \cdot P$. <input type="checkbox"/> Keine der Angaben ist zutreffend.

Basis-Test Präsenz 3

Aussage	Richtig o. falsch?	Begründung
Eine kryptographisch sichere Hashfunktion ist auch eine Einwegfunktion.	<input checked="" type="checkbox"/> richtig <input type="checkbox"/> falsch	
Eine Einwegfunktion ist auch eine kryptographisch sichere Hashfunktion.	<input type="checkbox"/> richtig <input checked="" type="checkbox"/> falsch	Die asymmetrische Kryptographie basiert auf solchen Einwegfunktionen, die keine Hashfunktionen sind.
Eine Prüfziffer schützt z.B. vor versehentlichem Verschreiben.	<input checked="" type="checkbox"/> richtig <input type="checkbox"/> falsch	
Für SHA-3 kann man vier zentrale Sicherheitseigenschaften aufzählen.	<input checked="" type="checkbox"/> richtig <input type="checkbox"/> falsch	Es sind dies die 3 zentralen Sicherheitseigenschaften. Die 4-te Eigenschaft ist, dass beim Ändern eines Bits im Input, im stat. Mittel die Hälfte der Bits im Hash ändern.
Eine Hashfunktion (= MDC) muss ab 2023 mindestens 240 Bit Output haben.	<input checked="" type="checkbox"/> richtig <input type="checkbox"/> falsch	
Den SHA-224 darf noch lange für Signaturen verwendet werden.	<input type="checkbox"/> richtig <input checked="" type="checkbox"/> falsch	Siehe vorherige Frage.
Die wichtigste Funktion eines Hashes ist die Komprimierung einer Meldung.	<input type="checkbox"/> richtig <input checked="" type="checkbox"/> falsch	Es wird nichts komprimiert. Die wichtigste Funktion ist die Stellvertreterfunktion beim Signieren.
Der Kollisionsangriff ist weniger aufwendig, aber auch weniger realistisch als ein Pre-Image Angriff.	<input checked="" type="checkbox"/> richtig <input type="checkbox"/> falsch	
Der Pre-Image Angriff ist weniger aufwendig, aber auch weniger realistisch als ein Kollisionsangriff.	<input type="checkbox"/> richtig <input checked="" type="checkbox"/> falsch	
Den CBC-MAC gibt es in verschiedenen Ausprägungen.	<input checked="" type="checkbox"/> richtig <input type="checkbox"/> falsch	

Basis-Test Präsenz 3, Fortsetzung

Aussage	Richtig oder falsch?	Begründung
Ein HMAC und ein CBC-MAC haben unterschiedliche Anwendungszwecke.	<input type="checkbox"/> richtig <input checked="" type="checkbox"/> falsch	
Mit Blockchiffren kann man Zufallszahlen erzeugen.	<input checked="" type="checkbox"/> richtig <input type="checkbox"/> falsch	
Mit Hashfunktionen kann man keine Zufallszahlen erzeugen.	<input type="checkbox"/> richtig <input checked="" type="checkbox"/> falsch	
Die asymmetrische Kryptographie basiert vornehmlich auf Hashfunktionen.	<input type="checkbox"/> richtig <input checked="" type="checkbox"/> falsch	
Die asymmetrische Kryptographie basiert vornehmlich auf Einwegfunktionen.	<input checked="" type="checkbox"/> richtig <input type="checkbox"/> falsch	
Einwegfunktion mit Trapdoor ist ein Widerspruch in sich.	<input type="checkbox"/> richtig <input checked="" type="checkbox"/> falsch	
Das Ziehen einer e-ten Wurzel mod N ($N = p \cdot q$ mit p, q grosse Primzahlen) ist eine Einwegfunktion ohne Trapdoor.	<input type="checkbox"/> richtig <input checked="" type="checkbox"/> falsch	
Der diskrete Logarithmus ist eine Einwegfunktion ohne Trapdoor.	<input checked="" type="checkbox"/> richtig <input type="checkbox"/> falsch	
Sei g ein Generator in \mathbb{Z}_p^* und p eine Primzahl, dann ist $y \equiv g^x \bmod p$ eine Einwegpermutation.	<input checked="" type="checkbox"/> richtig <input type="checkbox"/> falsch	Ein Generator g erzeugt per Definition alle Werte.

Danksagung

- Einige Folien entstammen aus der Vorlesung „Sichere Netzerwerkkommunikation“ von Prof. Dr. A. Steffen, Hochschule Rapperswil. An dieser Stelle ein recht herzliches Danke schön an Hr. Andreas Steffen.