
# Lektion 2

## Symmetrische Kryptosysteme

#### LERNZIELE

Nach der Bearbeitung dieser Lektion werden Sie wissen, …

… welche grundlegenden symmetrischen kryptografischen Algorithmen es gibt: Substitution und Transposition.

… welche die einzige kryptografisch vollkommen sichere Chiffre ist: das One-Time-Pad.

… was die modernen Algorithmen Data Encryption System und Advanced Encryption System ausmacht.

… welche vielfältigen Einsatzmöglichkeiten es für (kryptografische) Hashfunktionen gibt.

DL-E-DLMCSEAITSC02-U02

1. Symmetrische Kryptosysteme

### Einführung

Bis zum Ende der 1970er Jahre, vor der Veröffentlichung von Difﬁe und Hellman (1976) und Rivest et al. (1978), waren alle (bekannten) kryptografischen Algorithmen symmetrisch (oder Ein-Schlüssel-Verfahren), d. h. sie verwendeten denselben Schlüssel zum Ver- und Entschlüsseln. Jeder historische Algorithmus, so ausgeklügelt er auch gewesen sein mag, sei es die Caesar-Chiffre, die Skytale oder die Enigma, war symmetrisch.

Symmetrische Algorithmen verarbeiten die Eingabe als Zeichenkette (aus Bits oder Buchstaben) durch (iterierte) Substitutionen und Transpositionen. Im Gegensatz dazu beruhen asymmetrische Algorithmen auf einem rechnerisch schwierigen Problem, z. B. der Faktorisierung einer zusammengesetzten Zahl in ihre Primfaktoren, und betrachten die Eingabe als eine natürliche Zahl. Die einzige vollkommen sichere Chiffre ist das One-Time-Pad, bei dem der Schlüssel so lang wie der Klartext ist und der Geheimtext durch buchstabenweises Hinzufügen jedes Buchstabens des Schlüssels zu dem entsprechenden (d. h. an der gleichen Stelle befindlichen) Buchstaben des Klartextes gewonnen wird. Für komplexere Nachrichten, z. B. Text-, Bild- oder Videonachrichten, ist ein großer Schlüssel jedoch unpraktisch. Heute bedeutet das, dass zum Verschlüsseln einer Festplatte eine weitere Festplatte benötigt wird, auf der sich der Schlüssel befindet. Um die kürzere Schlüssellänge zu kompensieren, erzeugen moderne Algorithmen im Idealfall so viele Verflechtungen, dass sie eine nahezu perfekte Diffusion erreichen. Mit anderen Worten: Die Änderung eines einzigen Bits der Eingabe oder des Schlüssels bewirkt die Änderung von etwa der Hälfte der Ausgabebits. Moderne Algorithmen wie der Data Encryption Standard (DES) oder der Advanced Encryption Standard (AES) sind Blockchiffren mit Substitutions- und Permutationsverfahren, d. h., sie verschlüsseln jeweils ein Datenpaket durch wiederholte Transpositionen und Substitutionen.

### Substitution und Transposition

Die beiden grundlegenden Verfahren zur Verschlüsselung sind Transposition und Substitution:

* Eine Transposition verändert die Reihenfolge (d.h. transponiert oder permutiert) der Zeichen im Text, nicht aber die Zeichen selbst.
* Bei einer Substitution wird jedes Zeichen im Text durch ein anderes (eine andere Gruppe von) Zeichen ersetzt, jedoch nicht die Reihenfolge der Zeichen geändert.

Die historisch gesehen prototypischen Algorithmen für diese beiden Verfahren sind:

* Caesars Substitutionschiffre, die jeden Buchstaben im Klartext um drei Positionen vorrückt und z. B. A als D, B als E usw. verschlüsselt, sowie
* die Permutation des Klartextes durch die Skytale (Stab von Licurgo), bei der das Pergamentband um den Stab gewickelt wird und der Text horizontal darauf geschrieben wird.

Symmetrische Kryptosysteme

Wir werden sehen, dass selbst bei vielen möglichen Schlüsseln ein Algorithmus – wie z. B. der durch eine beliebige Permutation des Alphabets gegebene, der fast 280 Schlüssel hat – leicht geknackt werden kann, wenn er Regelmäßigkeiten, wie die Häufigkeit der Buchstaben, beibehält. Substitutions- und Permutationsnetze verbinden und iterieren diese beiden komplementären prototypischen Algorithmen, um dieses Ziel zu erreichen.

###### Substitutionschiffren

Bei einer Substitutionschiffre bestimmt der Schlüssel die Substitutionen des Klartextalphabets (betrachtet als eine Menge von Einheiten von Zeichen wie einzelne Buchstaben oder Buchstabenpaare) durch das Geheimtextalphabet. Wenn zum Beispiel die Einheiten des Klartextes und des Geheimtextes beide Buchstaben des lateinischen Alphabets sind, dann vertauscht eine Substitution die Buchstaben des lateinischen Alphabets. Handelt es sich bei der Substitutionschiffre um eine monoalphabetische Chiffre (z. B. die Caesar-Chiffre), so wird auf jeden Buchstaben des Klartextes unabhängig von seiner Position dieselbe Substitution angewendet. Wenn die Substitutionschiffre polyalphabetisch ist (wie die Enigma), dann variiert die Substitution mit der Position des Buchstabens im Klartext. Zum Verschlüsseln wird jede alphabetische Einheit des Klartextes durch die substituierte alphabetische Einheit ersetzt und zum Entschlüsseln entsprechend invertiert. Die monoalphabetische Substitution gilt als unsicher, weil die Häufigkeit der Zeichen, die im Klartext und im Geheimtext zu finden sind, gleichbleibt. Im Englischen zum Beispiel genügen etwa 25 Buchstaben des Geheimtextes für eine Kryptoanalyse. Das zentrale Verfahren zur Reduktion der Erhaltung der Häufigkeit von Einzelbuchstaben im Geheimtext, die so genannte polyalphabetische Substitution, besteht in der Verwendung mehrerer Geheimalphabete.

Verschiebung um einen festen Abstand

Die einfachste Substitutionschiffre ist eine zyklische Verschiebung des Klartextalphabets, die so genannte Caesar-Chiffre. Die römischen Kaiser Caesar (100-44 v. Chr.) und Augustus (63-14 v. Chr.) bedienten sich dieser Methode (Bauer, 2016). Wir legen einen Abstand d zwischen den Buchstaben in alphabetischer Reihenfolge fest, d. h. eine Zahl zwischen 0 und 25, und verschieben dann jeden Buchstaben des (lateinischen) Alphabets um diesen Abstand d (nach vorne). Wir stellen uns dabei vor, dass Alphabet sei kreisförmig, d. h. die Buchstaben sind in einem Ring angeordnet, so dass die Verschiebung eines Buchstabens am Ende des Alphabets zu einem Buchstaben am Anfang des Alphabets führt.

Substitutionschiffre Bei dieser Chiffre wird jede alphabetische Einheit des Klartextes durch eine entsprechende alphabetische Einheit ersetzt.

Caesar-Chiffre Diese Substitutionschiffre verschiebt die

alphabetische Position jedes Klartextbuchstabens um den gleichen Abstand.



Es gibt 26 Schlüssel (einschließlich des trivialen Schlüssels d = 0). Bei der Caesar-Chiffre wird jeder Buchstabe des Alphabets um den gleichen Abstand verschoben. Zur Entschlüsselung wird jeder Buchstabe um den negativen Abstand –d verschoben, d. h. d Positionen nach hinten.

Substitution durch Permutation der Buchstaben des Alphabets

Anstatt jeden Buchstaben durch einen um den gleichen Abstand d verschobenen Buchstaben zu ersetzen, können wir jeden Buchstaben durch einen beliebigen Buchstaben ersetzen, wie im folgenden Beispiel.

A B … Y Z

…

E Z … G A

Wir mischen die Buchstaben untereinander. Auf diese Weise erhalten wir 26 · 25 · 1 = 26! > 1026

Schlüssel (was in etwa der Anzahl von Passwörtern entspricht, die mit 80 Bit möglich sind).

###### Transpositions- bzw. Permutationschiffren

Eine Transpositionschiffre verschlüsselt den Klartext durch Permutation seiner Einheiten (und entschlüsselt ihn durch die umgekehrte Permutation). Jede alphabetische Einheit bleibt gleich; die Verschlüsselung hängt lediglich von der Position der Einheiten ab.

Die Skytale ist ein Stab, mit dem die Spartaner wie folgt verschlüsselt haben: (1) man wickelt ein Band um einen schmalen Stab, (2) man beschriftet dieses Band horizontal, also entlang der längeren Seite und

(3) man wickelt das Band ab (Bauer, 2016). Die auf dem Band transponierten Buchstaben konnten nur mit einem Stab gleichen Umfangs (dem Schlüssel) auf die gleiche Weise entziffert werden, wie der Text verschlüsselt wurde: Man wickelte das Band um den Stab und las den Text darauf horizontal, entlang der längeren Seite.

Symmetrische Kryptosysteme



In diesem Fall ist der Schlüssel die Zahl, die sich aus der Anzahl der Buchstaben ergibt, die um den Umfang des Stabes herum angeordnet sind.

###### Sicherheit historischer Beispiele

Wenden wir nun die festgelegten Sicherheitskriterien auf die Substitutionschiffre an.

Caesar-Chiffre

Diese einfache Substitutionschiffre verstößt gegen alle wünschenswerten Eigenschaften. Kerckhoffs‘ Prinzip besagt zum Beispiel, dass der Algorithmus öffentlich ist, sobald die Methode bekannt ist. In Anbetracht der geringen Anzahl von 25 Schlüsseln lässt sich der Geheimtext in kurzer Zeit durch einen Brute-Force-Angriff (Durchprobieren aller Möglichkeiten) knacken.

Substitution durch willkürliche Permutation der Buchstaben des Alphabets

Eine Ersetzung durch eine beliebige Permutation der Buchstaben des Alphabets hat 26 · 25 · 1 = 26! > 1026 Schlüssel, so dass ein Brute-Force-Angriff rechnerisch nicht durchführbar ist.

Dies verstößt jedoch gegen die Ziele der Diffusion und der Konfusion. Wenn der Schlüssel (Permutation des Alphabets) den Buchstaben α gegen den Buchstaben β austauscht, dann ergibt sich

* eine geringe Konfusion, da die Substitution von β im Schlüssel nur die Substitution jedes Buchstabens β im Geheimtext impliziert,
* eine geringe Diffusion, da die Substitution eines Buchstabens α im Klartext nur die Ersetzung des entsprechenden Buchstabens β im Geheimtext impliziert.

Brute-Force-Angriff Diese erschöpfende Schlüsselsuche überprüft jeden möglichen Schlüssel.

###### Produktchiffre

Produktchiffre Eine Zusammensetzung von Chiffren, bei der die Ausgabe einer Chiffre als Eingabe für

die nächste dient.

Eine Produktchiffre setzt Chiffren zusammen, d. h. bei einem zweifachen Produkt ist die Ausgabe der einen Chiffre die Eingabe der anderen. Der Geheimtext der Produktchiffre ist der Geheimtext der endgültigen Chiffre. Kombinieren wir Transpositionen nur mit Transpositionen oder Substitutionen nur mit Substitutionen, ist die erhaltene Chiffre wieder eine Transposition oder Substitution und kaum sicherer. Eine Mischung aus beiden, eine Transposition mit Substitutionen, kann die Chiffre jedoch tatsächlich sicherer machen.

Eine Fraktionierungschiffre ist eine Produktchiffre, bei der jedes Zeichen im Klartext durch eine Gruppe von Zeichen (in der Regel Paare) ersetzt wird und der erhaltene Geheimtext anschließend transponiert wird. Die bekannteste Fraktionierungschiffre war die ADFGVX-Chiffre, die von den deutschen Streitkräften während des Ersten Weltkriegs verwendet wurde (Bauer, 2016). Bei dieser Chiffre wurden die 26 Buchstaben des lateinischen Alphabets und zehn Ziffern in einer 6 × 6-Tabelle angeordnet und durch die Buchstabenpaare A, D, F, G, V und X ersetzt, die die Zeile und Spalte des Buchstabens oder der Ziffer angeben. Der resultierende Text wurde wie üblich von links nach rechts in die Zeilen einer Tabelle geschrieben und dann jede Spalte in der durch ein Schlüsselwort angegebenen Reihenfolge gelesen.

Wir werden uns nun ansehen, wie moderne Chiffren diese Idee einer Produktchiffre weiterentwickeln, um eine hohe Diffusion zu erreichen.

### Blockchiffren

Bei den klassischen Chiffren wurden in der Regel einzelne Buchstaben und manchmal auch Buchstabenpaare ersetzt. Systeme, die mit Dreiergruppen (Trigrammen) oder größeren Buchstabengruppen arbeiteten, wurden als zu umständlich angesehen und fanden keine breite Anwendung.

Stattdessen ist es sicherer, einen ganzen Block (von Buchstaben anstelle eines einzelnen Buchstabens) entsprechend dem Schlüssel zu substituieren. Das Alphabet dieses Austauschverfahrens wäre allerdings riesengroß, so dass dieses Ideal praktisch unerreichbar ist, insbesondere bei einer so beschränkten Hardware wie einer Chipkarte mit einem 8-Bit-Prozessor. Für einen Block von 16 Byte hätte diese Substitutionstabelle zum Beispiel die horrende Größe von 2128 • 16 Byte. In der modernen Kryptografie mit einem Schlüssel beträgt die Informationseinheit jedoch in der Regel 128 Bit, d. h. etwa 27 Buchstaben, während in der Kryptografie mit zwei Schlüsseln, die auf dem RSA-Algorithmus basiert, üblicherweise Einheiten von 2048 Bit, d. h. etwa 620 Buchstaben, verwendet werden.

So ersetzt AES zum Beispiel nur jedes Byte, jeden Eintrag in einem Block, eine Ersetzungstabelle von 28 = 256 Einträgen von 1 Byte (und transponiert anschließend die Einträge). Wir werden sehen, dass sich diese Verfahren so gut ergänzen, dass sie praktisch so sicher sind wie eine Substitution des gesamten Blocks.

Symmetrische Kryptosysteme

###### Block- und Stromchiffren

Bei einer Blockchiffre wird der Klartext in Blöcke gleicher Größe unterteilt und jeder Block mit einem gemeinsamen Schlüssel verschlüsselt; ein Block könnte zwar aus einem einzigen Zeichen bestehen, ist aber normalerweise größer. Bei DES beträgt die Blockgröße beispielsweise 64 Bit, bei AES 128 Bit.

Eine Stromchiffre unterteilt den Klartext in Einheiten, die normalerweise aus einem einzigen Zeichen bestehen, und verschlüsselt dann die i-te Einheit des Klartextes mit der i-ten Einheit eines Schlüsselstroms. Beispiele sind das One-Time-Pad, Rotor-Chiffriermaschinen (wie die Enigma) und Triple DES (bei dem die Ausgabe einer Verschlüsselung die Eingabe der nächsten Verschlüsselung ist). Bei einer Stromchiffre muss derselbe Abschnitt des Schlüsselstroms, der zur Verschlüsselung verwendet wurde, auch zur Entschlüsselung verwendet werden. Daher müssen die Schlüsselströme von Absender und Empfänger zu Beginn und danach ständig synchronisiert werden.

###### Feistelchiffren

Eine Feistelchiffre oder ein Substitutions- und Permutationsnetzwerk (SPN) gliedert den Text (Bytefolge) in n-Byte-Blöcke (z. B. n = 16 für AES) und verschlüsselt jeden Block durch Iteration (z. B. zehnmal bei AES und fünfmal in unserem prototypischen Modell) der folgenden drei Schritte:

* + 1. Addieren (XOR) des Schlüssels,
		2. Substituieren des Alphabets (das in Teilblöcken des Blocks arbeitet, z. B. auf jedem Byte), und
		3. Permutieren aller Teilblöcke eines Blocks.

Nach der Addition (XOR) des Schlüssels, der Ersetzung des Alphabets, z. B. im AES-Algorithmus (jedes Byte, hexadezimales Buchstabenpaar, durch ein anderes), und der Permutation des Textes (aus dem aktuellen Schritt, „Zustand“ genannt), z. B. im AES, werden die Einträge in jeder Zeile (und den Spalten) permutiert.

Diese beiden einfachen Verfahren, die Substitution des Alphabets und die Permutation des Textes, ergänzen sich gut. Sie führen nach wenigen Iterationen zu starker Konfusion und Diffusion. In der ersten und letzten Runde werden die Schritte vor und nach dem Hinzufügen des Schlüssels weggelassen, da sie die kryptografische Sicherheit nicht erhöhen. Da der Algorithmus öffentlich ist (Kerckhoffs' Prinzip), ist jeder Angreifer in der Lage, alle Schritte rückgängig zu machen, die keine Kenntnis des Schlüssels erfordern. Obwohl sich eine Feistelchiffre also von klassischen Kryptosystemen zu unterscheiden scheint, handelt es sich tatsächlich um eine Produktchiffre aus Transpositionen und Substitutionen.

Stromchiffre und Blockchiffre im Vergleich

Eine Stromchiffre arbeitet mit einzelnen Zeichen (z. B. einzelnen Bytes), während eine Blockchiffre mit Gruppen von Zeichen (z. B. jeweils 16 Bytes groß) arbeitet.

Substitutions- und Permutationsnetzwerk Bei dieser Chiffre wird jeder Block nach dem Hinzufügen eines Schlüssels iterativ substituiert und permutiert.

### Data Encryption Standard (DES)

Data Encryption

Standard Diese Blockchiffre hat eine Schlüssellänge von 56 Bit. Sie wurde von Horst Feis-

tel von IBM entwickelt.

Der Data Encryption Standard (DES) wurde 1977 zu einem öffentlichen Standard, nachdem er einen vom US National Bureau of Standards (heute National Institute of Standards and Technology, NIST) ausgeschriebenen öffentlichen Wettbewerb gewonnen hatte. Der DES umfasst 16 Iterationen, bei denen Substitution und Permutation (Transposition) durchgeführt werden. Die Blockgröße beträgt 64 Bit, und der Schlüssel besteht aus 64 Bit. Allerdings können nur 56 Bit vom Benutzer ausgewählt werden und bilden den Schlüssel: die restlichen acht sind redundante Paritätsprüfbits.

Wie der Name seines Erfinders vermuten lässt, handelt es sich um eine Feistelchiffre oder ein Substitutions- und Permutationsnetzwerk, ähnlich dem oben beschriebenen Prototyp. Der Algorithmus gruppiert den Text (eine Bytefolge) in 32-Bit-Blöcke mit Teilblöcken von vier Bit und verschlüsselt jeden Block in 16 Iterationen der folgenden drei Schritte, die als Feistelfunktion oder F-Funktion bezeichnet werden, in der angegebenen Reihenfolge

1. Addieren (XOR) des Schlüssels,
2. Substituieren jedes 4-Bit-Teilblocks des Blocks durch die S-Box (in hexadezimaler Schreibweise), und
3. Permutieren aller Teilblöcke.



Symmetrische Kryptosysteme

In jeder Runde i wird die Ausgabe der vorangegangenen Runde in die 32 am weitesten links liegenden Bits L(i) und die 32 am weitesten rechts liegenden Bits R(i) aufgeteilt. R(i) wird zu L(i + 1), während R(i + 1) die Ausgabe einer komplexen Funktion, L(i) + f(R(i)*,* K(i + 1)) ist, deren Eingabe der i + 1-te Block der Schlüsselbits, K(i + 1) und die gesamte vorangegangene Zwischenchiffre ist. Dieser Vorgang wird 16 mal wiederholt.

Entscheidend für die Sicherheit von DES ist die nichtlineare S-Box der vom NIST spezifizierten F-Funktion f. Sie ist sowohl nichtlinear, d. h. f(A) + f(B) ≠ f(A + B), als auch eine Maximierung der Konfusion und Diffusion im Sinne der Shannon-Kriterien für eine sichere Chiffre.

###### Schlüsselgröße und die Geburt von 3DES

Die Sicherheit des DES ist wie bei jedem Algorithmus nicht größer als der Aufwand für die Suche nach 256 Schlüsseln. Was 1977 noch als undurchführbare Aufgabe für einen Computer galt, schaffte eine spezielle DES-Suchmaschine zwei Jahrzehnte später in drei Tagen. Ein Workaround, Triple DES oder 3DES genannt, wurde entwickelt, das dem DES mit einem 112-Bit-Schlüssel die doppelte Schlüsselgröße gegenüber den DES-Schlüsseln verleiht, indem einfach zwei normale DES-Schlüssel verwendet werden.

Die Verschlüsselung sieht dann so aus: V(1) (E(2) (V(1); (1) verschlüsseln mit dem ersten Schlüssel, (2) entschlüsseln mit dem zweiten Schlüssel und (3) verschlüsseln mit dem ersten Schlüssel.

Die Verschlüsselung sieht dann so aus: V(1) (E(2) (V(1) – (1) verschlüsseln mit dem ersten Schlüssel, (2) entschlüsseln mit dem zweiten Schlüssel und (3) verschlüsseln mit dem ersten Schlüssel.

Wenn die beiden Schlüssel übereinstimmen, wird dieses Kryptosystem zu einem gewöhnlichen DES mit nur einem Schlüssel. Somit ist Triple-DES rückwärtskompatibel mit Anlagen, die für (Single-)DES implementiert wurden. DES war der erste kryptografische Algorithmus, der Kerckhoffs‘ Prinzip erfüllte.

### Advanced Encryption Standard (AES)

Im Januar 1997 kündigte das NIST einen öffentlichen Wettbewerb für einen sogenannten Advanced Encryption Standard (AES) an, der den früheren symmetrischen Verschlüsselungsstandard, den Data Encryption Standard (DES), ersetzen sollte. Nachdem der 56-Bit-Schlüssel des DES mit einer Brute-Force-Attacke geknackt wurde, verlangten die NIST-Spezifikationen für den AES eine größere Schlüssellänge. Der AES-Algorithmus war der Gewinner dieses Wettbewerbs (Bauer, 2016).

###### Beurteilung der Sicherheit

Die Erfinder von AES konnten nachweisen, dass sich diese beiden Verfahren so gut ergänzen, dass sie nach mehreren Iterationen das fehlende Ersetzen des gesamten Blocks durch einen anderen nahezu ausgleichen (Daemen & Rijmen, 1999; 2002).

3DES

Dieser Begriff bezeichnet die dreifache Anwendung von DES mit der doppelten Schlüsselgröße des DES-Algorithmus.

Advanced Encryption Standard

Dieses Substitutions- und Permutationsnetzwerk mit einer (variablen) Schlüssellänge von in der Regel 128 Bit wurde im Jahr 2000 als Nachfolger von DES zu einem kryptografischen Standard.

Wie es schon bei DES der Fall war, so ist auch AES Jahrzehnte nach seiner Einführung noch immer gegen alle Angriffe der Kryptoanalyse gewappnet, aber es ist nicht zu erwarten, dass es den Entwicklungen in der Computertechnik nachgibt, wie es bei DES der Fall war, da die Schlüsselgröße anpassbar ist.

###### Anwendungsfälle

Rijndael zeichnete sich durch seine Einfachheit und seine besonders rechenökonomische Implementierung aus. Es war nicht nur ein sicheres Verschlüsselungsverfahren, sondern dank seines eleganten und einfachen Designs auch klein genug, um auf Chipkarten implementiert zu werden (weniger als 10.000 Byte Code). Bis heute gilt er als der sicherste Algorithmus. Es besteht kein Bedarf an einem weiteren symmetrischen kryptografischen Standardalgorithmus. Er wird folgerichtig überall eingesetzt. Zur Verschlüsselung eines drahtlosen Netzwerks wird beispielsweise ein Schlüssel verwendet, so dass die Verschlüsselung symmetrisch ist. Die sicherste und daher empfehlenswerteste Option ist die Verschlüsselung mit AES.

###### Algorithmus

AES ist eine Blockchiffre, d. h., sie arbeitet mit Byte-Blöcken, in der Regel mit einer

Größe von 16 Byte (AES-128).

|  |  |
| --- | --- |
| SubBytes | Ersetzen jedes Bytes des Blocks gemäß einer Substitutionstabelle (S-Box) mit 28 = 256 Einträgen von 1 Byte. |
| ShiftRows | Zyklisches Verschieben der Einträge der letzten drei Zeilen |
| MixColumns | Hinzufügen der Spalten dazwischen. |

###### Blockweise Verschlüsselung

Der AES-Algorithmus gruppiert den Klartext (und die Schlüssel) in 4×B Byte-Rechtecke, wobei B die Anzahl der Spalten des Blocks ist: 4, 6 oder 8. Üblicherweise, und im Folgenden gilt das auch für uns, ist B = 4, d. h. das Rechteck ist ein Quadrat. Auf hexadezimaler Basis hat ein solches Quadrat zum Beispiel die Form

A1 13 B1 4A A3 AF 04 1E 3D 13 C1 55

B1 92 83 72

Symmetrische Kryptosysteme

###### Der Rijndael-Binärkörper

Um den AES zu verstehen, sind einige mathematische Definitionen von Gruppen und Feldern sowie Grundkenntnisse im Rechnen mit Polynomen nötig. Die nachfolgenden mathematischen Darstellungen erläutern die AES-Funktionen SubBytes, ShiftRows und MixColumns im Detail.

Gruppen und Felder

Eine Gruppe ist eine Menge G mit

* einer Operation • : G × G → G, die das Assoziativgesetz erfüllt,
* einem neutralen Element e, sodass gilt x • e = x = e • x für alle x in G), und
* einem inversen Element y für jedes Element x, sodass gilt x • y = e = y • x.

Allgemein wird die Operation mit •, das neutrale Element mit 1und das inverse Element von x

in G mit x-1 bezeichnet.

Beispiel

Die Menge der rationalen Zahlen ungleich Null \*\* mit der Multiplikationsoperation • ist eine Gruppe. Ist die Gruppe G kommutativ (die Operation erfüllt das Kommutativgesetz), so wird die Operation mit +, das neutrale Element mit Null und das inverse Element von x in G mit –x bezeichnet.

Beispiel

Die Menge der rationalen Zahlen \* mit der Additionsoperation + ist eine kommutative Gruppe. Ein „Feld“ ist eine Menge F mit einer Additions- und Multiplikationsoperation + und • so dass gilt:

* die Menge F mit + ist eine kommutative Gruppe,
* die Menge F\* = F - { 0 } mit • ist eine kommutative Gruppe, und
* das Distributivgesetz ist erfüllt.

Beispiel

Die Menge der rationalen Zahlen \* mit Addition + und Multiplikation • ist ein Feld.

Bytes als Polynome mit binären Koefﬁzienten vom Grad 7

Ein Byte, eine Folge b7 ... b1, b0 von acht Bits aus {0, 1} wird als Polynom mit binären Koeffizienten betrachtet durch

b7 . . . b1b0 b7X7 + ! + b1X+ b0

So entspricht beispielsweise die Hexadezimalzahl 0 x 57 oder die Binärzahl 0101 0111 dem Polynom x6 + x4 + x2 + x + 1.

Alle Additionen und Multiplikationen im AES finden im binären Feld IF28 mit 28 = 256 Elementen statt, einer Zahlenmenge mit Addition und Multiplikation, die dem Assoziativ-, Kommutativ- und Distributivgesetz (z. B. \*) entspricht, und wie folgt definiert ist. Sei

IF2 = 0,1

das „Feld zweier Elemente“ mit der Addition 1 + 0 = 0 + 1 = 1 und 0 + 0 = 1 + 1 = 0 (entspricht der ⊕-Addition durch XOR), und der (natürlichen) Multiplikation 1 • 0 = 0

• 1 = 0 • 0 = 0 und 1 • 1 = 1.

Sei

IF2 X = +/2+ = die Polynome über IF2,

d. h. die finiten Summen a0 + a1 X + a2 X2 + ·+ an Xn bis a0, a1, …, an bis IF2 und sei

IF28: = IF2 X / X8 + X4 + X3 + X + 1 .

Das heißt, das Ergebnis der beiden Operationen + und • in IF2[X] ist der Rest der Division durch X8 + X4 + X3 + X + 1.

Addition

Die + -Addition von zwei Polynomen ist die Addition in IF2 Koefﬁzient zu Koeffizient. Das heißt, dass die + -Addition als Bytes durch die XOR-Addition gegeben ist.

Multiplikation

Die Multiplikation • ist gegeben durch die natürliche Multiplikation, gefolgt von der Division mit Rest durch das Polynom

m x = x8 + x4 + x3 + x + 1 .

In hexadezimaler Notation ist beispielsweise 57• 83 = C1 , da

x6 + x4 + x2 + x + 1 x7 + x + 1 = x13 + x11 + x9 + x8 + x6 + x5 + x4 + x3 + 1

und

x13 + x11 + x9 + x8 + x6 + x5 + x4 + x3 + 1 = x5 + x3 + 1 x8 + x4 + x3 + x + 1 + x7

+ x6 + 1

Die Multiplikation mit dem Polynom 1 verändert nichts: Es ist das neutrale Element. Für ein beliebiges Polynom b(x) berechnet Euklids erweiterter Algorithmus die Polynome a(x) und c(x) , so dass

Symmetrische Kryptosysteme

b x a x + m x c x = 1 .

Das heißt, bei der Division mit dem verbleibenden a(x) b(x) für m(x) Rest 1. Dies bedeutet, dass

a der Kehrwert in IF 8 ist,

2

b−1 x = a x in IF28 .

Wenn wir ein Byte b in IF 8 invertieren, meinen wir Byte a = b–1.

2

###### Runden

Der AES verschlüsselt jeden Block iterativ, in Runden. R sei die Anzahl der Runden, die von B abhängt. Es gibt R = 10 Runden für B = 4 Spalten, R = 12 Runden für B

= 6 Spalten und R = 14 Runden B = 8 Spalten.

Für uns gilt R = 10. In diesen Runden werden Schlüssel erzeugt und der Klartext wird durch folgende Operationen ersetzt und transponiert:

|  |  |  |
| --- | --- | --- |
| 1. | Runde r = 0 | AddRoundKey zum Hinzufügen (durch XOR) des Schlüssels zum Klartextblock |
| 2. | Runde r = 1,…, R – 1 | Zum Verschlüsseln sind folgende Funktionen anzuwenden1. SubBytes zum Ersetzen jedes Bytes (= Folge von acht Bits) durch eine besser verteilte Folge von Bits
2. ShiftRows zum Permutieren der Einträge in jeder Zeile des Quadrats
3. MixColumn zum Hinzufügen der Spaltenboxen im Quadrat zwischen den Spalten
4. AddRoundKey zur Erzeugung eines Schlüssels aus dem Schlüssel der vorherigen Runde und dessen Hinzufügen (durch XOR) zum Quadrat
 |
| 3. | Runde r = R | Zum Verschlüsseln sind folgende Funktionen anzuwenden1. SubBytes
2. ShiftRows
3. AddRoundKey
 |

Im Vergleich zu früheren Runden entfällt die Funktion MixColumn. Es zeigt sich, dass MixColumn und AddRoundKey nach einer geringfügigen Änderung von AddRoundkey die Reihenfolge ändern können, ohne das Ergebnis der beiden Operationen zu verändern. In dieser äquivalenten Reihenfolge erhöht die Operation MixColumn nicht die kryptografische Sicherheit, da die letzte Operation ohne Kenntnis des Schlüssels invertierbar ist. Somit kann die letzte MixColumn-Operation entfallen.

Das CrypTool 1 bietet in seinem Menü „Individuelle Verfahren > Visualisierung von Algorithmen > AES“ einen Eintrag „Animation“ der Runden und einen Eintrag „Inspektor“ zum Experimentieren mit den Werten von Klartext und Schlüssel. Die verwendeten Funktionen werden im Folgenden näher beschrieben.

SubBytes

SubBytes ersetzt jedes Byte des Blocks durch ein anderes Byte, das in der Substitutionstabelle der S-Box angegeben ist.

Berechnung des Werts des Eintrags, durch den die S-Box die einzelnen Bytes ersetzt:

* + 1. Berechnung des Kehrwerts B in IF 8,

2

* + 1. Berechnung von ai = b1 + bi+4 + bi+5 + bi+6 + bi+7 + ci

mit i = 0, 1, …, 7 als Index jedes Bits eines Bytes, und

* + B = b7 b6 b5 b4 b3 b2 b1 b0 ist das Eingangsbyte,
	+ A = a7 a6 a5 a4 a3 a2 a1 a0 ist das Ausgangsbyte der Operation und
	+ c ist das konstante Byte 01100011.

In Matrixform:

a0 b0

1 0 0 0 1 1 1 1 1

a1 1 1 0 0 0 1 1 1 b1 1

a2 1 1 1 0 0 0 1 1 b2 0

a3 1 1 1 1 0 0 0 1 b3 0

= +

a4 1 1 1 1 1 0 0 0 b4 0

a5 0 1 1 1 1 1 0 0 b5 1

a6 0 0 1 1 1 1 1 0 b6 1

0 0 0 1 1 1 1 1 0

b

a7 7

ShiftRows

ShiftRows verschiebt die Zeilennummer l des Quadrats l Positionen nach links (wobei die Zeilennummern von Null an gezählt werden, d. h. l durchläuft 0, 1, 2 und 3, und die Verschiebung erfolgt zyklisch). *Anmerkung:* Die erste Zeile wird nicht verschoben.

MixColumn

MixColumn multipliziert jede Spalte des Blocks mit einer festen Matrix:

Symmetrische Kryptosysteme

* wenn Bj (mit den Koeffizienten b0,j, b1,j, b2,j und b3,j) der Spalte j des Eingangsblocks entspricht, und
* wenn Aj (mit den Koeffizienten a0,j, a1,j, a2,j und a3,j) der Spalte j des Ausgangsblocks der Operation entspricht,

dann

a0,j

a1, j

=

a2, j

a3, j

02 03 01 01

01 02 03 01

01 01 02 03

03 01 01 02

b0,j

b1, j

.

b2, j

b3, j

Das Byte a0,j wird zum Beispiel wie folgt berechnet:

a0,j = 2 · b0, j + 3 · b1, j + b2, j + b3, j .

AddRoundKey

AddRoundKey fügt durch die XOR-Operation den Schlüssel W(r) der aktuellen Runde r zum B-Quadrat des Geheimtextes hinzu, das heißt

B ⊕ W r .

Der Schlüssel wird spaltenweise erzeugt. Wir bezeichnen sie mit W(r)0, W(r)1, W(r)2 und

W(r)3, d. h.

W r = W r 0 ∣ W r 1 ∣ W r 2 ∣ W r 3 .

Da der Schlüssel 16 Byte hat, hat jede Spalte vier Byte. Das heißt, die letzte Spalte des vorherigen Rundenschlüssels plus das Ergebnis von ScheduleCore wird auf die erste Spalte des vorherigen Rundenschlüssels (den wir mit T bezeichnen) angewendet. Hier ist ScheduleCore die Zusammensetzung von Transformationen:

* SubWord ersetzt jedes der 4 Byte von T entsprechend der S-Box von SubBytes.
* RotWord verschiebt T ein Byte nach links (zirkulär, d.h. das letzte Byte wird das erste).
* Rcon(r) addiert (durch XOR) zu T den konstanten Wert, in hexadezimaler Schreibweise [(02)r–1 00 00 00] (wobei die Potenz, d. h. das iterierte Produkt, im Rijndael-Feld IF 8 berechnet wird). Das einzige Byte, das sich ändert, ist das Erste, indem entweder der Wert 2r–1 (für r ≤ 8) oder der Wert 2r–1 in F 8 für r = 9, 10 addiert wird.

2

2

1. Der Schlüssel der ersten Runde W(0) ist durch den Anfangsschlüssel W gegeben.
2. Für r = 1, ..., R (wobei R die Gesamtzahl der Runden ist, für uns R = 10), werden die vier Spalten W(r)0, W(r)1, W(r)2 und W(r)3 des neuen Schlüssels wie folgt aus den Spalten des alten Schlüssels W(r-1) erzeugt:
	1. Die erste Spalte W(r)0 ist gegeben durch

W r 0 = W r − 1 0 ⊕ ScheduleCore W r − 1 3 ;

* 1. Die nächsten Spalten W(r)1, W(r)2 und W(r)3 sind für i = 1, 2 und 3 gegeben durch

W r i = W r i − 1 ⊕ W r − 1 i;

* 1. Das heißt, die vorherige Spalte des aktuellen Rundenschlüssels plus die aktuelle Spalte des vorherigen Rundenschlüssels.

###### Diffusion

Wir stellen fest, dass die einzige Transformation, die nicht afﬁn ist, die Inversion im Feld IF 8 in der Operation SubBytes ist. Bei der Operation SubBytes werden angewandt: die Inversion in IF 8, eine lineare Applikation, und die Übersetzung durch einen konstanten Vektor (a) ShiftRows ist eine lineare Permutation, (b) MixColumn ist eine lineare Addition, und (c) AddRoundKey ist die Translation durch den Rundenschlüssel.

2

2

Im Hinblick auf die Ziele der idealen „Diffusion“ und „Konfusion“ ist zu beachten, dass in jedem Schritt etwa die Hälfte der Bits (in SubBytes) oder Bytes (in MixColumn und ShiftRows) ersetzt und transponiert werden. Die Komplementarität der einfachen Operationen für hohe Sicherheit, d.h. ihre Erzeugung von hoher Konfusion und Diffusion nach einigen Iterationen, wird durch die Substitution des Alphabets und die Permutation von Text deutlich. Bei Letzteren ist dies insbesondere bei der Permutation zwischen den Einträgen jeder Zeile und der Permutation zwischen den Spalten zu beobachten. Das Cryptool von Forma Estudio (2007) ist ein Beispiel für diese Eigenschaft.

Wir sehen, wie sich dieser kleine anfängliche Unterschied ausbreitet und nach nur vier Runden zu sehr unterschiedlichen Ergebnissen führt! Dies macht die Immunität von AES gegen differenzielle Kryptoanalyse plausibel.

Für den Fall, dass alle Schlüssel- und Klartexteinträge gleich 00 sind, verstehen wir auch die Auswirkung des Addierens der Konstante Rcon(r) zu dem Schlüssel in jeder Runde. Daher kommt die gesamte Konfusion!

### Kryptografische Hashfunktionen

Eine Hashfunktion wandelt beliebige Daten (z. B. eine Datei), d. h. eine Zeichenkette mit variabler Länge, in eine Zeichenkette mit fester Länge (in der Regel 16 oder 32 Byte) um. Sie wandelt eine große Datenmenge in eine kleine Informationsmenge um.

Symmetrische Kryptosysteme

###### Hash als ID

Eine Hashfunktion sollte auf alle möglichen Sequenzen mit fester Länge anwendbar sein und einer gleichförmigen Zufallsvariablen ähneln, d. h. die Wahrscheinlichkeit für jeden Funktionswert ist gleich groß.

Wenn die Ausgabe beispielsweise aus 256 Bits besteht, dann sollte im Idealfall jeder Wert die gleiche Wahrscheinlichkeit 2–256 haben. Das heißt, die Ausgabe identiﬁziert die Eingabe praktisch eindeutig (mit einer Kollisionswahrscheinlichkeit von idealerweise 2–256). So könnte man sich beispielsweise einen Datenhash einer Datei als ihren Personalausweis vorstellen. Ein Hash identiﬁziert viele Daten mit wenigen Informationen.

Da die Länge der Hash-Sequenz begrenzt ist (selten ≥ 512 Bits) und die Länge der Eingabesequenz unbegrenzt ist, kommt es zu Kollisionen, d. h. zu gleichen Hashes aus verschiedenen Dateien. Der Algorithmus minimiert jedoch die Wahrscheinlichkeit von Kollisionen, indem er die Werte so gleichmäßig wie möglich verteilt – oder intuitiv gesagt: so zufällig wie möglich macht. Genauer gesagt ist jede mögliche Sequenz mit fester Länge ein Wert, und die Wahrscheinlichkeit für jeden der Werte ist die gleiche.

###### Kryptografische Hashfunktionen

Für eine kryptografische oder Einweg-Hashfunktion muss der Algorithmus resistent sein

gegenüber

1. der Erstellung eines Urbildes, d.h. die Funktion ist unidirektional. Wenn eine Ausgabe gegeben ist, ist der schnellste Weg, eine Eingabe mit dieser Ausgabe zu finden, ein Brute-Force-Verfahren, d. h. die Prüfung aller möglichen Eingaben.
2. der Erstellung einer zweiten umgekehrten Abbildung, d. h. bei einer Eingabe ist der schnellste Weg, eine andere Eingabe mit demselben Ergebnis zu finden, ein Brute-Force-Verfahren.
3. der Erstellung von Kollisionen, d. h. der schnellste Weg, zwei (beliebige) Einträge mit demselben Ergebnis zu finden, ist ein Brute-Force-Verfahren.

Nach Kerckhoffs' Prinzip muss der Algorithmus zudem öffentlich sein. In der Praxis ist die wichtigste Eigenschaft die Resistenz gegen den Angriff über die Erstellung einer zweiten, umgekehrten Abbildung, und die am wenigsten wichtige ist die gegen Kollisionsangriffe. Es gibt verschiedene Algorithmen (z. B. MD4, MD5 und SHA-1), die nicht gegen Kollisionsangriffe resistent sind, aber dennoch verwendet werden. Der CRC-Algorithmus ist zum Beispiel eine (nicht kryptografische) Hashfunktion. Andere gängige kryptografische Hashfunktionen sind SHA-256 und SHA-3.

Die Ausgabe der Hashfunktion SHA-256 von IUBH lautet zum Beispiel 0a65cfb2c33b58749182d3eb965155336a52219793ff633016cb3942b2809a3f. Die Ausgabe einer Hashfunktion, aber auch Message Digest oder digitaler Fingerabdruck, je nach Eingabe, werden zum Beispiel für die Integrität und Authentifizierung von Nachrichten verwendet.

###### Gängige kryptografische Hashfunktionen und Anwendungen

Die am häufigsten verwendeten (kryptografischen) Hash-Algorithmen sind auch heute noch 16 Byte groß, wie MD4 oder MD5, oder SHA-1, das 20 Byte verwendet. Obwohl keine von ihnen Kollisionsangriffen standhält, sind sie nach wie vor beliebt. Die Einzelheiten ihrer Implementierung sind im Request for Comments (RFC) beschrieben. Ein RFC spezifiziert öffentlich die Einzelheiten eines geplanten Internetstandards oder einer neuen Version eines bestehenden Standards und wird in der Regel von Forschern an Universitäten und in Unternehmen verfasst, um Feedback einzuholen. Ein RFC wird online und in offiziellen Sitzungen der von der Internet Engineering Task Force (IETF) beauftragten Arbeitsgruppe diskutiert. So wurden beispielsweise Netzwerkstandards wie IP und Ethernet in RFCs dokumentiert.

Der Message-Digest-Algorithmus (MD4) wurde 1991 von Ron Rivest entwickelt, einem der drei Erfinder des RSA-Algorithmus. Er wird in RFC 1320 beschrieben, ist schnell, aber anfällig für Urbild-Angriffe.

MD5 wurde von RSA Data Security entwickelt. Das in RFC 132 beschriebene Verfahren ist anfällig für Kollisionen, aber nicht für die Erstellung eines zweiten Urbildes. Es wird häufig für Integritätsprüfungen, von Software mit Peer-to-Peer-Protokoll (P2P) und als Passwortspeicher verwendet.

Der Secure-Hash-Algorithmus (SHA-1) wurde vom NIST entwickelt. Er ist anfällig für Kollisionen, aber nicht für die Erstellung eines zweiten Urbildes. Die neueren Versionen, wie SHA-256 und SHA-3, sind gegenüber den alten SHA-1 oder MD4 und MD5 vorzuziehen.

Digitale Signaturen

Sind die Rollen des öffentlichen und des privaten Schlüssels vertauscht, so handelt es sich bei der Verschlüsselung um eine digitale Signatur. Während die verschlüsselte Nachricht nicht mehr geheim ist, kann jeder Besitzer des öffentlichen Schlüssels überprüfen, ob der private Schlüssel die ursprüngliche Nachricht verschlüsselt hat. Theoretisch würde jedoch das Signieren einer Datei (unter Verwendung des RSA-Algorithmus), die mit dem RSA-Algorithmus verschlüsselt wurde, diese entschlüsseln. In der Praxis wird ein kryptografischer Hash in der Regel mit dem privaten Schlüssel verschlüsselt, so wie eine Signatur ausreicht, um die signierte Datei eindeutig zu identifizieren (ihr Inhalt ist jedoch oft geheim, z. B. beim Signieren eines geheimen Schlüssels).

(H)MAC

Zur Nachrichtenauthentifizierung wird ein Message Authentication Code (MAC) verwendet. Wenn dieser MAC durch eine Hashfunktion gesichert ist, wird er als Keyed-Hash Message Authentication Code (HMAC) bezeichnet. Das American National Standards Institute (ANSI) hat einen Message Digest, einen Hash einer Nachricht, mit einem Passwort als Hash-Schlüssel unter Verwendung des DES-Algorithmus als neuen Stand der Kryptografie entwickelt. Ein HMAC wird also aus einer Nachricht und einem geheimen Schlüssel erzeugt, der in einer kryptografischen Hashfunktion wie SHA-1 oder MD5 verwendet wird. Der MAC-Wert schützt die Integrität und Authentizität einer Nachricht, indem er es den vorgesehenen Empfängern ermöglicht, Änderungen am Inhalt der Nachricht zu erkennen. Der HMAC liefert dem Benutzer außerdem sowohl eine Integritäts- als auch eine Authentizitätsprüfung (Krawczyk et al., 1997).

Datenspeicherung und -integrität

Hashfunktionen (sowohl kryptografische als auch nicht kryptografische) werden für die schnelle Datenabfrage, d. h. mit konstanter Dauer, in einer Hashtabelle und in einem Hash-Baum (auch Merkle-Baum) verwendet. Sie werden außerdem dazu eingesetzt, die Integrität einer Datei angesichts versehentlicher Änderungen zu gewährleisten, d. h. um Unterschiede zwischen der Datei und einer Referenzversion zu erkennen.

Passwörter

Kryptografische Hashfunktionen werden verwendet, um Werte gleichmäßig zu verteilen (Schlüsselstreckung, engl. Key Stretching), was sie intuitiv weniger vorhersehbar macht, bekannt als Key Derivation Function (KDF), und um Passwörter zu erzeugen und zu speichern, als PBKDF (Password Based Key Derivation Function). Kryptografische Hashfunktionen werden auch verwendet, um die Integrität einer Datei vor Manipulationen zu schützen, d. h. um Unterschiede zwischen der Datei und einer Referenzversion (in der Regel die Version, bevor die Datei übertragen wird) zu erkennen. Insbesondere, um die Echtheit einer Datei zu gewährleisten und Unterschiede zwischen der Datei und einer Version zu erkennen, die unter der Kontrolle einer bestimmten Person stand.

Beachten Sie den Unterschied zwischen Authentizität und Authentifizierung. Erstere garantiert die Gleichheit der von einer Person empfangenen und gesendeten Daten, z. B. bei der digitalen Signatur, Letztere die Identität dieser Person, z. B. bei einem sicheren Website-Zugang.

Die oben aufgeführten kryptografischen Hash-Algorithmen verteilen die Werte gleichmäßig, sind aber schnell und eignen sich daher nicht für die Erstellung von Passwörtern, da sie anfällig für Brute-Force-Angriffe sind. Um diese zu verhindern, sind die PBKDF-Algorithmen (a) absichtlich langsam, wie z. B. bcrypt, (b) erfordern absichtlich viel Speicher für die Berechnung, wie z. B. der scrypt-Algorithmus, oder (c) werden nur einmal für jeden Eintrag verwendet, garantiert durch ein „Salt“, ein zusätzliches, einzigartiges, normalerweise zufälliges Argument. Ohne Salt ist der Algorithmus anfällig für Angriffe durch „Rainbow Tables“, Tabellen mit den Hashes der gängigsten Passwörter.

Symmetrische Kryptosysteme

**Zusammenfassung**

Beim Chiffrieren gibt es zwei grundlegende Verfahren: Transposition und Substitution. Transpositionen ordnen den Text neu an, ohne die Zeichen zu verändern; Substitutionen ersetzen die Zeichen des Textes, ohne die Reihenfolge zu verändern. Feistelchiffren sind unsicher, weil sie die statistischen Daten des Klartextes erhalten. Bei modernen Chiffren werden im Gegensatz dazu Substitutions- und Permutations-Chiffren kombiniert, die als Substitutions- und Permutationsnetzwerke bezeichnet werden.

Es gibt zwar keine Methode der Kryptoanalyse, die schneller ist als eine erschöpfende Schlüsselsuche, aber moderne Chiffren wie DES (1976) oder AES (2000) erreichen in der Praxis dasselbe. Mit anderen Worten: Es ist keine kryptoanalytische Methode bekannt, die schneller als die erschöpfende Schlüsselsuche ist. Das entscheidende Kriterium ist eine hohe Diffusion im Sinne von Shannon.

Hashfunktionen wandeln beliebige Daten mit variabler Länge in eine Ausgabe mit fester Länge um. Der ausgegebene Hashwert sollte eine gute Diffusion aufweisen: die Umkehrung eines Eingabebits führt zu Änderungen bei etwa der Hälfte der Ausgabe; er sollte für alle möglichen Sequenzen mit fester Länge anwendbar sein und einer gleichförmigen Zufallsvariablen ähneln, d. h. die Wahrscheinlichkeit für jeden der Funktionswerte ist die gleiche. Bei einer Ausgabe mit fester Länge von 256 Bits sollte im Idealfall jeder Wert die gleiche Wahrscheinlichkeit 2-256 haben. Es handelt sich um eine kryptografische (oder Einweg-) Hashfunktion, wenn eine Umkehrung rechnerisch undurchführbar ist und wenn ähnliche Daten unterschiedliche Hashwerte ergeben. Empfohlen werden die neueren Versionen SHA-256 und SHA-3.