Friedrich-Schiller-Universität Jena
Fakultät für Mathematik und Informatik

Seminar Algebra

Symmetrische und asymmetrische Verfahren der Kryptographie

Jens Kubieziel
<jens@kubieziel.de>

Die vorliegende Ausarbeitung wurde auf der Grundlage des Buches „Codierungstheorie und Kryptographie“ von WOLFGANG WILLEMS erstellt. Es wird die Funktionsweise des symmetrischen Verschlüsselungsalgorithmus AES sowie des asymmetrischen Algorithmus' RSA und des Schlüsseltausch nach Diffie-Hellman beschrieben.

Inhaltsverzeichnis

Abkürzungsverzeichnis

AES Advanced Encryption Standard

CESG Communication-Electronics Security Group

DES Data Encryption Standard

DFN Deutsches Forschungsnetz e. V.

EFF Electronic Frontier Foundation

FIPS Federal Information Processing Standard

GCHQ Government Communications Headquarters

NBS National Bureau of Standards

NESSIE New European Schemes for Signatures, Integrity and Encryption

NIST National Institute of Standards and Technology

PGP Pretty Good Privacy

1 Einleitung

Diese Ausarbeitung entstand im Rahmen des Seminars Algebra unter der Leitung von Prof. Dr. Burkhard Külshammer. Im Rahmen der Veranstaltung wurde das Buch „Codierungstheorie und Kryptographie“ von WOLFGANG WILLEMS besprochen. Der Autor behandelte in seinem Vortrag die Kapitel 11 (Symmetrische Verfahren – die AES-Chiffrierung) und 12 (Public-Key-Kryptographie). Die Ausarbeitung gibt die wesentliche Inhalte des Vortrages wieder.

1.1 Grundbegriffe

Dieser Abschnitt soll einige der im folgenden verwendeten Begriffe definieren. Diese sind durchgängig in allen weiteren Abschnitten gültig.

Der Text setzt sich mit verschiedenen Methoden zur Verschlüsselung auseinander. Doch was bedeutet dieser Begriff genau? Um das zu erklären, werden die Begriffe des Klar- und des Geheimtextes benötigt:

Definition 1 (Klar-,Geheimtext)
*Ein **Klartext** ist der offene, interpretierbare Wortlaut eines Textes. Der **Geheimtext** hingegen ist der nicht interpretierbare, unleserliche Wortlaut eines Textes.*

Manchmal wird der Geheimtext auch als **Chiffrat** oder **Kryptogramm** bezeichnet. Mit der obigen Definition kann erklärt werden, was Verschlüsselung bedeutet:

Definition 2 (Verschlüsselung, Schlüssel)
*Als **Verschlüsselung** wird eine Maßnahme oder ein Vorgang bezeichnet, der aus einem Klartext einen Geheimtext erzeugt. Der **Schlüssel** wird benötigt, um eine Verschlüsselung durchzuführen.*

Umgangssprachlich ist klar, dass die **Entschlüsselung** die Wiedergewinnung des Klartextes ist. Bis nach dem 2. Weltkrieg waren nur Verfahren bekannt, bei denen Absender und Empfänger einer Nachricht im Besitz desselben Schlüssels sein mussten. Diese Verfahren bilden eine Klasse von Kryptosystemen:

Definition 3 (Symmetrisches Verschlüsselungsverfahren)
*Ein **symmetrisches Verschlüsselungsverfahren** ist ein Verfahren, welches zur Ver- und Entschlüsselung den gleichen Schlüssel k verwendet. Der Schlüssel k wird geheim gehalten, während Ver- und Entschlüsselungsfunktion allgemein bekannt sind.*

Gegen Ende der 1960er Jahre wurden Verschlüsselungsverfahren immer weiter eingesetzt. Mit dem steigenden Einsatz ergab sich das Problem der Schlüsselverteilung. Wenn zwei Personen miteinander geheim kommunizieren wollen, müssen diese sich einmal zum Austausch ihrer Schlüssel treffen. Kommt eine weitere Person hinzu, muss diese mit beiden Personen den Schlüssel tauschen. Insgesamt findet dreimal ein Schlüsseltausch statt. Kommt noch eine weitere Person hinzu, tauscht diese mit den schon

vorhandenen Personen den Schlüssel aus und die Zahl der Gesamtoperationen steigt auf 6. Insgesamt sind für eine Gruppe von n Personen $\frac{n(n-1)}{2}$ Operationen zum Schlüsseltausch (Nachweis siehe Unterabschnitt 3.1) notwendig. Somit erweisen sich symmetrische Verfahren bei einer hohen Zahl als unpraktikabel und die Forschung versuchte andere Möglichkeiten zu finden.

Definition 4 (Asymmetrische Verschlüsselungsverfahren)
*Ein **asymmetrisches Verschlüsselungsverfahren** ist ein Verfahren, bei dem jeder an der Verschlüsselung beteiligte Person ein Schlüsselpaar besitzt. Dieses besteht aus einem geheimen und einem öffentlichen Schlüssel.*

Der öffentliche Schlüssel dient dazu, um eine Nachricht an die Person zu verschlüsseln. Wie der Name sagt, unterliegt dieser keiner Geheimhaltung und kann auf beliebige Weise veröffentlicht werden. In der Praxis gibt es u. a. so genannte Keyserver[1], die Schlüssel auflisten.

1.2 Personen

Bei der Erklärung von Algorithmen zur Verschlüsselung werden meist bestimmte Rollen für die beteiligten Personen benutzt. RON RIVEST legte in (Rivest, Shamir, and Adleman 1978) folgende Bezeichnungen fest:

Alice und Bob Diese beiden Akteure wollen untereinander Nachrichten austauschen. Dies soll möglichst unbeobachtet geschehen. Sollten weitere Teilnehmer benötigt werden, so nutzt man in der Regel die Namen „Charlie“ und „Dave“.

Eve Eve ist eine Angreiferin, die nur die Möglichkeit hat, Nachrichten auf dem Übertragungsweg zu beobachten. Der Name leitet sich von dem englischen Wort „eavedropper“ ab.

Mallory Im Gegensatz zu Eve kann Mallory aktiv in die Übertragung eingreifen. Das heißt, er kann Nachrichten zurückhalten, verändern, neu einspeisen etc. Damit stellt er ein stärkeres Angreifermodell als Eve dar. Der Name leitet sich vom englischen Wort „malicious“ ab.

Für weitere Protokolle ist unter Umständen ein Notar, ein Polizist oder eine andere Rolle notwendig. Diese erhalten dann ebenfalls entsprechende Namen. Im Rahmen dieser Ausarbeitung werden sie nicht benötigt.

1.3 Klassifikation von Verfahren

Ein erstes Merkmal zur Klassifikation von Verschlüsselungsverfahren stellt die Einteilung in Block- und Stromchiffren dar.

[1] Ein Beispiel ist der PGP-Keyserver des Deutsches Forschungsnetz e. V. (DFN) unter der URL `http://pgpkeys.pca.dfn.de/`.

Definition 5 (Block-/Stromchiffre)
*Eine **Blockchiffre** ist ein Verschlüsselungsverfahren, welches den Klartext in gleichgroße Teile trennt und den Algorithmus auf jedes Teilstück anwendet.*

*Eine **Stromchiffre** wendet den Algorithmus auf jede eingegebene Informationseinheit an.*

Beispiele für Blockchiffren sind die Verschlüsselungsverfahren AES oder Data Encryption Standard (DES). Eine bekannte Stromchiffre ist das im Mobilfunk eingesetzte A5/1-Verfahren. Aber auch das XOR-Verfahren (Addition modulo 2) kann als Stromchiffre aufgefasst werden.

Eine weitere Unterscheidung wird nach symmetrischen und asymmetrischen Verfahren vorgenommen. Verschiedene derartige Algorithmen werden unten vorgestellt.

2 Advanced Encryption Standard (AES)

Das Verschlüsselungsverfahren AES ist eine Blockchiffre und wurde im Oktober 2000 von der US-amerikanischen Standardisierungsbehörde National Bureau of Standards (NBS)[2] als Federal Information Processing Standard (FIPS) 197 ins Leben gerufen. Damit löste es DES als amerikanischen Standard ab. Denn gegen Ende des letzten Jahrhunderts kamen ernsthafte Zweifel an der Sicherheit von DES auf. Die amerikanische Bürgerrechtsorganisation Electronic Frontier Foundation (EFF) zeigte 1998, dass es mit einem Einsatz von 250 000 US-Dollar möglich ist, einen Rechner zu bauen, der innerhalb von 56 Stunden den Schlüssel berechnet. Dies konnte durch aktuelle Forschungen der Ruhr-Universität Bochum und Kiel verbessert werden. Die als COPACABANA getaufte Maschine kostet etwa 10 000 US-Dollar und kann einen DES-Schlüssel innerhalb sechs Tagen berechnen.

Gegen AES sind bislang keine Schwächen bekannt. Es wird in vielfältiger Weise als Verschlüsselung eingesetzt. Bekannte Beispiele sind die Programme WinZip oder Pretty Good Privacy (PGP) bzw. GnuPG. Europaweit ist AES im Rahmen von New European Schemes for Signatures, Integrity and Encryption (NESSIE) standardisiert (NESSIE consortium 2003).

2.1 Standardisierung

AES wurde in einem neuen, offenen Verfahren diskutiert und standardisiert. Anfang 1997 rief das NBS Forscher in aller Welt auf, Vorschläge für ein Verschlüsselungsverfahren einzureichen. Von seiten der Behörde wurden hierbei folgende Vorgaben gemacht:

- symmetrische Blockchiffre
- Blocklänge 128 Bit, Schlüssellänge 128, 192 und 256 Bit
- leichte Umsetzung in Hard- und Software

[2] Diese firmiert heute unter dem Namen National Institute of Standards and Technology (NIST).

- überdurchschnittliche Performance
- sicher gegen bekannte Angriffe
- frei von patentrechtlichen Ansprüchen

Daraufhin gab es 15 Einreichungen. Im einzelnen waren das die Algorithmen in Tabelle 1. Die Finalisten des Wettbewerbs sind mit einem „X“ gekennzeichnet.

Name	Teilnahme in der Endrunde
CAST-256	
CRYPTON	
DEAL	
DFC	
E2	
FROG	
HPC	
LOKI97	
MAGENTA	
MARS	X
RC6	X
Rijndael	X
SAFER+	
Serpent	X
Twofish	X

Tabelle 1: Übersicht aller AES-Kandidaten

Im folgenden wurden mehrere Kongresse veranstaltet. Die Teilnehmer diskutierten die Algorithmen, versuchten Schwachstellen zu finden und es wurde auch die Schnelligkeit auf verschiedenen Plattformen gemessen. Insgesamt kamen fünf Algorithmen in die engere Wahl. Alle hatten ähnlich gute Eigenschaften bezüglich deren Sicherheit. Der Kandidat Rijndael von den Wissenschaftlern JOAN DAEMEN und VINCENT RIJMEN überzeugte wegen der Einfachheit des Designs und der Geschwindigkeit. Daher wurde dieser zum Sieger gekürt.

Die NIST wählte eine Variante von Rijndael. Diese umfasst eine feste Blockgröße von 128 Bit sowie die Schlüsselgrößen von 128, 192 und 256 Bit. Die Spezifikation des AES-Algorithmus ist als FIPS 197 (National Institute of Standards and Technology 2001) veröffentlicht und kann im Internet heruntergeladen werden[3].

[3] `http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf`

2.2 Der AES-Algorithmus

AES nimmt den Klartext entgegen und modifiziert ihn im Ablauf mehreren Runden. In neun Runden werden die vier Operationen `AddRoundKey`, `SubBytes`, `ShiftRows` und `MixColumns` angewendet. In der zehnten Schlussrunde werden nur die Operationen `AddRoundKey`, `SubBytes` und `ShiftRows` angewendet. Daneben wird in jeder Runde ein neuer Schlüssel berechnet.

2.2.1 Behandlung der Eingabe

Der Klartext, der als Eingabe in den Algorithmus einfließt, besteht aus einer Aneinanderreihung von 0 und 1. Diese Werte werden als Elemente des Körpers $\mathbb{F}$ betrachtet. Zu Beginn wird der Klartext in Datenblöcke zu je 128 Zeichen geteilt. Auf dieses Eingabewort wird der Algorithmus angewendet.

Definition 6 (Bit, Byte)
*Sei $b \in \mathbb{F}$. Man bezeichnet das Element b als **Bit** und eine Sequenz von 8 Bit als **Byte**.*

Zu Beginn erfolgt eine Teilung des 128 Bit großen Eingabewortes in 16 gleich große Teile zu je einem Byte. Jedes Byte wird durch ein Polynom mit Elementen aus einem endlichen Körper repräsentiert:

$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x^1 + b_0x^0 = \sum_{i=0}^{7} b_i x^i \tag{1}$$

So könnte beispielsweise die Folge 00100110 durch $x^5 + x^2 + x$ identifiziert werden. Mit dem irreduziblen Polynom $x^8 + x^4 + x^3 + 1$ und der Multiplikation modulo des Polynoms kann der Körper $\mathbb{F}_{2^8}$ repräsentiert werden.

In der Informatik greift man jedoch häufig auf hexadezimale Schreibweise zurück. Dazu wird jedem Bitmuster auf die in Tabelle 2 dargestellte Weise ein Wert zugewiesen.

Bitmuster	0000	0001	0010	0011	0100	0101	0110	0111
Wert	0	1	2	3	4	5	6	7
Bitmuster	1000	1001	1010	1011	1100	1101	1110	1111
Wert	8	9	A	B	C	D	E	F

Tabelle 2: Hexadezimale Repräsentation der Bitmuster

Die oben erwähnte Eingabefolge 00100110 hätte daher die Darstellung 26. In der Literatur wird sehr oft diese Schreibweise verwendet.

Nachdem das Eingabewort geteilt ist, werden die 16 Bytes als Einträge $e_0, \ldots, e_{15}$ in eine 4×4-Matrix der folgenden Form übernommen.

$$M = \begin{pmatrix} e_0 & e_4 & e_8 & e_{12} \\ e_1 & e_5 & e_9 & e_{13} \\ e_2 & e_6 & e_{10} & e_{14} \\ e_3 & e_7 & e_{11} & e_{15} \end{pmatrix} \in \mathbb{F}_{2^8}$$

Diese Matrix wird im Verlauf des Algorithmus in eine neue Matrix transformiert. Die transformierte Matrix stellt den Geheimtext dar.

2.2.2 Die Operation `SubBytes`

Das Wort `SubBytes` steht für „substitute bytes“. Schon daraus lässt sich vermuten, dass eine Substitution statt findet. Die Substitution wird auf jeden Eintrag $a_{ij} \in \mathbb{F}_{2^8}$ der Matrix M angewendet. Dabei sind $i, j = 0, \ldots, 3$. Die Substitutionsfunktion S besteht aus einer Verknüpfung zweier Funktionen f und g.

Die Funktion $f\colon \mathbb{F}_{2^8} \to \mathbb{F}_{2^8}$ bildet x auf sein multiplikatives Inverses x^{-1} ab, falls $x \neq 0$ ist. Sonst wird x auf 0 abgebildet. Die Funktion ist selbstinvers, d. h. $f(f(x)) = x$ und damit invertierbar.

KAISSA NYBERG beschrieb in (Nyberg 1994) verschiedene Substitutionsfunktionen. Die Designer des AES-Algorithmus' entschieden sich für die oben genannte Funktion, da diese eine minimale Wahrscheinlichkeit besitzt, Unterschiede im Klartext zu propagieren. Hierzu betrachtet man eine invertierbare, lineare Abbildung $f\colon \mathbb{F}_{2^8} \to \mathbb{F}_{2^8}$ und einen Schlüssel $k \in \mathbb{F}_{2^8}$. Mit einem Klartext $x \in \mathbb{F}_{2^8}$ lässt sich der Schlüssel durch $f(x+k) + f(x) = f(x) + f(k) + f(x) = f(k)$ gewinnen. Somit wird versucht, Funktionen zu finden, für die Anzahl $N_f = |\{f(x+k) + f(x)\colon x \in \mathbb{F}_{2^8}\}|$ für $k \neq 0$ groß ist.

Definition 7 (APN-Funktion)
*Funktionen, für die gilt, $N_f = |\{f(x+k) + f(x)\colon x \in \mathbb{F}_{2^n}\}| = 2^{n-1}$, heißen **APN-Funktionen**. Die Abkürzung APN steht dabei für Almost Perfect Nonlinear.*

Die zweite Funktion g, die für die Substitution verwendet wird, ist eine affine Transformation. Sie ist wie folgt definiert:

$$g\begin{pmatrix} a_7 \\ a_6 \\ a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a^7 \\ a^6 \\ a^5 \\ a^4 \\ a^3 \\ a^2 \\ a^1 \\ a^0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Es ist klar, dass die Multiplikationsmatrix in der Abbildung g invertierbar ist wie auch die Addition. Daher ist die gesamte Substitution $S = g \circ f$ invertierbar.

In der konkreten Umsetzung in Programmcode wird auf die Implementierung der Einzelfunktionen verzichtet. Da es sich um Abbildungen auf endlichen Körpern (konkret 256 Elemente) handelt, stellt das Programm eine feste Tabelle zur Verfügung. Dort wird nachgeschlagen, wie jede Bytefolge abgebildet werden muss. In Tabelle 3 ist die so genannte **S-Box** dargestellt. Die Eingabefolge 00100110 (binär) oder `26` (hexadezimal) wird auf die Hexadezimaldarstellung `F7` abgebildet.

x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Tabelle 3: Substitutionsbox in Programmcode

2.2.3 Die Operation `ShiftRows`

Der nächste Schritt in Algorithmus ist die Operation `ShiftRows`. Dabei werden die Zeilen der Matrix zyklisch nach links verschoben. Für $i = 0, \ldots, 3$ erfolgt dabei eine Verschiebung um i Stellen:

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \rightarrow \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{11} & a_{12} & a_{13} & a_{10} \\ a_{22} & a_{23} & a_{20} & a_{21} \\ a_{33} & a_{30} & a_{31} & a_{32} \end{pmatrix}$$

Es ist klar, dass dieser Schritt invertierbar ist. Weiterhin ist er diffusionsoptimal, d. h. statistische Strukturen des Klartextes werden gut aufgelöst. Die Diffusionseigenschaft wurde von CLAUDE E. SHANNON erstmals im Jahr 1949 in (Shannon 1949) beschrieben und ist wichtig, um gegen verschiedene Angriffe resistent zu sein.

2.2.4 Die Operation `MixColumns`

Schließlich erfolgt eine Permutation der Matrix M durch die Operation `MixColumns`. Jede Eingabespalte $a_{i.}$ wird mit einem Polynom $c(x) = 03 \cdot x^3 + 01 \cdot x^2 + 01 \cdot x + 02$ modulo $x^4 + 1$ multipliziert. Die Multiplikation mit dem Polynom kann als Multiplikation mit einer Matrix aufgeschrieben werden:

$$\begin{pmatrix} b_{i0} \\ b_{i1} \\ b_{i2} \\ b_{i3} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \cdot \begin{pmatrix} a_{i0} \\ a_{i1} \\ a_{i2} \\ a_{i3} \end{pmatrix}$$

Die Koeffizienten des Polynoms wurden von den Forschern so gewählt, dass der Algorithmus auf allen Computerarchitekturen schnell und effizient eingebaut werden kann. In Kapitel 4 des Buches (Daemen and Rijmen 2002) werden die Aspekte detailliert diskutiert.

Weiterhin trägt die Operation `MixColumns` wesentlich zur Diffusion bei. Zur Verdeutlichung betrachte man zwei Matrizen, die sich nur im Eintrag a_{00} unterscheiden. Im Ablauf der Operationen `SubBytes` und `ShiftRows` wird der Eintrag so geändert, dass sich die Änderung nur auf das Feld a_{00} beschränkt. Erst durch `MixColumns` wird die Änderung durch die Multiplikation auf die erste Spalte übertragen. Dies ist schematisch unten dargestellt.

$$\begin{pmatrix} * & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \xrightarrow[\text{Bytes}]{\text{Sub}} \begin{pmatrix} * & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \xrightarrow[\text{Rows}]{\text{Shift}} \begin{pmatrix} * & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \xrightarrow[\text{Columns}]{\text{Mix}} \begin{pmatrix} * & 0 & 0 & 0 \\ * & 0 & 0 & 0 \\ * & 0 & 0 & 0 \\ * & 0 & 0 & 0 \end{pmatrix}$$

In der folgenden Runde wird durch die erste Operation `SubBytes` keine Änderung an der Struktur vorgenommen. Durch `ShiftRows` erfolgt dann eine Verschiebung der Änderungen und schließlich sorgt `MixColumns` dafür, dass sich der ursprüngliche Unterschied im Eintrag a_{00} auf die komplette Matrix propagiert hat.

$$\begin{pmatrix} * & 0 & 0 & 0 \\ * & 0 & 0 & 0 \\ * & 0 & 0 & 0 \\ * & 0 & 0 & 0 \end{pmatrix} \xrightarrow[\text{Bytes}]{\text{Sub}} \begin{pmatrix} * & 0 & 0 & 0 \\ * & 0 & 0 & 0 \\ * & 0 & 0 & 0 \\ * & 0 & 0 & 0 \end{pmatrix} \xrightarrow[\text{Rows}]{\text{Shift}} \begin{pmatrix} * & 0 & 0 & 0 \\ 0 & 0 & 0 & * \\ 0 & 0 & * & 0 \\ 0 & * & 0 & 0 \end{pmatrix} \xrightarrow[\text{Columns}]{\text{Mix}} \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix}$$

2.2.5 Die Operation `AddRoundKey`

`AddRoundKey` ist eine Addition modulo 2 oder eine XOR-Operation mit der Matrix M und dem Rundenschlüssel.

2.2.6 Die Schlüsselexpansion

Der Anfangsschlüssel wird zu jeder Runde neu berechnet und der so berechnete im Schritt `AddRoundKey` (siehe Unterunterabschnitt 2.2.5) mit der Matrix M addiert. Zur Auswahl der Schlüsselexpansion waren wieder Effizienzüberlegungen entscheidend. Die Designer hatten insbesondere 8-Bit-Prozessoren im Blick (siehe Abschnitt 3.6 in (Daemen and Rijmen 2002)).

Bei der Schlüsselexpansion wird der Schlüssel in Form einer $4 \times i$-Matrix geschrieben. Dabei variiert i je nach Schlüssellänge zwischen $4, 6$ und 8. Für die weiteren Betrachtungen ist $i = 4$ gesetzt. Das bedeutet, dass die Matrix vier Spalten besitzt. Zu der Matrix werden im betrachteten Beispiel 40 weitere Spalten erzeugt. Die Spalten der Matrix werden mit $K[j]$ für $j = 0, \ldots, 43$ bezeichnet.

Zu Beginn stehen die Spalten $K[0], K[1], K[2]$ und $K[3]$ durch den Anfangsschlüssel fest. Bei der weiteren Expansion wird unterschieden, ob j durch 4 teilbar ist oder nicht.

Falls gilt, $j \mid 4$, so wird auf die Spalte $K[j-1]$ die Abbildung $R\colon \mathbb{F}_{2^8} \to \mathbb{F}_{2^8}$ angewendet und das Ergebnis mit der Spalte $K[j-4]$ addiert. Die Abbildung R ist definiert durch:

$$R\begin{pmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \end{pmatrix} = \begin{pmatrix} S(k_2) \\ S(k_3) \\ S(k_4) \\ S(k_1) \end{pmatrix} + \begin{pmatrix} a^{\frac{j-4}{4}} \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Dabei sind die k_1, k_2, k_3 und k_4 die Einträge der Spalten, S die Substitution aus der Operation `SubBytes` (siehe Unterunterabschnitt 2.2.2) und $a = 00000010 \in \mathbb{F}_{2^8}$. Das Element a wird in der Literatur häufig als Rundenkonstante `Rcon` (englisch: round constant) bezeichnet. Insgesamt gilt somit für $j \mid 4$:

$$K[j] = K[j-4] + R(K[j-1])$$

Falls $j \nmid 4$, so findet eine Addition modulo 2 mit der Spalte $K[j-4]$ statt:

$$K[j] = K[j-4] + K[j-1]$$

Für die Runden $r = 1, \ldots, 10$ besteht der Rundenschlüssel $k^{(r)}$ aus der Matrix $k^{(r)} = (K[4j]\ K[4j+1]\ K[4j+2]\ K[4j+3]) \in \mathbb{F}_{2^8}$.

2.2.7 Entschlüsselung

Die Entschlüsselung besteht aus der Anwendung der inversen Schritte. In der obigen Beschreibung wurde auf die Invertierbarkeit der Operationen eingegangen und gezeigt, dass jeder Schritt des Algorithmus' invertierbar ist. Damit ist auch die Verknüpfung der Operationen invertierbar. In der Literatur wird den Entschlüsselungsoperationen oft der Präfix `Inv` angehangen (z. B. `InvSubBytes` oder `InvMixColumns`) und die Operationen werden separat beschrieben.

2.3 Angriffe auf AES

Ein wesentliches Auswahlkriterium für den Rijndael-Algorithmus als neuen amerikanischen Standard war dessen Sicherheit. DAEMEN und RIJMEN berücksichtigten bei der Erschaffung von Rijndael bzw. von AES alle bekannten Angriffe. Forscher, die den Algorithmus auf mögliche Schwächen untersuchten, konnten im Rahmen des Auswahlprozesses keine finden.

Heutzutage ist es bei der Prüfung eines Verschlüsselungsverfahrens üblich, Klar- und Geheimtexte frei zu wählen und anhand von spezifischen Eigenschaften Schwachstellen zu finden. Weiterhin sollte, wie im Falle von AES, der Algorithmus selbst offenliegen. Dieses Prinzip erkannte bereits der niederländische Kryptologe AUGUSTE KERCKHOFFS. In seiner Veröffentlichung „La cryptographie militaire“ aus dem Jahr 1883 beschrieb er sinngemäß, dass die Sicherheit des Verfahrens auf der Geheimhaltung des Schlüssels und nicht des Verfahrens selbst beruhen sollte (Kerckhoffs 1883).

Die naheliegende Möglichkeit, aus einem gegebenen Geheimtext, den Klartext zu finden, ist alle möglichen Schlüssel zu probieren. Dies wird üblicherweise als **Exhaustive Key Search** oder **Brute-Force-Suche** bezeichnet. Bei AES hat der Schlüssel eine Länge von 128, 192 oder 256 Bit. Das bedeutet, es gibt 2^{128}, 2^{192} oder 2^{256} Kombinationen. Um die Anzahl etwas zu verdeutlichen, sei angenommen, dass die Anzahl der Menschen auf der Erde in etwa 2^{33} beträgt und jeder Mensch eine Computer besitzt, der 2^{34} Berechnungen pro Sekunde durchführen kann. Das entspricht etwa der nominellen Leistungskraft aktuell erhältlicher Rechensysteme. Eine weitere Annahme sei, dass in jeder Berechnung ein Schlüssel getestet werden kann und das die Schlüsselsuche auch in dem Maße parallel berechnet sowie koordiniert werden kann. Ein Jahr hat 31 536 000 Sekunden. Dies entspricht etwa der Zahl $2^{25} = 33554432$. Wenn alle Beteiligten sich an der Suche beteiligen, so dauert es mindestens $\frac{2^{127}}{2^{33} \cdot 2^{34} \cdot 2^{25}} = 2^{35} = 34359738368$ Jahre bis der korrekte Schlüssel gefunden wird. Somit ist bei AES von einem berechnungssicherem Code auszugehen.

Im Rahmen der Kryptoanalyse wird nun versucht, einen Angriff zu finden, der weniger Versuche benötigt. Dazu betrachten Forscher eine Version des Algorithmus', der weniger Runden hat und versuchen, Schwachpunkte zu finden. In weiteren Schritten erfolgen dann Versuche, die gefundenen Schwachpunkte auszubauen.

2.3.1 Differentielle Kryptoanalyse

Die differentielle Kryptoanalyse ist der Öffentlichkeit seit 1990 bekannt und erwies sich als eine mächtige Form, um Verschlüsselungen zu brechen. Die Forscher BIHAM und SHAMIR fanden damit einen effektiven Angriff gegen DES (Biham and Shamir 1991).

Im Rahmen des Angriffs werden zwei Klartexte betrachtet, die nur kleine Unterschiede aufweisen. Die differentielle Kryptoanalyse betrachtet die Veränderung des Unterschiede im Verlauf der Verschlüsselung. Ausgehend von den Unterschieden im Geheimtext werden verschiedenen Schlüsseln Wahrscheinlichkeiten zugewiesen. Durch

die Analyse von anderen Unterschieden kann dann ein Schlüssel wahrscheinlicher als andere erscheinen.

Dies ist nur eine grobe Beschreibung. Im Allgemeinen ist diese Form der Analyse komplizierter und zahlreiche Varianten wurden mittlerweile entwickelt. Für eine tiefergehende Beschreibung sei auf das Buch „Applied Cryptography“ von BRUCE SCHNEIER (Schneier 1996) verwiesen.

In 2.2.4 wurde gezeigt, wie sich der Unterschied in einem Eintrag der Matrix im Verlauf von zwei Runden ändert. Die Forscher bestimmten eine obere Schranke für die Wahrscheinlichkeit, dass sich Änderungen über vier Runden fortsetzen. Diese liegt bei 2^{-150} (Kapitel 9, (Daemen and Rijmen 2002)). Daher ist AES als sehr sicher gegen diese Form des Angriffs einzuschätzen.

2.3.2 Geschlossene Darstellung

Den Forschern FERGUSON, SCHROEPPEL und WHITING gelang es in (Ferguson, Schroeppel, and Whiting 2001), AES in einer geschlossenen Form darzustellen. Diese Darstellung liegt in Form eines Kettenbruchs vor. Nach fünf Runden ergibt sich folgendes vereinfachte Bild:

$$a_{i,j}^{(6)} = K + \sum_{\substack{e_5 \in \mathcal{E} \\ d_5 \in \mathcal{D}}} \cfrac{C_1}{K^* + \sum\limits_{\substack{e_4 \in \mathcal{E} \\ d_4 \in \mathcal{D}}} \cfrac{C_2}{K^* + \sum\limits_{\substack{e_3 \in \mathcal{E} \\ d_3 \in \mathcal{D}}} \cfrac{C_3}{K^* + \sum\limits_{\substack{e_2 \in \mathcal{E} \\ d_2 \in \mathcal{D}}} \cfrac{C_4}{K^* + \sum\limits_{\substack{e_1 \in \mathcal{E} \\ d_1 \in \mathcal{D}}} \cfrac{C_5}{K^* + p_*^*}}}}$$

Dabei ist $a_{i,j}$ der Eintrag in der Matrix M, die C_k Konstanten, $\mathcal{E} = \{0, \ldots, 3\}, \mathcal{D} = \{0, \ldots, 7\}$ und $*$ steht für einen bekannten Index. Dieser Index hängt jeweils von den Summationsvariablen ab.

Als Formel geschrieben, ergeben sich 2^{25} Terme der Form $C_k/K^*+p_*^*$. Das heißt, für die volle zehnrundige Version des Algorithmus ergeben sich 2^{50} Terme. Für diese Art Gleichung sind derzeit keine effizienten Lösungsmethoden bekannt.

2.3.3 Related-Key-Kryptoanalyse

Im Juli 2009 veröffentlichte BRUCE SCHNEIER einen Bericht[4], nach dem ein neuer Angriff auf AES gefunden wurde. Dabei wird das Ergebnis der Ver- oder Entschlüsselung bei verschiedenen Schlüsseln beobachtet. Der Angreifer kennt den Schlüssel nicht, kann jedoch Einfluss auf spezielle Eigenschaften des Schlüssels nehmen. In (Biryukov and Khovratovich 2009) wird die AES-Variante mit einer Schlüssellänge von 256 Bit

[4] `http://www.schneier.com/blog/archives/2009/07/new_attack_on_a.html`

angegriffen. Die Forscher zeigen, dass durch deren Angriff nur 2^{119} Schlüssel geprüft werden müssen. Dies ist der derzeit effektivste bekannteste Angriff gegen AES.

3 Public-Key-Kryptographie

3.1 Einleitung

Mit der Verbreitung von verschiedenen Verschlüsselungsverfahren kam recht schnell das Problem der Schlüsselverteilung auf. Durch die steigende Anzahl an Teilnehmern müssen überproportional viele Schlüsseltausche stattfinden. Insgesamt liegt quadratischen Wachstum vor. Denn bei einem System mit n Nutzern müssen $\frac{n(n-1)}{2}$ Transaktionen passieren. Bei zwei Nutzern ist $\frac{2(2-1)}{2} = 1$ Transaktion erforderlich. Sei nun die obige Formel für n Nutzer als gegeben angesehen und der Nutzer $n+1$ möchte den Schlüssel bekannt geben. Hierzu muss er sich mit allen bisherigen Nutzern zum Tausch treffen. Das heißt, es ist $\frac{n(n-1)}{2} + n = \frac{n^2-n}{2} + \frac{2n}{2} = \frac{n^2+n}{2} = \frac{(n+1)((n+1)-1)}{2}$. Damit ist die Gültigkeit der Formel nachgewiesen. Die Abbildung 1 zeigt eine grafische Übersicht des Wachstums.

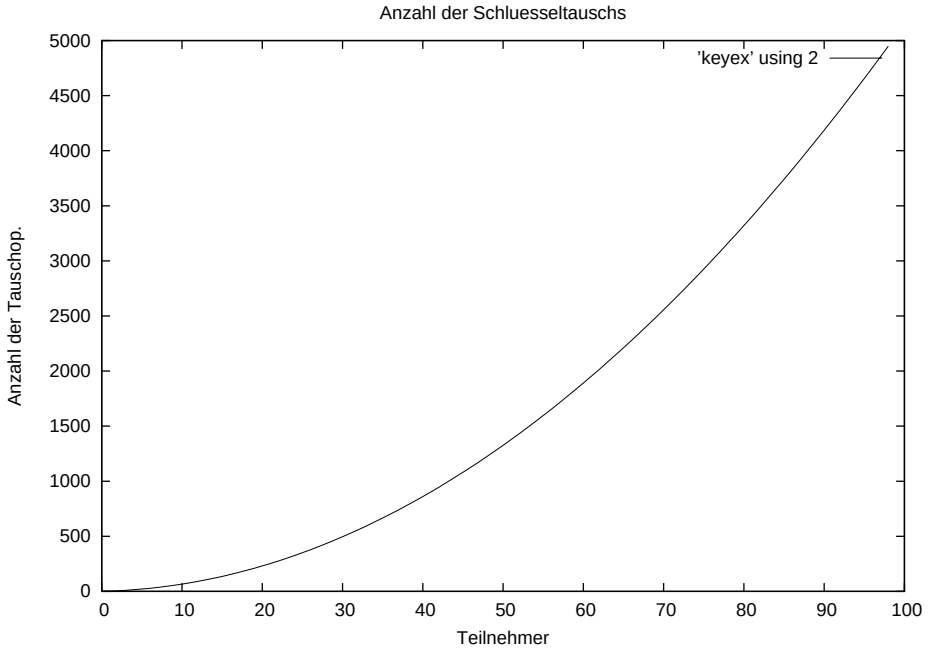

Abbildung 1: Grafische Darstellung der Anzahl an Tauschoperationen

Aus dem Grund begannen diverse Forschergruppen nach Lösungen zu suchen. Im Jahr 1969 wurde der britische Mathematiker JAMES ELLIS gebeten, eine Lösung für die Schlüsselverteilung zu finden. ELLIS arbeitete zu der Zeit für die Communication-Electronics Security Group (CESG) des britischen Geheimdienstes Government Communications Headquarters (GCHQ). In einem Bericht von Bell Telephone zu abhörsicherer Telefonen fand er die Grundidee. Darin wurde beschrieben, dass Alice ein zufälliges Rauschen in der Telefonleitung erzeugt. Nachdem das geschehen ist, sendet Bob seine Nachricht zu Alice. Da sie die Struktur des Rauschens kennt, kann sie das Rauschen rausrechnen und erkennt die Nachricht. Eve, die die Konversation belauscht, kann nichts über die gesendete Nachricht herausfinden. Dieses Verfahren inspirierte ELLIS zur so genannten „nichtgeheimen Verschlüsselung". Aus der theoretischen Idee erschuf ein weiterer Mitarbeiter des GCHQ, CLIFFORD COCKS, schließlich einen Verschlüsselungsalgorithmus.

Im Jahr 1977 fanden die Forscher RONALD L. RIVEST, ADI SHAMIR und LEONARD ADLEMAN ebenfalls das Verschlüsselungsverfahren. Nach den Initialen wurde es als **RSA** bezeichnet. Ihre Erkenntnisse basierten auf der Arbeit „New directions in cryptography" von WHITFIELD DIFFIE und MARTIN HELLMAN (Diffie and Hellman 1976). Danach gibt jeder Teilnehmer eines Kommunikationsnetzes seinen Schlüssel k sowie die Verschlüsselungsfunktion e_k bekannt. Dagegen bleibt die Entschlüsselungsfunktion d_k geheim und diese Funktion kann auch nicht aus e_k berechnet werden. Wenn nun Alice eine Nachricht an Bob senden will, wendet sie die Funktion auf ihre Nachricht x an: $e_k(x)$. Bob nimmt die Nachricht entgegen und wendet die Entschlüsselungsfunktion an: $x = d_k(e_k(x))$. Dies resultiert in der Definition der Einwegfunktion.

Definition 8 (Einwegfunktion)
*Seien X und Y Mengen. Eine injektive Funktion $f\colon X \to Y$ heißt **Einwegfunktion**, wenn für jedes $x \in X$ der Funktionswert $f(x) = y$ schnell berechnet werden kann und für jedes $y \in$ Bild f das Urbild $f^{-1}(y) = x$ nicht in vertretbarer Zeit gefunden werden kann.*

Weiterhin kann nun der Begriff des Public-Key-Kryptosystems definiert werden:

Definition 9 (Public-Key-Kryptosystem)
*Ein Kryptosystem $\mathbb{K}$ heißt **Public-Key-Kryptosystem**, falls alle Verschlüsselungsfunktionen e_k mit $k \in \mathcal{K}$ Einwegfunktionen sind.*

Bislang lässt sich die Eigenschaft der Einwegfunktion für keine Funktion nachweisen. Es wird angenommen, dass das Potenzieren von Elementen mit großer Ordnung in einer geeigneten Gruppe und die Multiplikation großer ganzer Zahlen Einwegfunktionen sind.

3.2 RSA

Der RSA-Algorithmus ist bis heute das Public-Key-Verfahren, welches sehr leicht zu verstehen und zu implementieren ist. Es wurde in den Jahren 1978 und 1979 veröffentlicht (Rivest, Shamir, and Adleman 1978) und (Rivest, Shamir, and Adleman 1979). Bis

heute wurden keine essenziellen Schwachstellen gefunden. Daher ist RSA immer noch der Quasi-Standard bei asymmetrischen Verfahren.

Der Ablauf des Algorithmus gestaltet sich wie folgt:

1. Wähle zwei Primzahlen p und q mit $p \neq q$.
2. Berechne $n = p \cdot q$ und $\varphi(n) = \varphi(p)\varphi(q) = (p-1)(q-1)$.
3. Wähle eine natürliche Zahl e mit $1 < e < \varphi(n)$ und $e \nmid \varphi(n)$ sowie
4. eine natürliche Zahl d mit

$$ed \equiv 1 \mod \varphi(n)$$

Die Verschlüsselungsfunktion von Bob ist dann:

$$e_{\text{Bob}} \colon \mathbb{Z}/n\mathbb{Z} \to \mathbb{Z}/n\mathbb{Z} \qquad x \mapsto y = x^e \mod n$$

und die Entschlüsselungsfunktion ist:

$$d_{\text{Bob}} \colon \mathbb{Z}/n\mathbb{Z} \to \mathbb{Z}/n\mathbb{Z} \qquad y \mapsto y^d \mod n$$

Nunmehr ist zu klären, ob die Verschlüsselungs- und Entschlüsselungsfunktion korrekt zusammenarbeiten. Gegenstand dessen ist der folgende Satz.

Satz 1
Für alle $x \in \mathbb{Z}/n\mathbb{Z}$ gilt:

$$d_{Bob}(e_{Bob}(x)) = x$$

BEWEIS:
Für alle Zahlen $x \in \mathbb{Z}$ gilt: $d_{\text{Bob}}(e_{\text{Bob}}(x)) = d_{\text{Bob}}(x^e \mod n) = (x^e \mod n)^d \mod n = x^{ed} \mod n$. Im Fall $x = 0$ ist alles klar. Daher muss nur der Fall $x \neq 0$ betrachtet werden. Hierfür müssen wieder zwei Fälle unterschieden werden.

1. Fall Sei $p \nmid x$ und $q \nmid x$. Im vierten Schritt des Algorithmus wurde die Zahl d mit $ed \equiv 1 \mod \phi(n)$ bestimmt. Die Äquivalenz lässt sich auch als $ed = 1 + z\phi(n)$ für eine natürliche Zahl z schreiben. Damit gilt, $x^{ed} = x^{1+z\phi(n)} = x(x^{\phi(n)})^z$. Der Satz von EULER besagt für eine natürliche Zahl n und eine ganze Zahl x, dass $x^{\phi(n)} \equiv 1 \mod n$ gilt. Diese Bedingung lässt sich anwenden und es ergibt sich, $x^{ed} = x(x^{\phi(n)})^z \equiv x(1 \mod n)^z \equiv x(1 \mod n) \equiv x \mod n = x$.

2. Fall Sei nun $p \mid x$ und $q \nmid x$. Es lässt sich die Gleichung $\phi(n) = \phi(q)\phi(p)$ und der Satz von EULER nacheinander anwenden. Insgesamt ergibt sich: $x^{ed} = x(x^{\phi(n)})^z = x(x^{\phi(q)})^{\phi(p)z} \equiv x(1 \mod q)^{\phi(p)z} \equiv x \mod q$. Wegen $p \mid x$ folgt, $p \mid x^{ed} - x$ und damit auch $n = pq \mid x^{ed} - x$. Damit ergibt sich insgesamt, $x^{ed} \equiv x \mod n$. Aus Symmetriegründen kann man die Teilbarkeitseigenschaft vertauschen und alle Fälle sind betrachtet. ∎

Wenn nun Mallory einen Geheimtext g abfängt und ohne Zusatzinformationen entschlüsseln möchte, so muss er $d_{\text{Bob}}(g) \equiv x \mod n$ berechnen. Hierzu benötigt er den Wert von d. Aufgrund der Kongruenz $ed \equiv 1 \mod \phi(n)$ müsste $\phi(n)$ bekannt sein, um d zu berechnen. Dies ist äquivalent zur Faktorisierung von n. In (Weis, Lucks, and Bogk 2002) wird der Aufwand für die Faktorisierung einer 1024 Bit langen Zahl diskutiert. Mit einem Aufwand von 50 Millionen US-Dollar soll es demnach möglich sein, die Zahl im Verlauf eines Jahres zu faktorisieren.

3.3 Der Diffie-Hellman-Schlüsseltausch

Wenn Alice und Bob kommunizieren wollen, müssen sie sich zunächst auf einen Schlüssel einigen. Da der Schlüssel der wichtigste Bestandteil des Kryptosystems ist, sollte der Austausch bzw. die Einigung sicher erfolgen. Gerade bei der Kommunikation über das Internet kann Abhörsicherheit nicht gewährleistet werden. Daher müssen Wege zur sicheren Einigung über einen gemeinsamen Schlüssel gefunden werden. Einen solchen Weg zeigten WHITFIELD DIFFIE und MARTIN HELLMAN im Jahr 1976 in (Diffie and Hellman 1976) auf. Hierzu sei eine Gruppe G sowie ein Element g aus G öffentlich bekannt. Zur Vereinbarung des Schlüssels unternehmen Alice und Bob folgende Schritte:

1. Alice wählt zufällig ein $a \in \{2, \ldots, n-1\}$ und sendet g^a zu Bob.
2. Bob wählt zufällig ein $b \in \{2, \ldots, n-1\}$ und sendet g^b zu Bob.
3. Bob berechnet $(g^a)^b = g^{ab}$ und Alice berechnet $(g^b)^a = g^{ba}$.

Der Schlüssel ist dann $g^{ab} = g^{ba}$.

Wenn nun Eve die Kommunikation mitlauscht, kennt sie $g \in G$ sowie g^a und g^b. Um nun g^{ab} zu berechnen, müsste sie den diskreten Logarithmus effizient ermitteln können. Dies wird derzeit als Einwegfunktion angesehen und es ist keine effiziente Berechnungsvorschrift bekannt.

Mallory als stärkerer Angreifer könnte sich jedoch zwischen die Kommunikationspartner stellen und im obigen ersten Schritt die Nachricht von Alice abfangen. Anschließend sendet er ein von ihm berechnetes g^a zu Bob, fängt die Antwort ab und schickt ein von ihm bestimmtes g^b zu Alice. Sowohl Alice als auch Bob sind der Meinung gegenseitig einen Schlüssel ausgetauscht zu haben. Jedoch geschah der Austausch in Wahrheit mit Mallory, der nun Eingriff in die gesamte Kommunikation hat. In der Praxis wird ein derartiger Angriff als **Man-in-the-middle-Angriff** bezeichnet. Bei der Benutzung von Public-Key-Kryptosystemen kann dem Problem einfach entgangen werden. Aber auch für andere Verfahren existieren Forschungsansätze, um dem Angriff Herr zu werden.

Literatur

Biham, E. and A. Shamir (1991). Differential Cryptanalysis of DES–like Cryptosystems. In *Advances in Cryptology–CRYPTO '90*, pp. 2–21. Springer-Verlag.

Biryukov, A. and D. Khovratovich (2009, Mai). Related-key Cryptanalysis of the Full AES-192 and AES-256. `https://cryptolux.uni.lu/mediawiki/uploads/1/1a/Aes-192-256.pdf`.

Daemen, J. and V. Rijmen (2002). *The Design of Rijndael* (1. Auflage ed.). Springer-Verlag.

Diffie, W. and M. Hellman (1976, November). New directions in cryptography. In *IEEE transactions on Information Theory*, Volume 22, pp. 644–654.

Ferguson, N., R. Schroeppel, and D. Whiting (2001). A simple algebraic representation of rijndael. In *Selected Areas in Cryptography*, Lecture Notes in Computer Science, pp. 103–111. Springer–Verlag.

Kerckhoffs, A. (1883, Januar). La cryptographie militaire. *Journal des sciences militaires Band 9*, 5–38.

National Institute of Standards and Technology (2001, Oktober). Federal Information Processing Standards Publication 197. Technical report, National Institute of Standards and Technology.

NESSIE consortium (2003, Februar). Portfolio of recommended cryptographic primitives. Technical report, NESSIE project, `http://www.cryptonessie.org/`.

Nyberg, K. (1994). *Differentially uniform mappings for cryptography* (LNCS 765 ed.). Advances in Cryptology, Proceedings Eurocrypt 1993. Springer-Verlag.

Rivest, R., A. Shamir, and L. Adleman (1978, Februar). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM 21*(2), 120–126.

Rivest, R., A. Shamir, and L. Adleman (1979, Januar). On Digital Signatures and Public Key Cryptosystems. Technical Report MIT/LCS/TR-212, MIT Laboratory for Computer Science.

Schneier, B. (1996). *Applied Cryptography* (2nd ed.). John Wiley & Sons.

Shannon, C. E. (1949). *Communication Theory of Secrecy Systems*, Volume 28–4. Bell System Technical Journal.

Weis, R., S. Lucks, and A. Bogk (2002). Sicherheit von 1024 bit RSA-Schlüsseln gefährdet. *DuD–Datenschutz und Datensicherheit 26*.