



Melita Birthälmer

Technische Universität Ilmenau  
Fakultät für Elektrotechnik und Informationstechnik

# Kryptografie

Hauptseminar im Fachgebiet Audiovisuelle Technik  
Sommersemester 2007

Name: Melita BIRTHÄLMER

Matrikel-Nr.: 34763

Studiengang: Medientechnologie

Betreuer: Dr. rer. nat. Eckhardt Schön

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Aufgaben der Kryptografie</b>	<b>2</b>
<b>3</b>	<b>Symmetrische Verschlüsselung</b>	<b>3</b>
3.1	Klassische symmetrische Verschlüsselung . . . . .	4
3.1.1	Skytale von Sparta (ca. 500 v. Chr.) . . . . .	4
3.1.2	Cäsar-Chiffrierung (ca. 100 v. Chr.) . . . . .	6
3.2	Stromchiffren . . . . .	7
3.3	A5-Algorithmus . . . . .	8
3.4	Blockchiffren . . . . .	9
3.5	Advanced Encryption Standard . . . . .	10
<b>4</b>	<b>Asymmetrische Verschlüsselung</b>	<b>13</b>
4.1	Einführung . . . . .	13
4.2	RSA-Verschlüsselung . . . . .	14
<b>5</b>	<b>Symmetrisch vs. Asymmetrisch</b>	<b>17</b>
<b>6</b>	<b>Hybride Verschlüsselung</b>	<b>18</b>
6.1	Einführung . . . . .	18
6.2	Digital Rights Management . . . . .	18
<b>7</b>	<b>Fazit</b>	<b>19</b>

# 1 Einleitung

Kryptografie ist die Lehre von Methoden zur Ver- und Entschlüsselung von Nachrichten zum Zweck der Geheimhaltung von Informationen gegenüber Dritten [3]. Ein wichtiges Werkzeug der Kryptografie ist die Mathematik, denn Verfahren zur sicheren Verschlüsselung von Daten können nur mit Hilfe von mathematischen Kenntnissen entwickelt werden. Die Welt der Kryptografie geht von einem einfachen Szenario aus, welches aus den Beteiligten Alice und Bob besteht, die Synonyme für Sender und Empfänger darstellen.

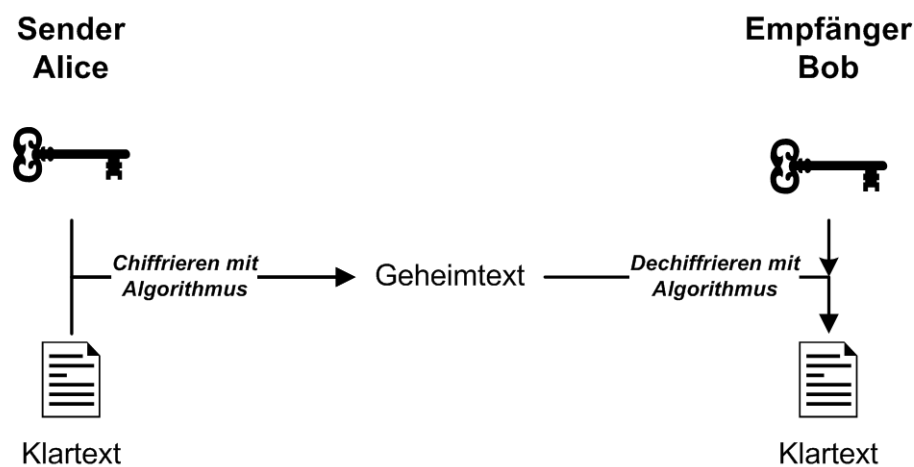


Abbildung 1: Einführung in die Kryptografie [2]

Alice möchte eine Nachricht an Bob übermitteln. Diese Nachricht nennt man Klartext. Alice chiffriert den Klartext mithilfe eines Schlüssels. Die verschlüsselte Nachricht wird als Geheimtext bezeichnet. Bob, der Empfänger der Nachricht muss den Geheimtext mithilfe eines Schlüssels dechiffrieren, um die Nachricht lesen zu können.

## 2 Aufgaben der Kryptografie

Kryptografie wird benötigt, um vertrauliche Informationen sicher auszutauschen. Neben Vertraulichkeit sind Authentizität, Integrität und Zurechenbarkeit die zentralen Sicherheitsziele der Kryptografie. Durch Verknüpfung unterschiedlicher kryptografischer Verfahren ist es möglich, diese Ziele zu erreichen.

### **Vertraulichkeit**

Kryptografische Methoden garantieren die Vertraulichkeit von elektronischen Daten, d. h., nur berechnigte Personen dürfen Zugriff auf Informationen haben. Ein Beispiel bei der Vertraulichkeit eine wichtige Rolle spielt, ist Homebanking, bei der nur Kunden einer Bank Konto-Transaktionen online durchführen können.

### **Authentizität**

Bei der Authentizität besteht das Ziel darin, dass die Herkunft einer übertragenen Information korrekt identifiziert werden kann. Tätigt man in einem Online-Shop einen Kauf, so ist es für den Betreiber des Online-Shops wichtig zu wissen, welcher Kunde den Kauf getätigt hat. Der Betreiber muss überprüfen können, ob der vorgegebene Absender dem tatsächlichen Absender entspricht.

### **Integrität**

Um die Integrität von Daten zu sichern, dürfen übertragene Informationen in keiner Weise manipuliert werden. Unter Manipulation versteht man dabei das Einfügen, Löschen oder Ersetzen von Daten. Beispielsweise muss eine Bank nachprüfen können, ob bei einer Online-Überweisung der Betrag nachträglich verändert wurde.

### **Zurechenbarkeit**

Kryptografische Methoden dienen auch dazu, elektronische Dokumente zurechenbar zu machen. Weder dem Sender noch dem Empfänger darf es möglich sein, die Übertragung einer Information abzustreiten. Wird beispielsweise online ein Kaufvertrag geschlossen, so müssen Käufer und Verkäufer nachweisen können, dass der Vertrag tatsächlich geschlossen wurde. Zu diesem Zweck können digitale Signaturen verwendet werden.

### 3 Symmetrische Verschlüsselung

Seit Beginn der Kryptografie gab es - bis in die 1970er Jahre hinein - ausschließlich symmetrische Verfahren. Die klassischen Verfahren wurden hauptsächlich mit der Hand oder mithilfe von mechanischen oder elektrischen Maschinen durchgeführt. Nach 1970 entstandene symmetrische Verfahren, wie z. B. AES, bedienen sich hingegen leistungsfähigen Computern. All diese Verfahren beruhen auf ein und demselben Szenario, welches in nachfolgender Abbildung dargestellt ist.

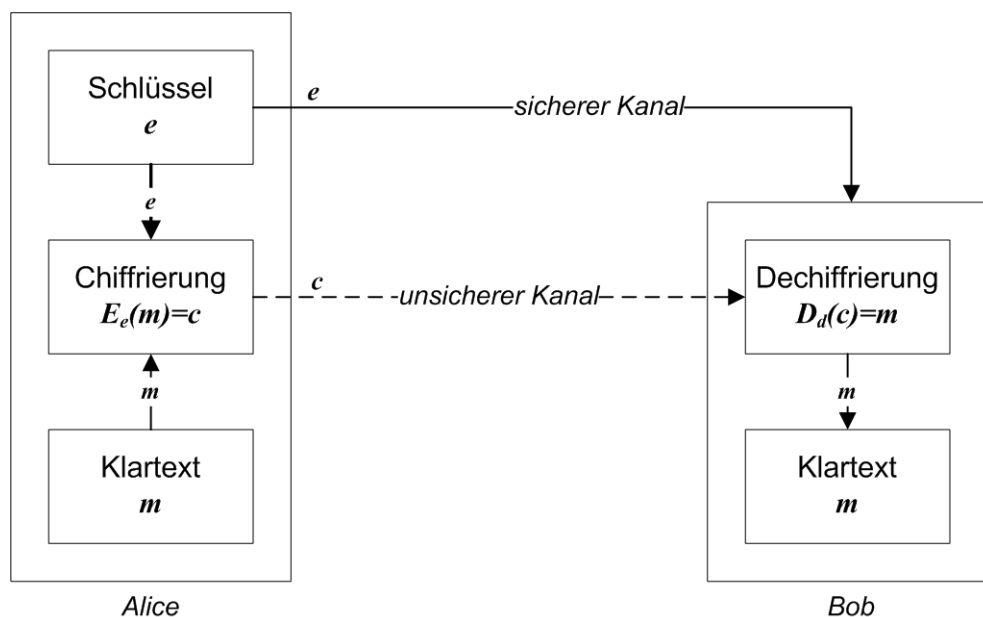


Abbildung 2: Zwei-Teilnehmer-Kommunikation mittels symmetrischer Verschlüsselung [1]

Alice möchte eine Klartextnachricht  $m$  an Bob schicken und chiffriert diese mit einem Schlüssel  $e$ . Zum Entschlüsseln des Geheimtextes  $c$  benötigt Bob den zugehörigen Schlüssel  $d$ . Man spricht von einem symmetrischen Verfahren, wenn  $d$  aus  $e$  leicht abgeleitet werden kann und umgekehrt. Im einfachsten Fall ist  $d = e$ . Wird ein solches Verfahren zur Verschlüsselung benutzt, so müssen Alice und Bob den Schlüssel  $e$  über einen sicheren Kanal austauschen und dann geheim halten. Der Geheimtextaustausch erfolgt über einen unsicheren Kanal. Unter einem unsicheren Kanal könnte man beispielsweise eine Postkarte verstehen. Ein sicherer Kanal wäre zum Beispiel ein versiegelter Brief.

### 3.1 Klassische symmetrische Verschlüsselung

Die Kryptografie hat eine weitreichende Geschichte. Kryptografische Verfahren wurden hauptsächlich in militärischen und diplomatischen Bereichen eingesetzt, um Informationen vertraulich übermitteln zu können. Das Ziel war schon damals wie heute die Überbringung von geheimen Botschaften.

Eines der ersten überlieferten Verschlüsselungsverfahren ist die Skytale von Sparta vor ca. 2500 Jahren.

#### 3.1.1 Skytale von Sparta (ca. 500 v. Chr.)

Mithilfe von Zylindern, sog. Skytale, übermittelte die spartanische Regierung geheime Nachrichten an ihre Generäle. Sowohl Sender als auch Empfänger besaßen einen Zylinder mit genau dem gleichen Durchmesser. Der Sender wickelte ein Pergamentband spiralförmig um den Zylinder und schrieb die Nachricht in Längsrichtung auf das Pergamentpapier. Nach dem Entfernen des Bandes befand sich die Nachricht in chiffrierter Form auf dem Papierstreifen. Nur eine Person, die eine Skytale mit gleichem Durchmesser hatte, konnte die Nachricht dechiffrieren, indem Sie das Band auf den Zylinder wieder aufwickelte.



Abbildung 3: Skytale von Sparta [19]

Die Skytale war somit der Schlüssel, mit dem man die Nachricht wieder entschlüsseln konnte. Dieses Verfahren wird als sog. *Transpositions-Chiffrierung* bezeichnet, bei der der Geheimtext eine Permutation der Buchstaben des Klartextes ist.

Die Verschlüsselung nach dem Skytale Prinzip, kann auch mithilfe eines einfachen Vorgehens, ohne Zylinder, durchgeführt werden.

Hierfür wird der Klartext zeilenweise in genau  $n$  Zeilen geschrieben. Durch spaltenweises Auslesen des Textes erhält man den Geheimtext. Um diesen dechiffrieren zu können, benötigt man den Schlüssel  $n$ , also hier die Anzahl der Zeilen. Man schreibt den Geheimtext spaltenweise mit genau  $n$  Zeilen auf und erhält dann durch zeilenweises

lesen den Klartext. Nachstehende Grafik zeigt die Entschlüsselung eines Geheimtextes mit verschiedenen Werten für  $n$ , wobei  $n=7$  den korrekten Schlüssel darstellt.

**Geheimtext:** VUEELYAELIIDPFRIITNETITCIZROERHSIKG.AKTERR

Entschlüsseln mit richtigem Schlüssel  $n = 7$ :

**Leserichtung  
Klartext**

→

<b>Leserichtung Geheimtext</b>	V	E	R	T	R	A
	U	L	I	C	H	K
	E	I	T	I	S	T
	E	I	N	Z	I	E
	L	D	E	R	K	R
	Y	P	T	O	G	R
	A	F	I	E	.	

Entschlüsseln mit falschem Schlüssel  $n = 6$ :

V	A	P	E	Z	S	K
U	E	F	T	R	I	T
E	L	R	I	O	K	E
E	I	I	T	E	G	R
L	I	T	C	R	.	R
Y	D	N	I	H	A	

Abbildung 4: Entschlüsselung der Skytale von Sparta [4]

Wird der Geheimtext mit dem richtigen Schlüssel ( $n = 7$ ) entschlüsselt, so erhält man nachstehenden Klartext:

**VERTRAULICHKEITISTEINZIELDERKRYPTOGRAPHIE.**

### 3.1.2 Cäsar-Chiffrierung (ca. 100 v. Chr.)

Ein weiteres bekanntes historisches Verfahren ist die Cäsar-Chiffrierung, benannt nach dem römischen Kaiser und Staatsmann Julius Cäsar. Für die geheime Überbringung von Botschaften hat Julius Cäsar die *Substitutions-Chiffrierung* entwickelt. Die Buchstaben des zu verschlüsselnden Textes wurden um 3 Positionen im Alphabet nach links verschoben und mit den neuen Buchstaben substituiert, d. h., aus A wird ein D, aus B wird ein E und so weiter. Folgende Tabelle ergibt sich aus dem erläuterten Verfahren:

Klartextalphabet:	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Geheimtextalphabet:	D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

Klartext:	C A E S A R
Geheimtext:	F D H V D U

Tabelle 1: Die Cäsar-Chiffrierung [4]

Wird beispielsweise der Klartext *CAESAR* chiffriert, so entsteht als Geheimtext *FDHVDU* (vgl. Tabelle 1). Um den Geheimtext dechiffrieren zu können, müssen die Buchstaben um je 3 Positionen im Alphabet nach rechts verschoben werden. Der Schlüssel, also die Anzahl 3, um welche die Buchstaben verschoben werden, ist von Cäsar rein zufällig gewählt worden. Da unser Alphabet aus eigentlich 26 Buchstaben besteht, gibt es 26 Möglichkeiten der Chiffrierung, wobei nur 25 davon nutzbar sind. Die Chiffrierung mit der Verschiebung um null Positionen ist für geheime Botschaften nicht zu gebrauchen, da dann Klartext- und Geheimtext-Alphabet identisch sind.

Die *Substitutions-Chiffrierung* ist zusammen mit der *Transpositions-Chiffrierung* noch heute die Basis für die eingesetzten symmetrischen Verschlüsselungsmethoden.

## 3.2 Stromchiffren

Stromchiffren sind Verschlüsselungsverfahren, bei denen jedes Zeichen des Klartextes einzeln verschlüsselt wird. Die Chiffrierung kann sich dabei für jedes Zeichen ändern. Stromchiffren haben ihren Vorteil in der kontinuierlichen Verschlüsselung des Klartextes, weil hierbei keine Verzögerungen auftreten. Des Weiteren kann bei Stromchiffren, aufgrund der einzelnen Verschlüsselung der Zeichen, keine Ausbreitung von Fehlern erfolgen. Stromchiffren werden in der Praxis in den Bereichen eingesetzt, bei denen ein kontinuierlicher Datenstrom von Bedeutung ist. Dies kann beispielsweise der Fall sein, wenn kein Speicher vorhanden oder der Daten-Puffer begrenzt ist. Die Mobiltelefonie ist dabei eines der wichtigsten Anwendungsgebiete.

Die Umsetzung von Stromchiffren erfolgt mithilfe linear rückgekoppelter Schieberegister, sog. LFSR (Linear Feedback Shift Register). Diese sind als Hardware leicht zu implementieren und arbeiten effizient.

Linear rückgekoppelte Schieberegister sollen anhand nachstehender Abbildung verdeutlicht werden.

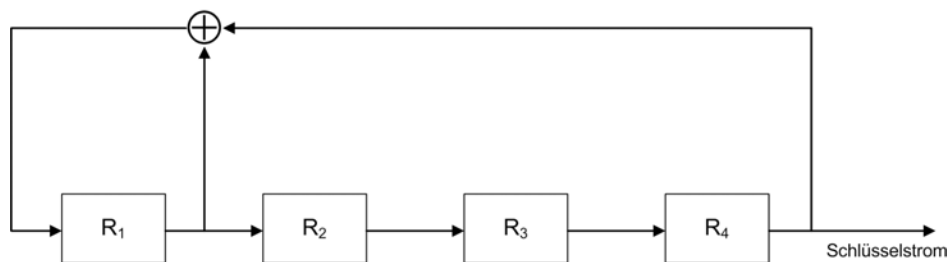


Abbildung 5: System aus vier linear rückgekoppelten Schieberegistern [1]

Das System besteht aus den Schieberegistern  $R_1$ ,  $R_2$ ,  $R_3$  und  $R_4$ . Der Wert von  $R_4$  bestimmt dabei den aktuellen Zufallswert des Schlüsselstroms.

Die Inhalte der jeweiligen Register werden für den Zeitpunkt  $t + 1$  nach rechts in das Nachbarregister geschoben, d. h.,  $R_2$  erhält dann den Wert von  $R_1$ ,  $R_3$  den von  $R_2$  und  $R_4$  erhält den Wert von  $R_3$ . Der neue Inhalt von  $R_1$  wird durch eine XOR-Verknüpfung der Werte aus den Registern  $R_1$  und  $R_4$  bestimmt.

Nachstehende Tabelle zeigt die Belegung der vier Register für 16 Zeiteinheiten, wenn für den Startwert  $[0, 1, 1, 0]$  gilt.

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$R_1$	0	0	1	0	0	0	1	1	1	1	0	1	0	1	1	0
$R_2$	1	0	0	1	0	0	0	1	1	1	1	0	1	0	1	1
$R_3$	1	1	0	0	1	0	0	0	1	1	1	1	0	1	0	1
$R_4$	0	1	1	0	0	1	0	0	0	1	1	1	1	0	1	0

Tabelle 2: Belegung der Register der Abb. 5 für 16 Zeiteinheiten [1]

Bei einem System aus  $n$  Registern wiederholt sich die Bitfolge nach  $2^n - 1$  Perioden. In unserem Beispiel mit  $n = 4$  Registern wiederholt sich die Bitfolge also nach 15 Durchgängen. Ein Startwert ausschließlich mit Nullen ist ungeeignet, da sich die Registerinträge nicht ändern würden. Die Länge des Schlüssels für die Chiffrierung entspricht der Anzahl der Register. Im angegebenen Beispiel besteht der Schlüssel aus vier Bit.

### 3.3 A5-Algorithmus

Ein wichtiger Stromchiffrier-Algorithmus, der auf linear rückgekoppelte Schieberegister basiert, ist der A5-Algorithmus. Der A5-Algorithmus ver- und entschlüsselt digitale Sprachdaten für die Funkübertragung im *Global System for Mobile Communications*-Netz. Auf dem Markt bestehen zwei Varianten der Verschlüsselung: A5/1 und A5/2. Die bessere Version der Beiden ist A5/1, die unter anderem in Europa eingesetzt wird [9].

Im Folgenden wird daher ausschließlich auf die A5/1-Version eingegangen.

Bei der A5/1-Variante werden verschiedene linear rückgekoppelte Schieberegister - mit 19, 22 und 23 Registern - verknüpft. Durch diese Kombination ergibt sich eine Schlüssellänge von 64 Bit. Der Schlüssel ist auf der SIM-Karte gespeichert und wird dadurch über einen sicheren Kanal übertragen, da der Kunde diese direkt vom Mobilfunkbetreiber erhält.

Für jede Zeiteinheit erzeugen die linear rückgekoppelten Schieberegister jeweils ein Zufallsbit  $s_1$ ,  $s_2$  und  $s_3$ . Jedes linear rückgekoppelte Schieberegister besitzt ein sog. *Clocking-Tap*-Register: Register 8 für  $R_1$  und jeweils Register 10 für  $R_2$  und  $R_3$ . Anhand der Inhalte der jeweiligen *Clocking-Tap*-Register werden die Zufallsbits  $s_1$ ,  $s_2$  und  $s_3$  miteinander XOR verknüpft und erzeugen den aktuellen Zufallswert des Schlüsselstroms.

Die schematische Funktionsweise des A5/1 Standards ist in nachstehender Abbildung wiedergegeben.

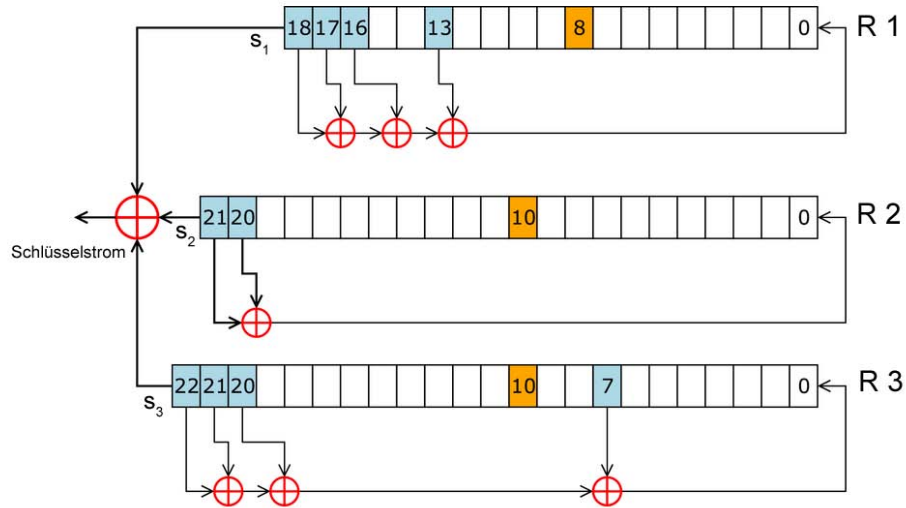


Abbildung 6: Funktionsweise der A5/1-Verschlüsselung [20]

Mit dem erzeugten Schlüsselstrom werden anschließend die digitalisierten Sprachdaten für die Funkübertragung in der Mobiltelefonie verschlüsselt.

### 3.4 Blockchiffren

Blockchiffren sind Verschlüsselungsverfahren, bei denen der Klartext in Blöcke fester Länge  $n$  unterteilt wird und dann chiffriert wird. Die Anzahl der Zeichen in einem Block wird als Blocklänge bezeichnet. Viele bekannte symmetrische Verschlüsselungsverfahren sind Blockchiffren, wie z. B. *Data Encryption Standard* (DES) und *Advanced Encryption Standard* (AES).

### 3.5 Advanced Encryption Standard

Das National Institute of Standards and Technology (NIST) gab im November 2001 den *Advanced Encryption Standard* offiziell als Standard bekannt. Dieser wurde von Joan Daemen und Vincent Rijmen entwickelt und beruht auf einer symmetrischen Blockchiffrierung mit einer festgelegten Blocklänge von 128 Bit. Der AES-Algorithmus hat eine variable Schlüssellänge von 128 (AES-128), 192 (AES-192) oder 256 (AES-256) Bit. Die Anzahl der Runden, die bei der Ausführung des Algorithmus durchgeführt werden, hängt von der Länge der Schlüssel ab und liegt bei 10 (AES-128), 12 (AES-192) oder 14 (AES-256) Runden.

	<b>Schlüssellänge [Wort]</b>	<b>Blocklänge [Wort]</b>	<b>Anzahl der Runden</b>
<b>AES-128</b>	4	4	10
<b>AES-192</b>	6	4	12
<b>AES-256</b>	8	4	14

Abbildung 7: AES: Schlüssel-Block-Runden-Zusammenhang [11]

Eine Runde besteht dabei aus vier einzelnen Operationen. Diese werden nacheinander auf den in 128-Bit-Blöcken unterteilten Klartext angewendet. Weiterhin wird in jeder Runde ein Rundenschlüssel erzeugt. Im Folgenden werden die Grundoperationen der Reihenfolge nach aufgelistet und näher erläutert:

- SubBytes()
- ShiftRows()
- MixColumns()
- AddRoundKey()

In der *SubBytes()*-Operation wird jedes Byte mithilfe einer Substitutionstabelle, sog. S-Box, ersetzt.

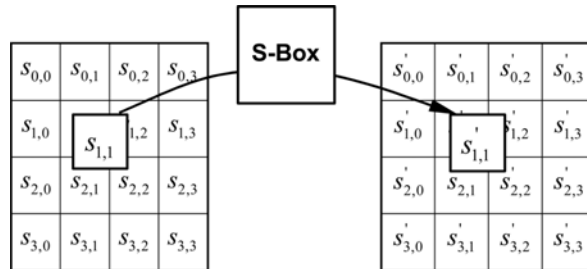


Abbildung 8: Schematische Darstellung der Operation *SubBytes()* [11]

Soll beispielsweise  $s_{1,1} = 42$  substituiert werden, so wird der neue Wert durch den Schnittpunkt der Reihe mit dem Index 4 und der Spalte mit dem Index 2 ermittelt. Damit ergibt sich für  $s'_{1,1}$  der Wert  $2c$ .

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	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

Abbildung 9: S-Box im Hexadezimal-Format [11]

Die *ShiftRows()*-Transformation rotiert die letzten drei Reihen zyklisch mit unterschiedlichen Werten. Die 2. Reihe wird um ein Byte rotiert, die 3. Reihe um 2 Byte und die 4. Reihe um 3 Byte. Die erste Reihe wird nicht verschoben.

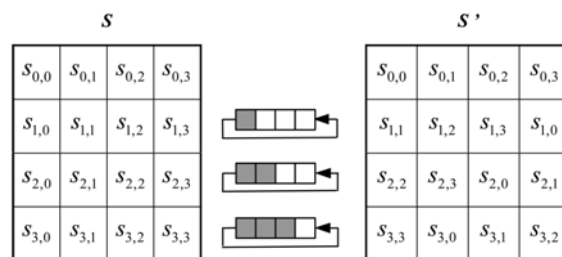


Abbildung 10: Schematische Darstellung der Operation *ShiftRows()* [11]

Die *MixColumns()*-Transformation multipliziert die einzelnen Spalten mit einer bekannten Matrix  $M$ , wobei die einzelnen Zellen miteinander multipliziert und die Ergebnisse anschließend mit XOR verknüpft werden. Die Multiplikation entspricht einer *Finite-Field-Multiplikation*.

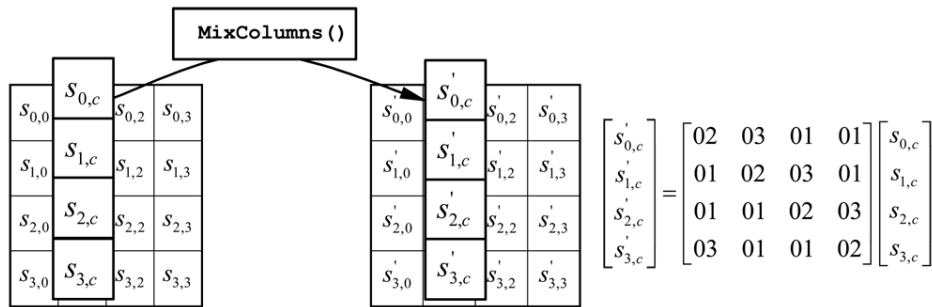


Abbildung 11: Schematische Darstellung der *MixColumns()*-Operation mit bekannter Matrix  $M$  [11]

In der *AddRoundKey()*-Operation findet die eigentliche Verschlüsselung statt. Hier wird der jeweilige Rundenschlüssel mit dem Block bitweise XOR verknüpft.

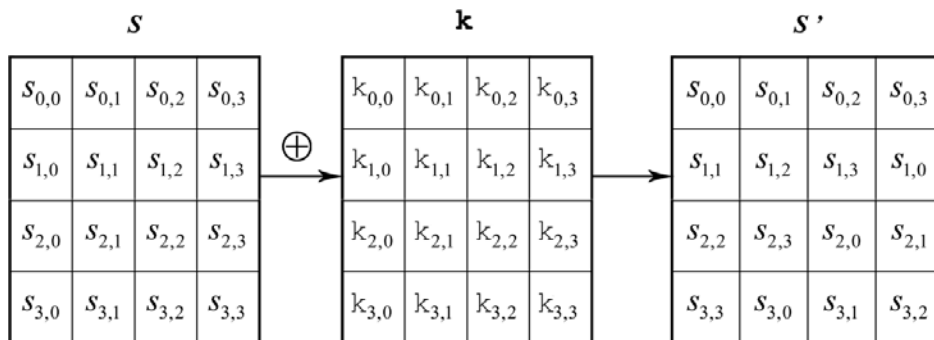


Abbildung 12: Schematische Darstellung der *AddRoundKey()*-Operation [11]

Der AES-Algorithmus wird unter anderem zum Verschlüsseln von Daten in Wireless-LAN genutzt und ist daher im IEEE 802.11i Standard implementiert.

## 4 Asymmetrische Verschlüsselung

### 4.1 Einführung

Bis zur Erfindung von asymmetrischer Verschlüsselung ging man davon aus, dass Sender und Empfänger einen gemeinsamen (geheimen) Schlüssel zum Verschlüsseln und Entschlüsseln benötigen. Bei asymmetrischen Verfahren ist der Verschlüsselungsschlüssel  $e$  nicht gleich dem Entschlüsselungsschlüssel  $d$ . Jedem Teilnehmer werden ein *privater Schlüssel*  $d$  und ein *öffentlicher Schlüssel*  $e$  zugeordnet, wobei nur der *private Schlüssel*  $d$  vom Besitzer geheim zu halten ist.

Die Chiffrierung mithilfe asymmetrischer Verfahren wird anhand nachstehender Grafik verdeutlicht.

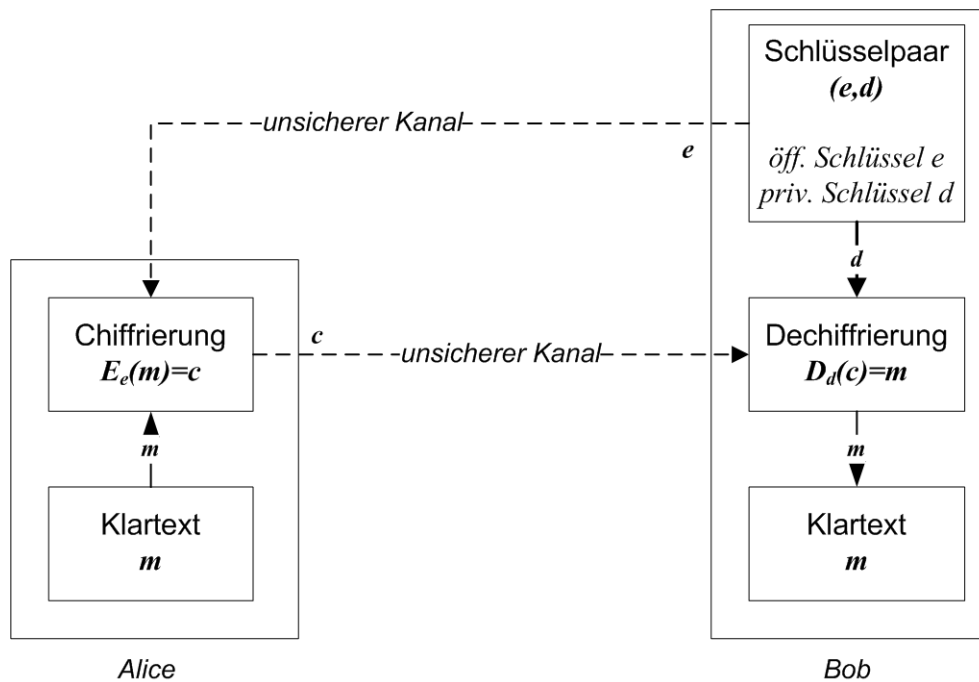


Abbildung 13: Zwei-Teilnehmer-Kommunikation mittels asymmetrischer Verschlüsselung [1]

Bob ist im Besitz des Schlüsselpaars  $(e, d)$ . Damit Alice an Bob eine geheime Nachricht schicken kann, muss Sie den *öffentlichen Schlüssel*  $e$  von Bob kennen. Bob sendet daher Alice den *öffentlichen Schlüssel*  $e$  über einen unsicheren Kanal, behält aber den *privaten Schlüssel*  $d$ . Alice chiffriert den Klartext  $m$  mit Bobs *öffentlichen Schlüssel*  $e$  und kann dann den Geheimtext  $c$  über einen unsicheren Kanal an Bob senden. Bob dechiffriert den Geheimtext mithilfe seines *privaten Schlüssel*  $d$  und erhält dadurch die Nachricht  $m$ . Der Unterschied zum symmetrischen Szenario in Abb. 2 liegt im unsi-

chere Kanal, über den der *öffentliche Schlüssel*  $e$  verschickt wird. Da  $e$  nicht geheim gehalten werden muss, kann  $e$  beispielsweise per E-Mail individuell verteilt werden oder auch öffentlich auf sog. *Key-Servern* abgelegt werden. Somit kann jede Entität eine verschlüsselte Nachricht an Bob schicken, die aber nur Bob entschlüsseln kann. Algorithmen, die nach dem beschriebenen asymmetrischen Szenario arbeiten, sind unter anderem die *Rabin*-, *RSA*- und *Elgamal*-Verschlüsselung. Das *RSA*-Verfahren ist heute das Wichtigste der genannten Verfahren und wird daher im nächsten Kapitel näher behandelt.

## 4.2 RSA-Verschlüsselung

Ronald Rivest, Adi Shamir und Leonard Adleman entwickelten 1977 ein neues kryptografisches Verfahren, welches nach den Anfangsbuchstaben ihrer Nachnamen benannt wurde: *RSA*. Der *RSA*-Algorithmus ist heutzutage das am weitesten verbreitete asymmetrische Verfahren und Teil vieler Standards wie z. B. *Pretty Good Privacy* (PGP), *Digital Signature Standard* (DSS) und *Secure Electronic Transaction-Spezifikation* von Visa und MasterCard.

Der *RSA*-Algorithmus nutzt das Faktorisierungsproblem großer Zahlen aus: Es ist einfach zwei große Primzahlen miteinander zu multiplizieren, jedoch sehr schwierig aus deren Produkt die verwendeten Primzahlen zu ermitteln.

Die Generierung eines Schlüsselpaars  $(e, d)$  erfolgt anhand des nachstehenden Algorithmus [1]:

1. Wähle zwei große Primzahlen  $p$  und  $q$
2. Berechne  $n = p \cdot q$
3. Berechne  $\phi = (p - 1) \cdot (q - 1)$
4. Bestimme eine Zufallszahl  $e$  mit  $1 < e < \phi$ , sodass  $e$  teilerfremd zu  $\phi$  ist
5. Bestimme mithilfe des erweiterten euklidischen Algorithmus  $d$  mit  $1 < d < \phi$ , sodass  $e \cdot d \equiv 1 \pmod{\phi}$
6. Verwende  $(n, e)$  als öffentlichen Schlüssel und  $d$  als privaten Schlüssel

Für die Auswahl von  $p$  und  $q$  ist zu beachten, dass die beiden Primzahlen eine gleiche Größenordnung besitzen, dennoch sollten Sie nicht zu dicht beieinanderliegen.

Die Verschlüsselung des Klartextes geschieht in zwei Schritten [1]:

1. Stelle den Klartext  $m$  als Integerwerte im Intervall  $[0, n - 1]$  dar
2. Berechne den Geheimtext  $c = m^e \bmod n$  mithilfe des *öffentlichen Schlüssels*  $(n, e)$

Die Entschlüsselung des Geheimtextes kann mit folgendem Schritt durchgeführt werden [1]:

- Berechne  $m = c^d \bmod n$  mithilfe des *privaten Schlüssels*  $d$

Für  $n = 221$ ,  $e = 71$  und  $d = 119$  wird beispielsweise die Nachricht  $m = 8$  chiffriert und dechiffriert [15].

- Verschlüsselung:  $c = 8^{71} \bmod 221 = 83$
- Entschlüsselung:  $m = 83^{119} \bmod 221 = 8$

Das durch den *RSA*-Algorithmus erzeugte Schlüsselpaar aus öffentlichem und privatem Schlüssel kann nicht nur zum ver- und entschlüsseln verwendet werden, sondern auch für die Erzeugung einer digitalen Signatur. Digitale Signaturen garantieren die Authentizität und Integrität von Informationen.

Die Signatur wird mithilfe des privaten Schlüssels  $d$  erstellt, indem eine Nachricht  $m$  signiert wird [4]:

- $sig = m^d \bmod n$

Die Verifizierung kann dann durch den öffentlichen Schlüssel  $e$  des Signaturinhabers erfolgen [4]:

- $m = sig^e \bmod n$

Damit *RSA* sicher ist, sollte laut dem Bundesamt für Sicherheit in der Informationstechnik (BSI) für  $n$  eine Mindestlänge von derzeit 1024 Bit gewählt werden. Empfohlen ist sogar eine Länge von 2048 Bit. Nachstehende Tabelle fasst die Mindestanforderungen für  $n$  bis zum Jahr 2012 zusammen. Eine jahresweise Staffelung ist nötig, da mit zunehmender technischer Entwicklung auch die Möglichkeiten der Kryptoanalyse zunehmen.

Zeitraum Parameter	bis Ende 2007	bis Ende 2008	bis Ende 2009	bis Ende 2010	bis Ende 2012
$n$	1024 (Mindestw.)	1280 (Mindestw.)	1536 (Mindestw.)	1728 (Mindestw.)	1976 (Mindestw.)
	2048 (Empf.)	2048 (Empf.)	2048 (Empf.)	2048 (Empf.)	2048 (Empf.)

Abbildung 14: Mindestlängenforderung für  $n$  bis 2012 nach BSI [16]

## 5 Symmetrisch vs. Asymmetrisch

Sowohl symmetrische als auch asymmetrische Verfahren haben Vor- und Nachteile.

- Vorteile symmetrischer Verschlüsselung:
  1. Symmetrische Verschlüsselung kann sowohl als Hardware, als auch als Software implementiert werden.
  2. Die Schlüssel sind im Vergleich zu asymmetrischer Verschlüsselung relativ kurz.
  3. Symmetrische Verschlüsselung hat eine weit greifende Geschichte und ist daher besser erforscht als asymmetrische Verschlüsselung.
  
- Nachteile symmetrischer Verschlüsselung:
  1. Sender und Empfänger müssen den Schlüssel geheim halten.
  2. Aus Sicherheitsgründen ist ein häufiges Ändern des Schlüssels notwendig.
  3. Der Schlüssel muss über einen sicheren Kanal übertragen werden.
  4. In einem großen Netzwerk muss eine große Anzahl an Schlüsseln verwaltet werden.
  
- Vorteile asymmetrischer Verschlüsselung:
  1. Nur der private Schlüssel muss geheim gehalten werden.
  2. Ein Schlüsselpaar aus öffentlichem und privatem Schlüssel kann über einen längeren Zeitraum verwendet werden.
  3. In einem großen Netzwerk ist die Anzahl der Schlüssel im Vergleich zu symmetrischer Verschlüsselung relativ gering.
  
- Nachteile asymmetrischer Verschlüsselung:
  1. Asymmetrische Verschlüsselung muss im Vergleich zu symmetrischer Verschlüsselung einen um den Faktor 10 oder mehr längeren Schlüssel besitzen, damit eine äquivalente Sicherheit gewährleistet ist.
  2. Asymmetrische Verschlüsselung existiert erst seit den 70er Jahren und ist daher weniger ausgeprägt.
  3. Die Durchsatzrate ist bis zu einigen Zehnerpotenzen langsamer als bei gleichwertiger symmetrischer Verschlüsselung.

## 6 Hybride Verschlüsselung

### 6.1 Einführung

Symmetrische und asymmetrische Verschlüsselung werden in der Praxis häufig zu sogenannten *hybriden Verfahren* kombiniert und nutzen dadurch die Vorteile beider Verschlüsselungsarten aus. Bei hybriden Verfahren wird ein symmetrischer Sitzungsschlüssel erzeugt, mit dem der Klartext  $m$  verschlüsselt wird. Dadurch wird eine schnelle symmetrische Verschlüsselung der Daten mit einem kurzen Schlüssel erreicht. Den Sitzungsschlüssel verschlüsselt man anschließend mithilfe eines asymmetrischen Verfahrens, wobei dieser dann über einen unsicheren Kanal übermittelt werden kann.

### 6.2 Digital Rights Management

Eine Anwendung für hybride Verschlüsselung ist *Digital Rights Management* (DRM). Durch dieses Verfahren kann die Distribution digitaler Medieninhalte, wie z. B. Video- und Audioinhalte, kontrolliert werden. Die Funktionsweise von Digital Rights Management wird nun anhand *Open Mobile Alliance DRM 2.1* (OMA DRM 2.1) näher erläutert.

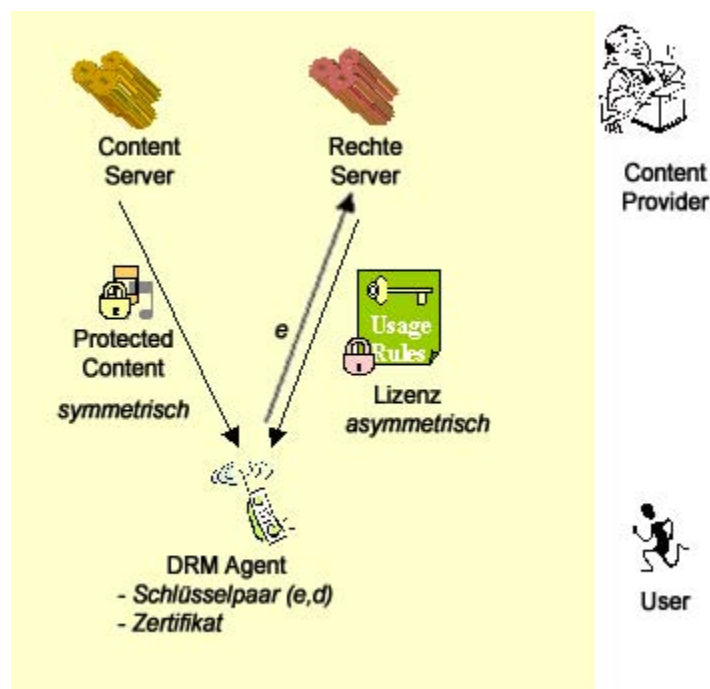


Abbildung 15: *Open Mobile Alliance DRM 2.1* [17]

Der Nutzer fordert eine Mediendatei (z. B. MP3-File) vom Content Provider an. Der Content Provider besitzt hierfür einen Content Server, der die angeforderte Datei symmetrisch verschlüsselt.

Der Nutzer erhält die verschlüsselte Datei auf seinem Endgerät, dem sog. DRM-Agent. Dieser DRM-Agent besitzt ein Schlüsselpaar  $(e, d)$  aus öffentlichem und privatem Schlüssel. Damit der User den geschützten Content entschlüsseln kann, benötigt er eine dazugehörige Lizenz vom Rechte Server des Content Providers.

Der Rechte Server verschlüsselt die Lizenz asymmetrisch mit dem öffentlichen Schlüssel  $e$  des DRM-Agents und übermittelt diese anschließend an den Agent. Mithilfe seines privaten Schlüssels  $d$  kann der DRM-Agent die Lizenz entschlüsseln. Die Lizenz ist ein XML-Dokument und enthält sowohl Nutzungsrechte für den Content als auch den symmetrischen Schlüssel, mit dem die Mediendatei verschlüsselt wurde.

Ein DRM-Agent besitzt zusätzlich ein Zertifikat. Dieses Zertifikat enthält zusätzliche Informationen wie z. B. Hersteller, Gerätetyp, Softwareversion und Seriennummer. Durch dieses Zertifikat ist es möglich, Authentizität und Integrität zu gewährleisten. *OMA DRM 2.1* verwendet AES-128 für die symmetrische Verschlüsselung und das RSA-Verfahren für die asymmetrische Verschlüsselung, um die Vertraulichkeit der Daten zu sichern.

## 7 Fazit

Mit der wachsenden Digitalisierung nimmt die Kryptografie einen immer wichtiger werdenden Platz in unserem Alltag ein. Durch die zunehmende Verbreitung von Medieninhalten, wie z. B. Audio- und Videoinhalte im World Wide Web, wächst das Verlangen der Urheber und Contentprovider diese Werke zu schützen. Mit kryptografischen Verfahren ist es möglich, digitale Medieninhalte auf einem kontrollierten Weg den Nutzern zur Verfügung zu stellen.

Kryptografische Verfahren, die heutzutage als sicher angesehen sind, können in einigen Jahren längst veraltet sein. Nur durch laufende Verbesserung bewährter Standards und Entwicklung neuer Verfahren können Vertraulichkeit, Authentizität, Integrität und Zurechenbarkeit digitaler Daten durch die Kryptografie gewährleistet werden.

## Literatur

- [1] A. Menezes, P. Oorschot, S. Vanstone:  
„*Handbook of Applied Cryptography*“  
CRC Press 1997
- [2] S. Singh:  
„*Geheime Botschaften*“  
Carl Hanser Verlag 2000
- [3] C. Eckert:  
„*IT-Sicherheit Konzepte-Verfahren-Protokolle*“  
Oldenbourg Verlag, 3. Auflage
- [4] A. Beutelspacher:  
„*Kryptologie*“  
Vieweg Verlag 2005
- [5] A. Beutelspacher:  
„*Moderne Verfahren der Kryptographie*“  
Vieweg Verlag 2004
- [6] K. Schmech:  
„*Kryptografie*“  
dpunkt Verlag, 2. Auflage
- [7] J. Buchmann:  
„*Einführung in die Kryptographie*“  
Springer Verlag 2004
- [8] D. Stinson:  
„*Cryptography*“  
CRC Press 2006
- [9] A. Biryukov, A. Shamir, D. Wagner:  
„*Real Time Cryptanalysis of A5/1 on a PC*“  
<http://cryptome.org/a5.ps>
- [10] E. Zenner, R. Weis, S. Lucks:  
„*Datenschutz und Datensicherheit: Sicherheit des GSM-Verschlüsselungsstandards A5*“,  
2000  
<http://www.cryptolabs.org/gsm/ZennerWeisLucksA5.pdf>

- [11] National Institute of Standards and Technology:  
„*FIPS PUB 197 Advanced Encryption Standard*“, 2001  
<http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>
- [12] Institute of Electrical and Electronics Engineers:  
„*IEEE Standard 802.11i*“, 2004  
<http://standards.ieee.org/getieee802/portfolio.html>
- [13] RSA Laboratories:  
„*PKCS #1 v2.1: RSA Cryptography Standard*“, 2002  
<ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-1/pkcs-1v2-1.pdf>
- [14] F. Bourseau, D. Fox, C. Thiel:  
„*Datenschutz und Datensicherheit: Vorzüge und Grenzen des RSA-Verfahrens*“, 2002  
<http://www.secorvo.de/publikationen/rsa-grenzen-fox-2002.pdf>
- [15] Finn Sohst:  
„*Verschlüsselung mit dem RSA-Verfahren*“  
<http://www.informatik.uni-hamburg.de/WSV/teaching/sonstiges/EwA-Folien/Sohst-Paper.pdf>
- [16] Bundesamt für Sicherheit in der Informationstechnik:  
„*Geeignete Algorithmen zur Erfüllung der Anforderungen nach §17 Abs. 1 bis 3 SigG vom 22. Mai 2001 in Verbindung mit Anlage 1 Abschnitt I Nr. 2 SigV vom 22. November 2001*“, 2006  
[http://www.bsi.bund.de/esig/dokumente/krypto/algo\\_entw2\\_07.pdf](http://www.bsi.bund.de/esig/dokumente/krypto/algo_entw2_07.pdf)
- [17] Open Mobile Alliance:  
„*DRM Architecture Draft Version 2.1*“, 04. Mai 2007  
[http://www.openmobilealliance.org/tech/wg\\_committees/drm.html](http://www.openmobilealliance.org/tech/wg_committees/drm.html)
- [18] Open Mobile Alliance:  
„*DRM Specification Draft Version 2.1*“, 25. Mai 2007  
[http://www.openmobilealliance.org/tech/wg\\_committees/drm.html](http://www.openmobilealliance.org/tech/wg_committees/drm.html)
- [19] „<http://de.wikipedia.org/wiki/Bild:Skytala£26EmptyStrip-Shaded.png>“
- [20] „<http://de.wikipedia.org/wiki/Bild:A5-1.png>“