Rev 2011-E2

Rolf Künzler, Jürg Füglistaler

INHALTSVERZEICHNIS

1 Code, Codierung

1.1 Definition

Definition nach DIN 44300: Ein Code ist eine Vorschrift für die eindeutige Zuordnung der Zeichen eines Zeichenvorrats zu den Zeichen eines anderen Zeichenvorrats.

Ein Code ist also eine Zuordnungsvorschrift zwischen zwei Zeichenmengen. Den Vorgang der Zuordnung nennt man Codierung bzw Decodierung. Zweck einer Codierung ist die Umformung einer gegebenen Zeichenmenge in eine nach bestimmten Kriterien geeignetere Zeichenmenge.

Diese Kriterien folgen aus den jeweiligen technischen Problemen bei der Übertragung, Verarbeitung, Speicherung und Geheimhaltung von Nachrichten bzw Daten.

Die Auswahl eines geeigneten Codes ist abhängig von den Forderungen, die an die binären Daten bezüglich ihrer Speicherung, Übertragung oder Verarbeitung gestellt werden.

1.1.1 Beispiele

- Codes zur effektiven Rechnung (z.B. Dualcode)
- Codes zur Datenübertragung und Speicherung (prüfbare Codes)
- Codes zur Messwertaufnahme (Bsp. Gray-Code)
- Alphabetische Codes (Bsp ASCII)
- Barcodes, Strichcodes
  (Bsp EAN: European Article Number ⇨ wird im Produktehandel gebraucht)

1.2 Codeklassen

Die Codes werden allgemein unterteilt in:

<table>
<tr><td colspan="6">Binärcodes für Dezimalzahlen</td></tr>
<tr><td colspan="2">Wortcodes</td><td colspan="4">Zifferncodes</td></tr>
<tr><td>Gewichtete Codes (g)</td><td>Anordnungs-Codes (a)</td><td colspan="2">Tetradische Codes (4 bit)</td><td>m aus n -Codes</td><td>Sonstige Codes</td></tr>
<tr><td>Mehrschrittig</td><td>Einschrittig</td><td>Mehrschrittig</td><td>Einschrittig</td><td></td><td></td></tr>
<tr><td>g Dual-Code</td><td>a Gray-Code</td><td>g 8421-Code
g White
g Aiken
g Jumpat2
a Stibitz (Exzess3)</td><td>a Glixon
a Tomkins
a O’Brian
a Reflektierter Exzess3</td><td>g 1-aus-10
g 2-aus-7 Biquinär
g 2-aus-7 Quibinär
a 2-aus-7 Walking</td><td>a Libaw-Craig-Code (Johnson-Code)</td></tr>
</table>

1.3 Gewichteter Code

Jeder Stelle des Codes kann ein Gewicht zugeordnet werden, die codierte Zahl ist dann die Summe aller Gewichte der Stellen, die den Wert 1 haben (Bsp BCD-Code 8421).

Achtung:

Bei gleicher Gewichtung von verschiedenen Stellen wird von der kleinstwertigsten Stelle (LSB: Least significant Bit) zur grösstwertigsten Stelle (MSB: Most significant Bit) gezählt (Bsp 4221-Code).

1.4 Anordnungscode

Den Stellen des Codes kann kein Gewicht zugeordnet werden.

1.5 Wortcode

Die zu codierende Zahl wird als ganzes Wort codiert. Nachteil: Die Ein- und Ausgabe ist aufwendig. Vorteil: Rechenoperationen sind einfach durchführbar. Meist wird der reine Binärcode (Dualcode) verwendet.

1.6 Zifferncode

Die zu codierende Zahl wird nicht als ganzes codiert, sondern ziffernweise. Man bezeichnet diese Codes als BCD-Codes (Binary-Coded-Decimal), sie codieren die Ziffern 0 bis 9.

1.7 Schrittigkeit

Bei **einschrittigen** Codes unterscheiden sich die benachbarten Codes jeweils nur in einer Stelle, bei **mehrschrittigen** in mehreren Stellen.

1.8 Redundanz

Redundanz bedeutet im Zusammenhang mit der Codierung, dass ein Code mehr Kombinationsmöglichkeiten besitzt als zur Codierung benutzt werden. Mit einem Code von n Bit Länge können binär 2^n Zustände (Kombinationen) dargestellt werden.

Beispiel Länge 4 Bit ⇨ maximal $2^4 = 16$ Zustände unterscheidbar

Wird der BCD Code verwendet, also ein Zifferncode für die Zahlen 0 bis 9 (dezimal) so ist der Code redundant, denn es werden nur 10 Zustände benötigt, aber 16 sind vorhanden.

1.9 Bewertbarkeit

Will man eine binär codierte Zahl in eine analoge Grösse umwandeln, so ist dies besonders einfach, wenn jeder Binärstelle ein Gewicht oder eine Wertigkeit zugeordnet ist.

1.10 Codebeispiele

1.10.1 BCD Code (Gewichtung 8421)

		CODE			
G		8	4	2	1
0		0	0	0	0
1		0	0	0	1
2		0	0	1	0
3		0	0	1	1
4		0	1	0	0
5		0	1	0	1
6		0	1	1	0
7		0	1	1	1
8		1	0	0	0
9		1	0	0	1

Der BCD-Code wird in der Informatik zur binären Darstellung von Zahlen eingesetzt. Er basiert auf dem Binärsystem (Zahlensystem der Basis 2). Dieser Code ist Ihnen bereits aus dem Kapitel Zahlensysteme bekannt.

1.11 Analog-Digital-Wandlung (ADC: Analog-to-Digital-Conversion)

In der Messtechnik werden Messwerte häufig durch Sensoren analog (als Spannungswerte) erfasst und müssen zur Weiterverarbeitung in Steuerung und Regelung in einen digital codierten Wert umgewandelt werden.

1.11.1 Umrechnung

Ein Sensor liefert eine Spannung im Bereich von 0 bis 1V. Für jeden umgewandelten Wert stehen 10 Bit zu Verfügung, dh es sind Werte zwischen 0 und 1023 (2^{10} Werte) möglich. Wenn die kleinste Einheit (1Bit) der Grösse 1mV entspricht, dann können Werte von 0V bis 1.023V dargestellt werden (0 bis 1023). Binär sind das die Werte 0000000000 bis 1111111111.

1.11.2 Aufgabe

Ein Signal kann Werte zwischen 0V und 6V annehmen. Wir möchten das Signal mit einer Genauigkeit von 10mV auflösen und binär codieren. Wieviele Werte sind in unserem Wertebereich? Wieviele Bit benötigen wir für einen Wert?

2 Vorzeichenbehaftete Grössen

Ein Signalbereich geht von -0.5V bis +0.5V. Wnnn wir dieses Signal in binäre Werte verwandeln wollen müssen wir auch das Vorzeichen der Werte berücksichtigen. Dazu könnte man ein zusätzliches Bit als Vorzeichenbit definieren (Bsp. 0: positiv und 1: negativ).

Wenn wir das Signal mit einer Genauigkeit von 1mV codieren benötigen wir für den Bereich von 0V bis 0.5V genau 501 Werte. Mit 9 Bit sind 512 Werte möglich. Wir brauchen mit dem Vorzeichen zusammen 10 Bit für die Darstellung.

Man könnte das vorderste Bit, also das MSB (most significant bit) für das Vorzeichen reservieren. Dadurch würden folgende Codierung möglich:

0V	00'0000'0000
1mV	00'0000'0001
511mV	01'1111'1111
-1mV	10'0000'0001
-511mV	11'1111'1111

2.1 Einerkomplement

Gebräuchlich ist das Einerkomplement. Hierbei wird für eine negative Zahl alle Bit der entsprechenden positiven Zahl umgekehrt (komplementiert).

0	0000'0000
1	0000'0001
2	0000'0010
3	0000'0011
127	0111'1111
-127	1000'0000
-126	1000'0001
-2	1111'1101
-1	1111'1110

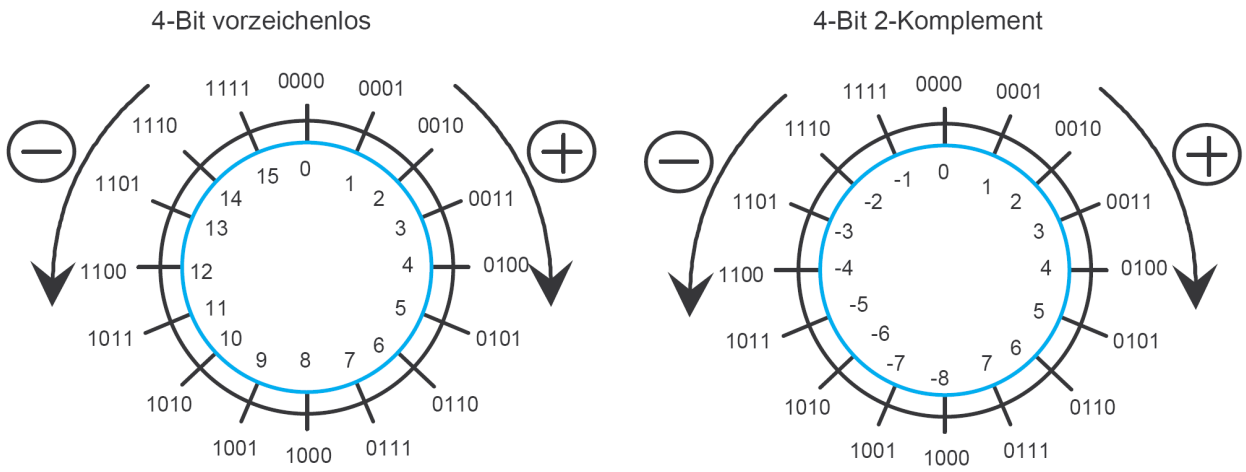
In diesem Fall wird die Zahl 1111'1111 nicht gebraucht. Deshalb ist diese Codierung nicht optimal

2.2 Zweierkomplement

Für die weitere Verrechnung von negativen Zahlen wird sinnvollerweise das Zweierkomplement eingesetzt. Das Zweierkomplement wird aus dem Einerkomplement+1 gebildet. Das Zweierkomp-lement wird überall in der Prozessortechnik für die Addition und Subtraktion von Zahlen verwendet.

0	0000'0000
1	0000'0001
2	0000'0010
3	0000'0011
127	0111'1111
-128	10000000
-127	1000'0001
-126	1000'0010
-2	1111'1110
-1	1111'1111

2.2.1 Graphische Darstellung anhand eines Zahlenkreises



2.3 Offset

Das Vorzeichen kann aber auch einfach durch einen Offset behandelt werden. Betrachten wir dabei wieder unser Beispiel mit den Werten von -0.5V bis +0.5V. Wir benötigen immer noch 10 Bit führen nun aber einen Offset von 512 ein, dh die Zahl 512 entspricht einem Wert von 0V, negative Werte sind kleiner, positive Werte sind grösser:

511mV	11'1111'1111
510mV	11'1111'1110
...	
2mV	10'0000'0010
1mV	10'0000'0001
0V	10'0000'0000
-1mV	01'1111'1111
-2mV	01'1111'1110
...	
-512mV	00'0000'0000

Die Werte sind durchgehend in aufsteigender Form möglich, aber der Nullpunkt wird verschoben. Dadurch ist ein negativer Bereich möglich.

3 Tetradische BCD-Codes

Tetradisch: 4 bit breit, die nicht genutzten Binärkombinationen heissen Pseudotetraden

3.1 8421-Code (Gewichtung)

Der Standard BCD Code ist ein tetradischer Code.

3.2 4221-Code (Gewichtung)

Tabellenabkürzung G: Gewicht

		CODE			
G		4	2	2	1
0		0	0	0	0
1		0	0	0	1
2		0	0	1	0
3		0	0	1	1
4		0	1	1	0
5		0	1	1	1
6		1	0	1	0
7		1	0	1	1
8		1	1	1	0
9		1	1	1	1

Wie bei diesem Code ersichtlich, wird der Wert 5 nicht durch 4+1 erreicht, sondern vom LSB her durch 2+2+1.

3.3 White-Code (Gewichtung: 5211)

5er-Bündelung, dadurch leichte Umsetzung in Dezimaldarstellung. Die obere Hälfter der Werte (5-9) sind direkt am MSB erkennbar.

		CODE			
G		5	2	1	1
0		0	0	0	0
1		0	0	0	1
2		0	0	1	1
3		0	1	0	1
4		0	1	1	1
5		1	0	0	0
6		1	0	0	1
7		1	0	1	1
8		1	1	0	1
9		1	1	1	1

3.4 Aiken (Gewichtung: 2421)

Besonders günstig für Rechenoperationen: Das Neunerkomplement der Dezimalzahl kann durch bitweise Invertierung erzeugt werden.

		CODE			
G		2	4	2	1
0		0	0	0	0
1		0	0	0	1
2		0	0	1	0
3		0	0	1	1
4		0	1	0	0
5		1	0	1	1
6		1	1	0	0
7		1	1	0	1
8		1	1	1	0
9		1	1	1	1

3.5 Stibitz-Code (Exzess-3-Code) - keine Gewichtung:

Exzess-3 bedeutet ein Überschuss von 3 gegenüber dem 8421-Code (Offset). Der Stibitz-Code vermeidet das sogenannte Nullwort (0000) und das Einswort (1111). Diese Bitkombinationen treten insbesondere im Fehlerfall, etwa bei Leitungsunterbrechungen oder Kurzschlüssen auf. Da sie nicht im Code enthalten sind, kann hier leicht ein Fehler festgestellt werden.

		CODE			
0		0	0	1	1
1		0	1	0	0
2		0	1	0	1
3		0	1	1	0
4		0	1	1	1
5		1	0	0	0
6		1	0	0	1
7		1	0	1	0
8		1	0	1	1
9		1	1	0	0

4 Einschrittige tetradische BCD-Codes

Keine Gewichtung

4.1 Glixon Code

		CODE			
0		0	0	0	0
1		0	0	0	1
2		0	0	1	1
3		0	0	1	0
4		0	1	1	0
5		0	1	1	1
6		0	1	0	1
7		0	1	0	0
8		1	1	0	0
9		1	0	0	0

4.2 Tompkins Code

		CODE			
0		0	0	0	0
1		0	0	0	1
2		0	0	1	1
3		0	0	1	0
4		0	1	1	0
5		1	1	1	0
6		1	1	1	1
7		1	1	0	1
8		1	1	0	0
9		1	0	0	0

4.3 O'Brian Code

		CODE			
0		0	0	0	0
1		0	0	0	1
2		0	0	1	1
3		0	0	1	0
4		0	1	1	0
5		1	1	1	0
6		1	0	1	0
7		1	0	1	1
8		1	0	0	1
9		1	0	0	0

5 BCD-Codes mit mehr als 4 Stellen

5.1 1-aus-10 Code (gewichtet)

		CODE									
G		9	8	7	6	5	4	3	2	1	0
0		0	0	0	0	0	0	0	0	0	1
1		0	0	0	0	0	0	0	0	1	0
2		0	0	0	0	0	0	0	1	0	0
3		0	0	0	0	0	0	1	0	0	0
4		0	0	0	0	0	1	0	0	0	0
5		0	0	0	0	1	0	0	0	0	0
6		0	0	0	1	0	0	0	0	0	0
7		0	0	1	0	0	0	0	0	0	0
8		0	1	0	0	0	0	0	0	0	0
9		1	0	0	0	0	0	0	0	0	0

Anwendung: Manuelle Zahleneingabe und -ausgabe

5.2 2-aus-5 Walking Code

		CODE				
0		0	0	0	1	1
1		0	0	1	0	1
2		0	0	1	1	0
3		0	1	0	1	0
4		0	1	1	0	0
5		1	0	1	0	0
6		1	1	0	0	0
7		0	1	0	0	1
8		1	0	0	0	1
9		1	0	0	1	0

Anwendung: Fernschreibtechnik

5.3 Generell: m-aus-n-.Code

Der Code hat n Stellen. Davon sind immer m Stellen gleich 1.

Man nennt diese Codes auch gleichgewichtete Codes

5.4 Johnson-Code oder Libaw-Craig-Code

		CODE				
0		0	0	0	0	0
1		0	0	0	0	1
2		0	0	0	1	1
3		0	0	1	1	1
4		0	1	1	1	1
5		1	1	1	1	1
6		1	1	1	1	0
7		1	1	1	0	0
8		1	1	0	0	0
9		1	0	0	0	0

6 Häufig verwendete Wortcodes

6.1 Dual Code

mit Gewichtung

		CODE			
G		2^3	2^2	2^1	2^0
0		0	0	0	0
1		0	0	0	1
2		0	0	1	0
3		0	0	1	1
4		0	1	0	0
5		0	1	0	1
6		0	1	1	0
7		0	1	1	1
8		1	0	0	0
9		1	0	0	1
10		1	0	1	0
11		1	0	1	1
12		1	1	0	0
13		1	1	0	1
14		1	1	1	0
15		1	1	1	1

6.2 Gray Code

Ohne Gewichtung

		CODE			
0		0	0	0	0
1		0	0	0	1
2		0	0	1	1
3		0	0	1	0
4		0	1	1	0
5		0	1	1	1
6		0	1	0	1
7		0	1	0	0
8		1	1	0	0
9		1	1	0	1
10		1	1	1	1
11		1	1	1	0
12		1	0	1	0
13		1	0	1	1
14		1	0	0	1
15		1	0	0	0

Anwendung: Binärcodierung von Winkeln und Strecken.

6.3 Übung Codevergleich

Die nachfolgende Tabelle vergleicht einige verschiedene Codes. Ergänzen Sie die freien Felder in der Tabelle.

Codeart \ Eigenschaften	Stellenzahl	redundant: J, N	E: einschrittig M: mehrschrittig
BCD-Code Beispiel	4	J	M
Aiken-Code			
Excess-3-Code oder Stibitz-Code			
White Code			
2-aus-5-Code (Walking Code)			
1-aus-10-Code			
Johnson-Code (0..9)			

7 Datenübertragung

Bei der Datenübertragung wird mit den effektiven Grössen gerechnet, somit entspricht 1kB genau 1000 Byte und nicht wie bei Speicherelementen oft wegen deren Sektorgrösse (512B) 1024 Byte. Wenn ein Paket von 32 Byte über ein Medium übertragen wird gehen nicht Cluster über das Medium sondern Bytes.

8 Kompression

Anna möchte an den Badestrand gehen. Dafür möchte sie vier Gegenstände mitnehmen: Eine Luftmatratze, einen Wasserball, ein Badetuch und ihr Bikini. In ihrem Zimmer legt sie alles auf einen Haufen. Sie merkt, dass der Haufen richtig gross geworden ist und dass sie diese Gegenstände niemals auf ihrem Velo transportieren kann. Irgendwie muss sie nun das Volumen reduzieren.

8.1 "Luft raus" / Komprimierung

Sie stopft als erstes ihr Badetuch und den Bikini in den Rucksack. Diese Dinge braucht sie ja unbedingt, hier ist nichts reduzierbar. Beim Wasserball und bei der Luftmatratze allerdings schon: Sie lässt die Luft ab, und die beiden Dinge passen nun auch in den Rucksack. Nun kann Anna endlich los fahren und baden gehen.

Am See angekommen, kann Anna den Ball und die Luftmatratze wieder aufpumpen. Die Gegenstände haben nun wieder ihre ursprüngliche Form.

Was Anna hier gemacht hat, nennt man **"Komprimieren"**.

In der Informatik werden häufig grössere Datenmengen über ein Kabel transportiert. Dabei möchte man die Datenmenge so gering wie möglich halten, damit die Übertragung schneller geht. Vor dem Versand **komprimiert** man also die Daten: Man lässt sozusagen die Luft heraus, um sie am anderen Ende der Leitung wieder hineinzupumpen, so dass sich die Daten wieder in ihrer ursprünglichen Form präsentieren.

8.2 Datenkomprimierung (-Kompression)

Bei der Datenkomprimierung wird die Repräsentation von Daten so geändert, dass sie anschliessend weniger Platz in Anspruch nehmen. Die wichtigste Motivation dazu ist heute, dass die Daten schneller übertragen werden können.

Es dürfen bei der Komprimierung auf keinen Fall Daten verloren gehen oder verändert werden, d.h. nach der Dekomprimierung müssen die ursprünglichen Daten wieder genau so verfügbar sein (Reversibilität).

8.3 Warum reicht die normale ASCII-Codierung nicht aus?

Wenn wir einen "reinen" Text als ASCII-Datei speichern, so belegt jedes gespeicherte Zeichen 8 Bit. Wir könnten auch sagen, es wird mit 8 Bit codiert. Ob dieses Zeichen nun 1000 Mal oder nur ein einziges Mal im Text vorkommt, spielt dabei keine Rolle. Das ist im Hinblick auf den Speicherbedarf einer Datei kaum eine effektive Codierungsmethode.

Besser wäre eine Codierung die Häufigkeiten der Zeichen im Text berücksichtigt. Wenn ein Zeichen häufig im Text auftritt, wählen wir einen möglichst kurzen Code dafür. Wenn ein Zeichen hingegen nur ganz selten vorkommt, spielt es keine Rolle wenn sein Code etwas länger ist. Das Ziel ist also ein Code mit folgender Eigenschaft:

Je grösser die Wahrscheinlichkeit, dass ein Zeichen im Text auftritt, desto kürzer soll sein Code sein (im Vergleich zu den Codes der anderen Zeichen).

Wir könnten auch sagen: je häufiger ein Zeichen im Text vorkommt, desto kürzer soll sein Code sein. Die Wahrscheinlichkeiten bzw. die Häufigkeiten der Zeichen spielen also eine zentrale Rolle.

8.4 Alphabet (deutsch, englisch)

In einem deutschen oder englischen Text kommt der Buchstabe e sehr viel häufiger vor als beispielsweise der Buchstabe q. Um den Text mit möglichst wenigen Bits zu codieren, liegt die Idee nahe, häufig vorkommende Zeichen durch möglichst kurze Codewörter zu codieren.

Diese Idee ist auch beim Morse-Code verwirklicht, in dem das Zeichen e durch ein Codewort der Länge 1, nämlich einen Punkt • ,das Zeichen q dagegen durch ein Codewort der Länge 4, nämlich **- - • -** codiert wird.

Problematisch bei Codes mit unterschiedlichen Codewortlängen ist, dass im allgemeinen eine eindeutige Decodierung von Zeichenfolgen nicht möglich ist: Im Morsecode könnte beispielsweise die Folge **•• - •••• -** sowohl zu "usa" als auch "idea" decodiert werden.

Das Problem kommt dadurch zustande, dass gewisse Codewörter Anfangswörter von anderen Codewörtern sind.

Dadurch kann bei der Decodierung nicht entschieden werden, wo ein Codewort endet und wo das nächste anfängt. Die o.a. Folge könnte mit e (•) oder mit i (••) oder mit u (•• -) oder sogar mit f (•• - •) anfangen. Im Morsecode wird das Problem gelöst, indem hinter jedem Codewort als Trennzeichen eine kurze Pause eingefügt wird.

Gilt dagegen die folgende Bedingung, so lassen sich codierte Zeichenfolgen direkt und eindeutig decodieren:

Wenn kein Codewort Anfangswort eines anderen Codewortes ist, dann ist jede codierte Zeichenreihe eindeutig decodierbar.

Um diese Bedingung erfüllen zu können wird der Algorithmus von Huffman eingesetzt. Dieser berücksichtigt auch die Häufigkeit eines Zeichens.

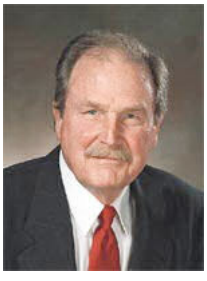
9 Huffman

Anna möchte das Wort **LIEGEBETT** über das Internet an Beat schicken.

Im Wort LIEGEBETT kommen 6 verschiedene Buchstaben vor: **L, I, E, G, B und T**.

Anna kann sich folgendes überlegen: Buchstaben, die oft im Wort vorkommen, sollen einen kurzen Code haben und solche, die selten vorkommen, einen langen. Das Verfahren, das wir anwenden, wird **Huffman-Codierung** genannt.

Es funktioniert vor allem dann gut, wenn es gewisse Buchstaben gibt, die viel häufiger als andere vorkommen. Wenn alle Buchstaben gleich häufig vorkommen, ist die Huffman-Codierung nicht besser als eine kompressionslose Codierung.



David A. Huffman (9.4.1925-7.10.1999), Computerpionier:

Huffman studierte an der Ohio State University in Columbus und promovierte 1953 am Massachusetts Institute of Technologie (MIT). Anschliessend wurde er dort Assistenzprofessor und 1962 ordentlicher Professor. Später wechselte er an die Universität Santa Cruz und gründete dort die Fakultät für Computer Science.

9.1 Komprimierung

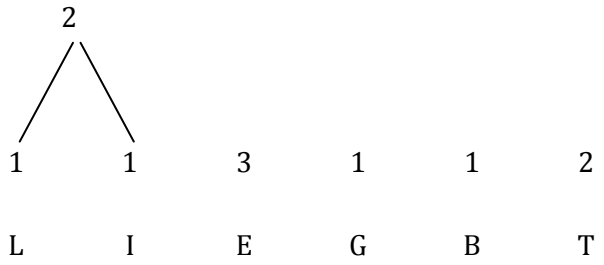
Als erstes erstellen wir eine Tabelle mit den Häufigkeiten der Buchstaben:

L:	1 Mal	0.11
I:	1 Mal	0.11
E:	3 Mal	0.33
G:	1 Mal	0.11
B:	1 Mal	0.11
T:	2 Mal	0.22
total	9	1.00

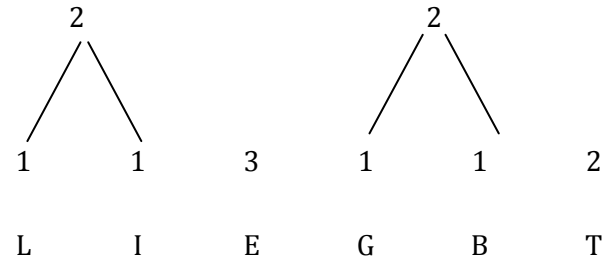
9.1.1 Vorgehen

Die beiden Buchstaben mit der **geringsten Häufigkeit** werden immer miteinander verbunden. Der entstandene Punkt erhält als Häufigkeit **die Summe der beiden Buchstaben**. Dies wird nun so oft wiederholt, bis nur noch ein Punkt übrig bleibt, die Wurzel des Baums. Haben mehrere Punkte die **gleiche Häufigkeit**, so können **zwei beliebige** Punkte gewählt werden (im Beipiel werden immer zuerst die linken genommen)

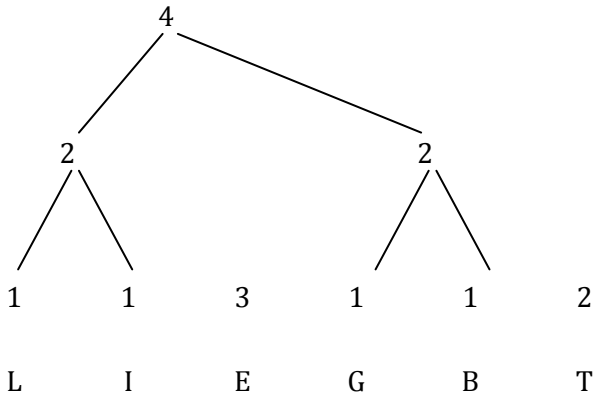
Schritt 1



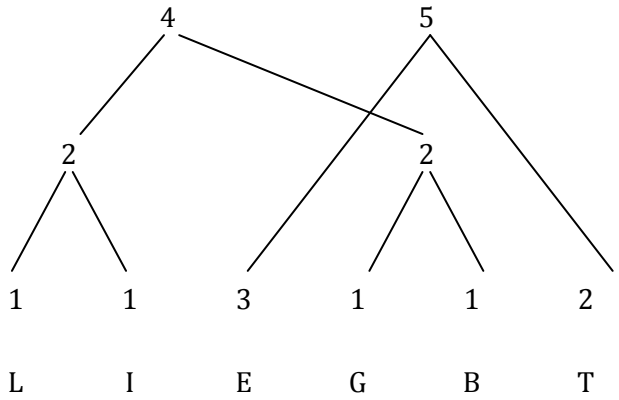
Schritt 2



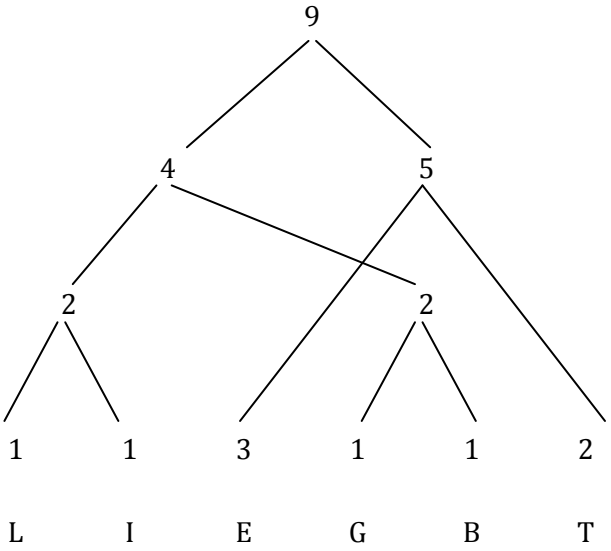
Schritt 3



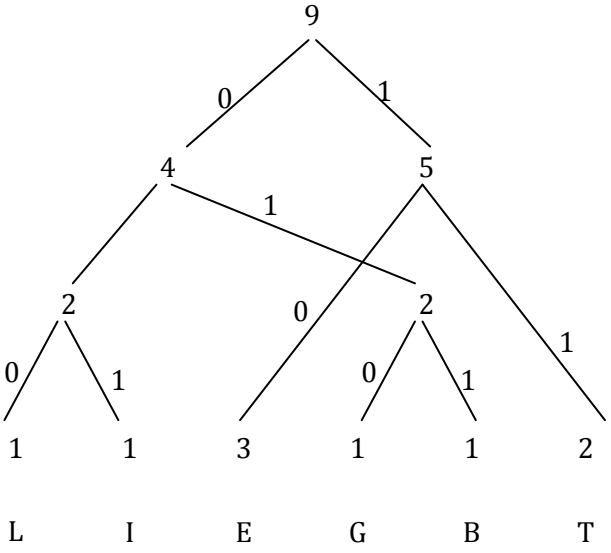
Schritt 4



Schritt 5



Huffman-Baum



9.2 Huffman-Baum

Es entsteht ein Baum. Um das Codewort für einen Buchstaben abzulesen, geht man von der "Wurzel" (zuoberst am Baum) nach unten. Ein Ast nach links bedeutet eine '0', eine nach rechts eine '1'. Also einfaches ablesen der Codewörter aus dem Baum.

L	⇨	000
I	⇨	001
E	⇨	10
G	⇨	010
B	⇨	011
T	⇨	11

9.3 Häufigkeit

Betrachten wir nochmals die Häufigkeiten, so sehen wir, dass 'E' und 'T' am häufigsten vorkommen (3 bzw. 2 Mal). Beide haben nun bei unserem Code tatsächlich ein kürzeres Codewort erhalten als die anderen vier Buchstaben.

Nun wollen wir die Anzahl Bits berechnen, die mit den neuen Codewörtern benötigt werden:

(Häufigkeit)·(Länge des Codes):

(L)1·3 + (I) 1·3 + (E) 3·2 + (G) 1·3 + (B) 1·3 + (T) 2·2 = 22

Mit der Huffman-Codierung braucht unser Text also noch 22 anstelle der 27 Bits mit der simplen Codierung

"LIEGEBETT" entspricht nun	000|001|10|010|10|011|10|11|11
oder einfach	0000011001010011101111

9.4 Dekomprimierung (Expandierung)

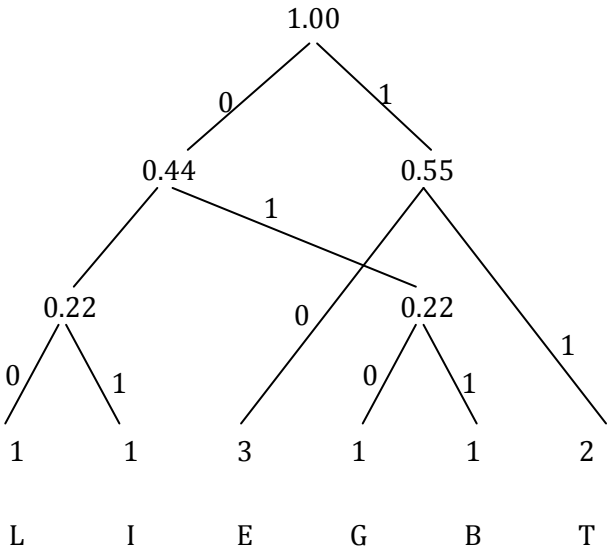
Beat erhält nun diese Reihe von Nullen und Einsen. Damit er sie entschlüsseln kann, muss er wieder die Codierung der Buchstaben kennen. Anna muss ihm diese mitteilen, damit er den oben beschriebenen Baum aufzeichnen kann. Da die Codewörter nun nicht mehr alle drei Buchstaben lang sind, kann er nicht mehr einfach drei Bits zusammenfassen wie vorher beim simplen Code.

9.4.1 Vorgehen

Er nimmt den Huffman-Baum zu diesem Codewort und beginnt bei der "Wurzel". Schickt ihm Anna eine '0', so geht er nach links, schickt sie ihm eine '1', so geht er nach rechts. Das macht er solange bis er zu einem Buchstaben kommt. Dann beginnt er wieder bei der Wurzel.

Bei unserem Beispiel geht er dreimal nach links (da nur Nullen kommen) und kommt zu 'L'. Dann beginnt er wieder oben, geht zweimal nach links (zwei Nullen) und einmal nach rechts (eine Eins) und kommt zu 'I', usw.

9.5 Huffman-Baum mit Häufigkeiten



9.6 Weiter komprimieren?

Nein

Man kann beweisen, dass der Huffman-Code optimal ist. Das heisst, es gibt keinen anderen Code, der weniger Bits braucht. Der Huffman-Code ist nur für die jeweils zugrunde liegenden Daten optimal. Der Code, den wir berechnet haben, funktioniert also nur für das Wort LIEGEBETT.

9.7 Variationen bei der Konstruktion

Die Konstruktion des Huffman-Codes zu einer Nachricht setzt voraus, dass man bei endlichen Nachrichten die Häufigkeiten der Zeichen ermitteln kann, bei unendlich langen Nachrichten die Wahrscheinlichkeiten der einzelnen Zeichen.

In einigen Fällen ist dies aber nicht möglich, weil etwa die Nachricht noch nicht vollständig bekannt ist oder noch während der Codierung vervollständigt wird.

In solchen Fällen verzichtet man auf die Garantie der kürzesten Codierung und greift auf empirische oder geschätzte Häufigkeiten beziehungsweise Wahrscheinlichkeiten der Zeichen zurück, um für eine Nachricht einen Code zu konstruieren. Ist die erwartete Nachricht zum Beispiel ein Text in einer Sprache, so kann man auf empirische Tabellen zurückgreifen, welche für diese Sprache erstellt wird.

9.8 Buchstaben-Häufigkeitsverteilung im deutschen Alphabet

	relative Häufigkeit		relative Häufigkeit
e	17.40%	m	2.53%
n	9.78%	o	2.51%
i	7.55%	b	1.89%
s	7.27%	w	1.89%
r	7.00%	f	1.66%
a	6.51%	k	1.21%
t	6.15%	z	1.13%
d	5.08%	p	0.79%
h	4.76%	v	0.67%
u	4.35%	j	0.27%
l	3.44%	y	0.04%
c	3.06%	x	0.03%
g	3.01%	q	0.02%

9.9 Nachteile von Huffman-Codes

Ein Nachteil der Huffman-Codierung ist die **Notwendigkeit** des Zugriffes auf die einzelnen Codezeichen in einer codierten Nachricht, also ein **Zugriff auf der Bitebene**, falls die Nachricht auf Rechnern verarbeitet wird.

Viele gebräuchliche Rechnerarchitekturen sind nur für den Zugriff auf Byte, Worte oder noch grössere Einheiten spezialisiert. Der direkte Zugriff auf Bits ist oft nicht vorgesehen und muss aufwendig durch boolesche Operationen realisiert werden, indem man zum Beispiel UND-Verknüpfungen und Bitverschiebungen zum Ausmaskieren einzelner Stellen von Binärzahlen verwendet.

Ein weiterer Nachteil ist die Notwendigkeit, dass man die Codierungstabelle mit der codierten Nachricht übertragen muss oder zumindest solche Informationen, dass die Codierung wieder vollständig rekonstruiert werden kann, weil sonst keine Decodierung mehr möglich ist.

Die **Übertragung der Codierungstabelle** verlängert vor allem bei kurzen Nachrichten die Länge der übertragenen Nachricht merklich. Bei längeren Nachrichten hingegen kommt die starke Vereinfachung zum Tragen. Da man nur die Häufigkeiten der Zeichen beachtet, aber nicht den Kontext, in welchem sie stehen, können sehr viele Redundanzen unausgenutzt bleiben. So lässt sich zum Beispiel eine zufällige Folge von gleich vielen Nullen und Einsen wesentlich schlechter komprimieren als eine Folge, welche abwechselnd Nullen und Einsen enthält. Der Huffman-Code codiert aber beide Nachrichten mit der gleichen Länge.

10 Run Length Encoding (RLE, RLC)

Ein einfacher und vor allem verlustfreier Kompressionsalgorithmus für digitale Daten ist die Lauflängencodierung. Dieser wird auch als Run Length Encoding (RLE) oder RLC (Run Length Coding) bezeichnet. Er ist besonders gut geeignet, Wiederholungen oder Sequenzen von gleichen Werten verkürzt darzustellen. Liegt eine Wiederholung vor, wird die Anzahl der Wiederholungen sowie der wiederholte Wert gespeichert.

Bei der Encodierung werden anstelle der uncodierten Daten sogenannte Runs gespeichert. Ein Run ist in der allgemeinen Form der RLE **eine Sequenz bestimmter Länge, die ein identisches Zeichen beinhaltet**. Die Länge der Sequenz wird als **Run Count** und der Inhalt als **Run Value** bezeichnet.

Wenn wir die Beispielzeichenkette **abddcda** RLE codieren, ergibt sich folgende Tabelle:

Run	Run Count	Run Value
a	1	a
b	1	b
dd	2	d
c	1	c
d	1	d
a	1	a

RLE codiert sich zu: 1a1b2d1c1d1a

Wie man aber jetzt schon in dem Beispiel sieht, ist die RLE nicht immer optimal. So haben wir die ursprünglich 7 Buchstaben in ganze 12 Buchstaben komprimiert, was sogar zu einer Vergrösserung der Datenmenge geführt hat. Abhilfe schafft hier eine Modifikation der Grundidee.

Zuerst einmal sollte ein Trennzeichen eingeführt werden, da bei der Komprimierung von Zahlen die Unterscheidung zwischen Multiplikator und Zahl nicht mehr möglich ist.

Ausserdem sollte man nur Runs mit einem Run Count grösser 3 komprimieren. Somit bietet sich dieses Verfahren nur bei Anwendungen an, bei denen häufig grosse Run Counts auftreten, wie das im Falle einer FAX- Übertragung der Fall ist.

10.1 Faxübertragung

Eine Faxübertragung erfolgt zeilenorientiert, dh. es werden in der Folge die eingescannten Zeilen nacheinander abgearbeitet und übertragen. Das Faxgerät liest eine DIN A4 Seite mit 1728 x 1143 Pixel (horizontal x vertikal) mit einem Bit Farbtiefe (schwarz oder weiss), die Rohdatenmenge beträgt ca. 241kByte. Nach RLE-Codierung ergibt sich eine Datenmenge von ca. 40-120 kByte, abhängig von dem Bildinhalt. Die Codierung wurde vom CCITT festgelegt.

Für ISO A4, Letter und Legal besteht eine Scanlinie aus 1728 Pixel auf einer Breite von 215 mm (± 1 % Toleranz). Standardmässig sind 2 Auflösungen vorhanden:

Niedrige Auflösung: 1728 Pixel bei 203 oder 204 dpi horizontal; 98 dpi vertikal
Hohe Auflösung 1728 Pixel bei 203 oder 204 dpi horizontal; 196 dpi vertikal

Als Beispiel nehmen wir eine Zeile aus einem Standardfax, die übertragen werden soll. Diese ist 1728 Pixel breit, jedes Pixel kann entweder schwarz (s) oder weiss (w) sein.

s	s	s	w	w	w	w	w	w....w	s	s	s	s	w	w	w	w	s	s	s

Aus unserem Beispiel ergibt sich folgende Zeichenkette, die wir nach RLE erhalten 3s1714w4s4w3s (es werden also nur die Farbwechsel und Längen gekennzeichnet)

Da nur schwarze oder weisse Pixel vorkommen, können wir uns auch die Farbe der Pixel bei der Übertragung sparen, wenn wir uns darauf einigen, mit welcher Farbe eine Übertragung beginnen muss. In dem Standard, der bei der Faxübertragung benutzt wird, beginnt eine Zeile immer mit den Runs von weissen Pixeln. Somit erhalten wir folgende Runs: 0 3 1714 4 4 3.

10.2 Modified Huffman Code für Run length Codewörter bei der Faxübertragung

Diese einzelnen Werte werden jetzt mit einer Codetabelle codiert, die mit Hilfe des Huffmann Algorithmus erstellt wurde. Diese Tabelle wurde wie unser Beispiel für die Huffman Codierung erstellt. Die Wahrscheinlichkeiten für das Auftreten von weissen und schwarzen Pixeln ist sehr unterschiedlich, deshalb gibt es für jede Farbe eine andere Tabelle. Die für unser Beispiel benötigten Werte sind hier in einer Tabelle angegeben.

Des weiteren muss man noch beachten, dass nur die Zahlen 0 - 63 direkt codiert werden. Danach werden die Werte bis 2560 in 64er Schritten codiert. Dies reduziert die Grösse des Huffman Codebaumes. So wird der Wert 1714 aus 1664+50 dargestellt. Es ist vollkommen ausreichend, die einzelnen Werte direkt hintereinander zu schreiben, ohne ein Trennzeichen zu benutzen. Auch die Addition kann erkannt werden, weil es unterschiedliche Werte für schwarze und weisse Pixel gibt und niemals 2 gleichfarbige Blocks direkt hintereinander codiert werden würden.

Run length	Weisse Bits	Schwarze Bits
0	00110101	0000110111
...		
2	0111	11
3	1000	10
4	1011	011
5	1100	0011
...		
49	01010010	000001100101
50	01010011	000001010010
...		
1600	010011010	0000001011011
1664	011000	0000001100100
...		

Am Ende jeder Zeile wird noch ein End of Line (EOL) eingefügt, um sicherzustellen, das keine Daten verloren gehen. So würde bei einem Fehler während der Übertragung nur maximal eine Zeile verloren gehen. Das EOL wird mit 000000000001 codiert.

Somit erhalten wir folgende Tabelle:

0	**weiss**	**00110101**
3	**schwarz**	**1000**
1714	**weiss**	**011000+01010011**
4	**schwarz**	**011**
4	**weiss**	**1011**
3	**schwarz**	**10**
EOL		**000000000001**

und den daraus folgenden Binärcode:

001101011000011000010100110111011100000000000001

Damit wird die 1728 Pixel breite Zeile mit lediglich 47 Bit codiert

11 Ziv-Lempel (ZIP)

Der Name Lempel-Ziv bezeichnet die Basis für eine Gruppe von sich in wenigen Details unterscheidenden Kompressionsmethoden. Der Name geht zurück auf die beiden Entwickler der "Urform" aller zur Lempel-Ziv-Familie gehörenden Algorithmen, Abraham Lempel und Jacob Ziv.

Jacob Ziv

Abraham Lempel

Alle heute gebräuchlichen (und existenten) Varianten basieren auf diesen Ende der 70er Jahre entwickelten und erstveröffentlichten Grundformen, die im folgenden als LZ-Algorithmen bezeichnet werden sollen. Alle ZIP-Verfahren verwenden Ziv-Lempel bzw. verwandte und erweiterte Verfahren.

Trotz der grossen Vielfalt haben alle LZ-Algorithmen einige gemeinsame Eigenschaften:

- Verlustfreie Datenkompression
- Die LZ-Algorithmen benötigen zur Dekompression keine zusätzlichen Informationen über die Daten. Vielmehr wird diese Information während des Decodiervorganges gewonnen. Das ist der Hauptunterschied der LZ-Algorithmen gegenüber anderen verlustfreien Verfahren wie Huffman etc
- Die LZ-Algorithmen eignen sich auf Grund dieser Eigenschaften sehr gut dazu, transparent für den Benutzer implementiert zu werden (zB. Modem-Übertragungsprotokoll v42.bis).

11.1 LZ77

LZ77 ist ein wörterbuchbasiertes Verfahren, dass anstelle der Originaldaten, Rückgriffe auf Sequenzen aus dem bisherigen Inhalt codiert. Im Prinzip existiert nur eine einfache Codierungsvorschrift. Alle Daten werden in der Form

- Adresse der bereits enthaltenen Sequenz
- deren Sequenzlänge und
- das erstes abweichende Zeichen

codiert. Sind die zu codierenden Daten noch nicht in den vorangegangenen Daten enthalten, tritt also ein neues Zeichen auf, so wird die Adresse 0, die Sequenzlänge 0 und das betroffene Zeichen codiert. LZ77 wird zum Teil auch Sliding-Window-Algorithmus genannt, weil der Text durch ein Fenster bearbeitet wird, das über den Text geschoben wird.

11.1.1 Beispiel "abrakadabra"

	Adr.	Länge	abw. Zeichen
abrakadabra	0	0	'a'
a brakadabra	0	0	'b'
ab rakadabra	0	0	'r'
abr akadabra	3	1	'k'
abrak adabra	2	1	'd'
abrakad abra	7	4	''

Bedingt durch die Ergänzung jeder Sequenz mit dem ersten abweichenden Zeichen erhöht sich der Werteumfang der codierbaren Zeichen, ohne von dem standardisierten Format abzuweichen. Das ermöglicht eine einfache Umsetzung des Algorithmus mit minimalen Anforderungen an die Resourcen von Encoder und Decoder.

Die erzielbare Kompressionsrate ist ausschliesslich von sich wiederholenden Sequenzen gleichen Inhalts abhängig. Deshalb ist die Leistungsfähigkeit von LZ77 für sich alleine betrachtet eher gering. Die Dekompression erfolgt gleich. Es muss kein Codebuch übertragen werden.

11.2 Dynamisch vs statisch

Im Gegensatz zu den statischen Verfahren, bei denen vor der Codierung das Wörterbuch oder die Übersetzungstabelle festgelegt werden muss handelt es sich bei Ziv-Lempel um ein dynamisches Verfahren. Das Wörterbuch wird während der Codierung aus den zu verschlüsselnden Daten als auch bei der Decodierung implizit aus den verschlüsselten Daten neu generiert. Hier liegt der Hauptunterschied zu Huffman.

11.3 Lempel-Ziv-Welch (LZW)

Das LZW-Kodierungsverfahren stellt eine Abwandlung der von Jacob Ziv und Abraham Lempel vorgestellten Kompressionsmethode LZ78 (Nachfolger von LZ77) dar. Publiziert wurde das Verfahren 1984 von Terry A. Welch.

LZW enwickelt während der Kodierung ein Wörterbuch, das die bereits kodierten Zeichen aufnimmt. In den kodierten Daten sind lediglich die Indizes der Einträge in diesem Wörterbuch enthalten. Die ersten 256 Einträge werden mit den einzelnen Zeichen vorbesetzt. Alle nachfolgenden Einträge repräsentieren längere Zeichenketten.

Der von Terry Welch vorgestellte Algorithmus definiert den Aufbau des Wörterbuchs und beinhaltet Mechanismen, die sicherstellen, dass die Wörterbücher sowohl bei der Enkodierung als auch bei der Dekodierung identisch sind.

12 Multimedia-Kompressionsverfahren

Die Verteilung von Medienprodukten geschieht mittels klassischer Broadcasttechnik (Radio / Fernsehen) oder neuerdings über Streaming in Netzwerken, was beides einen Echtzeitbetrieb und somit eine Liveübertragung ermöglicht. Die Alternative ist die Nutzung von Speichermedien wie Film, Videobänder, CD's DVD's oder Harddisk. Der Datenreduktion und den Datenformaten, unter denen die Inhalte ausgetauscht werden, kommen eine zentrale Bedeutung zu. Damit die Kommunikationspartner einander auch verstehen können (Kommunikation mit vertretbarem bzw. realisierbarem Aufwand und ohne merklichen Qualitätsverlust), ist eine Standardisierung sowohl der Datenreduktionsalgorithmen als auch der Fileformate erforderlich.

13 Analoges Videosignal

Unsere visuelle Umgebung ist dreidimensional und von unendlich feiner Struktur. Eine künstliche Bildübertragung in so feiner Auflösung würde jeden Kapazitätsrahmen sprengen. Die naheliegende Lösung, um den grossen Informationsgehalt eines Videosignales ökonomisch zu übertragen, besteht darin, eine Parallel-Seriell-Wandlung durchzuführen, also durch zeilenweise Abtastung eines Bildes aus dem räumlichen Nebeneinander der optischen Informationen ein zeitliches Nacheinander der elektrischen Signale zu kreieren. Die dafür Eingesetzte Technik musste trägheitslos sein, damit war die mechanische Lösung, Nipkow-Scheibe (1884, P.G. Nipkow) der Kathodenstrahl- oder Braunschen Röhre (1897, K.F. Braun) weitaus unterlegen. (CRT = Cathode Ray Tube). Heute ist die Visualisierungstechnik bei Flachbildschirmen mit LCD (Liquid Crystal Display) angelangt. Alle Techniken folgen jedoch den Bedingungen:

Voraussetzung für eine Televisionsübertragung sind

- genügend schnelle Bildabtastung
- genügend schnelle Übertragung und Wiedergabe
- Synchronisation zwischen Sender und Empfänger

Ein Bild aufzunehmen, bedeutet bereits eine erste Datenreduktion. Es wird nämlich zunächst in zweidimensionaler Abbildung auf eine lichtempfindliche Schicht transferiert.

13.1 Bildsignalparameter

Das Verhältnis Breite zu Höhe beträgt bei allen Standard-TV-Systemen B/H = 4:3.

Darüber hinaus lässt sich zeigen, dass mit der Bereitstellung von ungefähr 600 Zeilen den menschliche Anforderungen an Bildschärfe Genüge getan wird.

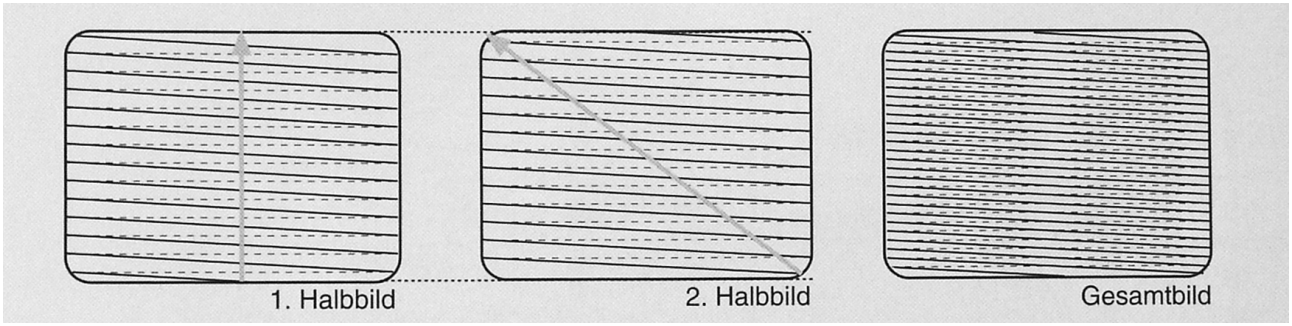
- Europäische Systeme (PAL) arbeiten mit 625 Zeilen (576 Zeilen sichtbar, die restlichen 49 Zeilen werden für andere Zwecke verwendet wie bsp Teletext))
- USA Systeme (NTSC) arbeiten mit 525 Zeilen (486 Zeilen sichtbar)

Unsere Fähigkeiten, die die zeitliche Abfolge betreffen, also das Sehen flüssiger Bildfolgen entsprechen die Frequenzen des Bildaufbaus. Die Trägheit des Auges lässt uns Bilder bei genügend schneller Aufeinanderfolge wie einen kontinuierlichen Bewegungsablauf erscheinen:

- Film : (Kino) 24 Bilder/Sek.
- TV Europa : 25 Bilder/Sek. (=halbe Frequenz des versorgenden Stromnetzes)
- USA : 29.97 (30) Bilder/Sek. (=halbe Frequenz des versorgenden Stromnetzes)

Die Austastung eines Elektronenstrahls auf einem TV-Schirm würde bei einem solche als progressiv bezeichneten Bildaufbau jedoch zu nur einer Dunkelphase zwischen zwei Bildern und damit zu einem störenden Grossflächenflimmern führen. Die mangelnden technischen Möglichkeiten zu Beginn der Fernsehentwicklung erlaubten aber nicht die Bereitstellung von geeigneten Speichern, um wie beim Kinofilm das gleiche Bild schlicht zweimal hintereinander zu zeigen. Der Trick dem man sich stattdessen bediente, nennt sich Zeilensprungverfahren, 2:1 oder Interlaced Mode:

Ein Bild wird in zwei Halbbilder zerlegt, wobei im ersten Schritt zunächst alle ungeradzahligen und nach dem Strahlrücksprung im zweiten Schritt alle geradzahligen Zeilen geschrieben werden. Das menschliche Gehirn und das Auge lassen sich soweit beeinflussen, dass der Mensch die Bilder als weniger flimmernd wahrnimmt.



Prinzipdarstellung des Zeilensprungverfahrens

13.2 Das Bild-Austast-Synchron-Signal (BAS)

Das Videosignal das zunächst nur aus der Information über die Leuchtstärke der einzelnen Pixel besteht, muss noch mit Synchronisationsimpulsen ergänzt werden. Die sichtbare, aktive Zeile ist etwas kürzer, weil eine gewisse Zeit, die sogenannte Austastlücke für den Rücksprung des Elektronenstrahls von rechts nach links reserviert sein muss. Um Sender um Empfänger zu Synchronisieren, ist in der Austastlücke ein Rechteckimpuls untergebracht. Ebenfalls muss um auch den vertikalen Rücksprung auszulösen ein entsprechendes Signal vorhanden sein.

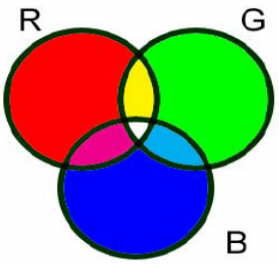
Die Bildauflösung ist auf 576 sichtbare Zeilen festgelegt worden und spiegelt sich über das Bildseitenverhältnis (4:3) in der Horizontalen wieder.

13.3 Farbvideosignal

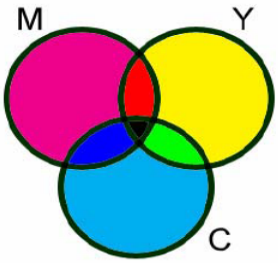
Zur Realisierung eines farbtauglichen Videosystems können wir der Einfachheit halber statt eines BAS-Signals drei BAS-Signale senden, generiert durch drei Bildwandler in der Kamera. Vorteilhaft ist beim RGB-Signal die hohe Qualität. Nachteilig sind allerdings der damit verbundene hohe Bandbreitenbedarf und die Notwendigkeit von drei separaten Leitungen. Verwendet wird diese Signalform daher nur auf kurzen Strecken wie beispielsweise zwischen Kameras und Einrichtungen der Bildtechnik. Ein weiterer Nachteil ist die fehlende Schwarz/Weiss Kompatibilität, so dass im Produktionsbereich das Komponentensignal bevorzugt wird, das auch die Basis der wichtigsten Digitalsignalform darstellt.

Hierbei wird schon die heutige Problematik eines Fernsehgeräts sichtbar, das alle Funktionen enthalten soll, wie herkömmliche Übertragung, HDTV in allen Formaten, 24 wie auch 25 Bilder/s, interlaced und noninterlaced Mode usw).

13.4 Farbbildung



Additives Farbsystem
(Fernseher)
RGB

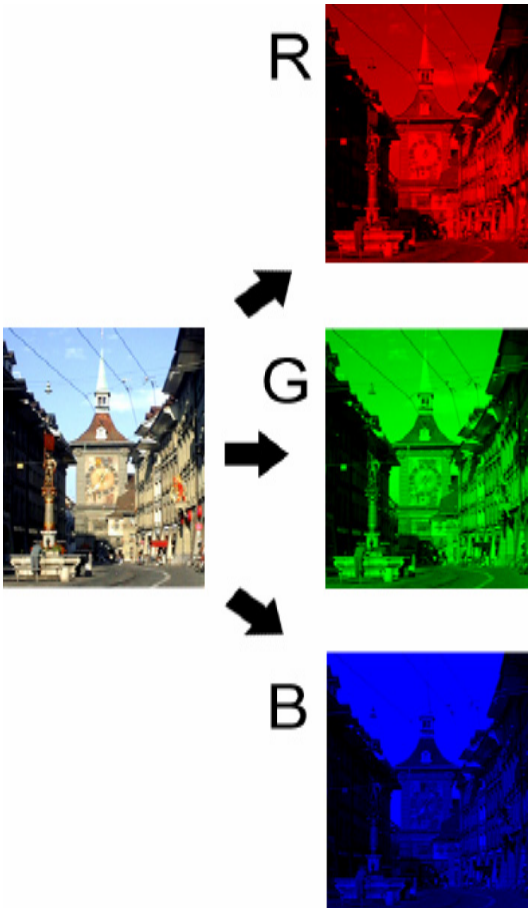


Subtraktives Farbsystem
(Drucker)

K=Keyfarbe (Schwarz)
CMYK

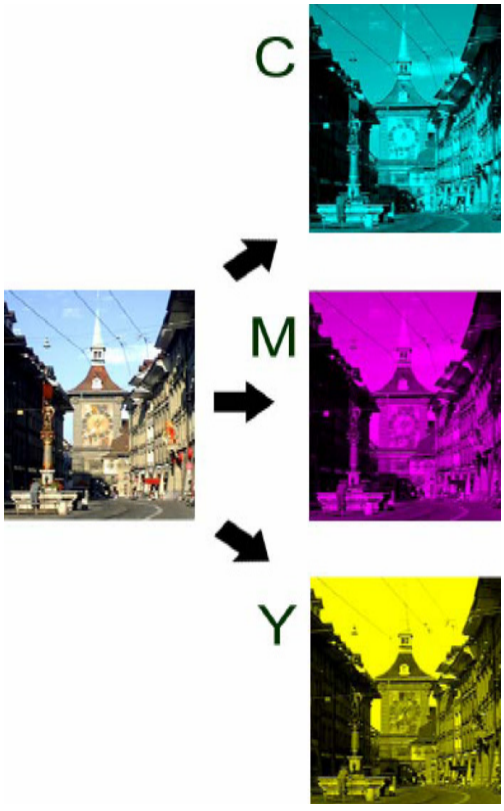
13.5 TV-Bildaufbereitung

Da es sich bei der Bilddarstellung um eine selbstleuchtende Quelle handelt wird dieadditive Farbmischung mit R: Rot, G: Grüb und B: Blau eingesetzt



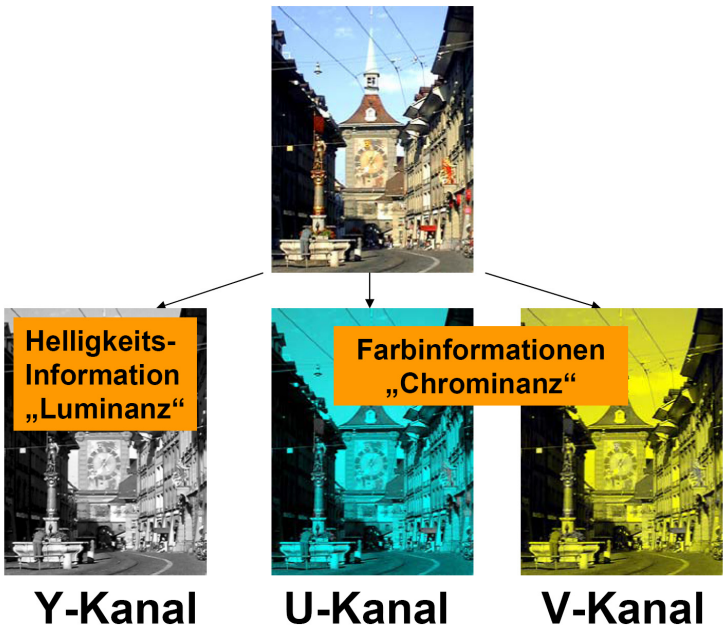
13.6 Druck-Bildaufbereitung

Für den Druck des Farbraums werden die Komplementärfarbender subtraktiven Farbmischung C: Cyan, M: Magenta und Y: Yellow benötigt.



13.7 Farbe und SW

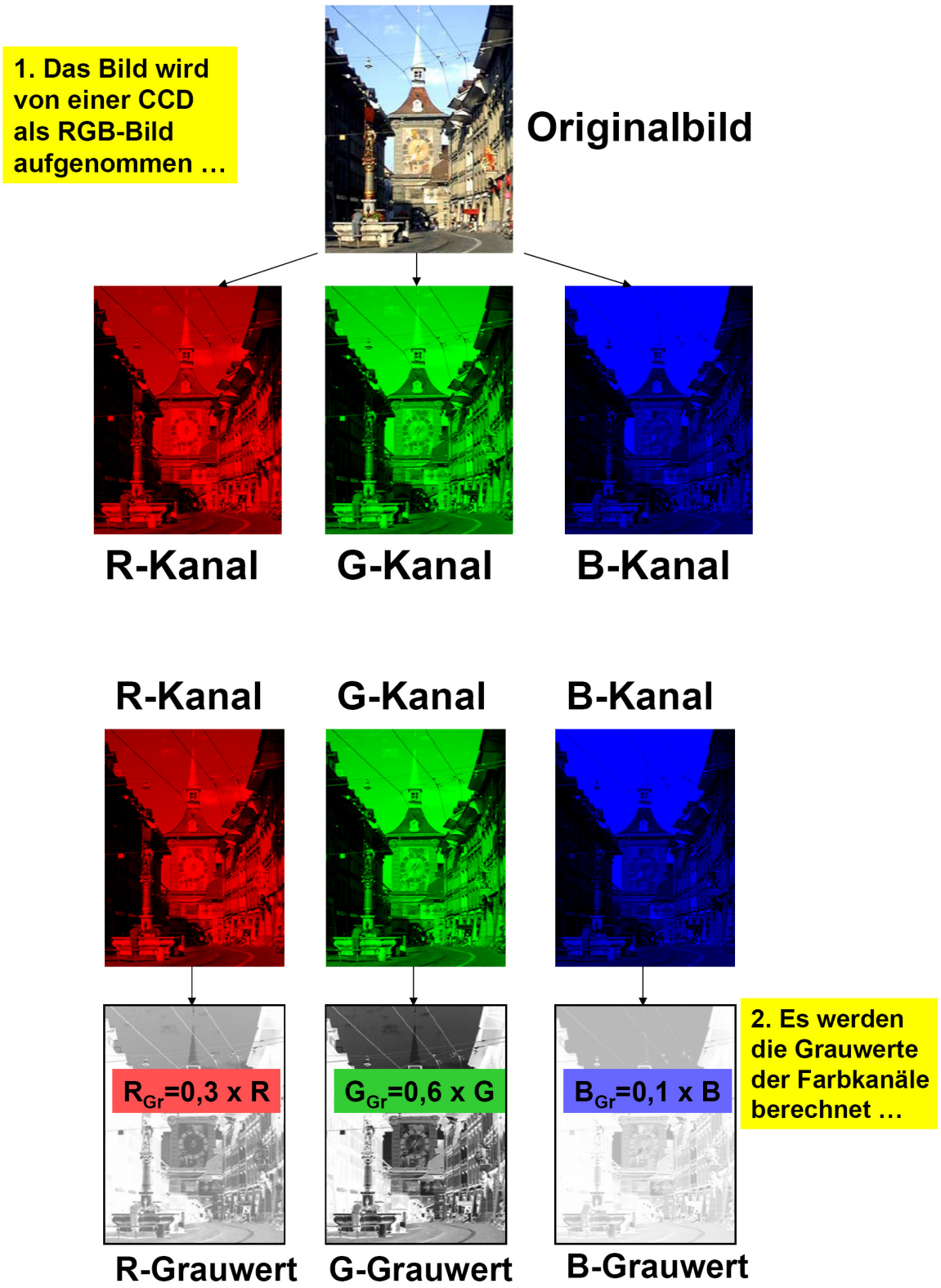
Kompatibilität zu S/W-Systemen bedeutet, dass aus dem Farbsignal das Leuchtdichtesignal, das als Luminanzsignal Y = Helligkeitssignal) bezeichnet wird, einfach ableitbar sein muss (Rückwärtskompatibilität RGB ⇨ SW).

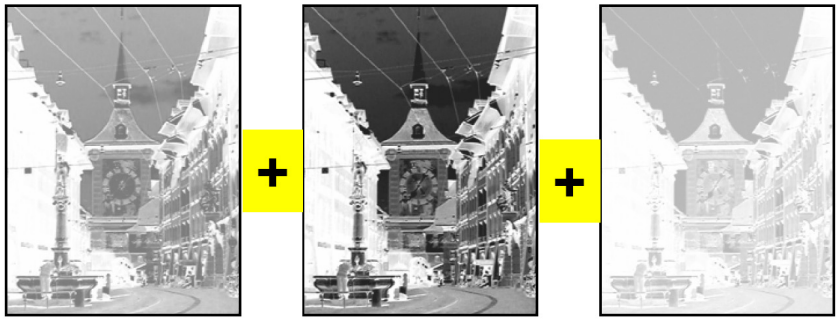


Dabei ist eine markante Eigenschaft des Auges zu berücksichtigen, nämlich das spektrale Helligkeitsempfinden, welches bei einer Wellenlänge von 555nm (Grün) ein ausgeprägtes Maximum besitzt. Daher gilt (näherungsweise):

$$Y = 0{,}299R + 0.587G + 0.114B$$

13.8 Umwandlung vom RGB-Format ins YUV-Format





3. Die drei Anteile werden addiert und man erhält das S/W-Bild

Graustufen-Bild

4. Die Chrominanz-anteile werden mit dem in Schritt 3 erhaltenen S/W-Bild berechnet.

Chrominanz (U)

Chrominanz (V)

13.9 Das Komponentenformat

Zusammengefasst : Das Komponentensignal

Y **U** **V**

Luminanz Chrominanz Chrominanz

13.10 Umwandlung vom YUV-Format ins RGB-Format

13.11 Umrechnung RGB in YUV und umgekehrt

Pro Kanal 8 Bit Auflösung, nach ITU-R Recommendation BT.601. Dies ist die genaue Umsetzung von RGB zu YUV und umgekehrt. Oft wird nur eine Näherung angegeben. Da die Farb/Helligkeits-Auflösung pro Kanal 8 Bit beträgt (ganzzahlig 0 .. 255) werden zusätzlich Offsets eingeführt

round: runden

$$Y = \text{round}(0.256788*R + 0.504129*G + 0.097906*B) + 16$$
$$U = \text{round}(-0.418223*R - 0.290993*G - 0.439216*B) + 128$$
$$V = \text{round}(0.439216*R - 0.367788*G - 0.071427*B) + 128$$

clip: abschneiden

$$R = \text{clip}(\text{round}(1.164383*(Y-16) + 1.596027*(V-128))$$
$$G = \text{clip}(\text{round}(1.164383*(Y-16) - (0.391762*(U-128)) - (0.812968*(V-128))$$
$$B = \text{clip}(\text{round}(1.164383*((Y-16) + 2.017232*((U-128))$$

clip() = Nur Werte aus dem Bereich 0..255.
Negative Werte werden zu 0, Werte grösser 255 werden zu 255

R, G, B: Signale für die Farben Rot, Grün und Blau

Y: Helligkeit, Leuchtdichte, Luminanz: Dieses Signal entspricht dem Videosignal eines Schwarz-Weiss-Fernsehers.
U, V: Farbart, Chrominanz: Farbton und Sättigung

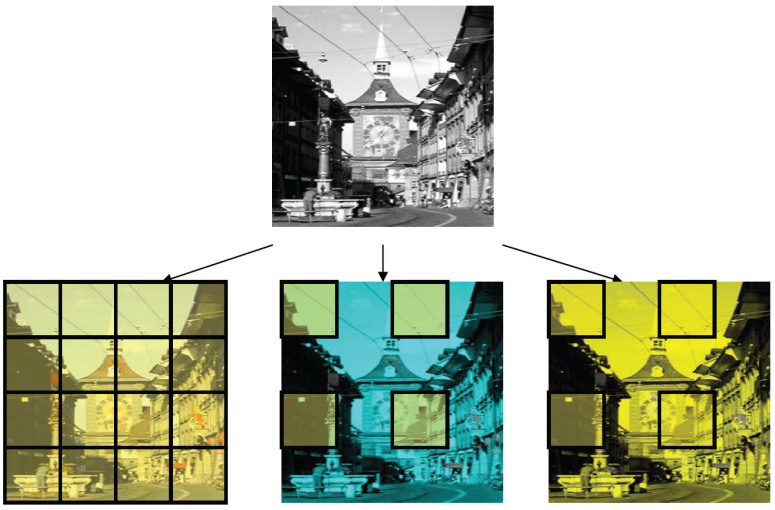
Aus U und V wird Signal C geschaffen.
Y und C auf zwei Leitungen : Y/C oder S-Video-Signal
Y und C zu einem Signal : FBAS oder Composite Signal

Neben dem Y-Signal würde es hiermit ausreichen, zB. Rot und Blau als einzige weitere Signale zu übertragen und dann den Grün-Anteil auf der Empfangsseite daraus zu berechnen. Um aber einen Informationsanteil zu gewinnen, der allein die Farbart (Buntheit/Chrominanz) betrifft, werden die Farbdifferenzkomponenten (R-Y) für Rot und (B-Y) für Blau verwendet, die zusammen mit Y mittels mathematischer Operationen einfach aus RGB gewonnen und ebenso einfach wieder zurückgewandelt werden können.

Die Differenzbildung ermöglicht auch eine Bandbreitenreduktion und zwar mit dem Argument, dass das menschliche Auge im Vergleich zur S/W-Auflösung für die Farbauflösung recht unempfindlich ist. Das bedeutet: Bei den Farben kann komprimiert werden. ⇨ weniger Samples in den Farbkanälen

Kennzeichnung	Unterabtastung	Bemerkung
4:4:4 Y/U/V	Keine	Keine Reduktion in den Farbkanälen (Chrominanz) -> Studiobereich
4:2:2 Y/U/V	Unterabtastung in beiden Chrominanzanteilen	2:1 Abtastverhältnis Norm ITU-R BT.601
4:1:1 Y/U/V	Unterabtastung in beiden Chrominanzanteilen	Die Chrominanz-Bandbreite und damit die horizontale Farbauflösung ist gegenüber dem 601-Signal halbiert
4:2:0 Y/U/V	Unterabtastung in beiden Chrominanzanteilen, alternierend.	Reduktion der Farbauflösung in der Vertikalen, da die Farbdifferenzsignale zeilenweise abwechselnd und nicht in jeder Zeile gleichzeitig übertragen werden. DV-Format

13.11.1 Beispiel: 4-1-1 Unterabtastung



Y-Kanal **U-Kanal** **V-Kanal**

Y-Kanal mit 16 Samples abgetastet

U- und V-Kanal mit je 4 Samples abgetastet

Das Synchronsignal wird hierbei einzig auf der Y-Leitung übertragen. Eine Verminderung des Kabelbedarfs ist mit dem Komponentensignal allerdings immer noch nicht gelungen (immer noch 3 Leitungen). Für eine Signalübertragung zum heimischen Fernsehempfänger sind noch erheblich grössere Reduktionen nötig, die uns schliesslich zum **FBAS-Signal (Farb-, Bild-, Austast-Synchronsignal)** führen.

Dabei werden aus den zwei Chrominanz-Anteilen (R-Y) und (B-Y) mittels Quadratur-Amplituden-Modulation (QAM) ein Signal geschaffen: das **Signal C**

- Liegen Y und C getrennt auf zwei Leitungen, spricht man vom Y/C oder auch S-Video-Signal. Verkämmen wir die beiden Signale (Y und C) zu einem Signal, erhalten wir das FBAS oder Composite Signal.

Aus U und V wird Signal C geschaffen.
Y und C auf zwei Leitungen : **Y/C** oder **S-Video-Signal**
Y und C zu einem Signal : **FBAS** oder **Composite Signal**

Das **FBAS**-Signal ist das **Standard-Videosignal** für den analogen Fernsehempfang auf der ganzen Welt. Für das Verfahren seiner Erzeugung existieren bezüglich dem Signal C aktuell drei weltweit dominante Verfahren :

- **NTSC** (USA, Japan, Kanada, Korea)
- **SECAM** (Frankreich, Aegypten, Algerien, Russland)
- **PAL** (Ganz Europa (ausser Frankreich), Brasilien, China, Indien)

NTSC	National Television System Commitee
SECAM	Séquentiel couleur à mémoire
PAL	Phase Alternation Line

Das YUV-Signal wird auch $Y/C_R/C_B$ -Signal genannt

14 Digitales Videosignal

Werden analoge Signale durch Störungen und Rauschen beeinträchtigt, ist das Nutzsignal im schlimmsten Falle unbrauchbar, mindestens aber im Dynamikbereich eingeschränkt. Bei der analogen Signalübertragung führt jede Signalverfälschung zu einer Informationsverfälschung (Fehlfarben und Pixel auf dem Bildschirm).

Die digitale Technik definiert daher für eine einwandfreie Rekonstuierbarkeit nur eine begrenzte Menge Spannungszustände, die durch die Zahlen 0 und 1 repräsentiert werden (Bit). Ihre Positionen liegen so weit auseinander, dass eine Störung keine Auswirkung hat, solange wir bei einer Signalverzerrung noch klar erkennen können, in welcher Hälfte des Signals (0 der 1) wir und befinden. Diese eindeutige Unterscheidbarkeitermöglicht auch, dass die Hardware ungenauer ausgeführt sein darf, als das bei der analogen Technik der Fall ist. Ausserdem treten bei der digitalen Form keine Generationsverluste (bei Kopien) auf.

14.1 Digitalisierung

Die Digitalisierung von analogen Video- und auch Audiosignal besteht im Grundsatz aus drei Abschnitten, nämlich der Abtastung (Sampling), einer Wertzuweisung (Quantisierung) und der Erzeugung einer digitalen Zahlenfolge (Codierung). Das Signal wird dabei doppelt diskretisiert, nämlich in ein festes Raster gleicher zeitlicher Abschnitte und ein mehr oder weniger feines Raster von zum jeweiligen Zeitpunkt gehörigen Werten eingeteilt.

Wie fein der Abstand zwischen den einzelnen Abtastzeitpunkten der entnommenen Proben (Samples), oder anders gesagt, wie hoch die Abtastrate (Sampling-Frequenz) sein muss, bestimmt sich nach der höchsten Geschwindigkeit, mit der sich das vorliegende Signal ändert, also aus der höchsten Signalfrequenz. Gemäss dem Shannon'schen Abtasttheorem lässt sich ein Signal auf der Empfängerseite genau dann eindeutig wieder zu einem analogen Signal mit ursprünglichem Verlauf zurückwandeln, wenn die Abtastrate F, also die Anzahl der entnommenen Proben pro Sekunde, mindestens doppelt so gross ist, wie die höchste erwünschte Signalfrequenz. Diese Forderung ist, zusammen mit einer Tiefpassfilterung für die Vermeidung des späteren Auftretens von Signalfehlern (Aliasing) unbedingt einzuhalten. Die Grundlage für die digitalen Videosignale sind die im vorigen Kapitel beschriebenen analogen Signalformen.

14.2 Digitales Komponentensignal

Mit der Norm ITU-R 601 wurde für die europäische und die US-Norm gleichermassen festgelegt, dass die aktive Zeile 720 Luminanzabtastwerte enthält sowie je 360 für die beiden Chrominanzanteile. Insgesamt befinden sich in der aktiven Zeile also 1440 Datenworte. Die Quantisierung der Signalabtastwerte wurde pro Komponente auf 10 Bit festgelegt. Ein digitalisiertes Vollbild mit 720 horizontalen und 576 vertikalen Bildpunkten (4:3) hat ohne Berücksichtigung der Daten in der Austastlücke einen Speicherbedarf von 1440 · 576=829'440 Bytes bzw. 6'635Mbit (=166Mbps). (Pro Zeile 1440 Werte = 720Y + 360 U + 360 V bei 4:2:2 Unterabtastung) Da im Datenstrom nicht nur die aktiven Bildwerte, sondern auch die Austastlücke übertragen werden, ergibt sich bei 10 Bit eine Datenrate von 270Mbs. Da bei der Bildung des digitalen Komponentensignals aus Gründen der einfachen Konvertier-barkeit auch die Abtastlücke abgetastet werden, sind die darin befindlichen Datenworte unbelegt und können zur Übertragung von Zusatzinformationen genutzt werden. Die häufigste Anwendung ist die Belegung mit Audiodaten (embedded audio)

14.3 High-Definition Videosignale

Es gibt seit Beginn der 90er Jahre Bestrebungen, das Erlebnis des Kinos auf das heimische Fernsehen zu übertragen. Eine grosse Darstellungsfläche, verbesserte Bild- und Tonqualität geben uns als High Definition Television (HDTV) das Gefühl, mehr in das Geschehen eingebunden zu sein. Das gelingt im Wesentlichen durch Zeilenzahlverdopplung, in Europa von 625 auf 1250 Zeilen, auf 1125 in den USA und Japan, und einer zusätzlichen Veränderung des Bildseiten-verhältnisses von 4:3 auf 16:9.

Neuere HDTV-Normierungen gehen von 1920 · 1080 Bildpunkten (Full HDTV) bei verschiedenen gängigen Bildraten zwischen 24Hz und 60Hz im Interlaced- so wie auch Progressive- Mode aus. (Norm ITU-R 709) Interessant gestaltet sich in jüngster Zeit auch die Frage nach der Verwendung der hochauflösenden Formate für Filmproduktionen.

HDTV 720p : 720 Zeilen, 1280 Linien, progressiv

HDTV 1080i : 1080 Zeilen, 1920 Linien, interlaced

14.3.1 MPEG4 Studio Profile :

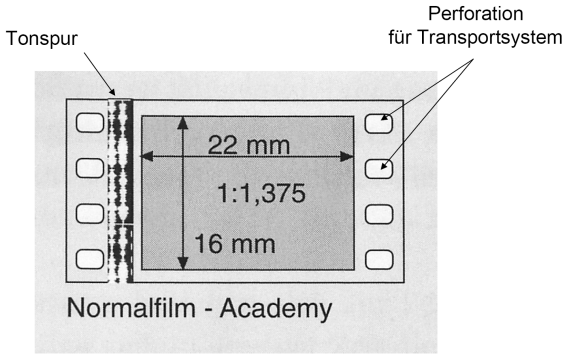
High (kompatibel mit MPEG-2 High)	1920 · 1080/60Hz	300Mbps
Highest	4096 · 4096/120Hz	2.4Gbps

15 Film

Das Medium Film kann auf eine noch ältere Geschichte als das Fernsehen resp. die Videotechnik zurückblicken. Seine Anfänge basieren auf Erkenntnissen im Bereich der Fotografie, die ihrerseits bereits zu Beginn des 19. Jahrhunderts gewonnen wurde. Geburtsstunde des Films : Erste öffentliche Filmvorführung 1895 (Gebrüder Lumière).

15.1 Filmformate und Bildfeldgrössen

35mm-Film ist seit Jahrzehnten das in Aufnahme und Projektion professionell verwendete Material für hochwertige Kinoproduktionen. In der Stummfilmzeit wurde hier eine Bildfeldgrösse von 24mm · 18mm genutzt. Mit der Entstehung des Tonfilms ging man dann zu einer verkleinerten Bildfläche, dem als Normal-Film bezeichneten Format mit 22mm · 16mm (1,375 : 1) über.



Kino-Film weltweit : **24** Bilder / Sek.

Die Präsentation von mindestens 20 diskreten Bildern in der Sekunde ist notwendig, um unserem Auge die Illusion eines flüssigen Bewegungsablauf zu suggerieren. Zur Vermeidung des Grossflächenflimmerns wird jedes Bild bei der Wiedergabe doppelt gezeigt.

Film arbeitet weltweit mit 24 fps (frames per second), wobei in jedem Bild die **gesamte Information** vorliegt. Dies ist der essentielle Unterschied zum Videosignal. Film ist das einzige wirkliche internationale Austauschformat im Gegensatz zu einer Vielzahl von Fileformaten in Zeiten von Multimedia, die ständig erweitert werden und wechseln.

15.2 Filmdigitalisierung/Bildauflösung

Wenn wir den Filmtransfer in die digitale Ebene betrachten, so stehen als wesentliche Charakteristika zur qualitativen Beurteilung das Auflösungsvermögen, der Kontrastumfang und die Umsetzung von Grauwerten und Farbe zur Verfügung.

Eine wesentliche Eigenschaft des Films ist die statistische Verteilung des Filmkorns bzw. der Farbstoffwolken beim Farbfilm. Dies führt zu einem Rauschen, welches neben der selektiven Schärfe den von uns als angenehm und typisch empfundenen Look des Films ausmacht. Dabei ist die Grösse des Korns bei unterschiedlichen Materialien von Bedeutung. Beim Filmmaterial besteht ein Antagonismus (Gegenspiel) zwischen den verschiedenen Leistungsmerkmalen. Hohe Empfindlichkeit ist mit grobem Korn verbunden.

Bei einem 200 ASA Film beträgt die Korngrösse ca. 5µm (5 tausendstel Millimeter).

Ziehen wir nun als Beispiel einen Super-35-Film heran, und wollen mit der Digitalisierung einem 200-ASA-Film entsprechen, ist dazu eine Auflösung von 4980 · 3752 Bildpunkten notwendig (4k-Abtastung).

Die Digitalisierung des Films führt zu sehr grossen Datenmengen.

Einige Zahlenbeispiele :

Bei einer 2k-Abtastung mit RGB 10 Bit und einem Bildseitenverhältnis von 4:3 ergibt sich für einen 90 minütigen Spielfilm eine formatunabhängige Rohdatenmenge von

(2048 · 1536)pix/frame · 24 frames/sec · 5400 sec · 3 · 10Bit · (1Byte/8Bit) =1,53TB

Ein Einzelbild hat dabei einen Datenumfang von 11,8MB. Bei 4k-Abtastung (4096 · 3112) wird die Datenmenge sogar vervierfacht. Sie beansprucht also 47.8MB pro Frame oder 1.15GB für eine Sekunde und hat die enorme Datenrate von 9,2Gbps. Dabei ist noch zu bedenken, dass wir es hier mit reinen Daten zu tun haben, die noch in keiner Weise in Files verpackt sind. Bei Files würde aufgrund der Header- und Zusatzdaten der Umfang noch weiter ansteigen.

16 Audiosignale

Während die Übertragung und Speicherung von Video- und Filmsignalen mit sehr hohem Datenaufkommen verbunden ist, haben Audiosignale den Vorteil, dass das Volumen im Vergleich nur etwa ein Hundertstel beträgt. Das liegt vor allem an dem beschränkten, vom Mensch wahrnehmbaren Frequenzbereich, der maximal zwischen 20Hz und 20kHz liegt. Dieser relativ schmale Bereich wird aber vom Gehör sehr effektiv genutzt, und im Gegensatz zu den Hell-Dunkel Differenzen bei Bilddaten werden die Laut-Leise-Differenzen, die so genannte Dynamik, weit umfangreicher ausgewertet. Das Gehör ist auch wesentlich empfindlicher für Signalfehler als das Auge. Die Audiosignalverarbeitung erfordert damit wesentlich mehr Störabstand, und gleichzeitig lässt sich die Datenreduktion nicht so effektiv durchführen wie bei Videosystemen.

Audiosignale sind immer mit Zeiträumen verknüpft. Anders als Bewegtbildsequenzen, die aus Folgen einzelner Bilder bestehen, die auch als Standbild betrachtet werden können, sind Audiosignale nicht zeitlich diskretisiert. Auf der akustischen Seite zeigen sie sich als kontinuierliche Folge von Luftdruckänderungen, auf die unser Ohr sehr empfindlich reagiert. Aufgrund der nichtlinearen Arbeitsweise des Gehörs sind wir in der Lage, Lautstärkedifferenzen über mehr als zehn Dekaden zu verarbeiten.

Daher wird als Lautstärkenmass auch der logarithmisch gebildete Schallpegel $L=20\log(p/p_0)$ angegeben, wobei bei $p_0=2x10^{-5}$ Pa der minimale Schalldruck für die Wahrnehmung, die Hörschwelle, liegt, der ein Pegel von 0dB zugeordnet ist. Bei 120dB liegt das zweite Extremum, die Schmerzgrenze, und mittlere Lautheiten empfinden wir bei 60dB, entsprechend $2x10^{-2}$ Pa. Die Empfindlichkeit des Gehörs ist stark frequenzabhängig. Die höchste Sensibilität hat es im mittleren Frequenzbereich bei 2..5kHz, sie nimmt zu höheren und tieferen Frequenzen in Form eines komplizierten Kurvenverlaufs ab, der zudem wiederum pegelabhängig ist.

16.1 1.5.1 Signalformen und Speicherverfahren

Mit der Einführung der optisch abgetasteten CD und der entsprechenden Abspielgeräte im Jahre 1983 fiel der Startschuss für den Einzug digitaler Audio-Medien. Ebenfalls im Jahre 1983 wurden die technischen Spezifikationen für das DAT (Digital Audio Tape) festgelegt, und 1992 wurde die MD (MiniDisc) vorgestellt, die auf magneto-optischer Basis arbeitet. Die Einführung der DVD (Digital Versatile Disc) sorgte dann erneut für eine Erweiterung des Qualitätsbegriffs im Heimbereich.

Soll den Ansprüchen des menschlichen Ohr genüge getan werden, welchem ein Hörvermögen von 20Hz bis 20kHz zu Eigen ist, ist unter Berücksichtigung des Abtasttheorems eine Sampling-Frequenz von mindestens 40kHz notwendig. In der Praxis wird zur Speicherung von Musik auf CD und MiniDisc standardmässig mit 44.1kHz abgetastet. Die Genauigkeit der Einteilung eines jeden Samples in 2^{16}=65536 Stufen stellt eine genügend feine Quantisierung dar, die dann bei der Codierung einem 16Bit breiten Binärwort zugewiesen wird. Bei der CD entsteht dadurch ein Datenstrom (Datenmenge pro Zeiteinheit) für Stereo-Ton von zweimal

44100 Samples/sec · 16Bit/Sample=1,41Mbps (Megabit pro Sekunde). Die Dynamik der CD kann mit etwa 96dB angegeben werden.

16.2 Digitalisierungsparameter

Telefonie :	8Bit bei 8kHz
CD :	16Bit bei 44.1kHz
DVD :	16Bit bei 48kHz
AudioDVD :	24Bit bei 192kHz
Studio :	16/20/24Bit bei 48/96/192kHz

17 Datenreduktion

Oftmals produzieren hochwertige digitale Signale einen so grossen Datenumfang, dass wir bei ihrer Speicherung oder Weitdistanzübertragung vor dem Problem stehen, vorhandene Systemresourcen und Kapazitäten nicht ökonomisch zu nutzen oder sie ggf. vollkommen zu überlasten. Dieser Umstand wird noch weiter verschärft, wenn wie bei einem Fernseh-Programm die Datenübermittlung in Echtzeit gefordert ist. Das digitale Komponentensignal etwa benötigt im einfachsten Codierungsfall (NRZ unipolar) einer Datenrate von 270Mbps. Eine solche Datenmenge würde niemals den Konsumenten als Fernsehsignal ins Haus geliefert werden, sondern dient ausschliesslich der professionellen Verarbeitung im Studiobereich. Eine Speicherung auf DVD-Video hätte in diesem Fall zur Folge, dass zB. eine DVD-5 mit 4.7GB Speicherplatz bereits nach 139 Sekunden gefüllt wäre.

17.1 Grundlagen

Ziel ist es, jede nicht benötigte Information im Originalsignal zu beseitigen und damit den Bandbreitenbedarf so weit wie möglich zu senken, ohne jedoch bei einer anschliessenden qualitativen Beurteilung wesentliche Unterschiede erkennen zu lassen. Basis für die Reduktion von Videosignalen ist das Vorhandensein von Ähnlichkeiten im Bild. Bei der Betrachtung eines einzelnen Bildes fällt auf, dass oft grosse Flächen mit ähnlichen Farb- und Grauwerten auftreten. Oft können wir feststellen, das diese Änderungen hauptsächlich von der Bewegung einzelner Objekten herrührt. Diese Wechselbeziehung des Bildes mit sich selbst oder folgenden Bildern aufgrund von Ähnlichkeiten nennt man örtliche bzw. zeitliche Korrelation (=Wechselbeziehung oder Ähnlichkeit) und nutzt sie zur Verkleinerung der Datenmenge. Datenreduktion wird auch als Kompression bezeichnet. Bei den einzelnen Verfahren unterscheiden wir grundsätzlich in Entropiecodierung und Quellencodierung (Entropie=mittlerer Informationsgehalt eines Symbols oder einer Symbolfolge), wobei wir in der gängigen Praxis zumeist die kombinierte Anwendung finden. Bei der Quellencodierung lässt sich noch eine Unterteilung in prädiktive und frequenzorientierte Kompression vornehmen. (Prädiktion=Vorhersage). Mit Ersterer können wir Korrelationen im Zeitbereich betrachten, mit Letzteren im Frequenzbereich, wie uns die DPCM (Differenziellen Puls Code Modulation) bzw. die DCT (Diskrete Cosinus Transformation) im Folgenden zeigen wird.

Grundlage der folgenden Abschnitte ist in den meisten Fällen das digitale Komponentensignal (4:2:2) mit 8Bit/Kanal und alleiniger Beschränkung auf die Pixel des aktiven Bildes. Zu diesem Signal gehört, und das ist als Referenzwert, insbe-sondere bei der Angabe von Kompressionsfaktoren wichtig, eine Datenrate von 166Mbps.

17.2 Redundanz und Irrelevanz

Jedes einzelne Symbol einer Nachricht ist Träger von Information. Dabei ist es gleichgültig, ob wir Sprache in Wort und Schrift analysieren, Datenströme oder Videobildsequenzen betrachten. Je seltener und unerwarteter ein Symbol ist, umso höher ist sein Informationsgehalt. Liefert eine Quelle z. B. die binäre Zeichenfolge '11111110', ist diese nach der ersten '1' so lange uninteressant, bis die '0' überraschend Neuigkeitswert bietet. Wenn im Idealfall jedes Symbol gleich unerwartet ist, so ist der mittlere Informationsgehalt oder Erwartungswert eines jeden Zeichens und einer Zeichenfolge, maximal.

Alltägliche Beobachtungen zeigen uns, dass einige Zeichen sehr viel häufiger als andere anzutreffen sind und der Informationsgehalt in der Praxis niemals ihren Maximalwert erreicht. In der Anwendung

der deutschen oder englischen Sprache finden sehr häufig die Buchstaben 'e' und 's', seltener hingegen z. B. das 'f' oder 'q', ein Umstand, den man sich schon bei der Erstellung des Morse-Alphabets zunutze gemacht hat.

In einer Nachricht liegt aufgrund dieses Sachverhalts stets Mehrinformation vor, auch Redundanz genannt, die nicht unbedingt notwendig ist sondern lediglich die beabsichtigte Grundinformation stützt. Ein Standbild muss nicht 25-mal in der Sekunde übertragen werden, und die gleichmässige Färbung eines Videobildes durch einen einheitlichen Hintergrund ist zwar zur Wiedergabe auf dem Bildschirm unabdingbar, jedoch nicht für die Übertragung der Daten.

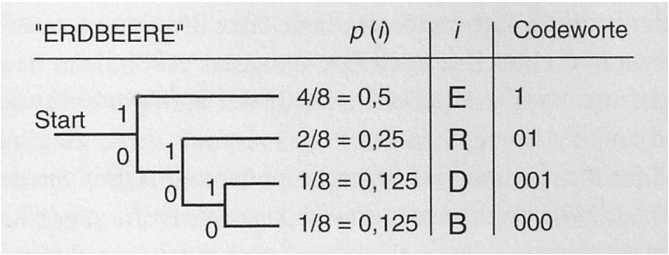
Es genügt, einzelne Informationen nur einmal zu verarbeiten. Die Beurteilung von Redundanz richtet sich nach objektiven Gesichtspunkten und ihre Beseitigung stellt einen vollständig reversiblen Prozess ohne Qualitätsverlust dar.

Irrelevanzreduktion (Reduzierung des Unnötigen) hingegen stellt bereits eine Beschneidung des Original-Informationsgehalts dar. Als Beispiel dafür haben wir bereits die Unterabtastung der Chrominanz-Anteile beim Komponentensignal kennen gelernt (4:2:2). Es werden Signalanteile beseitigt, die bei späterer Decodierung nicht wiederhergestellt werden können. Die Entscheidung, welche Informationen als irrelevant erachtet werden, wird praxisgerecht erwogen und hängt von der jeweiligen Anwendung ab. Wir werden dazu noch weitere Beispiele kennen lernen. Fehlen dem Empfänger Informationen, so sind auf Senderseite bereits relevante Anteile entfernt worden.

Obwohl alle hier vorgestellten Verfahren zur Datenreduktion eigenständig anwendbar sind und darüber hinaus verlustfrei arbeiten können, wird in der Praxis zur Speicherung und Übertragung eine Kombination aus mehreren verwendet und dabei meist zunächst eine Irrelevanz- und dann eine Redundanzreduktion durchgeführt.

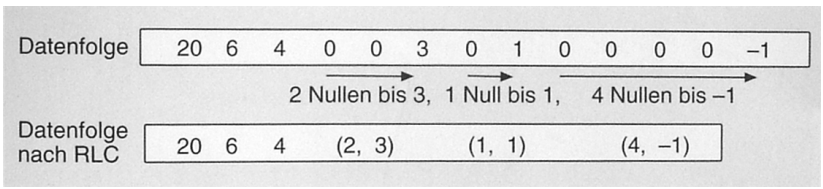
17.3 VLC und RLC (RLE)

Verfahren der nun vorgestellten Art werden gemäss des eben dargestellten Sachverhalts auch als Entropiecodierungen bezeichnet, ihre Aufgabe ist die Beseitigung von Redundanz. Bei der variablen Längencodierung (Variable Length Coding, VLC) wird zur Effizienzsteigerung wie beim Morsen ein Alphabet-Wechsel vorgenommen, d. h. dass durch eine Umcodierung den mit dem Original-Datenstrom angelieferten Symbolen neue Symbole zugewiesen werden, und zwar umso kürzere, je häufiger sie sind. Die verbreitetste Methode ist dabei die Anwendung des Huffman Codes, in der Abbildung einmal am Beispielwort 'ERDBEERE' demonstriert.



Zunächst werden für alle Zeichen i die Auftrittwahrscheinlichkeiten p(i) ermittelt und nach Häufigkeit geordnet. Die Einzelwahrscheinlichkeiten der seltensten Zeichen, hier D und B, werden zu einer gemeinsamen Wahrscheinlichkeit addiert (Knotenpunkt). Dieser Prozess wird so lange wiederholt, bis nur noch zwei Häufigkeiten übrig bleiben, denen die Codeworte I und 0 zugeordnet werden. Dann werden alle Stufen zurückverfolgt, wobei jeder Knotenpunkt die Zweigstelle zum nächst längeren Codewort bildet. Daraus resultiert eine Bitzuweisung, bei der wir den grössten Wahrscheinlichkeiten die kürzesten Codeworte zuweisen. Auf der Empfängerseite muss der Prozess zur Decodierung bekannt sein. Dies ist in Form einer festgelegten Huffman-Tabelle möglich, die dem Encoder bekannt ist oder im Datenstrom mitgeliefert wird.

Die Effektivität der Datenreduktion kann noch weiter durch die Lauflängencodierung (Run Length Coding, RLC oder RLE: Run Length Encoding) gesteigert werden, Dabei



wird mehreren gleichen Werten, die in Folge auftreten, ein einzelnes, zusammenfassend beschreibendes Symbol zugeordnet, anstatt jeden Wert separat zu übertragen. (⇨ siehe auch vorgängiges Kapitel RLC)

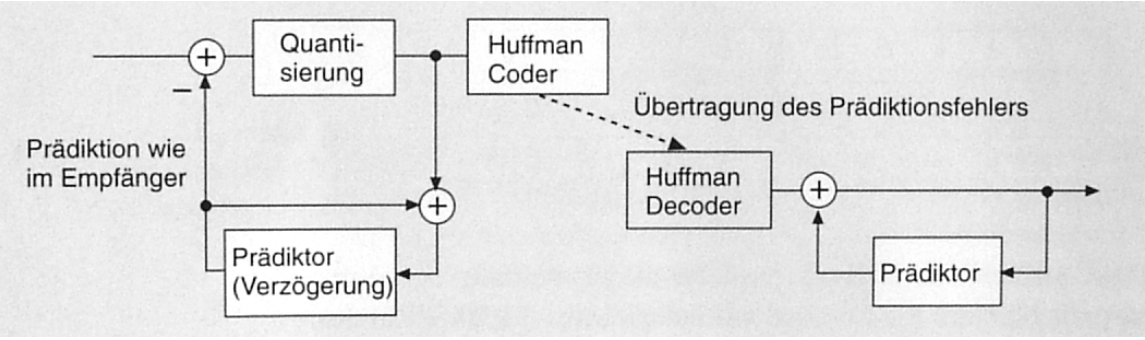
ZB. kann, wie wir gleich noch sehen werden, mit der Quellencodierung ein Datensatz für die RLC so vorbereitet werden, dass häufig viele Nullen aufeinander folgen. Die Datenrate kann dann durch Unterdrückung sich wiederholender Sequenzen erheblich reduziert werden, indem immer die Anzahl der Nullen mit dem nächsten Wert ungleich null zusammengefasst wird. Die Lauflängen-codierung finden wir nicht selten auch unter dem Begriff Run Length Encoding mit dem Kürzel RLE (oder RLC).

17.4 DPCM (Differenzielle Puls Code Modulation)

VLC und RLC können völlig unabhängig von anderen Verfahren eingesetzt werden. Für die weitere Steigerung der Leistungsfähigkeit ist es jedoch wünschenswert, vorbereitende Massnahmen zu treffen, die die Entropiecodierungen im Anschluss daran möglichst effizient arbeiten lassen. Dies können wir mittels Prädiktionscodierung im Zeitbereich oder mit Hilfe einer Transformationscodierung im Frequenzbereich erzielen. Die dazu heute am häufigsten eingesetzten Verfahren sind die DPCM bzw. die DCT.

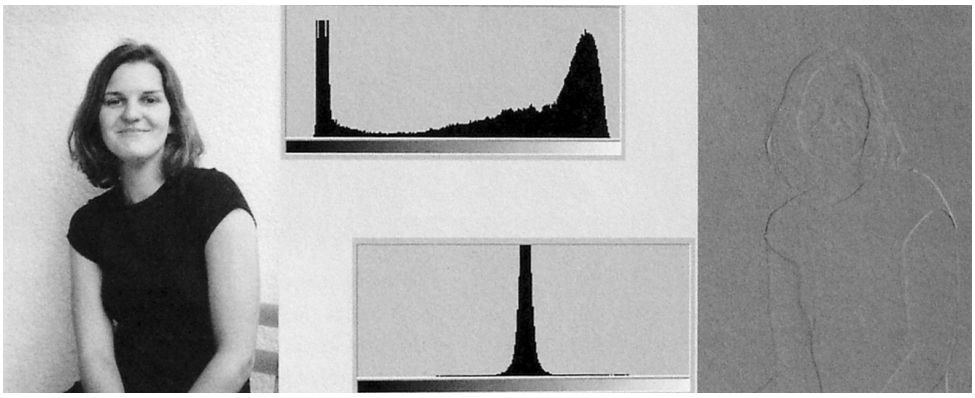
Bei der Differenziellen Puls Code Modulation (DPCM) nutzen wir hohe örtliche Korrelationen im Bild aus, Regelmässigkeiten, die sich über weite Teile des Bildes erstrecken. Das geschieht in der Art, dass nur die Differenzwerte zwischen benachbarten Bildpunkten ermittelt werden. Haben zwei Pixel beispielsweise die Grauwerte 194 und 209, die mit 8 Bit codiert werden können, so ergibt die Differenz einen Wert von 15, welcher mit nur 4 Bit dargestellt werden kann. Lediglich hoch-frequente Bildanteile, nämlich Kanten, verursachen wenige hohe Werte. Die Differenzbildung können wir so betrachten, dass aus den übertragenen Bildpunkten eine Vorhersage (Prädiktion) für den jeweils nächsten gebildet wird. Man nimmt im einfachsten Fall an, der nächstfolgende Wert wäre genau gleich dem, der gerade übertragen worden ist. Vom nächsten Signalwert (209) wird die Prädiktion (194) subtrahiert, und es wird nur die Abweichung vom wirklichen Wert, der so genannte Prädiktionsfehler (15), übertragen.

Bis hier arbeitet DPCM vollkommen verlustfrei. Da sich aber keine hohe Codier-Effizienz einstellt, werden die Daten zur weiteren Reduktion des Datenstroms noch quantisiert. Dadurch entsteht ein Fehler, der im Decoder aufaddiert wird und das Signal über kurz oder lang unbrauchbar macht. Um dies zu vermeiden, wird bereits im Sender eine Decodierung vorgenommen, die genau der Decodierung im Empfänger entspricht. Das decodierte Signal wird wieder der Additionsstufe zugeführt, so dass die Eingangsbedingungen am Decoder vorweggenommen sind.



Grundprinzip der DPCM mit Quantisierung

Als Beispiel für diese Art der DPCM zeigt das nächste Bild ein Graustufenbild und das zugehörige Differenzbild, sowie im Histogramm jeweils die zugehörigen Häufigkeitsverteilungen der Grauwerte:



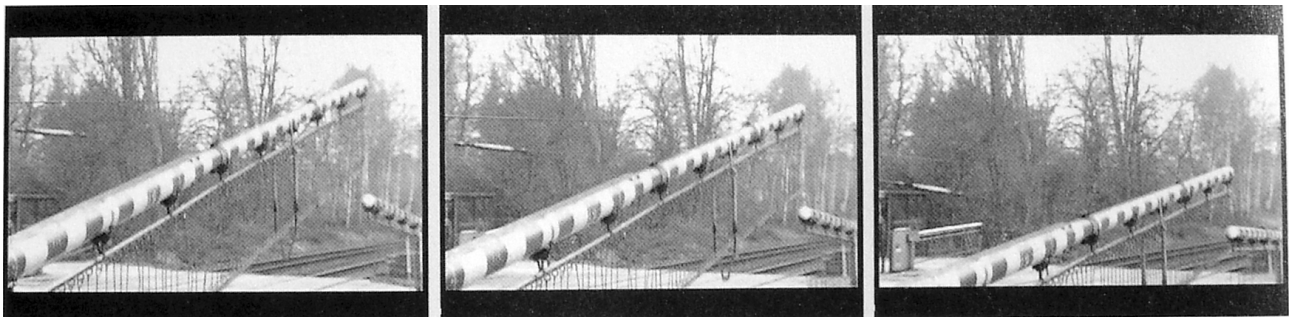
Zur Gewinnung des Differenzbild wurde das Bild horizontal um ein Pixel verschoben und die Differenz zwischen verschobenen und unverschobenen Bild gebildet. Die Statistik der Zeichen-häufigkeit ist zu einer starken Konzentration um mittlere Grauwerte verändert worden, und die variable Längencodierung (RLE) kann nun sehr effizient arbeiten.

17.5 Intraframe und Interframe

Die bisher dargestellte Form der DPCM bezog sich allein auf Ähnlichkeiten zwischen Nachbarpixeln in einem Bild, man spricht von **Intraframe-Prädiktion**. Die Effektivität von DPCM kann gesteigert werden, wenn wir **mittels Interframe-Prädiktion** auch die Ähnlichkeit zwischen zwei benachbarten Bildern ausnutzen, denn zeitliche Korrelationen sind für gewöhnlich noch weitaus stärker vorhanden als örtliche.

In sehr vielen Fällen hängt die Änderung des Bildinhalts lediglich von wenigen Objekten ab, deren Bewegung zumeist auch noch voraussagbar sind und somit nach einem gewissen Muster erfasst und kompensiert werden können. Hat zB. ein Flugzeug oder ein geworfener Ball bis zur Bildmitte eine gewisse Bahn beibehalten, so können wir mit grosser Sicherheit den weiteren Verlauf dieser Bahn in den folgenden Bildern bis zum anderen Bildschirmrand schätzen. Mit dieser Abschätzung verschobener Bildteile benachbarter Bilder können Bewegungsvektoren für verschiedene Bildbereiche gewonnen werden. Zum Aufsuchen ähnlicher Bildteile wird gegenwärtig häufig das Blockmatching-Verfahren verwendet. Dabei werden Blöcke von typisch 16x16Pixel betrachtet. Beim Bildvergleich wird im Nachbarbild der Block mit der grössten Ähnlichkeit gesucht und dessen relative Verschiebung zum Ausgangsblock als Bewegungsvektor gespeichert.

Die nächste Abbildung zeigt als Beispiel dazu eine sich senkende Bahnschranke.



Im dritten Bild stellen wir dabei allerdings als Problem fest, dass das Hauptobjekt eine Schranke im Hintergrund bislang noch verdeckt hat. In diesem Fall funktioniert zwar die Bewegungskompensation, eine einwandfreie Prädiktion versagt aber dennoch, wenn das nun sichtbare Objekt im davor liegenden Bild noch nicht freigegeben war.

Zur Optimierung wird daher eine bidirektionale Prädiktion durchgeführt, die das aktuelle Bild nicht nur aus dem vorangegangenen, sondern auch dem ihm folgenden ermittelt. Obwohl das folgende Bild

noch in der Zukunft liegt und eigentlich gar nicht bekannt sein kann, gibt es die Möglichkeit zur bidirektionalen Prädiktion durch zeitliche Verzögerung auf der Coderseite d.h. das nächste Bild wird vor der Übertragung berücksichtigt. Anschliessend kann die Bildfolge vor der Übertragung geändert werden, so dass zunächst beide Nachbarbilder des gewünschten Bildes am Decoder vorliegen, wobei die Bilder duch Inegration aus den Differenzen zurückgewonnen und danach wieder in die richtige Reihenfolge gebracht werden.

17.6 P-Frames, B-Frames und I-Frames

Bilder die auf einfacher, **unidirektionaler Prädiktion** beruhen, werden **P-Frames (P)** genannt, die aus **bidirektionaler Prädiktion B-Frames (B)**.

Intraframe codierte Bilder ohne Vorhersage werden als **I-Frames (I)** bezeichnet.

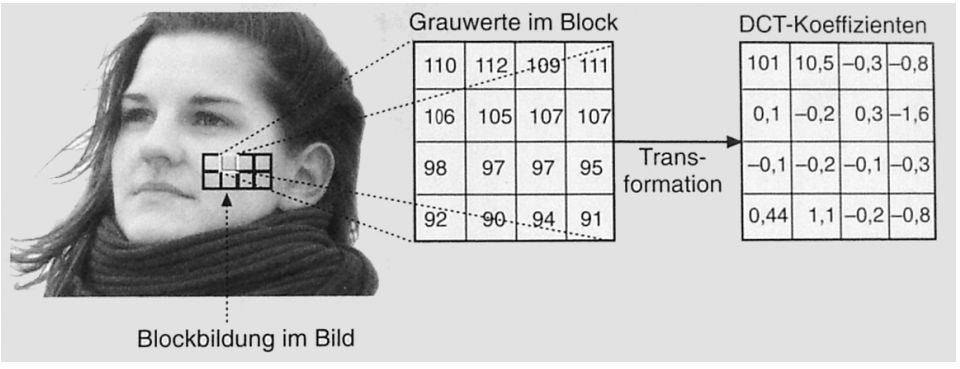
Im professionellen Videoschnitt ist zumeist der uneingeschränkte Zugriff auf jedes Einzelbild gefordert. Aufgrund der Umstellung der abfolgenden Bilder und ihre gegenseitige Abhängigkeit bei Interframe-Codierung können wir dieser Forderung nur mit reinen I-Frame-Codierung nachkommen. Da die DPCM dabei nicht effektiv genug arbeitet, wird dort zumeist die DCT angewendet.

17.7 DCT (Diskrete Cosinus Transformation)

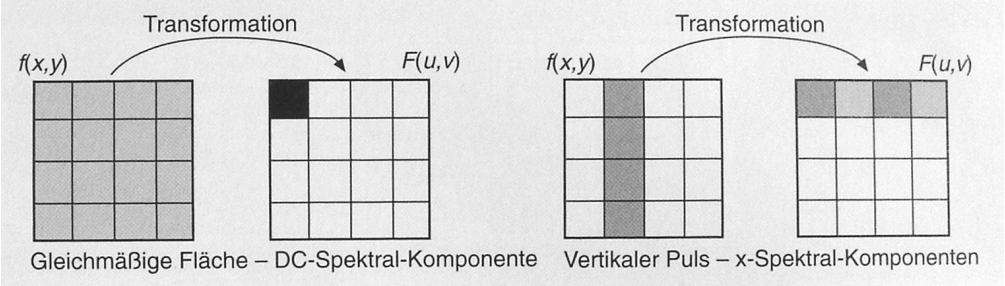
Während die DPCM Signale im Zeitbereich betrachtet, bereitet die DCT die Bearbeitung der Signale im Frequenzbereich vor. Sie basiert auf der Fourier-Reihen-Entwicklung, die als Frequenzanalysenverfahren den Verlauf periodischer Signale ausschliesslich aus der Summe unterschiedlich gewichteter Sinus- und Kosinus-Schwingungen annähert. Zur Approximation aperiodischer oder diskreter Signale existieren die kontinuierliche Fourier-Transformation bzw. die Diskrete Fourier-Transformation (DFT). Mit der DFT als Basis sind weitere mathematische Operationen entwickelt worden wie zB. die Fast-Fourier-Transformation (FFT) und eben die DCT. Deren Rechenvorschriften eignen sich besonders gut für die Bildverarbeitung.

Das Frequenzspektrum eines Videosignals kann dreidimensional dargestellt werden. Neben der gewohnten Frequenz bezüglich der Zeit (Bilder/sec) ergeben sich Ortsfrequenzen bezüglich der Koordinaten der Bildfläche. Tiefe Ortsfrequenzen repräsentieren grobe Bildstrukturen und "langsame" Bildübergänge, hohe Frequenzen werden dagegen durch feine Strukturen und harte Übergänge (Kanten) verursacht. Da in natürlichen Bildern grobe Strukturen weit häufiger vorkommen als feine, findet eine Konzentration auf niederfrequente Komponenten statt. Falls hochfrequente Anteile zur Darstellung mittels Quantisierung ungenauer dargestellt oder weggelassen ganz werden, ist der Fehler oft irrelevant.

Zur DCT wird das gesamte Bild in quadratische Blöcke, zB. 4x4 Pixel, aufgeteilt:



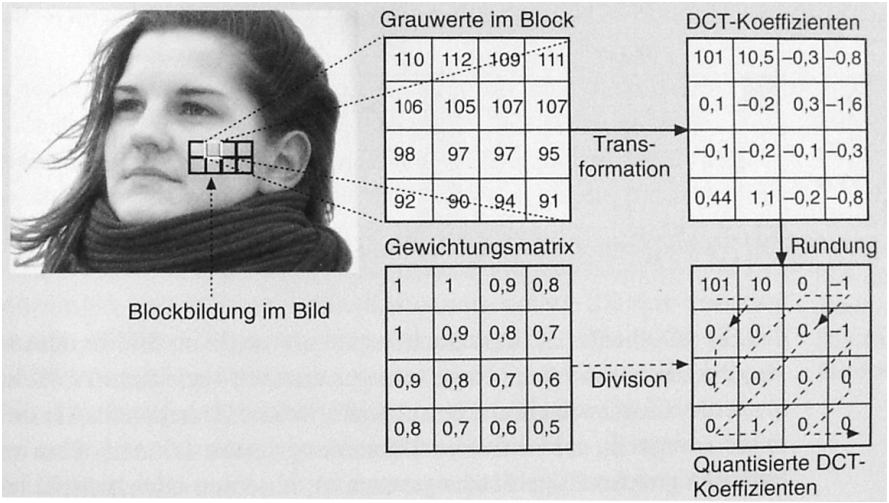
Die Transformation eines Blocks von Bildpunkten dient der Überführung in einen gleich grossen Block von Transformationskoeffizienten, die nun anstelle der eigentlichen Pixelwerte übertragen werden. Lesen wir die Matrizen im direkten Vergleich, mutet es zunächst so an, als ob jeder Grauwert einen zugehörigen Partner bei den DCT-Koeffizienten hätte. Dies ist aber keineswegs der Fall. Die Zahlenwerte lassen den direkten Vergleich nicht zu, denn in die Berechnung jedes einzelnen Transformationswertes gehen alle Werte des zu transformierenden Blocks ein.



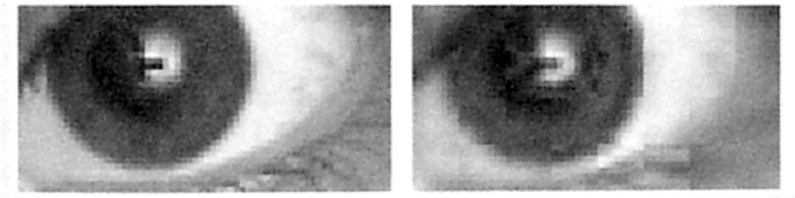
Obige Abbildung verdeutlicht noch einmal beispielhaft, welche Ursachen die Transformationswerte in der zeitlichen Darstellung haben können. Eine vollkommen gleichmässige Fläche verursacht nur einen Gleichanteil, während eine vertikale Kante gemäss einer Sprungfunktion in der Horizontalen des Koeffizientenfeldes abgebildet wird. Wir sehen, dass neben den grossen Werten der niederfrequenten Anteile nur kleine und dazu sich häufig wiederholende Werte zu finden sind. Letzteres schafft die Voraussetzung für eine effektiv arbeitende variable Längencodierung (RLC). Die kleinen Werte eignen sich hervorragend für einen weiteren Schritt der Reduktion, nämlich die Quantisierung.

17.8 Quantisierung

Bis zu diesem Zeitpunkt ist die DCT noch vollständig reversibel und verlustfrei. Ist aber grössere Effizienz gefragt, gibt es die Möglichkeit, die Werte der Transformationsmatrix einer gewichteten Quantisierung zu unterwerfen, d.h. die Werte werden geteilt und gerundet :



Dies ist keinesfalls so zu verstehen, dass zunächst die Quantisierung und dann eine Gewichtung erfolgt oder umgekehrt, Gewichtung bedeutet lediglich, dass alle Koeffizienten mit unterschiedlicher Stärke quantisiert werden können. Für die Werte der Gewichtung werden üblicherweise Tabellen verwendet, die anhand psycho-physiologischer Experimente ermittelt worden sind, so dass die Beschneidung der Frequenzbereiche nicht zu einem visuell störenden Problem wird. Den hochfrequenten Anteilen schenken wir dabei weniger Beachtung, sie werden mit hohen Divisoren bzw. kleinen Faktoren grob bewertet und häufig ganz zu Null herabgesetzt, so dass im Anschluss das Run Length Coding (RLC) optimal eingesetzt werden kann. Auch wenn die entstehenden Rundungsfehler klein sind, so wird das Signal durch den Quantisierungseffekt verfälscht. Ein typischer Fehler, der bei der DCT auftritt, ist der Blocking-Effekt, d.h. dass in Abhängigkeit von den Bewertungsmatrizen die Blockgrenzen sichtbar in Erscheinung treten.



Originalbild Artefakte (Blocking-Effekt) bei DCT

17.9 Hybride DCT

Zur weiteren Leistungssteigerung können wir eine Kombination aus DPCM und DCT verwenden, die als hybride DCT bezeichnet wird.

Eine dreidimensionale DCT (Bewegung als dritte Dimension) wäre so speicherintensiv und rechenaufwändig, dass man sich hier entschlossen hat, die DCT für die Intra- die DPCM für die Interframe-Codierung einzusetzen. Dazu wird zunächst die Differenz zweier aufeinanderfolgenden Bilder bestimmt und das Differenzenbild anschliessend transformationscodiert. Daraus resultiert eine sehr effektive Datenreduktion, die ermöglicht, dass Bilder in PAL-Qualität mit 5Mbps übertragen werden können. Bei akzeptabler Qualität sind sind Kompressionsfaktoren bis 100, bei geringen Qualitätsansprüchen auch Faktoren über 500 erreichbar, Videobilder geringerer Auflösung können dann mit 64kbps übertragne werden. Die hybride DCT ist die Basis der Video-Quellcodierung beim MPEG-Standard und damit auch die Basis für das digitale Fernsehen. Ein Zugriff auf die Videodaten und somit ihrer Bearbeitung ist nach dieser Codierung nur dann noch möglich, wenn in regelmässigen Abständen ein unabhängiges Bild, sprich ein rein intraframe-codiertes Bild eingefügt wird.

Die DCT ist heute die Bildcodierungsvariante mit der grössten Bedeutung.

Sie wird bei MPEG und JPEG ebenso eingesetzt, wie bei den speziellen Datenreduktionsverfahren für digitale Videorekorder, zB. Digital Betacam, IMX oder DV.

17.10 Wavelet-Transformation

Ein Nachteil der Video-Codierung mittels DCT ist die Tatsache, dass hierbei eine Blockbildung mit zweidimensionaler Transformation zum Einsatz kommt, die eine gezielte Rekonstruktion einzelner Pixel, Pixelgruppen oder gar des gesamten Bildes ohne vorherige vollständige Rücktransformation nicht unmittelbar zulässt. Bei einer groben Quantisierung des Bildinhaltes und einer daraus resultierenden geringen Datenrate sind zudem deutliche Blockartefakte zu erkennen.

Ein neueres Verfahren zur Bildkompression ist die so genannte Diskrete Wavelet-Transformation (DWT). Wie auch bei der DCT wird das Bild dabei zur Codierung in seine Spektralanteile zerlegt. Eine Blockbildung ist aber nicht vonnöten, denn die DWT wird auf das vorliegende Bild als Ganzes angewendet. Bei DWT-codierten Bildern ist es möglich, ein Bild mit wachsender Grösse oder steigender Qualität aufzubauen, oder aber sortiert nach bestimmten hervorzuhebenden Bildbereichen, sogenannten "Regions of Interest".

18 JPEG

Die Joint Photographic Experts Group (JPEG) ist eine Arbeitsgruppe der internationalen Standardisierungsorganisation ISO/IEC, die es sich mit ihrer Gründung im Jahre 1986 zum Ziel gesetzt hat, in einem offenen Standard einen universellen Algorithmus für die Einzelbildcodierung zu schaffen. Besondere Rücksicht auf Videobilder wurde dabei nicht genommen.

Das 1992 eingeführte Verfahren beruht auf reiner Intraframe-Codierung. Das Bild wird in Blöcke von 8x8Pixeln Grösse zerlegt, mit der DCT in den Frequenzbereich transformiert, anschliessend werden die Koeffizienten mit der Quantisierungsmatrix gewichtet. Als Besonderheit werden bei JPEG gewisse Koeffizienten aus der Quantisierungsmatrix einer DPCM unterzogen.

Der Datenreduktionsfaktor wird zunächst durch die Stärke der Quantisierungsmatrix bestimmt. Um die Koeffizienten für die VLC (Variable Lengt Coding) und RLC (Run Length Coding) in eine günstige, eindimensionale Reihenfolge zu bingen, werden die Werte mit Hilfe des so genannten Zick-Zack-Scannings ausgelesen. Auf diese Weise erreichen wir, dass zum Ende der Zahlenfolge sehr viele Nullen auftreten, die mit der RLC sehr effektiv bearbeitet werden können. Schliesslich wird die Huffmann-Codierung durchgeführt, um einen optimalen, den Auftrittswahrscheinlichkeiten angepassten Code zu finden.

JPEG beschreibt ein Einzelbildcodierverfahren, bei dem keine Echtzeitverarbeitung gefordert ist. Soll die Datenreduktion aber doch für Videobilder eingesetzt werden, so müssen die Bilder nacheinander in Echtzeit komprimiert werden, was als Motion JPEG (M-JPEG) bezeichnet wird.

18.1 JPEG2000

Bereits im März 1997 wurde mit JPEG2000 auf Grundlage der Wavelet-Transformation die Entwicklung eines Standards für die Kompression von Standbildern gewonnen. Hier sollte nicht nur eine höhere Kompressionsrate als beim Vorgänger JPEG ermöglicht werden, sonder auch andere Kriterien wie die Möglichkeit verlustfrei oder verlustbehaftet zu komprimieren oder einen wahlfreien Zugriff auf den Datenstrom. (Beliebiger Zugriff auf bestimmte Bildinhalte). Die wichtigste Eigenschaft aber ist die progressive Übertragung, die es ermöglicht, ein Bild mit steigender Auflösung oder Grösse zu übertragen, mit der Option, den Datenstrom jederzeit nach Belieben abbrechen zu können oder aus einem bestehenden Bild mit hoher Auflösung eine stärker komprimierte Datei ohne den Aufwand einer Neucodierung zu extrahieren. Beim älteren JPEG–Format ist eine Bild-Decodierung nicht möglich, ohne die Daten in ihrer Gänze vorliegen zu haben.

Vergleich von Original (Links), JPEG (Mitte) und JPEG2000 (Rechts)

18.2 Formatübersicht Photo und Graphik

FileType MAC/WIN	Farbmodell	Icc-Profile	Anz. Kanäle	Bittiefe	Kompression	max. Bildgrösse	Plattform
TIFF (Tagged Image File Format)							
TIFF/.TIF	S/W, Graustufen RGB, CMYK, Lab, Indizierte Farben, Volltonfarben	ja	1,3 oder 4 Farb-, 20 Alphakanäle	1 bis 16Bit pro Kanal	ohne, RLE, LZW, CCITT Gr.3,4, JPEG	2^{32} Pixel	MAC, WIN, UNIX, u.a.
JPEG (Joint Photographic Expert Group)							
JPEG/.JPG	Graustufen, RGB, CMYK	ja	3 oder 4 Farb-, keine Alphakanäle	8Bit pro-Kanal	DCT	-	MAC, WIN, UNIX
JPEG2000 (Joint Photographic Expert Group 2000)							
JP2/.JP2	Graustufen, RGB, CMYK	ja	256	1 bis 16Bit pro Kanal	Wavelet	-	MAC, WIN, UNIX
GIF (Grafics Interchange Format)							
GIF/.GIF	Indizierte Farben	nein	1	256 Farben	LZW	-	MAC, WIN, UNIX
BMP (Bitmap Format)							
BMP/.BMP	RGB, Graustufen, S/W, Indizierte Farben	nur WIN ICM	1 oder 3 Farb-, keine Alphakanäle	1 bis 8Bit pro Kanal 32Bit Pixel	keine oder RLE	32000 x 32000 Pixel	MAC, UNIX, MS-DOS, WIN, OS2

PSD (Photoshop Dokument)							
8BPS/.PSD	S/W, Grau-stufen, RGB, CMYK, Lab, Duplex, Mehrka-nal, Volltonfarbe	ja	1,3 oder 4 Farb-, 20 Alphakanäle	1 bis 16Bit pro Kanal	ohne oder RLE	30000 x 30000 Pixel	MAC, WIN, UNIX
EPS (Encapsulated PostScript)							
EPS/.EPS	S/W, Graustufen RGB, CMYK, Lab, Duplex	ja	1,3 oder 4 Farb-, keine Alphakanäle	1 bis 8Bit pro Kanal	ohne, JPEG	-	MAC, WIN, UNIX
PNG (Portable Network Graphics)							
PNG/.PNG	RGB, Graustufen, Indizierte Farben	be-dingt	3 + 1 Alpha-kanal	1 bis 16Bit	LZ77 Variante	2^{31} Pixel	MAC, WIN, UNIX

Im Laufe der Zeit wurden sehr viele Formate entwickelt. Die obige Tabelle zeigt nur die wichtigsten und gängigsten Formate zur Zeit der Drucklegung.

18.3 DV-Algorithmus

Dieser Algorithmus ist für das DV-MAZ-Verfahren entwickelt worden und daher speziell auf die für die Bandformate erforderliche konstante Datenrate ausgelegt. Da sich das Format DV (Digital Video) im Laufe der Zeit in den Varianten DVCAM und DVCPro auch im Produktionsbereich (zB. TeleZüri) etabliert hat, ist der DV-Algorithmus inzwischen ein wichtiges Datenreduktionsverfahren neben MPEG geworden.

Wie JPEG beruht auch DV auf reiner Intraframe-Codierung mittels DCT und Blöcken von 8x8 Bildpunkten. Im Gegensatz zu JPEG ist DV aber für Videoanwendungen optimiert. DV arbeitet mit 8Bit Auflösung. Aus dem Eingangssignal nach ITU-R 601 wird zunächst ein 4:2:0 (PAL) oder 4:1:1 (NTSC) unterabgetastetes Signal gewonnen. Für DV in Europa (PAL) erzeugt die Y-Abtastung $720 \cdot 576$ und die UV-Abtastung $360 \cdot 288$ Bildpunkte, so dass pro Bild $90 \cdot 72$ Luminanz- und $45 \cdot 36$ Chrominanzblöcke mit je $8 \cdot 8$Pixel vorliegen.

Zur effektiven Verarbeitung wird bei DV bei schnellen veränderlichen Bildinhalten vom Voll- auf den Halbbildmodus umgeschaltet. Der Vollbildmodus verwendet die ineinander geschobenen Halbbilder bei der Blockbildung, während im Halbbildmodus der 8x8Block in zwei 4x8Blöcke aufgeteilt wird, die jeweils nur die Information aus einem Halbbild tragen, so dass die Ähnlichkeit innerhalb des Bildes maximiert wird. Die Umschaltung wird von einem Bewegungsdetektor gesteuert und kann für einzelne Bildbereiche separat vorgenommen werden. Um die YUV-Verarbeitung zu vereinfachen, fassen wir 4 Luninanzblöcke und je einen Chrominanzblock zu einem Makroblock zusammen. Ein Bild enthält damit $45 \cdot 36$ Makroblöcke, die im so genannten Intra-Frame-Shuffling-Prozess systematisch verwürfelt werden. Je 5 Makroblöcke aus weit entfernten Bildbereichen werden damit zu einem Superblock zusammengefasst. Dies hat den Zweck, schon vor VLC und RLC eine Entropiecodierung zu erreichen, indem die durchschnittliche Redundanz angeglichen wird.

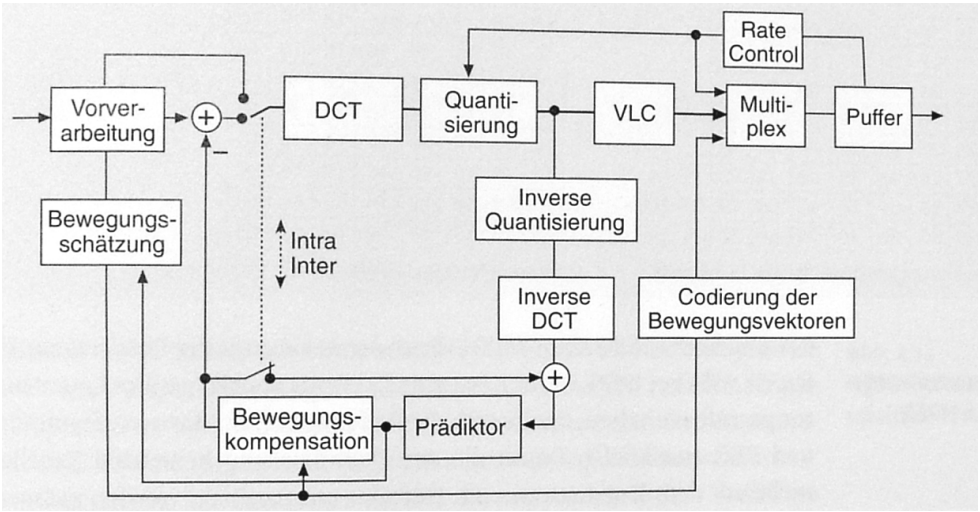
Die eigentliche Datenreduktion stammt wie bei den meisten Verfahren aus der kombinierten Anwendung von DCT, VLC und RLC, wobei die Datenrate vom Bildinhalt abhängig ist. Um die Datenrate konstant zu halten, wurde bei DV die so genannte Feed-Forward-Steuerung implementiert, die vor der eigentlichen Datenreduktion bestimmt, welche Wahl aus einer vorhandenen Menge definierter Quantisierungstabellen getroffen wird, um diesbezüglich ein optimales Ergebnis zu erzielen, d.h. eine Datenrate, die am dichtesten an dem durch das Aufzeichnungsverfahren gesetzten Limit liegt. DV definiert wie JPEG eine reine Bildcodierung, der Begleitton wird per Stereo-PCM mit 16Bit und 48kHz bei 2 Kanälen bzw. 12Bit und 32kHz bei 4 Kanälen aufgezeichnet.

19 MPEG

Wie JPEG ist MPEG ebenfalls ein Komitee der ISO/IEC, gegründet im Januar 1988. Seine Begriffsfindung leitet sich aus dem Arbeitsziel ab: **MPEG heisst "Moving Picture Experts Group"**. Mit MPEG sind Codierbedingungen definiert und sie legt ein Fileformat fest, dass die universelle Anwendbarkeit und Austauschbarkeit in Multimedia-Systemen anstrebt. Ursprünglich war MPEG rein für die Videobild-Verarbeitung gedacht, alsbald wurden jedoch auch Verfahren für die Audio-Codierung und die Multiplexbildung von Video und Audio zu gemeinsamen Datenströmen erarbeitet.

19.1 MPEG-Codierung

Die MPEG-Videocodierung beruht auf dem hybriden DCT. Um gegenüber JPEG und DV eine höhere Datenreduktionseffizienz zu ermöglichen, werden hier nun mittels Interframe-Codierung neben den örtlichen auch die zeitlichen Korrelationen per DPCM ausgenutzt.



MPEG2-Coder

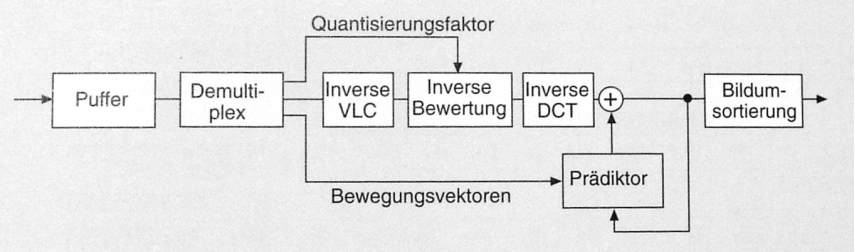
Wie bei JPEG ist im Kern eine DCT mit nachfolgender Quantisierungsstufe enthalten, an die sich die Redundanzreduktionsstufe mit VLC und RLC anschliesst. Damit die zur Quantisierung benutzten Tabellen nicht mit dem Datenstrom zum Decoder mitgeschickt werden müssen, sind für MPEG Standard-Tabellen festgelegt worden. Nach der Quantisierungsstufe kann das Signal verzweigt werden, und mit reziproker Gewichtung und inverser DCT werden die Bildpunkte zurückgewonnen. Im Encoder wird, wenn keine I-Frame-Codierung gefordert ist, aus dieser Information ein Prädiktionsbild ermittelt, woraus zusammen mit dem nächsten Bild am Coder- Eingang die DPCM-Differenz gebildet wird. Wie bereits im Abschnitt über DPCM erläutert, kann die Prädiktion mit Hilfe von Bewegungsvektoren optimiert werden (Bewegungskompensation).

MPEG-2 unterstützt diesbezüglich auch die dort beschriebene bidirektionale Prädiktion, die auf Einbeziehung der Nachbarbilder vor und hinter dem aktuellen Bild beruht. Hierdurch steigt die Datenreduktionseffizienz, da ein bidirektional prädizierter Block mit etwa der Hälfte eins unidirektional prädizierten codiert werden kann.

Die Information der Bewegungsinformation stammen aus einer Verarbeitungsstufe zur Bewegungsschätzung, die zumeist mit dem bereits beschriebenen Blockmatching-Verfahren arbeitet.

Bei MPEG ist , anders als bei DV mit seiner Feed-Forward-Steuerung, am Ende der Coder-Verarbeitung meist ein Puffer zu finden, der bei Über- oder Unterlastung den ankommenden Datenstrom in der Art steuert, dass bei zu hohem bzw. zu niedrigen Datenaufkommen eine gröbere bzw. feinere Quantisierung veranlasst wird, um einen gleichmäsigen Datenstrom zu erhalten.

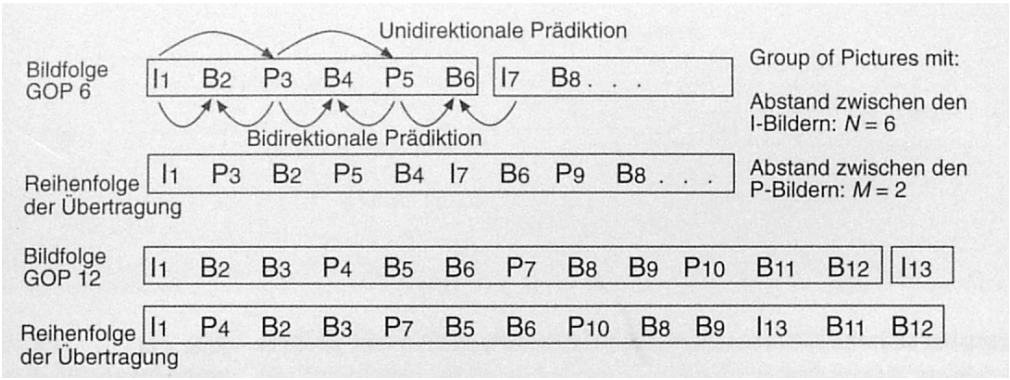
MPEG stellt im Gegensatz zu DV ein unsymmetrisches Verfahren dar, was bedeutet, dass der Arbeitsaufwand im Coder um ein Vielfaches höher ist als im Decoder :



MPEG2-Decoder

Obige Abbildung zeigt das Blockschaltbild des MPEG-Decoders. Auch hier werden die Daten zunächst in einen Pufferspeicher übernommen. Im Demultiplexer werden dann die Bewegungs-vektoren und die Quantisierungsinformationen abgetrennt und dem Prädiktor bzw. der Verar-beitungsstufe zur inversen Quantisierung zugeführt. Nachdem die Koeffizienten der DCT zurücktransformiert sind und die Vorhersage-werte zur Invertierung der DPCM aufaddiert wurden, steht die Bildinformation wieder zur Verfügung.

19.2 Group of Pictures (GOP)



Am Ausgang des Decoders wird, wie genau vor der Encodierung, eine Stufe zur Bildumsortierung durchlaufen. Sie ist erforderlich, weil nicht alle Bilder gleich codiert werden. Wie bereits beschrieben, ist die effizienteste Form die, die auf bidirektionaler Prädiktion beruht. Da die B-Frame-Codierung aber von Nachbarbildern abhängt, kann sie nicht für alle Bilder durchgängig beibehalten werden, sondern im Datenstrom müssen in regelmässigen Abständen auch unidirektional prädizierte P-Frames und I-Frames auftauchen, die nicht von Nachbarbildern abhängen, da der Decoder sonst keinen Einstiegspunkt finden kann. Auf diese Weise entstehen **zyklische wiederkehrende Bildergruppen**, die als Group of Pictures (GOP) bezeichnet werden.

Die GOP-Struktur wird bei der Encodierung festgelegt, je mehr P- und B-Frames auftauchen, umso effektiver ist die Datenreduktion. In der Praxis wird häufig mit einer Folge von 12 Bildern (GOP12) mit einem I-, drei P- und viermal zwei B-Frames gearbeitet. Die Bildreihenfolge wird nun für die Übertragung mit einer Bildumsortierungsstufe verändert, denn dem Decoder müssen zunächst die Nachbarbilder eines B-Frames bekannt sein, damit diese decodiert werden kann.

Die Verwendung sehr langer GOP's ist eingeschränkt, da sich mit der Erhöhung der Anzahl der B-Frames auch die Verzögerungszeit zur Decodierung erhöht.

Um die Effizienz der Verwendung zeitlich abhängiger Bilder abzuschätzen, kann man zB. den Vergleich mit einem DV-Datenstrom heranziehen. Hier liegen bei 25Mbps nur I-Frames vor, die mit einem Kompressionsfaktor von 5:1 reduziert sind, so dass jedes Bild 1Mbit benötigt und dabei eine hohe Qualität aufweist. Wenn nun bei MPEG jedes Bild 1/3 des I-Frames und jedes B-Bild noch einmal die Hälfte eines P-Bildes umfasst, so entsteht für GOP12 ein Datenumfang von

$(I + 3P + 8B) = (I + 3 \cdot 1/3 + 8 \cdot (1/6) = (I + 3/3 + 8/6) = 3{,}33; \quad 3{,}33 \cdot 1\text{Mbit} = 3{,}33\text{Mbit}$

bzw. eine Datenrate von 6,9Mbps, bei mit dem DV-Ausgangsmaterial vergleichbarer Bildqualität.

19.3 MPEG-1

Der Standard MPEG-1 (ISO/IEC 11172) wurde 1992 eingeführt und zwar auf kleine Datenraten bis zu 1,5Mbps ausgelegt. Da die Videoanwendung auf ein Bildformat von 352 · 288 (50Hz) bzw. 352 · 240 (60Hz) Bildpunkten beschränkt ist, welches auch als Source-Input-Format (SIF) bezeichnet wird, ist MPEG-1 mit dem Aufkommen der weiteren MPEG-Standards relativ unbedeutend geworden. Die Chrominanz wird sogar auf 176 · 144 bzw. 176 · 120 Pixel reduziert. Auf der Wiedergabeseite werden die 720 · 576 Bildpunkte dann durch Interpolation aus den übertragenen Werten zurückgewonnen.

19.4 MPEG-2

MPEG-2 (ISO/IEC 13818) stellt seit 1994 das solide Rückgrat der MPEG-Familie dar. Viele praktische Anwendungen speichern heutzutage im Vertrauen auf robuste Qualität Videosignale in MPEG-2, so zB. die DVD-Video, die Super VideoCD (SVCD) und das noch recht junge digitale MAZ-Format IMX für den professionellen Studiobereich. Auch bei Übertragung per Digital Video Broadcasting kommt es zum Einsatz, denn es ist eigentlich für die Distribution optimiert, wird aber zunehmend auch in den Bereichen Produktion und Editing verwendet.

Die Video-Codierung nach MPEG-1 und MPEG-2 beruht auf den in diesem Abschnitt allgemein vorgestellten Codierverfahren auf Basis der hybriden DCT und Blockbildung mit 8 · 8 Bildpunkten. Im Gegensatz zu MPEG-1 sind bei MPEG-2 aber mehrere Bildauflösungen zwischen dem SIF von MPEG-1 über Standardauflösung (720 · 576 Pixel) bis hin zu HDTV (1920 · 1152 Pixel) zugelassen. In den weitaus meisten Fällen wird mit 4:2:0-Abtastung gearbeitet.

MPEG-1 erlaubt nur die progressive **(progressiv = auf der Vollbildverarbeitung beruhend)** Bildverarbeitung.

MPEG-2 kann wie der DV-Algorithmus bei der Encodierung zwischen dem Voll- und dem Halbbildmodus umschalten.

Der Vollbildmodus nutzt die Ähnlichkeiten der Nachbarzeilen aus zwei Halbbildern, die bei ruhender Bildvorlage auftreten und beruht daher auf 8 · 8 Bildblöcken, die aus beiden Halbbildern stammen. Bei starker Bildbewegung im Blockbereich sind die Halbbilder jedoch so stark unterschieden, dass es günstiger ist, auf die Teilbildverarbeitung überzugehen und die Zeilen umzuordnen, dass nur Ähnlichkeiten innerhalb eines Teilbildes ausgenutzt werden.

MPEG-2 stellt bezüglich der Bildauflösung und weiterer Funktionen eine so reiche Auswahl an Möglichkeiten zur Verfügung, dass dafür gesorgt wurde, diese auf einige wenige, häufig verwendete Funktionalitäten (Profiles) bei verschiedenen Auflösungsstufen (Levels) zu beschränken.

	Simple Profile 4:2:0 (no B-Frames)	**Main Profile** 4:2:0	SNR Scalable Profile 4:2:0	Spatially Scalable Profile 4:2:0	High Profile 4:2:0 or 4:2:2
High Level < 60 fps		1920 · 1152 < 80 Mbps			1920 · 1152 < 100 (80) Mbps
High 1440 < 60 fps		1440 · 1152 < 60 Mbps		1440 · 1152 < 60 (40) Mbps	1440 · 1152 < 80 (60) Mbps
Main Level < 30 fps	720 · 576 < 15Mbps	***720x576 < 15 Mbps***	720 · 576 < 15 Mbps		720 · 576 < 20 (15) Mbps
Low Level < 30 fps		352 · 288 < 4 Mbps	352 · 288 < 4 (3) Mbps		

Levels und Profiles bei MPEG-2 (Werte in Klammern geben Alternativen bezüglich der Skalierung an)

19.5 MPEG-4

Wir können nun die Vermutung anstellen, dass die Moving Picture Experts Group im Anschluss an die Entwicklung von MPEG-2 entweder mit den erreichten Ergebnissen zufrieden war oder einen weiteren Standard namens MPEG-3 in Angriff genommen hat. Beides ist nicht der Fall.

MPEG-3 war zwar zunächst als Erweiterung des MPEG-2-Standards im Hinblick auf HDTV-Signale geplant, jedoch umfasste MPEG-2 im Laufe seiner Entwicklung diese Fähigkeiten bereits, so das das vorhaben MPEG-3 gänzlich eingestellt worden ist. Dennoch hat die MPEG bereits 1993, also vor der letztendlichen Fertigstellung von MPEG-2, mit der Entwicklung eines Verfahrens begonnen, das gegenwärtig den neusten Stand dieser Art der audiovisuellen Codierung darstellt, nämlich MPEG-4 (ISO/IEC 14496). Version 1 wurde Anfang 1999 verabschiedet, Version 2 Anfang 2000. Die Abwärtskompatibilität bis hinab zu MPEG-1 wurde dabei gewahrt. MPEG-4 nutzt eine Szenenbeschreibungssprache, die Binary Format of Scenes (BIFS) genannt wird und auf der Skriptsprache VRML (Virtual Reality Modeling Language) basiert.

19.6 MPEG-7

MPEG-7 knüpft an die Errungenschaften von MPEG-4 an. Das Ziel ist dabei aber weder weiter erhöhte Codiereffizienz noch die Repräsentation von Inhalten überhaupt. Hier wird also keine Datenreduktion mehr verfolgt, sondern das Content-Management, die Verwaltung der Inhalte.

20 MPEG-Audio

Das hauptsächlich durch die Verbreitung über das Internet enorm bekannt gewordene Musikformat mit dem Dateikürzel **.mp3** ist Bestandteil der MPEG-Standardisierung.

Die technische Entwicklung dieser Form der Audio-Codierung wurde hauptsächlich vom Fraunhofer-Institut für Integrierte Schaltungen (IIS) durchgeführt, wo man ab 1987 bemüht war, für das DAB-Projekt EUREKA (Digitaler Hörfunk) einen effizienten Audio-Codec (Coder/Decoder) zu entwickeln. Einige Leute gehen irrtümlicherweise davon aus, dass mit MP3 die Abkürzung für MPEG-3 gemeint ist, obwohl es diesen Standard nicht gibt, oder sehen MP3 generell als Oberbegriff für die gesamte Audio-Codierung die MPEG zu bieten hat.

Die Moving Picture Experts Group aber definiert, obwohl ihr Name zunächst einen reinen Standard zur Videokompression vermuten lässt, in jedem ihrer Codierungsstandards mehrere Teile, unter denen sich immer mindestens die Parts Video, Audio und Systems finden lassen. Video und Audio können also gemäss MPEG separat codiert oder über die Multiplex-Strukturierung des Systems-Parts zu einem gemeinsamen Datenstrom zusammengefasst werden.

Das 1991 vorgestellte und heute bekannteste Audioformat der Familie mit dem Kürzel **MP3** steht für **MPEG-Audio Layer-3 (ISO/IEC 11172-3)**, es handelt sich dabei um die derzeit komplexeste und leistungsfähigste Ebene der Audio-Codierung gemäss MPEG, abgesehen vom neusten Verfahren **AAC. (Advanced Audio Coding)**

20.1 MPEG-Audio-Codierung

Bei Audiosignalen besteht prinzipiell die Möglichkeit, ähnlich wie bei Videosignalen, Datenreduktionsverfahren zu nutzen, welche die Autokorrelationen des Signals im Zeitbereich nutzen, wie zB. die Differentielle Pulse Code Modulation (DPCM) oder artverwandte Verfahren wie die Adaptive DPCM (ADPCM), die Delta-Modulation (DM) oder die Adaptive DM (ADM), bei denen, vereinfacht gesagt, zur Einsparung lediglich die Differenz zwischen zwei benachbarten Abtastwerten übertragen wird.

20.2 MPEG-2-AAC

Das 1997 eingeführte und nachträglich in den MPEG-2-Standard übernommene **Advanced Audio Coding (AAC, ISO/IEC 13818-7)** stellt laut Fraunhofer Institut die zweite Generation dieser Codierungsart dar. Zwar implementiert AAC grundsätzlich die gleichen Funktionsprinzipien wie sein Vorgänger Layer-3, ist jedoch **nicht mehr abwärtskompatibel**.

20.3 Dolby AC-3 (Dolby Digital)

Die Dolby Laboratories haben ein eigenes Codierverfahren entwickelt, dessen dritte Version AC-3 (Audio Coding Nr. 3) genau wie MP3 in der Allgemeinheit momentan einen grossen Bekanntheitsgrad geniesst. Dieses Verfahren unterscheidet sich in Idee und technischer Ausführung nicht wesentlich von denen anderer Entwickler. Dolby sah seine Aufgabe bezüglich Audiokompression im Zusammenhang mit Mehrkanalton für High Definition Television (HDTV). Seinen grossen Durchbruch erlangte das Verfahren jedoch beim digitalen Lichtton für Kinofilme, und ist heute des Weiteren, zusammen mit dem verstärkt nachrückenden DTS, die dominante Toncodierung für die DVD-Video, da die Filmproduktion auf diese Weise bequem in einem Durchgang auf dieses Medium umgesetzt werden kann.

20.4 ATRAC (Adaptive Transform Acoustic Coding)

Im Jahre 1992, also etwa 10 Jahre nach der Einführung der Compact-Disc (CD), stellte die Firma Sony ein Audiospeichersystem namens MiniDisc vor, eine 64mm Durchmesser umfassende Scheibe, die in ein quadratisches Kuststoffgehäuse (Cartridge) eingelassen ist und ca. 140MByte Speicherkapazität (=74Min. Spielzeit) aufweist. Das Arbeitsprinzip der Geräte beruht auf magneto-optischer Technologie, wie sie bereits von früheren Medien zur Datensicherung bekannt war, d.h. ein Laser erwärmt das Speichermedium auf eine Temperatur von knapp 200°C, und ein magnetisches Feld sorgt für das Beschreiben mit den Daten. Die Datenkompression von ATRAC kann mit dem Faktor 5:1 angegeben werden. Das ATRAC-Reduktionsverfahren findet übrigens auch beim SDDS-Kinoton (Sony Dynamic Digital Sound) Verwendung.

21 Multimedia-Fileformate für den PC

Die beiden am häufigsten anzutreffenden Formate im Consumer-Bereich sind Apple-Quicktime und Microsoft Windows Media, des weiteren werden oftmals auch RealMedia-Files genutzt. Die Wiedergabe-Software (Player) für diese Formate sind für die gängigen Plattformen (MAC-OS, Windows, Linux, SUN-Solaris) erhältlich, zum Teil sogar bereits vorinstalliert. Seit Beginn der 90-er Jahre haben sich sowohl Microsoft wie auch Apple darum bemüht, Heimanwender, Internetbenutzer wie auch der professionelle Sektor mit leistungsfähigen Anwendungen zu versorgen. Von keiner der beiden Firmen lässt sich behaupten, sie hätte das definitiv bessere Format. Die Entscheidung für das eine oder andere sollte je nach Anwendungsfall und Zielgruppe abgewogen werden, besonders mit Blick auf die jeweils unterstützten Codecs, die massgeblich Einfluss auf die Qualität des Produkts haben. Die Anwendungsszenarien können vielfältig sein, vom Abspielen auf einem lokalen Rechner bis zum Echtzeit-Streaming in Netzwerken.

21.1 Quick-Time

Quick-Time (QT) ist ein von Apple Computer Inc. entwickeltes und 1991 auf den Markt gebrachtes Format für multimediale Inhalte. Neben einem wachsenden Funktionsumfang hat Apple bis einschliesslich seiner 2003 veröffentlichten Version 6 stets die Abwärtskompatibilität bewahrt, ohne das das Format dabei rückständig oder ausgedient wirkt. Das Format bietet die Möglichkeit, eine Referenz auf verwendete Daten zu erstellen, ohne die Notwendigkeit, diese als neuen Medien-Datenstrom innerhalb des erstellten Files abzuspeichern. So können beispielsweise für einen Videoschnitt oder ein Layering, also einer schichtweisen Überlagerung mehreren Videoströme, lediglich Metadaten über das Zusammenspiel der verwendeten Bilder angelegt werden. Beim Abspielen muss dann auf deren externen Referenzquellen zugegriffen werden. Der Vorteil dieser Art des Umgangs mit Content-Daten (Content=Medialer Inhalt) liegt darin, dass eine mit QuickTime erstellte Datei u.U. sehr klein ausfallen kann, obwohl sie mehrere Gigabyte an Mediendaten verwaltet. Selbstverständlich ist weiterhin auch die herkömmliche, komplette Neuspeicherung aus dem Resultat des Editings möglich.

QuickTime weist eine kar nachzuvollziehende Entwicklung auf, so ist ab QT4 zB. der Import von Shockwave-Daten erlaubt. Neben der ebenfalls von Apple entwickelten QuickTime Virtual Reality (QuickTime VR, spezialisiert auf Rundum-Panorama) bietet QT viele aufbauende Applikationen (zB.

FinalCut), die es selbst im Datenaustausch bei professionellen Produktionen zur beliebten Anwendung haben reifen lassen.

Aufgrund der Tauglichkeit auf dem professionellen Gebiet begann sich auch die ISO (International Organization for Standardization) bei der Suche nach einem Grundstein für MPEG-4 für das vielseitige Format von Apple zu interessieren. Sie übernahm es dann 1998 als Schlüssel-Komponente zur Entwicklung ihres Standards, weshalb QuickTime 6 als erster wirklicher MPEG-4-Player genannt wird.

21.1.1 Daten in Quicktime

Ein QuickTime-File kann jede Art zeitbasierter Daten enthalten und kombinieren. Die Daten werden Spuren (Tracks) zugeordnet, in erster Linie natürlich Audio- und Videodaten, die über ihre Sample- bzw. Frame-Rate immer einen direkten Zeitbezug aufweisen. Als feste Referenz kann ihnen des weiteren auch ein Timecode zugeordnet werden. Dies ist nicht selbstverständlich, denn es lassen sich zwar viele Formate bei kritischen Übertragungssituationen synchronisieren, meistens weisen sie aber ihre eigenen paketierten Synchrondaten auf, so dass selten eine externe

Zeitreferenz herangezogen wird. Beim sogenannten Interleaving (=Verschachtelung), also im Falle, dass Audio- und Videodaten alternierend im Datenstrom vorliegen, ist das Abspielen ohne Referenz unkritisch, bei einer getrennten Speicherung des Audio- und Videostroms kann es durchaus passieren, dass Bild und Ton nach einiger Zeit auseinander laufen. Neben AV-Daten nimmt QuickTime auch Textebenen auf, denen ein Zeitbezug zugewiesen ist, und die dann als zB. Untertitel dienen können. Grafiken und Animationen können als QuickDraw eingebunden werden.

21.2 Windows Media

1991 startete Microsoft mit der ersten **AVI-Fileformat**-Version. AVI bedeutet "**Audio Video Interleave**" und beinhaltet einen Audio- und Videostrom. Seine Stärke liegt in der Einfachheit und brachte ihm darum eine grosse Verbreitung auf dem Markt: Es ist leicht zu adaptieren und es ist einfach den Export in eine Anwendung zu implementieren aber die Aufnahme von Datenmaterial beschränkt sich lediglich auf Audio und Video, und zwar jeweils genau auf einen Strom. Ein weiteres Manko ist die Tatsache, dass sich die beiden Spuren kaum synchronisieren lassen. Sollte die Möglichkeit des Interleaving nicht genutzt werden, besteht die Gefahr, dass Audio- und Videospur zwar gleichzeitig gestartet werden, aber nach einiger Zeit auseinanderlaufen. Ein weiterer Nachteil, den AVI mit sich bringt, ist die Tatsache, dass ein aus Rohdaten komponiertes Video komplett neu in einem AVI-File gespeichert wird, wodurch eine immense Redundanz verursacht wird, da die Rohdaten neben dem fertigen File auf der Festplatte liegen. In diesem Fall wäre ein Verweis auf die Rohdaten inklusive Schnittliste Zeit und Kapazität sparender. Ausserdem ist AVI sehr anfällig für Fehler. Sollte dem File nur der am Ende der Datei liegende Index-Teil fehlen oder sollte er defekt sein, versagen viele Player das Abspielen der Datei oder verhindern einen wahlfreien Zugriff.

21.3 DivX

Der AV-Codec DivX hat die Raubkopiererszene des Videomarktes revolutioniert. Die meisten gerippten DVD's sind mit dem beliebten Codec verschlüsselt. Spannend ist die Entstehungs-geschichte von DivX: Ein Franzose hat einst die von Microsoft geschriebene MP43C32.DLL entschlüsselt. Ursprünglich sollte diese DLL (Direkt Linked Library) den MPEG-4-Codec Version-3 für Windows bereitstellen, der allerdings keinen ISO-konformen MPEG-4-Strom erzeugt. Da Microsoft sich gerade von seinem AVI-Format verabschiedete und den Codec dem neueren Advanced Streaming Format bereitstellen wollte, patchten (engl. flickten) der Franzose zusammen mit einem deutschen Hacker den Code und gaben in als DivX;-) 3.11 Alpha für AVI heraus. Damit der Codec mit seinem Original zusammen auf einem Rechner lauffähig war, wurde noch der von Microsoft verwendete FourCC von MP43 auf DIV3 bzw. DIV4 geändert. Während Microsoft eigentlich Windows Media Audio V2 als Toncodierung vorsah, entschieden sich die DivX-Entwickler für das allseits beliebte MP3.

Der auch als DivX4 bekannte Codec OpenDivX hat eigentlich mit seinem Namensvetter nichts gemeinsam. Es handelt sich hierbei nämlich um einen echten bzw. legalen MPEG-4-Codec der Firma DivXNetworks.

22 Kryptologie, Verschlüsselung

Schon immer haben Leute versucht, Nachrichten so zu übermitteln dass niemand ausser dem beabsichtigten Empfänger sie lesen kann. Die Diskretion ist auf der ehrlichen wie auf der unehrlichen Seite ein grosses Bedürfnis. Dazu gibt es unterschiedliche Möglichkeiten. Man kann einen vertrauenswürdigen (wer ist nicht bestechlich?) Boten einstellen. Man kann einen gewissen Schutz anbringen wie beispielsweise eine geeignete Verpackung (Siegel mit Wappen) um die Unberührtheit der Nachricht zu forcieren. Oder man kann das Vorhandensein der Nachricht selbst verheimlichen. Die Wissenschaft der Kryptographie befasst also sich mit der Kunst, einen Text so zu verändern, dass ihn jeder sehen, aber nur der bestimmte Empfänger ihn auch verstehen kann.

22.1 Kryptologie

Die Kryptographie bildet zusammen mit der Kryptoanalyse die Kryptologie. Der Begriff **Kryptographie** stammt aus dem griechischen *kryptós*, "verborgen", und *gráphein*, "schreiben" ab. Er beschreibt die Wissenschaft der Verschlüsselung von Informationen.

22.2 Kryptographie

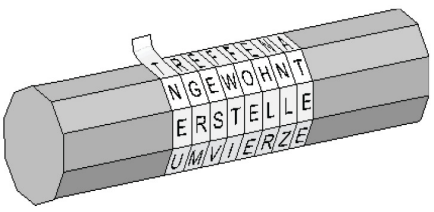
Die Kryptographie wird in symmetrische und asymmetrische Verfahren unterteilt. Die symmetrischen Verfahren werden weiter in Transpositions- und Substitutionsverfahren unterteilt.

22.3 Symmetrische Verfahren

Grundsätzlich kann man sagen, dass man eine mathematische Vorschrift verwendet, um aus einem Klartext einen Ciphertext zu erzeugen. Mit Hilfe eines Schlüssels wird jedes Zeichen oder Zeichengruppen umgewandelt. der Empfänger benötigt den gleichen Schlüssel und kehrt diese Operation um. Dies nennt man symmetrisch. Die symmetrischen Verfahren werden in zwei Vorgehensweisen, Methoden eingeteilt: Transposition und Substitution.

22.3.1 Skytale von Sparta (5. Jh.v.Chr.)

Dies ist die wohl älteste Überlieferung eines echten kryptographischen Verfahrens. Die Idee bestand darin, dass man ein Schreibband spiralförmig auf die sogenannte Skytale aufwickelte. Die war im wesentlichen ein n-eckiges Holzstück mit variablem Durchmesser. Danach wurde die Nachricht quer zum Spiralverlauf auf das Band geschrieben (siehe Abb). Nach Entfernen des Bandes befand die Nachricht in verschlüsselter Form auf dem Band. Für die Entschlüsselung musste man das Band auf eine ebensolche Skytale mit gleichem Durchmesser aufwickeln. Die Skytale selbst war somit der Schlüssel, ohne deren Besitz das Entschlüsseln nicht möglich war. Genau genommen war die Skytale eine Transpositions-Chiffrierung.



SIEIYHITIEPICINLTEHSZDO!ETIEGRDGRRHAEKAESZRP

n=5

SHPLZTDAS
IIITDIGEZ
ETCEOERKR
IIIH!GRAP
YENSERHE

n=6

SICHERHE
ITISTDAS
EINZIGEZ
IELDERKR
YPTOGRAP
HIE!

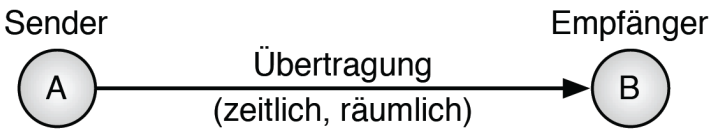
n=7

STNDGAR
IILOREP
EET!DK
IPEEGA
YIHTRE
HCSIRS
IIZEHZ

Die Kryptographie ist überall dort ein wichtiges Hilfsmittel, wo die Sicherheit beim Übertragen von Information garantiert werden muss. Man unterscheidet zwischen

- zeitlicher Übertragung (z.B. auf einer Festplatte) und
- räumlicher Übertragung (von A nach B, meist über ein Netzwerk)

In kryptographischen Texten bezeichnet man den Sender oft als Alice oder A und den Empfänger als Bob oder B.

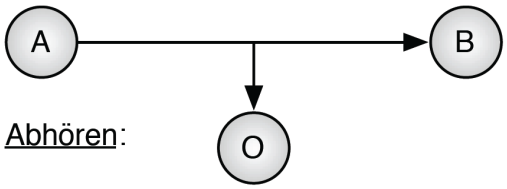


22.4 Angriffe auf die Informationssicherheit

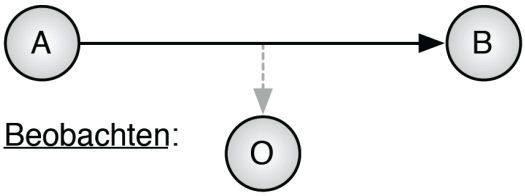
Grundsätzlich kann man zwischen passiven und aktiven Attacken unterscheiden. Im ersten Fall spricht man auch von Spionage, während man im zweiten Fall von Sabotage spricht.

22.4.1 Passive Attacken

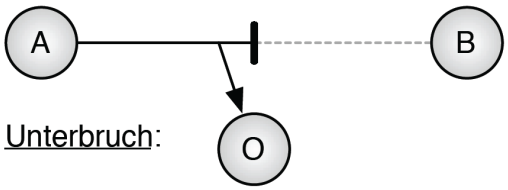
(Release of Message Contents)

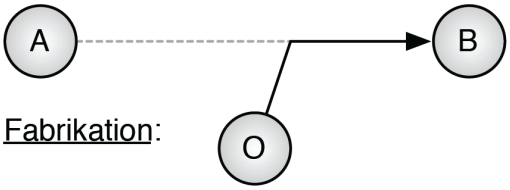


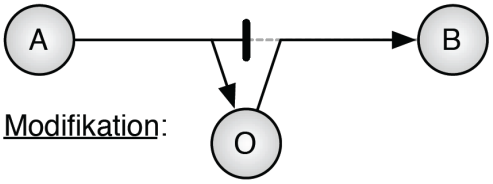
(Verkehrsfluss-Analyse, Traffic Analysis),



22.4.2 Aktive Attacken







Passive Attacken verletzen die Vertraulichkeit, aktive Attacken verletzen die Integrität und die Authentizität der Information.

23 Transposition (Rotation) - symmetrisch

In einer Nachricht wird jeder Buchstabe im Alphabet durch denjenigen ersetzt, der n Stellen später im Alphabet folgt. Beim Entschlüsseln muss man jeden Buchstaben des chiffrierten Textes durch den um n Stellen vorgestellten Buchstaben ersetzen. Das Alphabet wird um eine bestimmte Anzahl Buchstaben verschoben. Da die Verschiebung über das Alphabet hinausreicht wird diese verschiebung am Anfang fortgesetzt, das Alphabet also um n Stellen rotiert. Deshalb spricht man in diesem Sinne auch von Rotationchiffrierung. Das Alphabet enthält nur eine Darstellung von Buchstaben – Klein- und Grosschreibung werden nicht getrennt. Die Leerstellen bleiben im chiffrierten Text erhalten.

23.1 Beispiel Caesar-Chiffre (=ROT-3)

(nach dem römischen Feldherrn Julius Caesar)

Klartext	**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**
Chiffre	**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**

Der Klartext "SCHULE" lautet somit verschlüsselt "VFKXOH"

23.2 Kryptoanalyse von ROT-Chiffren

Das Caesar-Verfahren als eine ROT-Chiffrierung ist schon über 2000 Jahre alt. Es ist deshalb nicht erstaunlich, dass es sehr leicht zu brechen ist. Eine Analysemethode nennt sich "Brute-Force", was nichts anderes heisst als "rohe Gewalt". Der Trick besteht darin, dass alle möglichen Schlüssel ausprobiert werden, und man schaut, was dabei jeweils als Klartext herauskommt.

Wie viele Schlüssel-Möglichkeiten gibt es beim Rotations-Verfahren?

23.2.1 Aufgabe (2 Teams à 2 Partner)

1) Jedes Team schreibt einen sinnvollen Klartext.
2) Verschlüsseln Sie den Text mit einem dem Partnerteam unbekannten ROT-Schlüssel.
3) Tauschen Sie die Geheimnachricht mit ihrem Partnerteam.
4) Versuchen Sie die Nachricht zu knacken.

23.3 Häufigkeitsanalyse

Eine weitere Methode den ROT-Code zu brechen besteht in der Auswertung der Häufigkeitsverteilung der vorkommenden Buchstaben. Wenn die verschlüsselte Nachricht eine natürliche Sprache ist, so treten die einzelnen Buchstaben mit unterschiedlicher Häufigkeit auf.

23.3.1 Buchstaben-Häufigkeitsverteilung im deutschen Alphabet

	relative Häufigkeit		relative Häufigkeit
e	17.40%	m	2.53%
n	9.78%	o	2.51%
i	7.55%	b	1.89%
s	7.27%	w	1.89%
r	7.00%	f	1.66%
a	6.51%	k	1.21%
t	6.15%	z	1.13%
d	5.08%	p	0.79%
h	4.76%	v	0.67%
u	4.35%	j	0.27%
l	3.44%	y	0.04%
c	3.06%	x	0.03%
g	3.01%	q	0.02%

Das Rotations-Verfahren ist ein monoalphabetisches Verfahren. Aus einem bestimmten Klartextbuchstaben wird immer der gleiche Geheimtextbuchstabe. Deshalb kann man folgendermassen vorgehen:

1. Man zählt wie oft die einzelnen Buchstaben auftreten.
2. Der Buchstabe, der am häufigsten auftritt, ist wahrscheinlich ein "e"
3. Kommt ein Wort mit nur 2 Buchstaben vor, bei dem der erste Buchstabe wahrscheinlich ein "e" ist, so ist der 2. Buchstabe vermutlich ein "r". Begründung: "er" ist ein häufiges Bigramm (Buchstabenpaar).
4. Es können auch Häufigkeitsdaten von Trigrammen benutzt werden. Trigramme sind Folgen von 3 Buchstaben.
5. Hat man erst einmal ein paar Buchstaben erraten ist es meist nicht allzu schwierig aus dem Kontext noch weitere Buchstaben zu erraten und in den Rest der Botschaft zur Verifikation einzusetzen.

23.3.2 Aufgabe (Zweierteam)

Erstellen Sie mit "rotation.exe" einen verschlüsselten Text (der Klartext soll in einer Datei anyname.txt vorliegen). Der Dateiname des verschlüsselten Textes sollte die Endung ".txt" aufweisen (kann mit dem Editor geöffnet werden). Die Anzahl Stellen, die rotiert werden, bleiben aber vorerst das Geheimnis des Programmierers. Versuchen sie anhand der Häufigkeitsanalyse des chiffrierten Textes (analyse.exe) rauszufinden, um wie viele Stellen rotiert wurde.

24 Substitution - symmetrisch

Irgendwann haben Caesars Nachfahren herausgefunden, dass dieses Verfahren nicht sehr sicher ist. Wenn jedoch die Anzahl der möglichen Schlüssel erhöht werden könnte, dann würde die Kryptoanalyse massiv erschwert werden. Dies gelingt mit dem Substitutionsverfahren. Dabei wird jeder Buchstabe des Klartextes durch einen beliebigen Buchstaben aus dem Geheimtext ersetzt ("substituieren" = "ersetzen"). Es werden nicht mehr alle Buchstaben um gleich viele Stellen verschoben wie beim Caesar-Verfahren.

Die Zahl der Schlüssel kann mit folgendem Trick erhöhen. Anstatt das Alphabet nur um eine feste Anzahl stellen zu verschieben, kann man die Buchstaben beliebig permutieren (vertauschen). Konkret geht man folgendermassen vor: Man schreibt das Klartextalphabet in eine Reihe, danach verteilt man alle Buchstaben des Alphabets wild gemischt auf eine zweite Reihe. Das sieht dann so aus:

Beispiel

abcdefghijklmnopqrstuvwxyz

CFILORUXADGJZPSVYBEHKNQTWM

Ein Text wird nun verschlüsselt, indem jeder Buchstabe durch den darunter stehenden Buchstaben ersetzt wird. "hallo" wird also zu "XCJJS".

Eine andere Variante sieht folgendermassen aus:

Die Buchstaben des Klartextalphabetes werden der Reihe nach hingeschrieben. Darunter wird zuerst das Schlüsselwort (im Beispiel: "KRYPTO") geschrieben, und dann kommen der Reihe nach alle im Schlüsselwort nicht benutzten Buchstaben des Alphabetes.

Beispiel mit Schlüsselwort "KRYPTO":

abcdefghijklmnopqrstuvwxyz

KRYPTOABCDEFGHIJLMNQSUVWXZ

Damit wird aus dem Klartext "hallo" der Geheimtext " BKFFI".

Das Substitutionsverfahren ist ein symmetrisches Verfahren. Deshalb funktioniert die Entschlüsselung wie beim Caesar-bzw. Rotationsverfahren.

24.1 Kryptoanalyse des Substitutionsverfahrens

Das Substitutionsverfahren scheint ziemlich sicher zu sein, gibt es doch theoretisch 26! = 403’291’461’126’605’635’584’000'000 mögliche Schlüssel. Eine Brute-Force-Analyse würde je nach Rechner einige Zeit in Anspruch nehmen. Die Kryptoanalytiker benutzen hier vorteilhafter die "Häufigkeitsanalyse". Das Substitutionsverfahren ist wie das Caesar-Verfahren ein monoalphabetisches Verfahren.

24.1.1 Aufgabe

1) Schreiben Sie einen Klartext mit mindestens 200 Zeichen.
2) Verschlüsseln Sie den Text mit einem Substitutions-Schlüssel.
3) Tauschen Sie die Nachricht mit ihrem Partnerteam.
4) Führen Sie eine Analyse mit Hilfe der Häufigkeitsverteilung durch. Dazu können sie das Programm "analyse.exe" benutzen.

24.1.2 Frage

Wieso sollte der Text mindestens 200 Zeichen umfassen?

25 Die Vigenère-Chiffre

Das Problem der ROT-Verschlüsselung ist, dass die Zeichen zwar durch andere ersetzt werden, die Häufigkeit von einzelnen Zeichen und die Reihenfolge des Chiffrealphabets aber erhalten bleibt. In der ROT-Chiffre gibt es nur ein Chiffrealphabet. Häufige Zeichen (wie zB. das "e") werden also immer durch dieselben anderen Zeichen ersetzt, die dann genau so häufig sind.

Das Ziel ist es also so zu Verschlüsseln, dass in der Chiffre alle Buchstaben gleich häufig vorkommen.

25.1 Das Vigenère-Quadrat

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

Die oberste Zeile wird auf den Klartext angewandt, die linke Kolonne auf das Passwort.

25.2 Verschlüsselung

Sender und Empfänger benötigen als Schlüssel ein **Passwort**. Dies wird über den Klartext geschrieben und wenn nötig so vielmal wiederholt, bis die Länge der Nachricht erreicht ist:

Passwort : `KRYPTOGRAFIEKRYPTOGRAFIE` (Jeweils Buchstabe in oberster Zeile)

Klartext : `BERUFSSCHULEN` (Jeweils Buchstabe in erster Spalte)

Chiffre : **`LVPJYGYTHZTIX`** (Ergibt jeweils Buchstabe im Schnittpunkt)

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

Es wird Buchstabe für Buchstabe verschlüsselt. Vigenère ist ein polyalphabetisches Verfahren, dh. pro Klartextbuchstaben gibt es mehrere mögliche Geheimtextbuchstaben und umgekehrt.

25.2.1 Aufgabe

1. Wählen Sie ein Schlüsselwort
2. Schreiben Sie eine kurze Nachricht.
3. Verschlüsseln Sie die Nachricht.
4. Tauschen Sie die Nachricht und Schlüssel mit ihrem Partnerteam.
5. Entschlüsseln Sie die Nachricht.
6. Untersuchen Sie die verschlüsselte Nachricht mit "analyse.exe"

25.2.2 Kryptoanalyse des Vigenère-Verfahrens

Das Vigenère-Verfahren wurde auch als "Le chiffre indéchiffrable" bezeichnet.

25.3 Ist denn nichts sicher?

Das Knacken von Vigenère gelingt, da es wegen der fixen Schlüssellänge zu Wiederholungen kommt. Nimmt man einen Schlüssel der gleich lang ist wie der zu verschlüsselnde Text gibt es keine Wiederholungen. Dieses Verfahren heisst "One-Time-Pad".

Das One-Time-Pad ist ein 100% sicheres Verfahren, denn jeder Schlüsselbuchstabe wird nur einmal (one-time) verwendet.

Es hat nur einen leider relativ grossen Nachteil: Im vornherein muss ein riesengrosser geheimer Schlüssel vereinbart werden. Auch beim One-Time-Pad gilt: Der Schlüssel muss über einen sicheren separaten Kanal übermittelt werden. Ist der Schlüssel dem Gegner bekannt ist die Sicherheit dahin.

26 Moderne Verfahren

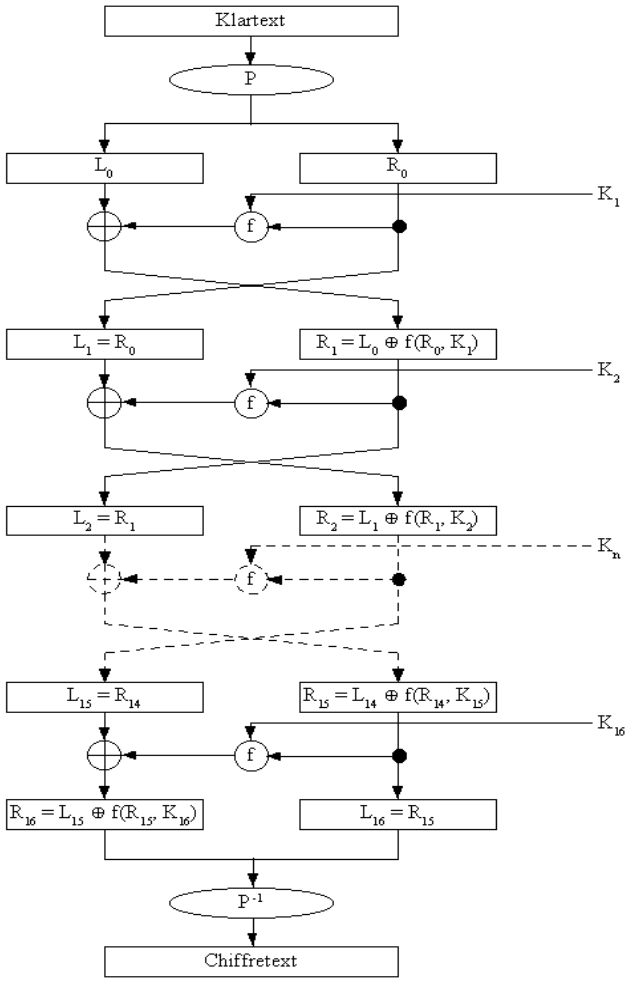
Das heute im kommerziellen Gebrauch am häufigsten eingesetzte Verfahren heisst **DES**. DES steht für **Data Encryption Standard**. Es funktioniert im Prinzip wie ein mehrfach hintereinander angewandtes Substitutionsverfahren

26.1 Sicherheit von DES

Der DES erlaubt mit 56 Bits Schlüssellänge 2^{56} = 72'057'594'037'927'936 = 72 Billiarden mögliche Schlüssel.

Dennoch ist dies heute nicht mehr ausreichend: DES kann in wenigen Tagen mittels der Brute-Force-Methode geknackt werden.

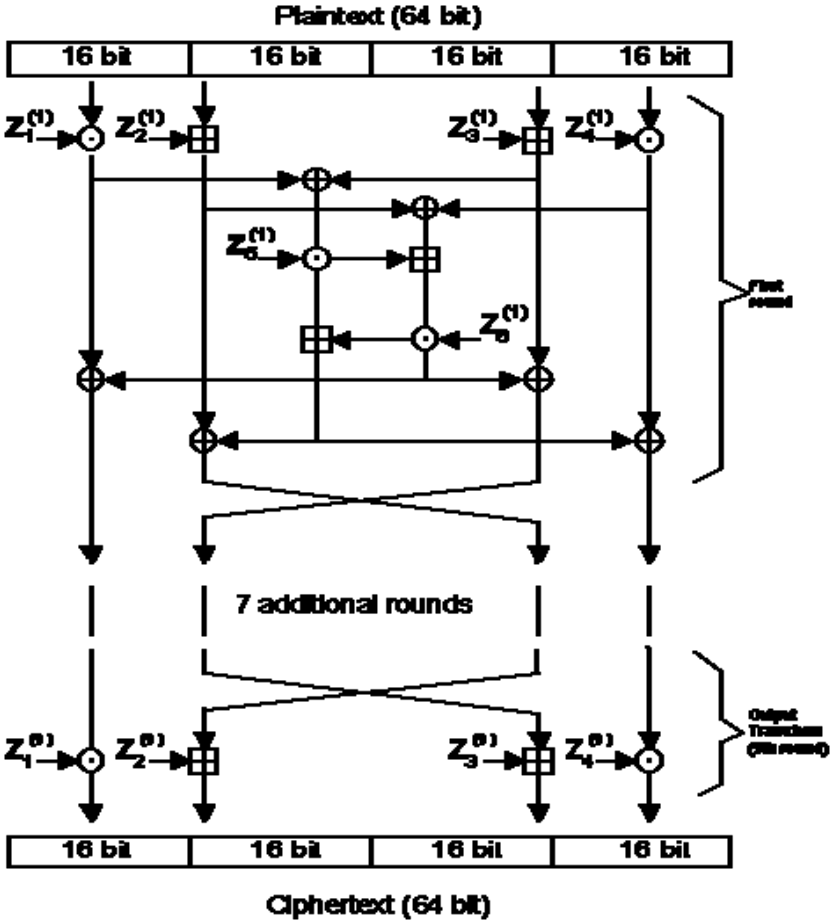
26.1.1 Abbildung: Struktur von DES



26.2 IDEA (International Data Encryption Algorithm)

Der IDEA arbeitet mit 128 Bits Schlüssellänge, sonst ähnlich wie der DES. $2^{128} = 3.43669 \cdot 10^{38}$. Deshalb gilt der IDEA heute als sicher.

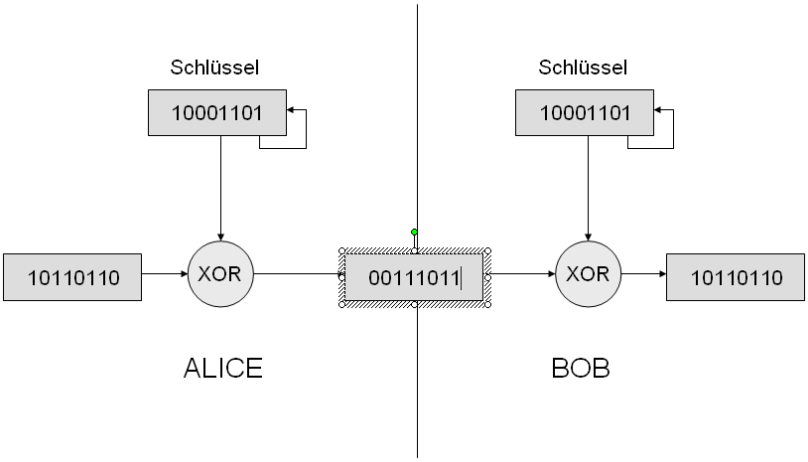
26.2.1 Struktur von IDEA



27 Stromchiffrierung (RC4, XOR)

Eine Stromchiffre betrachtet den Datenstrom bitweise und verschlüsselt diesen, indem der Datenstrom mit den Bits des Passwortstroms im Exklusiv-ODER-Verfahren verknüpft wird. Die Schlüsselbits machen scheinbar zufällige Modifikationen an den Datenbits. Somit ist der verschlüsselte Datenstrom für Dritte nicht entzifferbar. Die Qualität der Stromchiffre hängt von der Qualität der Zufallszahlen im Schlüssel ab. Ist die Bitfolge im Schlüssel nicht vorhersehbar und wiederholt sich nicht, so ist die Chiffre sicher.

RC4 (Ron's Code 4) ist eine von Ronald L. Rivest in den 1987 für die RSA Security entwickelter Stromchiffrierungsalgorithmus, der 7 Jahre später bekannt (Sicherheitsleck) wurde aber dennoch sehr weit verbreitet ist. Die häufigste Anwendung ist die Verschlüsselung des Datenstroms im Webbrowser, wenn eine gesicherte Verbindung mit SSL hergestellt wird.

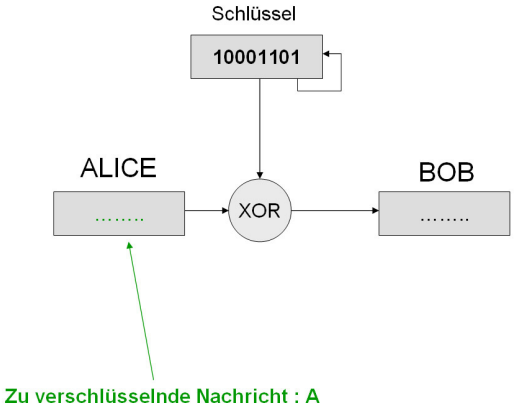


RC4 verschlüsselt immer ein Byte auf einmal. Die Schlüssellänge kann bis zu 2048 Bit betragen

27.1.1 Aufgabe zu dem symmetrischem Kryptoverfahren XOR:

Verschlüsseln Sie die Nachricht mit dem obigen Verfahren. Achtung: Es handelt sich um eine Stromchiffre. Der Buchstabe wird als ASCII-Zeichen binär übertragen.

Zur Kontrolle entschlüsseln Sie die Nachricht auf dieselbe Art, wie sie verschlüsselt wurde. Sie sollten dabei wieder die ursprüngliche Information erhalten.



Dezimal	Oktal	Hex	Binär	Zeichen
64	100	40	0100 0000	@
65	101	41	0100 0001	A
66	102	42	0100 0010	B
67	103	43	0100 0011	C
68	104	44	0100 0100	D
69	105	45	0100 0101	E
70	106	46	0100 0110	F
71	107	47	0100 0111	G
72	110	48	0100 1000	H
73	111	49	0100 1001	I
74	112	4A	0100 1010	J
75	113	4B	0100 1011	K
76	114	4C	0100 1100	L
77	115	4D	0100 1101	M
78	116	4E	0100 1110	N
79	117	4F	0100 1111	O
80	120	50	0101 0000	P
81	121	51	0101 0001	Q
82	122	52	0101 0010	R
83	123	53	0101 0011	S
84	124	54	0101 0100	T
85	125	55	0101 0101	U
86	126	56	0101 0110	V
87	127	57	0101 0111	W
88	130	58	0101 1000	X
89	131	59	0101 1001	Y
90	132	5A	0101 1010	Z

28 Asymmetrische Verfahren

Das Manko der symmetrischen Verfahren liegt darin, dass der benötigte Schlüssel geheim bleiben muss, aber der berechtigte Empfänger der Nachricht ihn dennoch haben muss. Dies erfordert einen geheimen Transport des Schlüssels. Ferner ist eine geheime Kommunikation mit jedem einzelnen einer grösseren Anzahl von Teilnehmern dadurch erschwert, dass eine Vielzahl von Schlüsseln erforderlich wäre. Bei drei Teilnehmern wären es

$\binom{3}{2} = \frac{3 \cdot 2}{1 \cdot 2} = 3$ ausgesprochen: Drei tief zwei = 3 Schlüssel ⇨ A-B, A-C, B-C

$\binom{10}{2} = \frac{10 \cdot 9}{1 \cdot 2} = 45$ Schlüssel bei 10 Teilnehmern

Dieses Problem umgehen asymmetrische Verfahren, auch Public-Key-Verfahren genannt.

Hierbei hat jeder Teilnehmer ein Schlüsselpaar bestehend aus einem geheimen und einem öffentlichen Schlüssel. Letzterer kann und soll frei an alle Kommunikationsteilnehmer verteilt werden, daher öffentlich. Die Verschlüsselung des Klartextes erfolgt mit dem öffentlichen Schlüssel des Empfängers. Die Nachricht kann nur mit dem geheimen Schlüssel entschlüsselt werden.

Die Sicherheit dieser Verfahren basiert im Grunde auf der Tatsache, dass es im Moment als unmöglich gilt, in vernünftiger Zeit das Produkt zweier sehr grosser Primzahlen ohne Kenntnisse dieser wieder zu zerlegen, zu faktorisieren. In anderen Fällen basiert die Sicherheit darauf, dass es zur Zeit als unmöglich gilt, in absehbarer Zeit den diskreten Logarithmus einer Zahl zu ermitteln, die aus dem Potenzieren zweier grosser Zahlen entstanden ist (Diffie-Hellmann, El-Gamal). Mit anderen Worten: Es ist einfach, zwei grosse Zahlen miteinander zu multiplizieren oder zu potenzieren, aber sehr schwierig, den Weg umzukehren.

Diese Verfahren gelten heute als sicher, aber eine Grundregel der Kryptologie sagt, dass kein Code unbrechbar ist. Bei den asymmetrischen Verfahren steht und fällt die Sicherheit mit den Fortschritten der Mathematik bzw. der Leistungszunahme in der Rechentechnik.

28.1 Voraussetzungen für Public Key Kryptographie: die Einwegfunktion

Voraussetzungen für Public Key Kryptographie ist die **Einwegfunktion**. Aus der Mathematik kennen wir verschiedene Funktionen, zum Beispiel Multiplizieren oder Potenzieren. Für beide Funktionen besteht aber eine Umkehrfunktion. Bei der Multiplikation die Division, beim Quadrieren das Radizieren (Wurzelziehen). Gefordert ist nun eine Funktion, zu der keine Umkehrfunktion existiert. Allerdings, wenn tatsächlich keine Umkehrfunktion besteht , wie das bei echten Einwegfunktionen der Fall ist, wird's schwierig. Gefordert ist darum eine Einwegfunktion mit Trapdoor (Falltüre, bzw. geheimen Zusatzinformationen). Die gewöhnliche Exponentialfunktion $\mathbf{y = a^b}$ ist zwar keine Einwegfunktion. Ergänzt man diese aber noch mit der Funktion Modulo (Restwertfunktion) wird es zu einer:

$y = a^b \mod p$

a = Erzeugendes Element; zB. 2
p = Primzahl, zB. 19
Das erzeugende Element a kann aus der Primzahl p hergeleitet werden.

28.2 Diffie-Hellmann

In der Tat eignet sich das 1976 beschriebene Verfahren Diffie-Hellmann nicht zur Ver- und Entschlüsselung, sondern vielmehr zum Austausch von Schlüsseln über "unsichere" Kanäle. Seine Sicherheit basiert wie El-Gamal auf der Problematik, dass es keine eindeutigen, sicheren mathematischen Verfahren gibt, den diskreten Logarithmus zu berechnen, es aber umgekehrt sehr einfach ist, eine Zahl zu potenzieren.

Die Darstellung ist wie üblich möglichst kurz und einfach gehalten und richtet sich nicht an Mathematiker oder Krypto-Experten.

28.2.1 1. Schritt

Alice und Bob möchten miteinander verschlüsselt kommunizieren. Dazu möchten sie gern die Schlüssel für den Verschlüsselungsvorgang miteinander austauschen. Leider besteht keine gesicherte Verbindung zwischen beiden und ein persönliches Treffen als sicherste Alternative ist nicht möglich. Also wählen sie folgende Vorgehensweise:

Zunächst denkt sich einer von beiden eine möglichst grosse Primzahl P sowie eine Zahl z aus. Für diese muss gelten, dass z kleiner als P ist. (Tatsächlich gibt es noch eine weitere Einschränkung, die wir aber hier vernachlässigen; bei weiterem Interesse sei auf die Literatur verwiesen).

Diese Daten werden offen an den anderen Partner gesendet.

28.2.2 2. Schritt

Nun denkt sich jeder der beiden eine geheime Zahl aus; Alice nimmt a und Bob nimmt b. Nun berechnet jeder:

Alice	Bob
$X = z^a \mod P$	$Y = z^b \mod P$

Die Werte X und Y werden wie auch z und P miteinander ausgetauscht. Die Werte a und b dagegen sind die geheimen Schlüssel. Jeder der beiden berechnet daraus den Schlüssel:

Alice	Bob
$K_A = Y^a \mod P$	$K_B = Y^b \mod P$

Damit erhalten beide den gleichen Wert, denn es gilt: $z^a \mod P = z^b \mod P = z^{ab} \mod P$

Beispiel:

Alice denkt sich für P=11 und für z=4 aus. Dieses teilt sie Bob mit. Ausserdem hat sie sich für a den Wert 3 ausgedacht, während Bob seinen geheimen Schlüssel auf den Wert 5 gesetzt hat.

Alice	Bob
$X = 4^3 \mod 11$	$Y = 4^5 \mod 11$
$X = 64 \mod 11$	$Y = 1024 \mod 11$
X = 5 Rest 9	Y = 93 Rest 1

Alice teilt Bob noch den Wert X=9 mit und Bob den Wert Y=1.
Nun kann jeder seinen Schlüssel berechnen:

Alice	Bob
$K_A = 1^3 \mod 11$	$K_B = 9^5 \mod 11$
$K_A = 1 \mod 11$	$K_B = 59049 \mod 11$
K_A = 0 Rest 1	K_B = 5368 Rest 1

Dieses Beispiel ist natürlich sehr vereinfacht und trivial gehalten. Es soll lediglich die Vorgehensweise und die Funktion von Diffie-Hellmann beschreiben. Aus diesem Zweck wurden unrealistisch kleine Werte eingesetzt und ggf. vorgegebene Restriktionen (s. weiter oben) nicht beachtet.

Diffie und Hellmann gelten zusammen mit Merkle aufgrund ihrer Veröffentlichungen in 1976 als Entdecker der Public-Key-Verfahren. In jüngerer Zeit wurde aber bekannt, dass nur wenige Zeit zuvor drei andere Wissenschaftler ein nahezu identisches Verfahren entdeckt hatten, es aber nicht veröffentlichen durften.

Aufgrund sehr hoher Kosten bei der Schlüsselverteilung hatte das britische Militär dem Government Communication Headquarter (GCHQ) schon in den 60er Jahren den Auftrag erteilt, andere Wege zu finden. Die von James Ellis und Clifford Cocks geäusserten Ideen ähnelten denen von Diffie und Hellmann, nur etwa sechs Jahre früher. Diese Ideen wurden zusammen mit Williamson 1975 vervollstän-

digt und entsprachen dem nur ein Jahr später vorgestellten Diffie/Hellmann/Merkle-Verfahren. Das GCHQ hat einerseits aus Gründen der Geheimhaltung und andererseits wegen des für die Briten aus Sicht der frühen 70er Jahre fraglichen Nutzens nie ein Patent beantragt. Im Dezember 1997 wurde dieser Sachverhalt bekannt.

James Ellis starb einen Monat zuvor und reiht sich in die Liste britischer Kryptologen wie Babbage und Turing ein, denen öffentlicher Ruhm zu Lebzeiten aus Gründen der Geheimhaltung vorenthalten blieb.

28.3 El-Gamal

El-Gamal ist eine Krypto-Anwendung die auf dem Diffie-Hellmann-Schlüsseltausch basiert.

28.3.1 Ablauf von El-Gamal (SENDER)

1. Sämtliche Teilnehmer einigen sich auf Primzahl p und natürliche Zahl g.
2. Jeder Teilnehmer wählt eine geheime natürliche Zahl t.
3. Jeder Teilnehmer berechnet mithilfe seiner Zahl t den Wert $T=g^t \bmod p$ und stellt diesen " " öffentlichen" Schlüssel T in das öffentliche Verzeichnis.
4. Will Alice nun einem Teilnehmer T eine Nachricht zusenden, so sucht sie im öffentlichen Verzeichnis seinen Schlüssel T.
5. Alice wählt zufälligen Wert a und rechnet : $x=g^a \bmod p$
6. Alice rechnet $K=T^a \bmod p$ (K=Key)
7. Alice verschlüsselt ihre Nachricht "symmetrisch" mit dem Key k
8. Alice schickt die verschlüsselte Nachricht und den Wert x an den Teilnehmer T

28.3.2 Ablauf von El-Gamal (Empfänger)

1. Empfänger muss zuerst den Schlüssel K errechnen. Dazu verwendet er den Wert X, den er von Alice erhalten hat: $K=x^t \bmod p$ (t ist die geheime Zahl des Empfängers)
2. Der Empfänger entschlüsselt die Nachricht von Alice mit seinem Key K.

28.4 RSA

RSA wurde 1977 von Ron Rivest, Adi Shamir und Leonard Adleman entwickelt. Das Verfahren verwendet grosse Primzahlen; seine Sicherheit basiert auf der Schwierigkeit, grosse Zahlen zu faktorisieren. Das Verfahren ist einfach zu verstehen und ebenso leicht in Applikationen zu implementieren. RSA wurde intensiv der Kryptoanalyse unterzogen, die zwar das Verfahren nicht brechen konnte, allerdings auch keinen Beweis für die Sicherheit liefern konnte.

Die folgenden Zeilen sollen darstellen, wie simpel und doch zugleich effektiv der RSA-Algorithmus funktioniert, der neben den Algorithmen El-Gamal und Diffie-Hellmann in verschiedenster Software zum Einsatz kommt. Diese Darstellung basiert auf der Veröffentlichung: The Mathematical Guts of RSA Encryption und "Angewandte Kryptographie". Sie richtet sich nicht an den Mathematiker unter den Lesern, sondern an den Anwender von PGP, GnuPG oder S/MIME. Die durchzuführenden Schritte wurden nur mit einem Minimum an mathematischen Formeln beschrieben. Für das Beispiel wurden bewusst kleine und einfache Zahlen verwendet, die natürlich für die Realität keineswegs hinreichend sind.

28.4.1 1. Schritt

Wir benötigen zwei sehr grosse Primzahlen; für die reale Verschlüsselung sollten die Primzahlen 1024 bit gross sein, also 100 oder mehr Ziffern haben. Wir begnügen uns mit P=3 und Q=5. In PGP sollen P und Q so gewählt werden, dass das Produkt $P \cdot Q$ mehr als 300 Stellen hat.

28.4.2 2. Schritt

Nun sollen wir eine Zahl E finden, die folgende Bedingungen erfüllen muss:

E muss kleiner sein als P·Q

E darf keine gemeinsamen Faktoren mit dem Produkt (P-1)·(Q-1) haben

E braucht keine Primzahl zu sein, muss aber ungerade sein

Das Produkt (P-1)·(Q-1) ist übrigens zwangsläufig gerade. Das Produkt P·Q nennen wir N. N sollte etwa doppelt soviele Ziffern wie die Primfaktoren P und Q haben.

P·Q=3·5=15 daraus folgt für (P-1)·(Q-1)=2·4=8

Eine Primfaktorenzerlegung zeigt für 8 die Faktoren 2·2·2. E darf also keine 2 als Teiler haben, was ja ohnehin gegeben ist, da E ungerade sein soll. Der für unser Beispiel einfachste Wert wäre die 3; also E=3.

28.4.3 3. Schritt

Der Algorithmus schreibt nun vor, dass wir noch eine Zahl D bestimmen:

Diese muss die Bedingung erfüllen, dass (D·E-1) ohne Rest durch (P-1)·(Q-1) teilbar ist:

(D·E-1) mod ((P-1)·(Q-1))=0 respektive (D·E) mod ((P-1)·(Q-1))=1 und nicht gleich dem Wert von E sein soll, also D·E wird durch (P-1)·(Q-1) so geteilt, dass Rest 1 bleibt.

Wie macht man das? Nun, eigentlich ganz einfach: man muss nur eine ganzzahlige Zahl X finden, die dafür sorgt, dass die folgende Gleichung ein ganzzahliges Ergebnis liefert:

D= (X·(P-1)·(Q-1)+1)/E

Eine derartige Gleichung nennt man diophantische Gleichung, die in unserem vorliegenden Fall exakt lösbar ist. Im Rahmen dieser Beschreibung wollen wir aber aus Gründen der einfachen Darstellung nicht auf den Lösungsweg eingehen.

In unserem einfachen Fall kann X=4 gewählt werden, dann wird D zu

D=(4·2·4+1)/3=33/3=11.

Damit gilt für D·E=33. Die Probe ist 33/8=4 Rest 1.

Fassen wir mal unsere einzelnen Zahlen zusammen:

P=3, Q=5, N=15, D=11, E=3

Die Zahlen N und E bilden den Public Key. Die Zahlen N und D bilden den Secret Key. Dies erklärt auch die Zusatzforderung, dass E und D nicht den gleichen Wert haben sollten, da sonst Public und Secret Key gleich wären. Sind diese Zahlen einmal berechnet, so müssen alle Hilfsgrössen wie P, Q, (P-1)(Q-1) und X gelöscht werden.

28.4.4 Verschlüsselung

Die Rechenvorschrift für die Verschlüsselung lautet (senderseitig)

$C=K^E \bmod N$	C steht für Ciphertext, K steht für Klartext, E und N sind der **Public Key** des Empfängers.

Die Zeile sagt lediglich: Potenziere den Klartext mit E, teile das durch N und gib den Rest als Ciphertext aus. Nehmen wir mal an, wir wollen folgende Zahlenreihe verschlüsseln:

12130508.

Die einzelnen zu verschlüsselnden Elemente müssen in Stücke zerlegt werden, die kleiner als N sind: 12 13 05 08

Die Ciphertexte wären demnach:

123= 1728	1728/15 = 115 Rest 03
133= 2197	2197/15 = 146 Rest 07
053= 125	125/15 = 8 Rest 05
083= 512	512/15 = 34 Rest 02

Das Ergebnis ist dann: 03 07 05 02

28.4.5 Entschlüsselung

Die Rechenvorschrift für die Entschlüsselung lautet (empfängerseitig):

$K=C^D \bmod PQ$ C steht für Ciphertext, K steht für Klartext
D und N sind der **Secret Key** des Empfängers

Die Mathematik ist exakt die gleiche wie bei der Verschlüsselung. Rechnen wir mal den Ciphertext zurück: 03 07 05 02

0311= 177147	177147/15= 11809 Rest 12
0711= 1977326743	1977326743/15= 131821782 Rest 13
0511= 48828125	48828125/15= 3255208 Rest 05
0211=2048	2048/15=136 Rest 08

Das sehr einfach gewählte Beispiel führt zu sehr kleinen zu verschlüsselnden Textelementen. Der entstandene Code liesse sich vermutlich schnell brechen. Es ist allerdings nicht Aufgabe dieses Beispiels gewesen, die Unbrechbarkeit des Codes zu beweisen, sondern vielmehr die Vorgehensweise und den Algorithmus verständlich zu erläutern.

28.4.6 Zusammenfassung RSA

Das war jetzt etwas ausführlich! Darum das Verfahren nun noch einmal aber stark vereinfacht:

Alice wählt zufällig zwei Primzahlen $p \neq q$, die etwa gleich lang sein sollten und
berechnet deren Produkt $N = p \cdot q$. zB. p=11, q=17, n=11·17=187
Danach berechnet sie $x = (p-1) \cdot (q-1)$ x=(11-1)·(17-1)=160
Alice berechnet x + 1 x+1=161
Der Wert e wird ausgewählt,
wobei d ein Teiler aus x+1 sein muss.
Der zweite Teiler aus x+1 ist d.
Es muss gelten: d*e=x+1. d·e=161=7*23
Unzulässige Werte von d und e (z.B. d=e)
werden als unsicher abgelehnt. 7 und 23 sind geeignete Kandidaten
für e und d
Alice erhält N , e : public key von Alice
d , N : secret key von Alice

Jeder kann Alice eine verschlüsselte Nachricht senden, aber nur Alice kann sie entschlüsseln. Das Verfahren verwendet grosse Primzahlen. Seine Sicherheit basiert auf der Schwierigkeit, grosse Zahlen zu faktorisieren.

28.4.7 RSA-Verschlüsselung

Die Rechenvorschrift für die Verschlüsselung lautet:

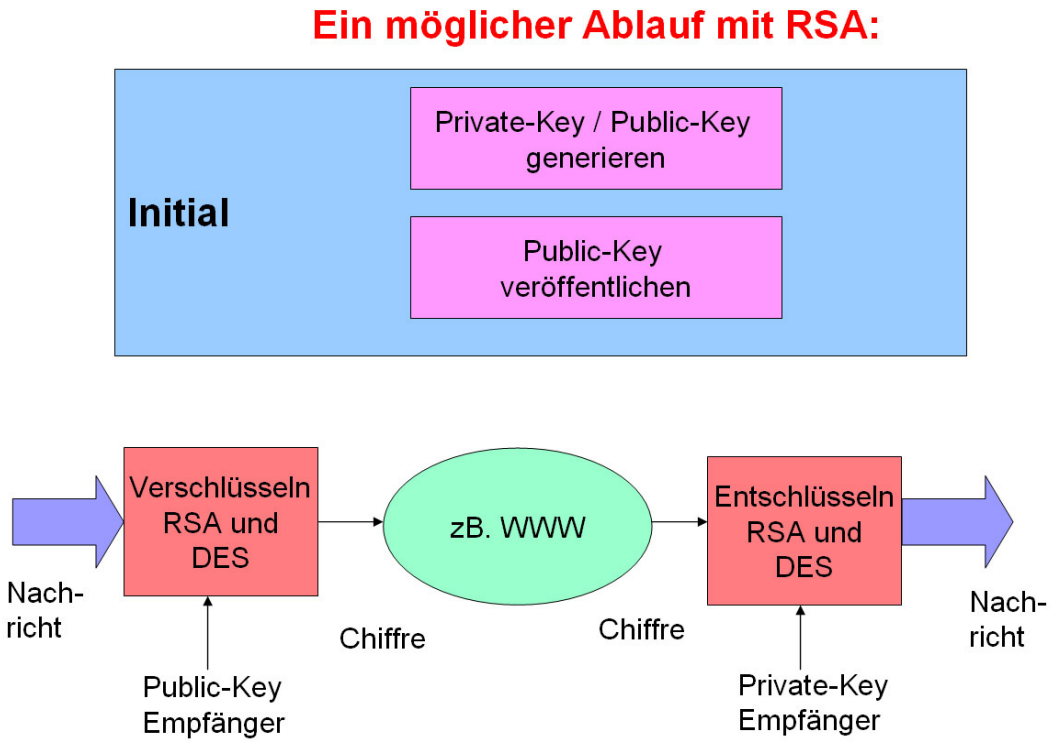
$c=k^e \bmod n$ c steht für Ciphertext, k steht für Klartext
e und n sind der Public Key des Empfängers

28.4.8 RSA-Entschlüsselung

Die Rechenvorschrift für die Entschlüsselung lautet:

$k=c^d \bmod n$ c steht für Ciphertext, k steht für Klartext
d und n sind der Private Key des Empfängers

Die Mathematik ist exakt die gleiche wie bei der Verschlüsselung.



28.5 Digitale Signatur

Bei Informationsaustausch über unsichere Kanäle besteht folgende Problematik:

- Wer bestellt in meinem E-Shop ?
- Wer schickt mir eine E-Mail ?
- Hat er wirklich das geschrieben was ich lese ?
- Woher kommt das Applet, das gerade auf meinen PC geladen wird ?
- Ist das Update wirklich das “richtige, unmanipulierte” Original ?
- Wohin wird meine Kreditkartennummer übermittelt ?

- Wer gibt bei einer Wahl gerade seine Stimme ab ?

- Stammt der Inhalt von “www.admin.ch” wirklich von unserer Regierung ?

28.5.1 Ausgehend von seinem Grundprinzip ist das Internet nicht sicher. Gefahren wären da zum Beispiel.:

- Mitlesen von Daten (Sniffing)

- Vortäuschen falscher Identitäten (Spoofing)

- Angriffe auf die Verfügbarkeit (Denial-of-Service)

- Übertragen von Programmen mit Schadfunktion (Viren, Würmer...)

- Menschliches Fehlverhalten (Preisgabe von geheimen Daten)

28.5.2 Um sicher zu Kommunizieren müssen folgende Ziele verfolgt werden:

- **Authentisierung**
  Sicherstellung der Identität eines Kommunikationspartners

- **Vertraulichkeit**
  Zugänglichkeit der Nachrichteninhalte nur für einen bestimmten Empfängerkreis

- **Integrität**
  Schutz vor Verfälschung von Nachrichten bei der Übermittlung

- **Autorisierung**
  Prüfung der Zugriffsberechtigung auf Ressourcen

- **Verfügbarkeit**
  Schutz vor Verlust von Daten, Sicherstellung des laufenden Betriebs

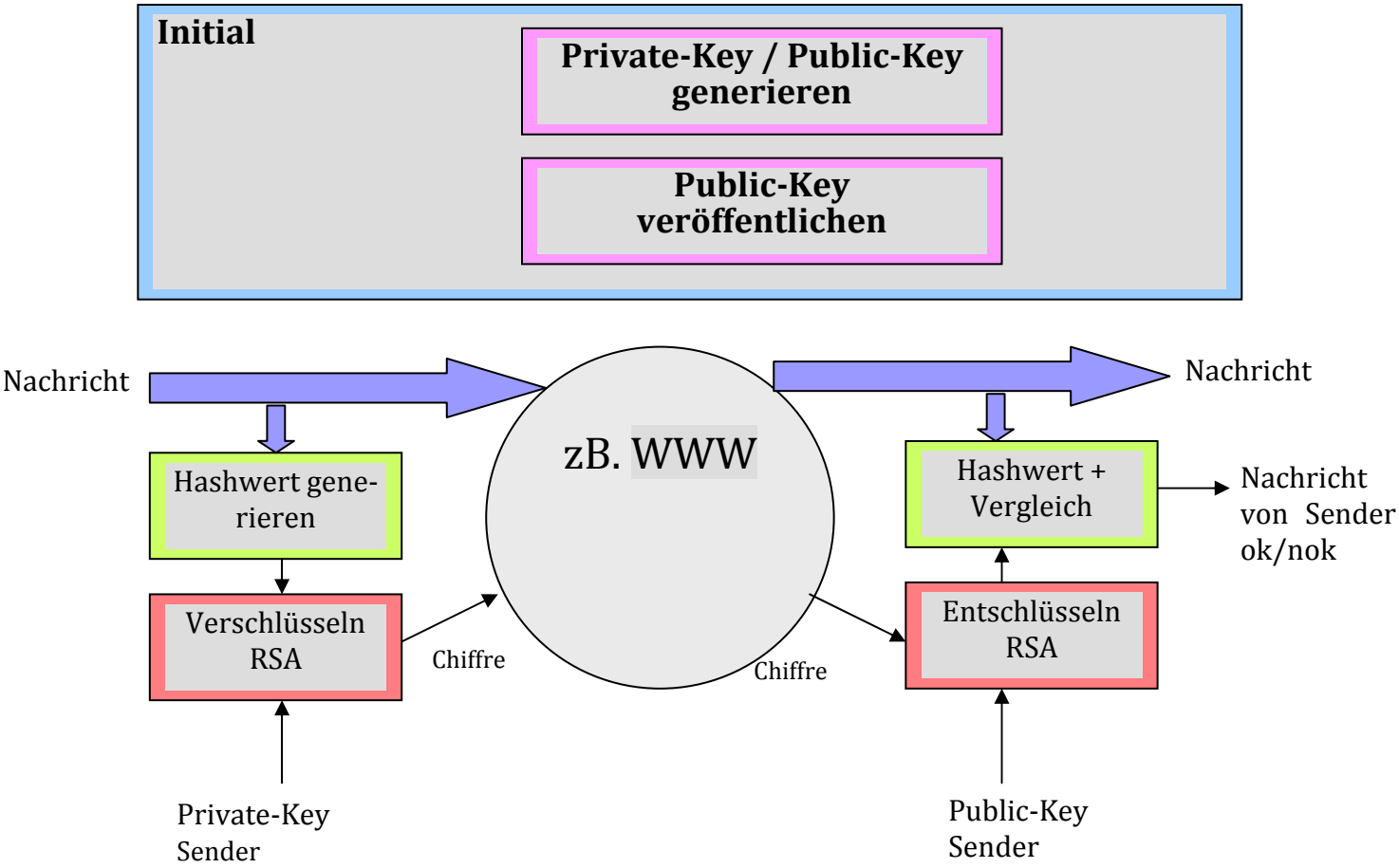
- **Verbindlichkeit**
  Sicherer Nachweis der Absendung/des Empfangs von Nachrichten

28.5.3 Der Ablauf einer digitalen Signatur

Soll sichergestellt werden, dass eine Nachricht auch wirklich von einem Sender stammt, wird digital signiert. Dabei wird vor dem Versenden der Nachricht von dieser einen Hashwert ermittelt. Dies ist eine Art Prüfsumme, und hat immer dieselbe Länge.

Die Nachricht geht unverschlüsselt übers Netz. Der Hashwert der Originalnachricht wird allerdings mit dem Privat-Key des Senders verschlüsselt. Der Empfänger erhält die Nachricht im Klartext und berechnet vorerst einmal den Hashwert. Ausserdem entschlüsselt der Empfänger nun mit dem Public-Key des Senders den verschlüsselten Hashwert. Beide Hashwerte sollten übereinstimmen. Andernfalls wurde die Nachricht irgendwo auf dem Weg zum Empfänger verändert.

Digitale Signatur mit RSA :



28.5.4 Der Hash-Algorithmus

- Der Hash-Algorithmus bildet einen beliebig langen Nachrichtentext auf einen Wert vorgegebener (kurzer) Länge an (Hashwert, “Prüfsumme”)
- Aus dem Hashwert kann die ursprüngliche Nachricht nicht errechnet werden (Irreversibilität)
- Die Konstruktion von Nachrichten mit gleichem Hashwert muss praktisch unmöglich sein
- Die zufällige Übereinstimmung von Hashwerten beliebiger Nachrichten ist unwahrscheinlich (Kollisionsresistenz, Integrität prüfbar)
- Die Hashwertbildung und anschliessende Verschlüsselung gewährleistet eine sichere Identifizierung des Absenders eines Dokumentes
- Die Hashwertbildung und anschliessende Verschlüsselung gewährleistet eine Sicherheit vor nachträglichen Manipulationen des Dokumentes
- Elektronisch signierte Dokumente sind mit unterschriebenen Papierdokumenten gleich gesetzt! Die Rechtssprechung anerkennt heute auch digitale Unterschriften.
- Bekannte Hashfunktionen sind MD5, SHA-1, Tiger und RIPEMD

28.5.5 Wer traut wem?

In der heutigen globalisierten Geschäftswelt ist es nicht immer mehr möglich, seinen Kommunikationspartner persönlich zu kennen. Man muss sich auf die öffentlichen Schlüsseln der Teilnehmer verlassen können. Dazu gibt es zwei Ansätze!

28.5.6 Public Key Infrastruktur

- OpenPGP (ein dezentraler Standard) Anfangs 90‘ Jahre durch Phil Zimmermann entwickelt: Jeder Teilnehmer am Verschlüsselungssystem, der ein gültiges Schlüsselpaar besitzt, kann die Echtheit anderer Schlüssel durch seine Unterschrift bestätigen.

- X.509 (zentral und hierarchisch organisiert) Der Besitzer hat seine Identität/Glaubwürdigkeit durch ein X-509 Zertifikat von einer **CA** (=**C**ertification **A**uthority) bescheinigen lassen Das heisst : Personenangaben werden mit dem Schlüsselpaar gekoppelt (Elektronischer Ausweis), ausgestellt von einer vertrauenswürdigen Instanz **TC** (=**T**rust **C**enter) oder CA

28.5.7 Aufgaben von TC’s bzw. CA's (X.509)

- Zertifikate ausstellen: Korrekte Identifikation des Teilnehmers, Zertifizieren von User, Server, Sub CA's Erstellung des Zertifikates durch digitale Signatur über Identifikationsdaten und öffentlichen Schlüssel eines Teilnehmers

- Bei Ausgabe eines Zertifikates an einen Teilnehmer ist dieser über Funktionalität und Gefahren von zertifizierten Signaturschlüsseln zu unterrichten

- Veröffentlichen der Public-Keys der User

- Publizieren eigenes Zertifikat

- Verwalten bzw. Archivierung angelaufener Zertifikate

- Zertifikate zurückziehen oder für ungültig erklären

- Festlegung der Gültigkeitsdauer von Zertifikaten

- Zeitstempeldienst Absicherung im Falle des Diebstahls des geheimen Schlüssels.Nachweis einer zeitpunktsbezogenen Willensbekundung. Nachweis und Sicherheit, dass die digitale Signatur nicht nach Ablauf der Gültigkeit des Zertifikates erstellt wurde.

28.6 Schwachstellen

28.6.1 Chosen Ciphertext

Der Ablauf:

1. Mallory fängt verschlüsselte Nachricht an Bob ab und besorgt sich den Public-Key von Bob.
2. Mallory wählt Zufallszahl r (r<n und teilerfremd zu n)
3. Mallory : $x=r^e \bmod n$ (e=Public-Key von Bob)
4. Mallory : $y=x * c \bmod n$ (c=Nachricht)
5. Mallory überredet Bob dazu, die errechnete Nachricht y digital zu signieren.
6. Bob : $u=y^d \bmod n$ (d=Private Key von Bob)
7. Mallory : $r^{-1} * u \bmod n = r^{-1} * y^d \bmod n$
   $= r^{-1} * x^d * c^d \bmod n$
   $= r^{-1} * (r^e)^d * c^d \bmod n$
   $= r^{-1} * r * c^d \bmod n$
8. Die Exponenten e und d heben sich gegenseitig auf wie auch r und r^{-1}.
   Es bleibt übrig : $c^d \bmod n$, somit die Klartextnachricht !

Fazit

Unterschreiben sie nie irgendwelche binären Dateien unbekannten Inhalts mit ihrem private Schlüssel. Es könnte sich um einen Chosen-Ciphertext-Angriff handeln. (=Man in the middle-Attack)

28.6.2 Der Geburtstagsangriff

Idee : Zwei Dokumente mit gleichem Hashwert finden

Der Ablauf:

1. Mallory erstellt zwei Versionen eines Dokumentes DokuA "Alice an Bob" und DokuB "Alice an Mallory"
2. Mallory erstellt von beiden Dokumenten verschiedene Varianten mit gleichem Inhalt aber mit geringfügigen Änderungen (zB. Leerschläge etc.)
3. Mallory sucht DokuA und DokuB Varianten mit demselben Hashwert (Wahrscheinlichkeit für Treffer ist relativ hoch)
4. Mallory legt Alice die gefundene Variante von DokuA zur Unterschrift vor.
5. Alice unterschreibt die DokuA für Bob
6. Mallory trennt nun die Signatur von DokuA ab und fügt sie DokuB an.
7. DokuB ist nun ebenfalls gültig!

Fazit

Unterschreiben Sie niemals ein Dokument, (auch wenn es für sie im Klartext abgefasst ist), das sie nicht selber verfasst haben. Es sei denn, sie nehmen vor der Unterzeichnung einige subtile Änderungen vor. ZB: Vor der Unterzeichnung Einfügen von Leerzeichen; Damit ändern sie wiederum den Hashwert der ihnen vorgelegten Daten und machen den Geburtstagsangriff zunichte.

29 Hybride Verfahren

Bei hybrider Verschlüsselung werden die Vorteile von beiden Verfahren, symmetrische und asymmetrische, ausgenutzt :

- Asymmetrisches oder Public-Key-Verfahren für Schlüsselmanagement, zB. RSA
- Symmetrisches Verfahren zum Versenden der eigentlichen Nachricht, zB. RC4, DES

Für weitere Experimente steht Ihnen CrypTool.exe zur Verfügung

29.1.1 Aufgabe zu asymmetrischem Kryptoverfahren

Installieren sie PGP (BSCW) auf ihrem PC. Studieren sie die Handbücher und Anweisungen, die sie dazu auf dem Internet finden. Nun suchen sie sich ein Partner aus, mit dem sie verschlüsselte Texte austauschen werden. Dazu müssen sie sich vorerst Ihre Schlüssel generieren und anschliessend die Public-Keys gegenseitig bekannt machen. Ziel ist, dass jede Gruppe zwei Verschlüsselungs- und Entschlüsselungsvorgänge durchgespielt hat. Zusatzaufgabe: Versuchen sie, anstatt die Botschaft zu verschlüsseln, diese mit einer digitalen Signatur zu versehen. Die Aufgabe ist erfüllt, wenn ihr Partner die Richtigkeit ihres Dokumentes anhand der Signatur verifizieren kann.

Dokumentieren sie alle Arbeiten in einem Journal, das Sie nach der Übung im BSCW-Ordner ablegen.

30 Quellen

ETH Zürich
Universität Bern
Swisseduc
Johann Wolfgang Goethe Universität
Datenformate im Medienbereich, Arne Heyna, Marc Briede, Ulrich Schmidt