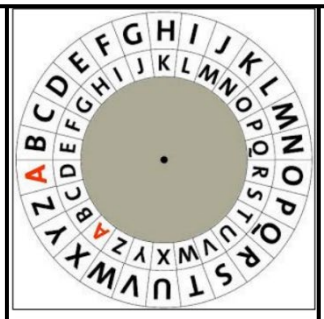
- Vertraulichkeit (Confidentiality)
  Inhalte können von unbefugten nicht eingesehen werden
  (sichere Verschlüsselung)

Hash-Funktion

- Integrität (Integrity)
  Die Inhalte sind unverfälscht und vollständig
  (gute Hash-Funktion)

- Authentizität (Authenticity)
  Sender und Empfänger einander klar bekannt sein
  (gute digitale Signatur-Funktion)

	01010101	(Klartext)
XOR	11001100	(Schlüssel)
	10011001	(Chiffrat)
	10011001	(Chiffrat)
XOR	11001100	(Schlüssel)
	01010101	(Klartext)



DAS IST STRENG GEHEIM → GDV LVW VWUHQJ JHKHLP (Verschlüsselt mit Schlüssel «3» oder «D»)

Ziel:	Identifikation eines Objektes x anhand eines «Fingerabdruckes» (für Signaturen und Prüfsummen)
Umsetzung:	Nicht umkehrbare mathematische Abbildung H(x) - x kann beliebige Länge haben - H(x) hat eine feste Länge - H(x) ist eine Einweg-Funktion - H(x) soll möglichst keine Kollisionen liefern
Standards:	~~MD5~~ (veraltet), CRC-32, MD2, Sha1 u.a.
Beispiel:	x = Beliebige natürliche Zahl → H(x) = x mod 2 (Diese Funktion erfüllt nur das letzte Kriterium nicht...)

Ziel:	Algorithmus für Signaturen **und** Verschlüsselung
Umsetzung:	- Verschlüsseln mit Public-Key / Entschlüsseln mit Private-Key - Signieren mit Hash → siehe Folie 5
Verwendung:	IPSec, SSH, OpenPGP u.v.m.
Vertiefung:	In der Aufgabensammlung

RSA(Rivest, Shamir, Adlemann)

$n = p \bullet q$

$\Phi(n) = (p - 1)(q - 1) = \text{phi}(n)$

$e \bullet d \bmod \Phi(n) = 1$

c = m^e mod n = chiffrat

m = c^d mod n = message

öffentlicher Schlüssel = {d, n} (Bearbeitet)

Privater Schlüssel = {e, n}

Einheit	Zusammenhang	Anzahl Bit
1 Bit (b)	Grundeinheit	$1 \cdot 10^0$ Bit
1 Byte (B)	8 Bit	$8 \cdot 10^0$ Bit
1 Kilobyte (KB)	1000 Byte	$8 \cdot 10^3$ Bit
1 Megabyte (MB)	1000 Kilobyte	$8 \cdot 10^6$ Bit
1 Gigabyte (GB)	1000 Megabyte	$8 \cdot 10^9$ Bit
1 Terabyte (TB)	1000 Gigabyte	$8 \cdot 10^{12}$ Bit
1 Petabyte (PB)	1000 Terabyte	$8 \cdot 10^{15}$ Bit

IEC-Bezeichnung	Zusammenhang	Anzahl Bit
Bit	Grundeinheit	$1 \cdot 2^0$ Bit
Byte	8 Bit	$8 \cdot 2^0$ Bit
Kibibyte (KiB)	1024 Byte	$8 \cdot 2^{10}$ Bit
Mebibyte(MiB)	1024 Kibibyte	$8 \cdot 2^{20}$ Bit
Gibibyte (GiB)	1024 Mebibyte	$8 \cdot 2^{30}$ Bit
Tebibyte (TiB)	1024 Gibibyte	$8 \cdot 2^{40}$ Bit
Pebibyte (PiB)	1024 Tebibyte	$8 \cdot 2^{50}$ Bit

Samplingrate: Auch: „Abtastrate“
Wie oft wird das Signal pro Sekunde abgetastet.
Angabe in Hertz (16kHz = 16'000 mal pro Sekunde)

Samplingtiefe: Auch: «Abtasttiefe»
Wie viele Bit stehen für die Messskala zur Verfügung
Z.B. 8 Bit für eine Skala mit 256 Einheiten
oder 16 Bit für eine Skala mit 65'536 Einheiten.

WAV-Dateien: Samplingraten von 1Hz bis 4.3GHz

MP3: Samplingraten 32kHz, 44.1kHz und 48kHz

Welchen Speicherplatz würde die .wav-Datei von der vorletzten Folie benötigen, wenn sie 3.5 Minuten dauern würde?

210 sec x 44'100 Abtastungen x 2 Byte = **18'522'000 Byte,**
das Ganze dann auch noch in Stereo (also doppelt):

also ca. **37 MB**

- Ausführbare Dateien — Anwendungen oder Dateien mit Befehlen / Scripts (.exe, .bat, .php, etc.)
- Systemdateien — Dienen zur Konfiguration von Hard- und Software (Conf-Files) (.bin, .drv, .ini, .sys, etc.)
- Bibliotheken — Enthalten Programmierwerkzeuge für Anwendungen (.cls, .dat, .dll, etc.)
- Nutzerdaten — Vom Nutzer (mit verschiedenen Anwendungen) erstellte Dateien

Text	Bild	Audio	Video	MS Office	Andere
.txt .docx .rtf	.bmp .jpg .gif .svg .png	.wma .mp3 .wav	.avi .wmv .mov .mp4	.docx .xlsx .pptx	.zip .rar .tar .html .csv
.pdf					

```
<svg height="200" width="200">
  <path d="M5 100 L100 5 L195 100 L100 195 L5 100"
        fill="none" stroke="black"/>
</svg>
```

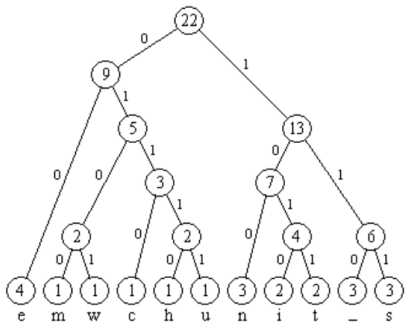
Erklärung:

<svg> definiert die Arbeitsfläche
<path> definiert einen Zeichnungspfad
M X Y Springe zu Punkt X Y (Move)
L X Y Zeichne eine Linie zu Punkt X Y

Anleitung zur Huffman – Codierung

Zu komprimierender Text: IM WESTEN NICHTS NEUES

1. Schreibe jedes vorkommende Zeichen unten auf ein Blatt.
2. Schreibe über jedes Zeichen seine Häufigkeit.
3. Verbinde immer die beiden tiefsten (freien) Häufigkeitswerte (oder Knoten nach oben zu einem Summenknoten.
4. Wiederhole Schritt 3 bis Du den Stammknoten gebildet hast (Totalsumme)
5. Male unter jeden Knoten links eine null und rechts eine eins.
6. Nun kannst Du den Binärcode jedes Zeichens vom Stammknoten her ablesen.



E	00
M	0100
W	0101
C	0110
H	01110
U	01111
N	100
I	1010
T	1011
[ ]	110
S	111

Codierter Bitstrom (73 Bit):

1010010011001010011110110010011010010100110011101011111111010000011111001111

Merke: Die Kompressionsrate ist der Kehrwert des Kompressionsfaktors!

$KF = \frac{komprimierte\ Grösse}{ursprüngliche\ Grösse}$ $KR = \frac{ursprüngliche\ Grösse}{komprimierte\ Grösse}$

«Real Color» 5 Bit/Farbe, 15 Bit/Pixel 2^{15} = 32'768 Farben
«True Color» 8 Bit/Farbe, 24 Bit/Pixel 2^{24} = 16'777'216 Farben

GROSS: Absolute Koordinatenangaben (bezogen auf den Ursprung des Koordinatensystems)

KLEIN: Relative Koordinatenangaben (bezogen auf aktuelle Cursor-Position)

```
<svg height="400" width="400">
  <path d="M150 0 L75 200" stroke="blue" stroke-width="5" />
</svg>
```

```
<svg height="400" width="400">
  <circle cx="200" cy="200" r="150" stroke="black" stroke-width="1" fill="red" />
</svg>
```

SVG