

# Steganographie in Audio- und Videodateien

René Frerichs

# Steganographie mit Audio- und Videodateien

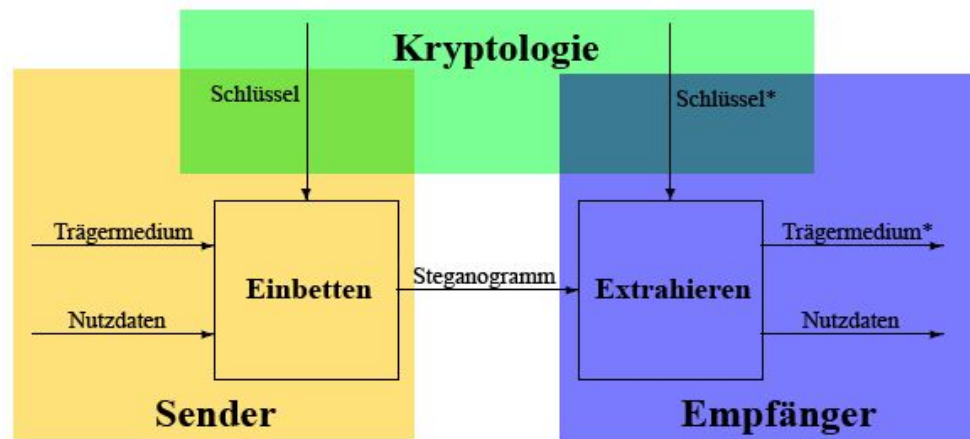
## Ein erster Überblick

1. Steganographie und Kryptographie
2. Audio- und Videodateien als Trägermedium
3. LSB-Verfahren
4. Steganographie in Videodateien
5. Steganographische Programme
6. Fazit

# Steganographie mit Audio- und Videodateien

## 1. Steganographie und Kryptographie

- Steganographie: Verstecken von Informationen vor Dritten
- Heute häufig rechnergestützte Steganographie
- Kryptographie: Verschlüsseln von Informationen



- Kombination:
  - Doppelte Sicherheit
  - Verschlüsselte Nachricht ähnelt Rauschen

# Steganographie mit Audio- und Videodateien

## 2. Audio- und Videodateien als Trägermedium

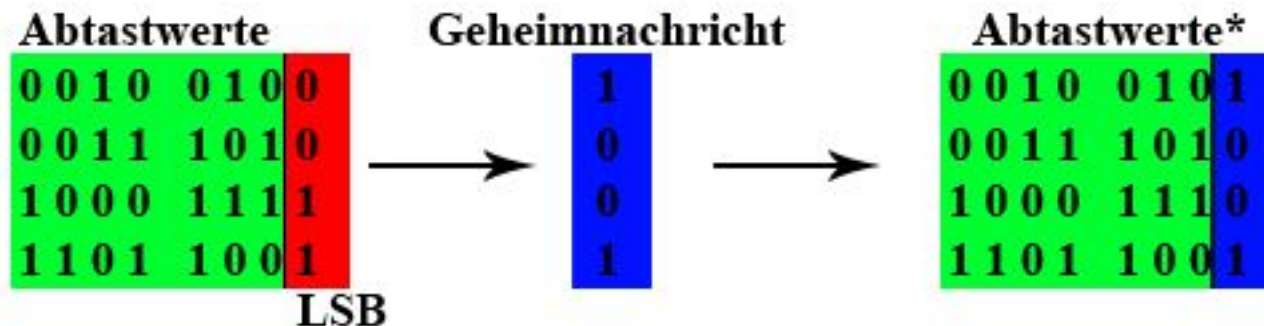
- "gewöhnlicher" Dateityp
- Große Datenmengen
- Kapazität abhängig von Wahrnehmungsschwäche
- Zufallsanteil

# Steganographie mit Audio- und Videodateien

## 3. LSB-Verfahren

LSB = Least Significant Bits (niedrigstwertige Bit)

- Audio- und Bilddateien



- Keine Größenänderung durch Bitvertauschen
- Im Beispiel eine Kapazität von 12,5%
- S-Tools

# Steganographie mit Audio- und Videodateien

## 4. Steganographie in Videodateien

- Sowohl Audio- als auch Videoanteil
- Im Folgenden nur Video
- JPEG als Beispiel
  - M-JPEG
  - MPEG1 – MPEG2

# Steganographie mit Audio- und Videodateien

## 4. Steganographie in Videodateien

Beispiel nach Andreas Westfeld

JPEG – Komprimierung

- Helligkeitsmatrix
- Diskrete Kosinustransformation (DCT)
- Quantisierung

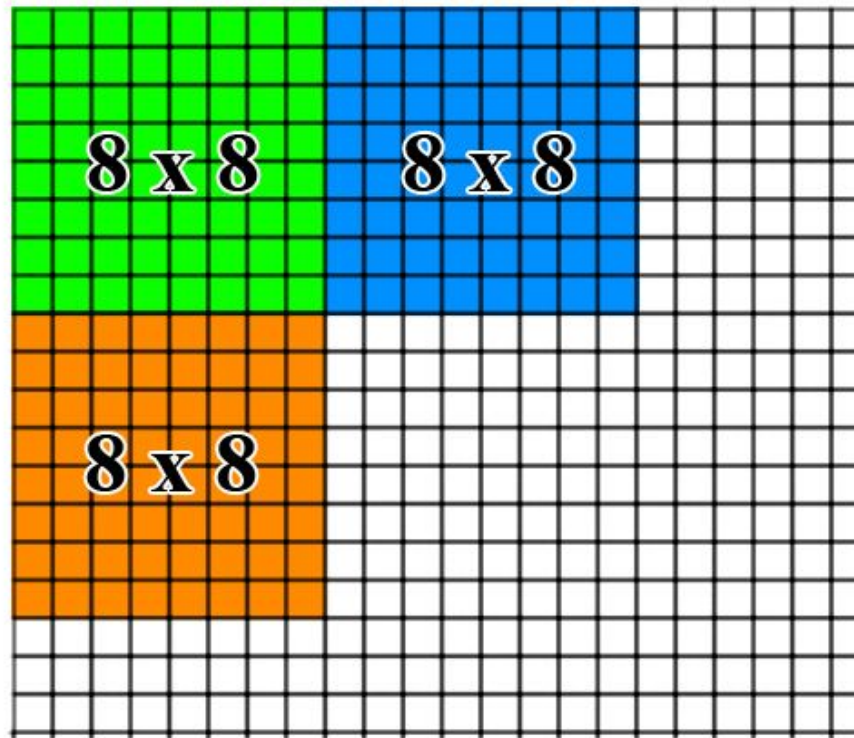
Einbettung

- Blockqualität überprüfen
- Einbetten als Parität – Bit ändern wenn nötig

# Steganographie mit Audio- und Videodateien

## 4.1 JPEG - Komprimierung: Blockbildung

Einteilung des Quellbildes in 8x8-Blöcke





# Steganographie mit Audio- und Videodateien

## 4.2 JPEG - Komprimierung: Helligkeitsmatrix

Helligkeitswerte eines Blocks als Matrix



=

20	20	36	49	50	38	20	20
20	51	65	65	65	65	53	21
36	65	65	65	65	65	65	39
49	65	65	65	65	65	65	52
50	65	65	65	65	65	65	53
38	65	65	65	65	65	65	41
20	53	65	65	65	65	55	21
20	21	39	52	53	41	21	20

# Steganographie mit Audio- und Videodateien

## 4.3 JPEG - Komprimierung: Transformation

Zweidimensionale diskrete Kosinustransformation  
(DCT)

- Transformation
- Rücktransformation

$$f(k, n) = \frac{C(k)}{2} \frac{C(n)}{2} \sum_{x=0}^7 \sum_{y=0}^7 F(x, y) \cos\left(\frac{\pi(2x+1)k}{16}\right) \cos\left(\frac{\pi(2y+1)n}{16}\right)$$
$$F(x, y) = \sum_{k=0}^7 \sum_{n=0}^7 \frac{C(k)}{2} \frac{C(n)}{2} f(k, n) \cos\left(\frac{\pi(2x+1)k}{16}\right) \cos\left(\frac{\pi(2y+1)n}{16}\right)$$

# Steganographie mit Audio- und Videodateien

## 4.4 JPEG - Komprimierung: Transformation

DCT-Matrix (8 x 8)

$$\begin{bmatrix} 20 & 20 & 36 & 49 & 50 & 38 & 20 & 20 \\ 20 & 51 & 65 & 65 & 65 & 65 & 53 & 21 \\ 36 & 65 & 65 & 65 & 65 & 65 & 65 & 39 \\ 49 & 65 & 65 & 65 & 65 & 65 & 65 & 52 \\ 50 & 65 & 65 & 65 & 65 & 65 & 65 & 53 \\ 38 & 65 & 65 & 65 & 65 & 65 & 65 & 41 \\ 20 & 53 & 65 & 65 & 65 & 65 & 55 & 21 \\ 20 & 21 & 39 & 52 & 53 & 41 & 21 & 20 \end{bmatrix} \xrightarrow{\text{DCT}} \begin{bmatrix} 50 & -4 & -84 & -2 & -31 & -2 & -12 & -1 \\ -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -84 & 0 & -29 & 2 & 15 & 0 & 10 & 0 \\ -2 & 0 & 2 & 0 & 1 & 0 & 0 & 0 \\ -31 & 0 & 15 & 1 & 28 & -1 & 10 & -1 \\ -2 & 0 & 0 & 0 & -1 & 0 & -2 & 0 \\ -12 & 0 & 10 & 0 & 10 & -2 & 0 & -1 \\ -1 & 0 & 0 & 0 & -1 & 0 & -1 & 0 \end{bmatrix}$$

# Steganographie mit Audio- und Videodateien

## 4.5 JPEG - Komprimierung: Transformation

- Linearkombination der DCT-Matrix mit DCT-Basisbildern ergeben Helligkeit des Quellblocks.

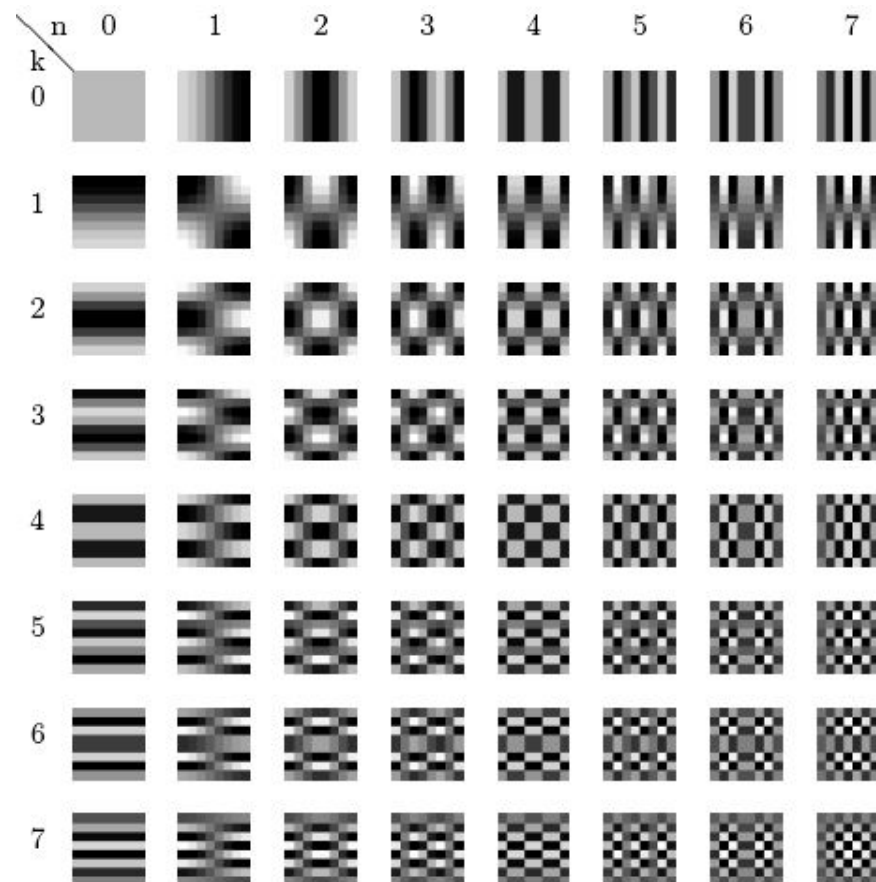
The diagram illustrates the DCT transformation of an 8x8 pixel block. On the left, a tilted rectangle is labeled "8x8-Bild". This is followed by an equals sign and a series of terms representing a linear combination of DCT basis images. The first term is  $f(0,0)$  multiplied by a tilted rectangle representing the DC component (a uniform gray). This is followed by a plus sign,  $f(0,1)$  multiplied by a tilted rectangle representing the first AC component (a horizontal gradient). This is followed by a plus sign, an ellipsis, a plus sign, and  $f(7,7)$  multiplied by a tilted rectangle representing the last AC component (a checkerboard pattern).

$$\text{8x8-Bild} = f(0,0) \cdot \text{Basisbild}_0 + f(0,1) \cdot \text{Basisbild}_1 + \dots + f(7,7) \cdot \text{Basisbild}_N$$

# Steganographie mit Audio- und Videodateien

## 4.6 JPEG - Komprimierung: Transformation

DCT-Basisbilder



# Steganographie mit Audio- und Videodateien

## 4.7 JPEG - Komprimierung: Quantisierung

- Quantisierungsfaktor Q für ganzen Block außer (0,0)
- Für Dekompression wird Q mit übertragen
- Verringerung der Werte – mehr Nullen
- Verlustbehaftet

$$\begin{bmatrix} 50 & -4 & -84 & -2 & -31 & -2 & -12 & -1 \\ -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -84 & 0 & -29 & 2 & 15 & 0 & 10 & 0 \\ -2 & 0 & 2 & 0 & 1 & 0 & 0 & 0 \\ -31 & 0 & 15 & 1 & 28 & -1 & 10 & -1 \\ -2 & 0 & 0 & 0 & -1 & 0 & -2 & 0 \\ -12 & 0 & 10 & 0 & 10 & -2 & 0 & -1 \\ -1 & 0 & 0 & 0 & -1 & 0 & -1 & 0 \end{bmatrix} \xrightarrow{\text{DCT}} \begin{bmatrix} 2 & 0 & -8 & 0 & -3 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -8 & 0 & -2 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3 & 0 & 1 & 0 & 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

# Steganographie mit Audio- und Videodateien

## 4.8 Einbettung: Blockqualitätsbestimmung

Eignung auf Einbettung testen:

- Koeffizienten aus quant. DCT-Matrix multiplizieren mit Wert aus Tabelle
  - Wenn größer 100%
    - Block geeignet → Bit wird eingebettet
  - Wenn kleiner 100%
    - Block ungeeignet → keine Einbettung

# Steganographie mit Audio- und Videodateien

## 4.9 Einbettung: Blockqualitätsbestimmung

Eignung auf Einbettung testen:  
(Spalte 0 wird nicht betrachtet)

$$|-8| * 6,75\% = 54\%$$

54% < 100%  $\rightarrow$  Block ungeeignet

2	0	-8	0	-3	0	-1	0
0	0	0	0	0	0	0	0
-8	0	-2	0	1	0	1	0
0	0	0	0	0	0	0	0
-3	0	1	0	2	0	1	0
0	0	0	0	0	0	0	0
-1	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0

$k$	$n$	$k'$	$n'$	$\frac{\Delta f(k',n')}{f(k,n)}$
0	0	-	-	(0,00) %
0	1	0	2	1,51 %
0	2	0	1	6,75 %
0	3	0	2	9,19 %
0	4	0	3	12,27 %
0	5	0	4	15,16 %
0	6	0	5	18,90 %
0	7	0	6	24,60 %
1	0	3	0	(0,27) %
1	1	1	0	3,61 %
1	2	1	1	6,79 %
1	3	1	2	9,28 %
1	4	1	3	12,28 %
1	5	1	4	15,13 %
1	6	1	5	18,98 %
1	7	1	6	24,55 %



# Steganographie mit Audio- und Videodateien

## 4.10 Einbettung: Blockqualitätsbestimmung

Eignung auf Einbettung testen:  
(Spalte 0 wird nicht betrachtet)

$$|-16| * 6,75\% = 108\%$$

$108\% > 100\%$   $\rightarrow$  Block geeignet

4	0	-16	0	-6	0	-2	0
0	0	0	0	0	0	0	0
-16	0	-5	0	3	0	2	0
0	0	0	0	0	0	0	0
-6	0	2	0	5	0	2	0
0	0	0	0	0	0	0	0
-2	0	2	0	2	0	0	0
0	0	0	0	0	0	0	0

$k$	$n$	$k'$	$n'$	$\frac{\Delta f(k',n')}{f(k,n)}$
0	0	-	-	(0,00) %
0	1	0	2	1,51 %
0	2	0	1	6,75 %
0	3	0	2	9,19 %
0	4	0	3	12,27 %
0	5	0	4	15,16 %
0	6	0	5	18,90 %
0	7	0	6	24,60 %
1	0	3	0	(0,27) %
1	1	1	0	3,61 %
1	2	1	1	6,79 %
1	3	1	2	9,28 %
1	4	1	3	12,28 %
1	5	1	4	15,13 %
1	6	1	5	18,98 %
1	7	1	6	24,55 %

# Steganographie mit Audio- und Videodateien

## 4.11 Einbettung: Parität berechnen

Parität des Blockes berechnen:

Summe der Beträge der Koeffizienten modulo 2

$$4 + |-16| + |-6| + |-2| + \dots = 72$$

$$72 \text{ modulo } 2 = 0$$

$$\begin{bmatrix} 4 & 0 & -16 & 0 & -6 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -16 & 0 & -5 & 0 & 3 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -6 & 0 & 2 & 0 & 5 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 0 & 2 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

# Steganographie mit Audio- und Videodateien

## 4.12 Einbettung: Parität

Geheiminformation wird als Parität von guten Blöcken übermittelt.

Parität = Geheimes Bit

→ Block wird unverändert übertragen

Parität  $\neq$  Geheimes Bit

→ Block wird angepasst

# Steganographie mit Audio- und Videodateien

## 4.13 Einbettung: Parität anpassen

Koeffizient  $(k,n)$  ist der, der laut Tabelle die größte Änderung zulie.

- Wenn  $|(k, n-1)| \geq 1 \rightarrow (k, n-1)$  verringert um 1
- Wenn  $|(k, n-1)| = 0 \rightarrow (k, n-1) = 1$

Die Parit kippt.

$$\begin{bmatrix} 4 & 0 & -16 & 0 & -6 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -16 & 0 & -5 & 0 & 3 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -6 & 0 & 2 & 0 & 5 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 0 & 2 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 1 & -16 & 0 & -6 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -16 & 0 & -5 & 0 & 3 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -6 & 0 & 2 & 0 & 5 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 0 & 2 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

# Steganographie mit Audio- und Videodateien

## 4.14 Empfänger

- unterscheidet mittels gleicher Tabelle zwischen „guten“ und „schlechten“ Blöcken
- bekommt Geheimbotschaft durch die Parität der „guten“ Blöcke
- Kann Trägervideomaterial zurückgewinnen:
  - Inverse Quantisierung
  - Rücktransformation

# Steganographie mit Audio- und Videodateien

## 4.15 Eigenschaften

- Maximale Kapazität  $\leq (\text{Anzahl der Blöcke}) * \text{Bilder}$
- Kapazität abhängig von Quelle

# Steganographie mit Audio- und Videodateien

## 5. Programme

### S-Tools

- Audio (WAV, AU), Bilder (BMP, GIF)
- LSB-Verfahren

### Steghide

- Audio (WAV, AU), Bilder (BMP, GIF)
- Erweitertes LSB-Verfahren

### MP3-Stego

- WAV → MP3
- nur Textdateien (.txt)
- Einbettung während Komprimierung

# Steganographie mit Audio- und Videodateien

## 5.1 MP3Stego: Funktionsweise

- Verschlüsselung mit TripleDES
- Einbettung bei Komprimierungs-Vorgang
- Änderung der Quantisierungsschleife
  - Nach Quantisierung:
    - Parität der Blocklänge = Geheimbit
      - Block nicht verändern
    - Parität der Blocklänge  $\neq$  Geheimbit
      - ein weiterer Quantisierungsschritt



# Steganographie mit Audio- und Videodateien

## 5.2 MP3Stego: Anmerkungen

- Verkürzte Blöcke überlassen Ressourcen folgenden Blöcken
  - > gleiche Datengröße wie normal MP3-Komprimierung
- Parität der Blocklänge „kippt“ bei Quantisierung nicht immer
  - > mehrere zusätzliche Quantisierungsschritte
  - > einige Blöcke stärker komprimiert (Qualitätsverlust)
- Angriff durch Encoder-Analyse naheliegend.

# Steganographie mit Audio- und Videodateien

## 5.3 Anmerkungen zu den Programmen

- Kapazität
  - S-Tools 12,5%
  - Steghide 5%
  - MP3Stego 0,1%
- Alle Programme sind nicht Robust gegen Formatänderung/Dateiänderung.

# Steganographie mit Audio- und Videodateien

## 6. Fazit