Steganographie

WS 01/02, Belegarbeit, Prof. Dr. Blakowski

Ulrike Nehls,
Frank Gleichmann,
03. Januar 2002

Inhalt

1. Einleitung

1.1. Aktueller Bezug

Im Zuge der aktuellen Ereignisse nach dem 11. September, ist ein spezielles Gebiet der Informatik in den Fokus der öffentlichen Aufmerksamkeit gerückt worden – die Steganographie.

Die Geheimdienste konnten die US-amerikanische Bevölkerung nicht vor den terroristischen Angriffen auf das World-Trade-Center und das Pentagon warnen.

Warum dies so ist, dazu gibt es verschiedene Theorien. Eine davon beschäftigt sich mit dem Thema Steganographie.

Abbildung 2 "ground zero" Das zerstörte WTC
Quelle: www.spiegel.de

Es wird vermutet, dass die Gruppe, die für die Durchführung der Attentate zuständig war, ihre Befehle über eMails und Web-Seiten erhalten hatte. In harmlos aussehenden Inhalten, Bildern oder Audio-Dateien wurden mit Hilfe der Steganographie wichtige Botschaften versteckt. Diese Botschaften sind schwer ausfindig zu machen. Allein die Menge von Millionen eMails und Web-Pages erschweren den Scannern der Geheimdienste solche Nachrichten ausfindig zu machen.

Dies impliziert nun einen negativen Eindruck. Deshalb beginnen wir im Anfangsstadium der Steganographie. Diese Anfänge sind viel älter als heutige Computer.

1.2. Geschichte der Steganographie

Seit vielen Jahrhunderten werden wichtige Nachrichten so verschickt, dass diese für uneingeweihte Augen nicht zu erkennen sind. Dies gilt besonders in sensiblen Bereichen wie dem Militär oder der Wirtschaft. Bei der Steganographie geht es also hauptsächlich darum, Daten nicht als verschlüsselt und somit wichtig darzustellen. Die Informationen werden „verharmlost" und suggerieren etwaigen Angreifern, dass eine Untersuchung des Inhaltes nicht nötig ist.

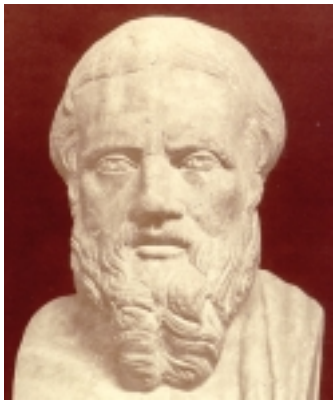


Abbildung 3 Herodot | Quelle: http://www. phil.uni-Erlangen.de/ ~p1altar/photo_html /portraet/ griechisch/ geschichtsschreiber/ herodot/herodot.html

„ Schon der griechische Geschichtsschreiber Herodot (490-425 v. Chr.), berichtet von einem Adligen, der seine Geheimbotschaft auf den geschorenen Kopf eines Sklaven tätowieren ließ. Nachdem das Haar nachgewachsen war, machte sich der Sklave unbehelligt zu seinem Ziel auf, wo er zum Lesen der Nachricht wiederum kahlrasiert wurde.

In einem anderen Bericht von Herodot geht es um Wachstafeln, auf die man damals schrieb. Als eine sensible Nachricht überbracht werden sollte, entfernte der Absender das Wachs, gravierte den Text in das Holz darunter und füllte das Wachs wieder auf. Den kontrollierenden Wachen erschienen die Tafeln leer.

Der Gebrauch *unsichtbarer Tinte* war bereits zur Zeit des römischen Schriftstellers Plinius der Ältere (23-79 n. Chr.) bekannt: Mit Urin, Milch, Essig oder Fruchtsäften wurde die Nachricht auf Papier oder Pergament geschrieben und war nach dem Trocknen der Flüssigkeit nicht mehr zu sehen. Der Empfänger musste nur das Dokument über einer Kerzenflamme erhitzen - schon tauchte die Schrift wieder auf.

Im Zweiten Weltkrieg ... entwickelten deutsche Spione den sogenannte *Microdot*, ein Stück Mikrofilm in der Größe eines I-Punktes, der in unverdächtigen Schreibmaschinenseiten als Satzzeichen oder oberhalb des Buchstabens "i" eingeklebt wurde. Solche Microdots konnten riesige Datenmengen einschließlich technischer Zeichnungen und Fotos enthalten.

Um Spionen das Übermitteln versteckter Informationen zu erschweren, reglementierten die Regierungen von Großbritannien und den USA im Zweiten Weltkrieg die internationalen Postsendungen. Verboten war das Verschicken von Schachaufgaben, Kreuzworträtseln, Zeitungsausschnitten, Strickmustern, Liebesbriefen und Kinderzeichnungen. Blumengrüße, Musikwünsche im Radio und Chiffreanzeigen waren suspekt und wurden eingeschränkt. Teilweise formulierten die Zensurbehörden der Regierungen sogar abgefangene Briefe um oder klebten die Briefmarken auf den Umschlägen an andere Positionen.“[1]

[1] Artikel: Sag's durch die Blume, Marit Köhntopp, http://www.koehntopp.de/marit/publikationen/steganographie/

2. Grundkonzept der Steganographie

An folgendem Beispiel kann man das Grundkonzept der Steganographie sehr gut demonstrieren:

Abbildung 4 Postkarte mit Steganogramm

Der Text der Nachricht klingt harmlos. Wenn man jedoch die jeweils ersten Buchstaben der einzelnen Worte betrachtet, erhält man die Nachricht: HILFE.
Dies zeigt sehr gut die wichtigsten Konzepte der Steganographie:

1. Die Nachricht ist versteckt, d.h. es ist nicht zu erkennen, dass eine andere Nachricht als die Offensichtliche übermittelt werden soll.
2. Es wurde ein harmlos erscheinendes Trägermedium, hier der nette Urlaubsgruß, zum Verstecken der Botschaft benutzt.
3. Die Entzifferung des Geheimnisses ist nur demjenigen möglich, der den entsprechenden „Algorithmus" zur Auflösung der Mitteilung besitzt.
4. Bei Verdacht auf eine versteckte Nachricht lassen sich immer noch verschiedene Botschaften herauslesen.
5. Die Menge der versteckten Informationen ist zumeist sehr viel geringer, als die Nachricht in der sie verpackt ist.

Noch einmal zusammenfassend:
Steganographie ist die älteste Methode, um Nachrichten für unbefugte Personen nicht zugänglich zu machen. Diese ist aber nicht zu verwechseln mit der Kryptographie. Kryptographie beinhaltet das Verschlüsseln von Informationen. Es ist definiert als erkennbare Benutzung von Kryptographie-Systemen.
Im Gegensatz dazu verstehen wir unter Steganographie den verdeckten Gebrauch eines Verfahrens, mit dessen Hilfe eine Botschaft in einem un-

scheinbaren Trägermedium versteckt wird. Das bedeutet nicht zwingend, dass die Daten verschlüsselt sein müssen. Sie werden nur in einer bestimmten Art und Weise in einem Trägermedium eingebettet. Steganographie ist nicht so robust gegenüber Entschlüsselung wie die Kryptographie. Ist erst mal entdeckt worden, das die vorliegenden Daten ein Steganogramm sind, ist dieses schnell entschlüsselt. Deshalb sollten die Daten vorher verschlüsselt werden, sozusagen, als doppelte Sicherheit.

> ***Steganographie* ist die Wissenschaft, Information unsichtbar zu verbergen.**
> **Ein *Steganogramm* ist ein mit einer Nachricht versehenes Trägermedium.**

3. Computergestützte Steganographie

In der heutigen Gesellschaft benutzt man natürlich keine Sklaven mehr, denen der Kopf rasiert wird um eine Nachricht einzutätowieren. Hierbei würde gegen diverse Menschenrechte verstoßen. Also benötigt man andere Verfahren und Trägermedien.
In nachfolgender Abbildung wird der schematische Ablauf einer Informationsübermittlung per Stenographie beschrieben:

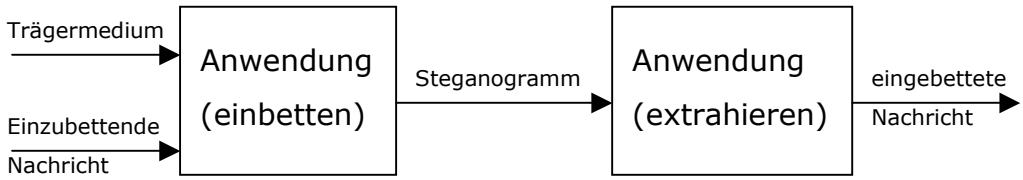


Abbildung 5 schematische Darstellung Steganographie, Quelle: in Anlehnung an: Artikel von Andreas Westfeld aus der C't 9/2001

3.1. Steganographie – Wozu ?

Wer benötigt heute eigentlich Steganographie und warum? Im Moment denkt man natürlich, Menschen verschlüsseln ihr Nachrichten nur deshalb, um „böse" Absichten zu verschleiern. Mitnichten! Es werden hauptsächlich Nachrichten versteckt, um den Missbrauch von Informationen zu vermeiden.
So gibt es in der Wirtschaft viele Bereiche, in denen Geheimhaltung ein wichtiges Thema ist. Wirtschaftsspionage ist eine wirkliche Bedrohung. Wo es zum

Beispiel gilt Patentrechte zu schützen, sind Forschungsdaten natürlich anfällig für Lauschangriffe, da viele über elektronische Medien ausgetauscht werden. In diesen Bereich fällt auch das Abhören durch das amerikanische Echelon-System. Angeblich wurden durch dieses „Spionage-System" europäische Firmen, die neue Geschäftskonzepte entwickelt haben, schon um die Patentierung ihrer Ideen gebracht.
Zum Schutz dieser Daten kann man sie natürlich verschlüsselt übermitteln. Doch dies kennzeichnet natürlich entsprechende Dateien als wichtig. Um einem potentieller Späher auch noch diesen Hinweis zu nehmen, nutzt man Steganogramme.
Weitere Anwendungsgebiete sind:

- Militärdaten
- Wirtschaftsdaten
- Vertragsverhandlungen
- Forschung- und Entwicklung
- eBanking
- eMail
- Politische Kommunikation
- Bankdaten ...

Durch das große Interesse von Wirtschaft, Militär, Regierung usw. an der sicheren Übertragung von Informationen, sind natürlich viele Programme für diesen Sektor entwickelt worden.
Aber auch Privatleute können diese Software nutzen. Viele Entwicklungsfirmen stellen ihre Produkte als Free- und Shareware ins Internet.

3.2. Trägermedien

Zum Verstecken der Daten können verschiedene Trägermedien genutzt werden:

- Text:
  Textnachrichten, wie eMail-Texte, elektronische Briefe, Skripte usw.
- Bild: [jpeg-, gif-, bmp-Dateien]
  alle Formate, für die es Stenographie-Verfahren gibt
- Ton: [wav-Dateien]
  alle Formate, für die es Stenographie-Verfahren gibt

Diese Trägermedien können offensichtlich oder als Anhang an entsprechende eMails verschickt werden.

3.2.1. Eignung der Trägermedien

Die oben genannten Arten von Trägermedien bieten unterschiedlich gute Eigenschaften, die für die Steganographie ausgenutzt werden können.

- **Textdateien** eignen sich sehr schlecht für die Steganographie. Erstens sind Textdateien relativ klein, wodurch sich nur wenig Information verstecken lässt, zweitens ist es sehr schwer, in einer Textdatei Informationen zu verstecken, ohne diese erkennbar zu verändern.
- **Bilder** eignen sich sehr gut und werden derzeit wohl als häufigstes Medium für die Steganographie eingesetzt. Bildern können ohne merkliche Veränderungen bequem Daten hinzugefügt werden. Ähnlich wird ja auch beim JPEG-Format vorgegangen, wo Daten vom Bild absichtlich nicht mitgespeichert werden, um die Größe der Datei zu senken. Der Unterschied ist mit freiem Auge nicht erkennbar.
- **Audio** eignet sich auch sehr gut zur Steganographie. Hier können die Hintergrundgeräusche (Rauschen etc.), die durch das Analog/Digital-Wandeln entstanden sind, genutzt werden, um Information darin zu verstecken.[2]

3.2.2. Techniken bei den verschiedenen Trägermedien

Bit ins Bild

Zur Speicherung von Bilddateien existieren viele verschiedene Formate. Für die Steganographie unterteilen wir Sie in

- nicht verlustbehaftete Formate
  z.B. bmp
- Indexfarbenbilder
  z.B. gif
- verlustbehaftet komprimierende Formate
  z.B. jpeg,.

Diese Unterscheidung ist wichtig, da bei den jeweiligen Formaten unterschiedliche Schemata zur Erstellung eines Steganogramms genutzt werden.

[2] Artikel : "Hinters Licht geführt", Jürgen Rinks, aus c't 6/97, Seite 330

Computerfarben

Eine Computerfarbe besteht aus den drei Farbanteilen: Rot, Grün, Blau (RGB). Die Farbe eines Pixels (Bildpunktes) ergibt sich als Mischungsverhältnis daraus.

Dabei werden die Farbanteile in Intensitäten ausgedrückt, wobei 0 für nicht vorhanden und 255 für volle Intensität steht. Bei einem Byte pro Farbanteil belegt also der Farbwert eines Bildpunkt drei Byte.

Pixelfarbe	Binär			Intensität			Hexadezimal		
	Rot	grün	blau	r	g	b	r	g	b
Weiß	1111 1111	1111 1111	1111 1111	255	255	255	FF	FF	FF
Schwarz	0000 0000	0000 0000	0000 0000	0	0	0	0	0	0
Rot	1111 1111	0000 0000	0000 0000	255	0	0	FF	0	0
Grün	0000 0000	1111 1111	0000 0000	0	255	0	0	FF	0
Blau	0000 0000	0000 0000	1111 1111	0	0	255	0	0	FF

Schema für verlustfreie Formate

Einleitung

Bei diesen Verfahren wird die zu übermittelnde Nachricht in den niederwertigsten Bits der Farbwerte eines Bildpunktes versteckt, da hierbei die geringsten Veränderungen gegenüber dem Original-Bild entstehen.

Ein Beispiel zum Verständnis:

Bits	Wert
111	7
011	3
101	5
110	6

Bei voller Belegung der ersten 3 Bits in der Binärdarstellung, ergibt sich der Wert 7. In diese Kombination soll nun eine Nachricht eingebettet werden. Nutzt man dafür das Bit auf der 2^2 Position, verändert sich der entsprechende Wert auf 3. Nutzt man das letzte Bit, ist der Unterschied am geringsten.[3]

Ein Bildpunkt, der durch 3 Byte beschrieben wird, kann also 2^{24} Farbnuancen darstellen (16,8 Mio. Farben) Das niederwertigste Bit hat dabei, wie oben beschrieben, den kleinsten Einfluss auf die Farbgebung.

[3] „Tarnkappe kontra Krypto-Verbot" Ralf Schneider, aus C't, 19.04.01

Einfügen der Nachricht

Zum Einfügen der Nachricht muß man nun entscheiden, in welchem Farbanteil das Nachrichtenbit versteckt werden soll. Bei einer Binärnachricht geht man folgendermaßen[4] vor.

Die Ausgangspixelfolge heißt in unserem Beispiel:

FF FF FF	07 07 07	F0 E0 A7	F0 E0 A7	E0 FF 0B	90 A0 00	00 00 00	90 10 A0

Für den Anfang soll das niederwertigste Bit des Blau-Anteils jedes Bildpunktes ein Nachrichtenbit tragen. Der Blau-Anteil entspricht folgender Tabelle:

Hex.	FF	07	A7	A7	0B	00	00	A0
Bin.	1111 1111	0000 0111	1010 0111	1010 0111	0000 1011	0000 0000	0000 0000	1010 0000

Als Nachricht wähle wir 01010010 (dez. 82), was dem ASCII-Code für den Buchstaben "R" entspricht.

0	1	0	1	0	0	1	0
1111 1110	0000 0111	1010 0110	1010 0111	0000 1010	0000 0000	0000 0001	1010 0000

Daraus ergibt sich die Farbkombination

FF FF FE	07 07 07	F0 E0 A6	F0 E0 A7	E0 FF 0A	90 A0 00	00 00 01	90 10 A0

Die Blauwerte des Bildes verändern sich damit allenfalls um eine Farbstufe. Unser Auge kann diese leichte Veränderung nicht bemerken.

Ergiebigkeit der Methode

„Eine kleine Überschlagsrechnung zeigt welche Datenmengen auf die beschriebene Weise in eine Bilddatei eingebracht werden können.

In einem Bild mit 320 mal 240 Pixeln haben so 76800 Bit Platz. Bei einer Zeichenlänge von 8 Bit können 9600 Buchstaben versteckt werden.*

**Allerdings ist die Bilddatei mindestens 320(Px) * 240(Px) * 3(Farben) = 230400 Byte groß.*

[4] in Anlehnung an: „Tarnkappe kontra Krypto-Verbot" Ralf Schneider, aus C't, 19.04.01

Selbstverständlich ist man nicht gezwungen jeden Bildpunkt zu verändern. Außerdem können andere oder gleichzeitig verschiedene Farben "geimpft" werden. Dem Einfallsreichtum sind hier keine Grenzen gesetzt. Im Blick behalten werden sollte lediglich dabei, dass die Vorgehensweise ja auch dem Empfänger zur "Entblätterung" der Nachricht mitgeteilt werden muss. Hier liegt, wie oben schon erwähnt, die eigentliche Schwachstelle der Steganographie. Ist dem Lauscher der Schlüssel einmal bekannt, ist der Aufwand zum Herausfiltern sehr gering, auch wenn er viele harmlose Nachrichten bearbeiten muss.“ [5]

Abbildung 6 Steganogramm
Quelle: Chip 01/2002 Seite 40

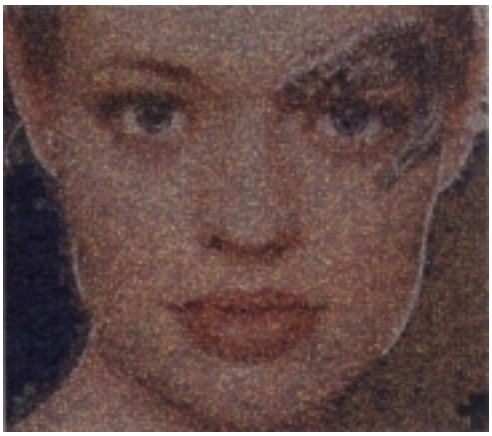


Abbildung 7 Differenz zum Original

Schema für Indexfarbenbilder

Einleitung

Die Indexfarbenbilder sind nur rund ein Drittel so groß wie True Color Bilder. Sie speichern nur die 256 Farben wichtigsten Farben in einer eigenen Palette.

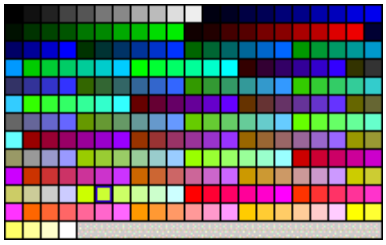


Abbildung 8 Mac-Farbtabelle

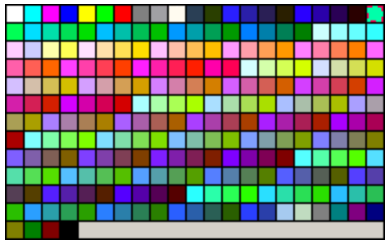


Abbildung 9 Windows-Farbtablle

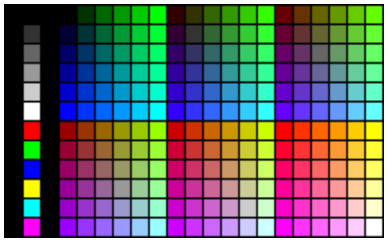


Abbildung 10 websafe-Tabelle | 216 Einträge zzgl. 40 Systemfarben

[5] in Anlehnung an: „Tarnkappe kontra Krypto-Verbot“ Ralf Schneider, aus C’t, 19.04.01

Jeder Bildpunkt verweist dann durch einen Index auf den entsprechenden Palleteneintrag. Somit muß nur noch ein Byte pro Bildpunkt gespeichert werden.[6]

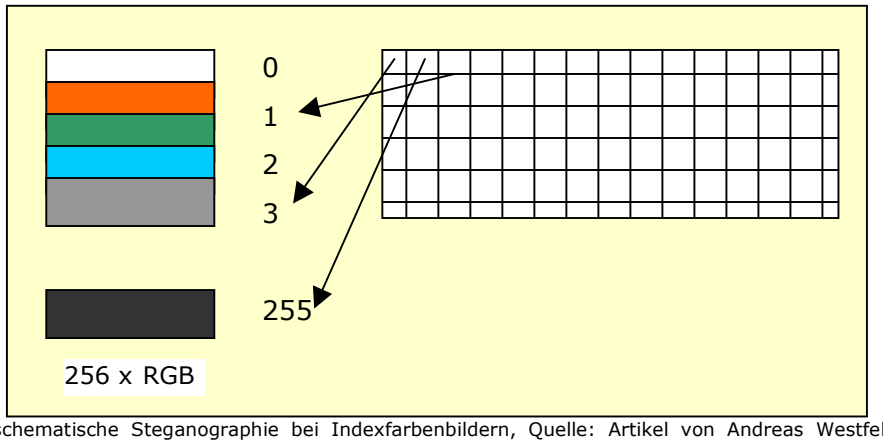


Abbildung 11 schematische Steganographie bei Indexfarbenbildern, Quelle: Artikel von Andreas Westfeld aus der C't 9/2001

Einfügen der Nachricht

Um Nachrichtenbits in Indexfarbenbilder einzufügen, werden, wie bereits oben beschrieben, die niederwertigsten Bits geändert.

Da die Farben beim GIF-Format, im Gegensatz zum BMP-Format, in einer Farbtabelle gespeichert und die Farbpunkte durch den Verweis auf die in der Tabelle gespeicherten Farbeinträge entstehen, ist es notwendig, die Farbtabelle nach Einfügen der Nachricht in die niederwertigsten Bits, neu zu sortieren.

Ändert man nur ein Bit der Adresse des Verweises, so wird dem Bildpunkt sofort eine andere Farbe zugewiesen.

Beispiel:

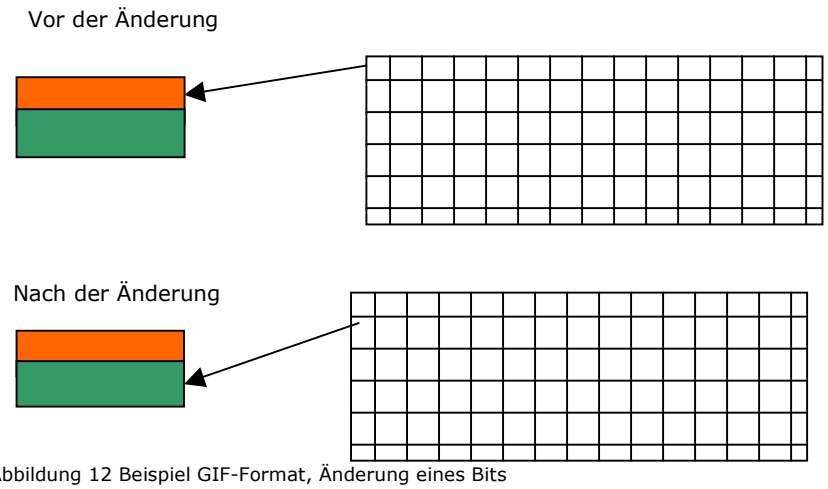


Abbildung 12 Beispiel GIF-Format, Änderung eines Bits

[6] Artikel „Unsichtbare Botschaften" von Andreas Westfeld Quelle: C't 9/2001

Im oberem Beispiel kann man erkennen, welche gravierende Abweichungen bei der Veränderung der Bits eines GIF-Bildes auftreten können. Diese Effekte werden durch das Neusortieren der Farbtabelle vermieden.

„Eine andere Möglichkeit, Daten in einem GIF-Bild zu verstecken, besteht darin nicht die vollen 256 Farben zur Darstellung des Bildes zu nutzen. Das heißt, es werden 128 Farben ausgesucht, die den Hauptfarben im Bild entsprechen. Die anderen 128 „Farbplätze" werden mit solchen Farben belegt, die den 128 „Hauptfarben" am ähnlichsten sind. Anschließend wird die Farbtabelle so belegt, dass bei einer Bitänderung der Zeiger für den Farbpunkt auf eine Farbe zeigen kann, die der Ausgangsfarbe am ähnlichsten ist. Hierzu existiert natürlich ein entsprechender Algorithmus, der dass Zuordnen der Farben vornimmt. Dieses Verfahren kann auch angewendet werden, wenn man mehr als 1 Bit verändert."[7] Dies ist allerdings nur begrenzt möglich, da durch die Reduktion der Farbe Abbildungsqualität verloren geht. Je weniger Farben zur Darstellung verwendet werden, um so offensichtlicher ist die entsprechende Veränderung.

Abbildung 13 Grafik mit 256 Farben
Quelle des Originals: www.artofpics.com

Abbildung 14 Grafik mit 16 Farben

Ergiebigkeit der Methode

Bei akzeptabler Qualität passt in der Regel in ein Indexfarbenbild Höhe x Breite [Bits], also ein Bit je Bildpunkt. Allerdings wird diese Regel stark von der Qualität (Größe, Auflösung) und der Anzahl der verwendeten Farben im Original beeinflusst.

[7] Artikel: Sag's durch die Blume, Marit Köhntopp, http://www.koehntopp.de/marit/publikationen/steganographie

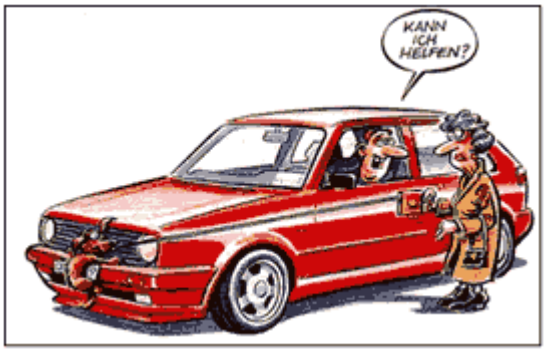


Abbildung 15 Grafik mit 16 Farben
Quelle des Originals: www.witze.de

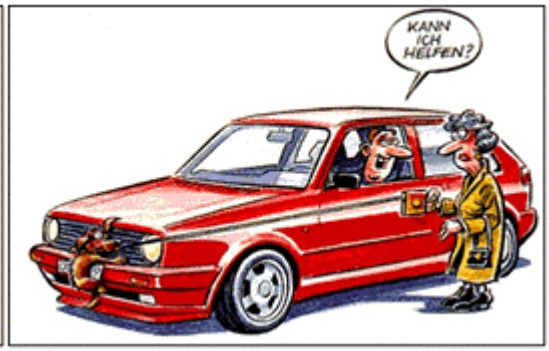


Abbildung 16 Grafik mit 256 Farben

Wenn also bereits das Original nur wenige Farben verwendet, stehen entsprechend viele Farben zur Einbettung einer Nachricht zur Verfügung, ohne dadurch sichtliche Veränderungen der Abbildung zu verursachen.
Leider hat das Verfahren für Indexfarbenbilder einen Nachteil. Die Veränderung (Impfung) kann sehr leicht erkannt werden, wenn die jeweilige Farbtabelle durchsucht wird.

Schema für nicht verlustfrei komprimierende Formate

Einleitung

Diese Bildformate erreichen sehr hohe Kompressionsraten, wobei das Bild geringfügig geändert wird. Es ist natürlich offensichtlich, dass diese Änderung die niederwertigsten Bits und somit auch die Nachrichtenbits beeinflussen, deshalb muß hierbei eine andere Technik verwendet werden.

Abbildung 17 einfaches jpg

Quelle des Originals:www.artofpics.com

Abbildung 18 jpg mit 50% Komprimierung

Abbildung 19 jpg mit 90% Komprimierung

Einfügen der Nachricht (Beispielhaft am JPEG-Format)

JPEG komprimiert Dateien in zwei Schritten, Kosinus-Transformation und Quantisierung.

Zuerst wird das zu bearbeitende Bild in Blöcke von 8x8 Bits unterteilt. Dann wird die Kosinus-Funktion über diese Pixel gelegt, um diese darzustellen. Nun werden die Frequenzkoeffizienten (Häufigkeiten) dieser Kosinus-Funktion zur Beschreibung der Pixelblöcke als Abfolge von Nullen und Einsen gespeichert (Quantisierung).
In dieser Frequenz können jetzt wiederum die niederwertigsten Bits genutzt werden, um Informationen zu speichern. Zum Abschluss erhält das Bild eine Farbtabelle mit sehr ähnlichen, benachbarten Farbeinträgen (siehe Exkurs: Huffmann-Kodierung).

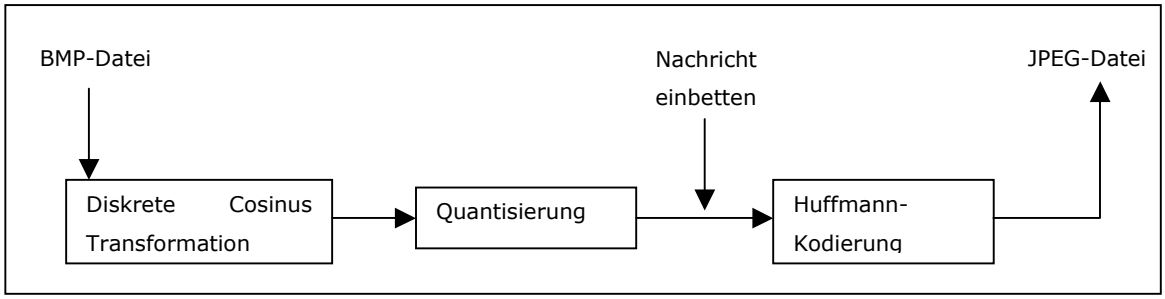


Abbildung 20 Entstehung eines Steganogramms bei einer JPEG-Datei, Quelle: Artikel von Andreas Westfeld aus der C't 9/2001

Eine andere Möglichkeit Informationen in JPEG's zu verstecken besteht darin, die Frequenzen einer diskreten Kosinus-Funktion, die einen 8x8 Pixel-Block repräsentiert, zu verändern. Der Unterschied zur Benutzung der einfachen Kosinus-Funktion ist der, dass die diskrete Kosinus-Funktion schon eine Annäherung/ Rundung beinhaltet, so dass Veränderungen der Annäherungen noch weniger auffallen.
Bei dieser sogenannten diskreten Kosinustransformation (DCT) vom Pixelbereich in einen Frequenzraum, entstehen genau so viele Frequenzkoeffizienten wie Bildpunkte vorhanden sind.
Da bei dieser „Rundung" die meisten Frequenzen mit einem verschwindend kleinen Anteil auftreten, entsteht am häufigsten der Wert 0. Die Koeffizienten lassen sich wegen der vielen Nullen verlustfrei und besonders platzsparend abspeichern. An dieser Stelle greifen Steganographie-Anwendungen ein, indem sie nach der Quantisierung die niederwertigsten Bits dieser Koeffizienten überschreiben.[8]

[8] Artikel „Unsichtbare Botschaften" von Andreas Westfeld Quelle: C't 9/2001

Bit in den Ton

Rauschen

Digitalisierte Töne haben einen bedeutsamen Anteil an Rauschen aus ihrer ursprünglichen Herstellung (Digitalisierung). Erstens stammt dieses Rauschen aus den analogen Daten (Hintergrundgeräusche). Zweitens sind die Schaltkreise/Programme die Stimmen und Töne zu Bits konvertieren nicht ganz perfekt. Drittens gehen durch die zeit- und wertdiskrete Aufnahme bei der Digitalisierung Ausprägungen der Töne verloren. Etwas elektrisches Rauschen verändert die Bits ein bisschen und es gibt keine Möglichkeit dies zu verhindern.
Dieses Rauschen ist auch eine Chance für das Verstecken von Informationen. Digitalisierte Töne werden durch eine Matrix aus Zahlen dargestellt, die für die Intensität von Schallwellen stehen, so wie es zu einer bestimmten Zeit an einem bestimmten Ort zu hören war. Digitale Töne sind nur Aufzählungen des Drucks der auf ein Mikrofon in bestimmten Zeitscheiben ausgeübt wurde.
Eben jene Bits die das Rauschen abbilden, können steganographisch genutzt werden.

Auch bei Audio-Dateien gibt es verlustbehaftet komprimierte und unkomprimierte oder verlustfrei komprimierte Datei-Formate.

WAV-Dateien

Audiodateien eignen sich durch das schon vorhandene Rauschen hervorragend für steganographische Zwecke. Die Methode der LSB's[9] läßt sich auch hier sehr gut anwenden. Das gilt vor allem für das WAV-Format, welches nicht nach dem Prinzip der Datenreduktion (vgl. MP3) arbeitet. Dabei können diese Dateien etwa 1/8 oder 1/16 ihrer Größe an steganographischen Daten aufnehmen können.
Zusätzliche Methoden des *information hiding* sind hier:

- Eine Passage wird ausgeschnitten und durch akustisch gleichwertige ersetzt.
- Eine Passage wird mehrfach dupliziert.
- Ein künstliches Echo wird eingefügt, welches in Wirklichkeit (z.B. durch die Anzahl der Wiederholungen) Informationen enthält

[9] LSB...least significant bits, zu deutsch: Veränderung der niederwertigsten Bits

- Es werden Verfahren angewendet, die aus der Erforschung der menschlichen Psychoakustik hervorgehen:
  - Lautere Töne überschatten (bei nicht allzu großer Differenz der Frequenzen) leisere Töne
  - Kurz nach einer lauten Passage bzw. Tons ist das Gehör für wenige Millisekunden "taub".

MP3-Dateien

Das MP3 (MPEG3) Verfahren benutzt das Huffmann-Verfahren um Audioinformationen nicht nur zu komprimieren, sondern auch zu reduzieren. Dies geschieht dadurch, dass der komplette Soundtrack als ganzes untersucht wird: Sind Passagen gleichen oder annähernd gleichen Inhalts, werden sie nur einmal ein einer Tabelle abgespeichert und danach bei jedem Auftreten nur noch referenziert. Ein Paar der möglichen Verfahren der Implementierung von Steganographie-Daten wären hier:

- Anlegen von akustischen "dummies" in den Tabellen - das Verstecken von Daten erfolgt dann durch Referenzieren auf "lautlose" Passagen mit sehr kurzer Länge
- Ersetzen von psychoakustisch unrelevanten Sequenzen durch Steganographie -Sequenzen
- Anlegen von Sequenzen in den Tabellen, die jedoch nicht gespielt werden
- Einarbeitung von Information ("Hintergrundrauschen") in die Sequenztabellen[10].

Steganographie bei Texten

Auch in Texte können Informationen eingebettet werden. Hierbei kann nicht, wie in den vorangegangenen Beispielen, die Methode der niederwertigsten Bits benutzt werden.

Es können zum Beispiel Eigenschaften eines vorhandenen Textes bzw. eines extra für das Steganogramm erdachten Textes benutzt werden. Hierbei gibt es viele Möglichkeiten. Um Sie erkennen zu können, benötigt man nicht unbedingt komplizierte, von einem Computer umgesetzte Algorithmen. Informationen können sehr einfach versteckt werden.

[10] Pacher, R. Vortrag Steganographie, TU München, http://home.in.tum.de/~pacher/stego.html#Audiodateien

Ein Beispiel:

```
Liebe Kolleginnen! Wir genießen nun endlich unsere Ferien auf
dieser Insel vor Spanien. Wetter gut, Unterkunft auch, ebenso
das Essen. Toll! Gruß, M. K.
```

[11]

Dieser Text auf einer Postkarte klingt ziemlich harmlos und vermittelt ein positives Gefühl. Nun zählt man die Anzahl der Zeichen bis zum nächsten Leerzeichen. Ergibt sich eine ungerade Summe, setzt man eine 0, für eine gerade Anzahl dementsprechend eine 1. Hieraus ergeben sich im Binärcode verschiedene Zahlenkombinationen. Die ersten acht Wörter ergeben: 01010011, die nächsten acht: 01001111 und die letzten acht: 01010011. Im Dezimalsystem stehen diese Null/Einsen-Kombinationen für die Zahlen 83,79,83. In der ASCII-Code-Tabelle erhält man für die Zahl 83 ein S und für 79 ein O.

Voila, unsere Nachricht heißt also **SOS**.

Informationen können nicht nur in den Buchstaben eines Textes versteckt werden. Steganographie-Anwendungen für Text ersetzen auch Wörter durch Synonyme, verändern den Zeilenumbruch oder fügen Leerzeichen und Tabulatoren am Zeilenende ein.[12]

[11] Artikel: Sag's durch die Blume, Marit Köhntopp, http://www.koehntopp.de/marit/publikationen/steganographie/

[12] Andreas Westfeld „Unsichtbare Botschaften", Artikel C't 9/2001

4. Exkurs

4.1. Huffmann-Kodierung

„Das Huffman-Verfahren geht von der Annahme aus, daß eine Datei nicht alle 255 möglichen ASCII-Zeichen enthält, also daher auch nicht unbedingt acht Bit nötig sind um diese darzustellen. Falls doch, so ist meist die Häufigkeit der einzelnen Zeichen recht unterschiedlich verteilt, so daß man Platz sparen kann, wenn man für die häufiger vorkommenden Zeichen kürzere Bitmuster verwendet als für die "Raritäten". Also verteilt man die Zeichen in einem sogenannten Binärbaum, die häufig vorkommenden nach oben. An der Spitze beginnend hangelt man sich über die Knotenpunkte rechts-links-eins-null nach unten, bis man das Gesuchte gefunden hat. Das auf dem Weg entstandene Bitmuster ersetzt dieses Zeichen."[13]

4.1.1. Eigenschaften

Nachfolgende Eigenschaften zeichnen die Huffmann-Kodierung aus:

- „verlustfrei, d.h. dekomprimierte Datei entspricht 100%ig den Originaldaten, daher geeignet für Backups, Dateikompression allgemein (arj, zip, rar ...)
- Idee des Morsealphabets, häufig vorkommende Zeichen erhalten einen kurzen Code, seltene eine längeren
- Zusammenfassung der Codes und der entsprechenden Zeichen in einer Tabelle, dadurch Wiederherstellung der ursprünglichen Information möglich"[14]

Die bereits 1952 von David Huffmann beschriebene Methode ist immer noch die effektivste Kodierungs-Möglichkeit.

[13] Datenkompression unter M – Die Methode nach David Huffmann von Burkhard Kasten, BEWIDATA, Mainz | http://www.mug-d.de/mboerse/1997_1/3.htm

[14] Datenkompression - Fakten im Überblick | http://www.htw-dresden.de/~htw8962/belege/kompression.htm

4.1.2. Verfahren[15]

Schritt 1: Ermittlung der Häufigkeit des Auftretens jedes Zeichens der zu kodierenden Zeichenfolge (z.B.: 11 Leerzeichen, 3 A, 3 B usw.)

Schritt 2: Ermittlung der Häufigkeitstabelle

Schritt 3: Aufbau eines den Häufigkeiten entsprechenden Kodierungsbaumes

3a: Erzeugung eines Knotens für jede von 0 verschiedene Häufigkeit

3b: Auswahl zweier Knoten mit der kleinsten Häufigkeit

3c: Erzeugung eines Vorgängers zu den beiden Knoten, dessen Häufigkeit der Summe ihrer Werte entspricht

3d: Wiederholung 3b & c mit den nächsten kleinsten Häufigkeiten

3e: Wenn ein Summenknoten kleiner oder gleich der nächst kleinsten Häufigkeit ist, wird der Vorgänger aus der kleinsten Häufigkeit und dem Summenknoten (Teilbaum) gebildet

3f: die Punkte 3b – 3e werden wiederholt, bis ein vollständig zusammenhängender Baum entstanden ist.

Schritt 4: Ersetzen der Knotenwerte (Häufigkeiten) durch die entsprechenden Buchstaben (dabei entspricht 0=links und 1=rechts)

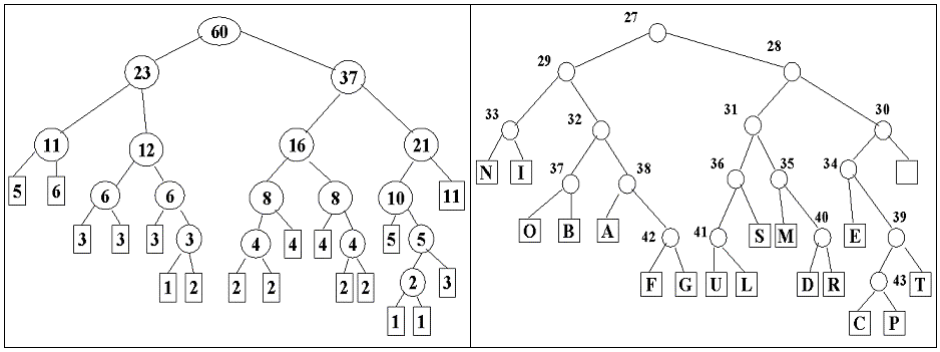


Abbildung 21 Skizze eines huffmann-kodierten Baumes nach: Algorithmen und Datenstrukturen, G.Heyer | http://www.informatik.uni-leipzig.de/ifi/lehre/Heyer2000/ADKapitel10

Schritt 5: Ableitung des Huffmann-Codes (Beispiel siehe Abb.: N=000, C=110100)

[15] Verfahren nach Algorithmen und Datenstrukturen, G.Heyer | http://www.informatik.uni-leipzig.de/ifi/lehre/Heyer2000/ADKapitel10/sld018.htm

5. Vorstellung von Steganographie Software

5.1. Steganos Security Suite 4

Diese Software wird von der Steganos GmbH, Eckenheimer Landstraße 17, 60318 Frankfurt am Main hergestellt und die Vollversion ist für Euro 49,95 erhältlich. Wer über das Internet bestellt und auf Box und Handbuch verzichten kann, zahlt für den Freischaltcode nur Euro 39,95. Eine 30 Tage- Testversion kann auf der Firmenhomepage www.steganos.com gedownloadet werden.

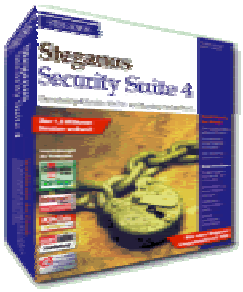


Abbildung 22 Steganos Packung, Quelle: www.steganos.com

Wir haben uns für die Vorstellung dieses Produktes entschieden, da es in diversen Test-Berichten verschiedener Fachzeitschriften als sehr zuverlässig und schwer zu entschlüsseln gelobt wird. Ein weiteres Kriterium war die einfache Bedienung und die Zusatzkomponenten.

Abbildung 23 Testsiege der Software, Quelle:www.steganos.com

Anfänglich war Steganos ein reines Steganographie-Tool. Heutzutage findet man im Lieferumfang verschiedene Anwendungen wie z.B.:

- Blitzschneller Schutz eines ganzen Laufwerks – Verschlüsselung von 1 GB Daten in weniger als einer Sekunde!
- Mit Automatic Crash Protection - damit auch bei einem Systemabsturz keine Daten unverschlüsselt bleiben
- Beseitigung von Internet-Spuren
- Leicht verständliche E-Mail-Verschlüsselung
- verschlüsselte E-Mail-Attachments
- Verstecken und Verschlüsseln von Dokumenten
- Endgültiges Vernichten von Dateien
- Sperren des Computers

Die Steganos Security Suite 4 bietet eine übersichtliche grafische Oberfläche. Die Benutzerführung erfolgt intuitiv und so ist es sehr leicht, sich in das Produkt einzuarbeiten.
Von den technischen Hintergründen bekommt man als Anwender nichts mit. Somit ist es für den Einsatz in verschiedensten Bereichen geeignet.

5.1.1. Technische Daten

Die Steganos Security Suite ist für PCs mit Windows 95, 98, Me, NT 4.0, Windows 2000 oder XP geeignet. Sie benötigt etwa 9,4 MB Platz auf der Festplatte. Weitere Systemvorausetzungen sind: Pentium-Prozessor oder vergleichbare CPU, 32 MB RAM (oder mehr), Bildschirmauflösung 640x480 oder höher bei einer Farbtiefe von 256 Farben (oder höher), normale Schriftgröße.
Die Steganos Security Suite verwendet verschiedene Algorithmen zur Verschlüsselung und Steganographie der Daten.

Kryptographische Anwendung

Zur Verschlüsselung werden mehrere Algorithmen eingesetzt.

Advanced Encryption Standard (AES)

„Der AES-Algorithmus wurde im Oktober 2000 vom US-amerikanischen *National Institute of Standards and Technology* (NIST) zum Nachfolger des von IBM entwickelten *Data Encryption Standard* (DES) ernannt. Der DES gilt heute als veraltet – er war fast 30 Jahre lang der Standard für verschlüsselte Informationen."[16] Der AES beruht auf dem Krypto-Algorithmus "Rijndael" der belgischen Kryptologen Dr. Joan Daemen und Dr. Vincent Rijmen. Er gilt als absolut sicher. „AES ist wie DES ein symmetrisches Verschlüsselungsverfahren und unterstützt 128-, 192- und 256-Bit-Schlüssel. Damit werden laut NIST Brute-Force-Angriffe, die bei den 56-Bit-Schlüsseln von DES mit spezieller Hardware schon nach wenigen Stunden zum Ziel führten, auf Jahre hinaus unmöglich sein. Der Algorithmus ist nicht patentiert und darf von jedermann kostenlos eingesetzt werden.[17]" Er wird im Safe, bei der Datei-Verschlüsselung und im Passwort-Manager verwendet.

Blowfish

„Blowfish ist genau wie DES eine Blockchiffrierung und wurde von Bruce Schneier, *Counterpane Internet Security,* entwickelt. Die Blockgröße beträgt 64-Bit. Jedoch ist die Schlüssellänge variabel und kann bis zu 448 Bit betragen. Blowfish wurde speziell für solche Anwendungen entwickelt, bei denen

[16] http://www.steganos.com/de/s3/inside.htm

[17] Quelle: (ju/c't), http://www.heise.de/newsticker/data/ju-05.12.01-001/

sich der Schlüssel selten ändert (zum Beispiel automatische Verschlüsselung von Dateien). Weiterhin ist Blowfish auf 32-Bit-Prozessoren wesentlich schneller als DES.[18]" Auch Blowfish gilt als absolut sicher. Der Blowfish-Algorithmus kommt beim Verstecken von Daten in Dateien und bei der E-Mail-Verschlüsselung zum Einsatz.

Kryptografische Hashvalues

„Zur Generierung von Hashvalues (kryptografischen Prüfsummen) kommt der SHA-1-Algorithmus zum Einsatz. Jede Modifikation, ob mutwillig oder nicht, wird erkannt.[19]" „Der Secure Hash Algorithm 1 (SHA-1) erzeugt die 160-bit lange Textprüfsumme (Hashwert, Message Digest).[20]"

Secure Hash Standards: http://www.itl.nist.gov/fipspubs/fip180-1.htm

Steganographische Anwendung

„Durch die sogenannte Matrixkodierung müssen besonders wenige Bits in den Trägerdateien verändert werden. Dadurch sind die versteckten Informationen besonders unauffällig. Steganografie wird in der Steganos Security Suite immer zusammen mit Kryptografie verwendet.[21]"

Schlüsselvereinbarung

Der unsichtbare Schlüsselaustausch InKA (Invisible Key Agreement) erfolgt unter Verwendung des Diffie-Hellman-Algorithmus bei 2048-Bit.

„Whitfield Diffie und Martin E. Hellman haben 1976 ein Key-Austausch-Protokoll entwickelt, das zum sicheren Austausch von Schlüsseln dient und dessen Bestandteil der Einsatz eines Schlüsselpaares aus Public- und Private-Key ist. Es begründete das asymmetrische Verschlüsselungsverfahren mit Public-Key Kryptografie.[22]"

[18] http://home.t-online.de/home/jerry.luitwieler/krypto4.htm

[19] http://www.steganos.com

[20] http://home.arcor.de/kraven/pgp/pgpindex.html

[21] http://www.steganos.com/de/s3/inside.htm

[22] http://home.arcor.de/kraven/pgp/pgpindex.html

Datenvernichtung

Entspricht der Norm des US-Militärs DOD 5220.22-M/NISPOM 8-306 und geht darüber hinaus. Nicht nur Dateiinhalt, sondern auch Dateiname, Grösse, Datum und Attribute werden vernichtet."[23]

5.1.2. Steganographie mit Steganos

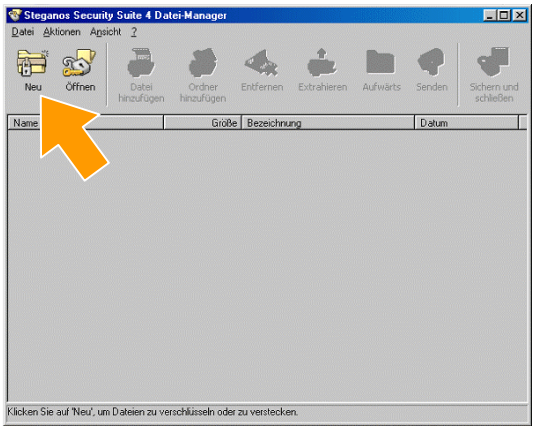
Im folgenden möchten wir erklären, wie eine Datei mit der Steganos Security Suite 4 kodiert wird:

Man beginnt mit dem Steganos Datei-Manager.

Abbildung 24 Startmenü der Steganos Security Suite 4

Als nächstes erscheint der Bedienerbildschirm zur Erstellung des Steganogramms:

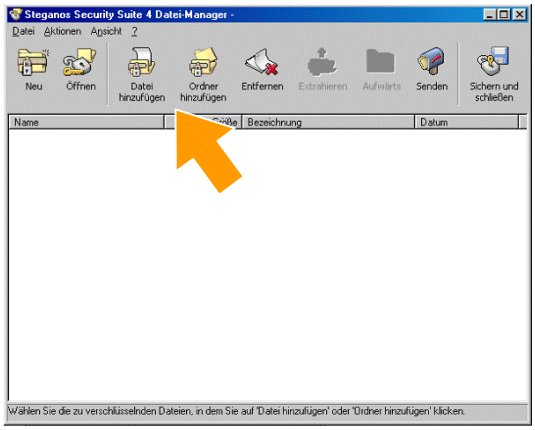
[23] Verwendete Algorithmen, Steganos Security Suite Hilfe



Um ein neues Steganogramm zu erstellen klickt man auf NEU.

Der Bildschirm verändert sich und es sind nicht mehr alle Buttons ausgegraut.

Abbildung 25 Steganos- Dateimanager



Nun kann man das zu versteckende Dokument/ den zu versteckenden Ordner über den Button DATEI/ ORDNER HINZUFÜGEN auf den Bildschirm und in die Anwendung holen.

Abbildung 26 Steganos- Dateimanager II

In unserem Fall haben wir ein Word-Dokument ausgewählt. Möchte man diese Datei nun verschlüsseln, klickt man auf den Button SENDEN.

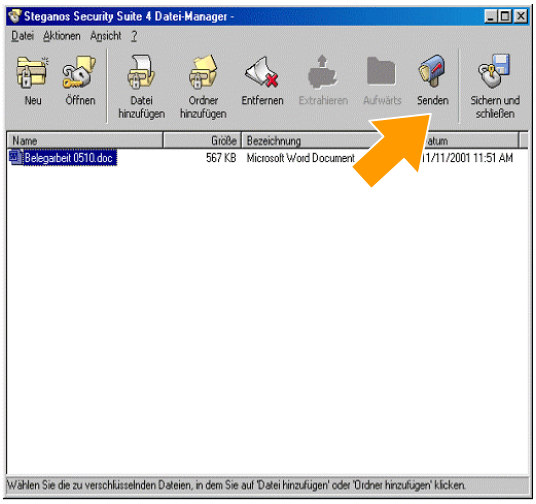


Abbildung 27 Steganos- Dateimanager mit Datei

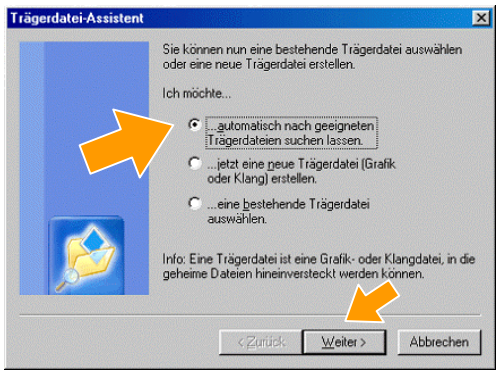
Man wird aufgefordert, die Daten zuerst zu sichern.

An dieser Stelle muß man sich nun entscheiden, ob die Daten nur verschlüsselt oder verschlüsselt und versteckt werden sollen.

Abbildung 28 Steganos- Verschlüsselungsassistent

Wir entscheiden uns für verschlüsseln und verstecken um die Steganographie-Anwendung zu benutzen. Man folgt dem Assistenten durch einen Klick auf WEITER.

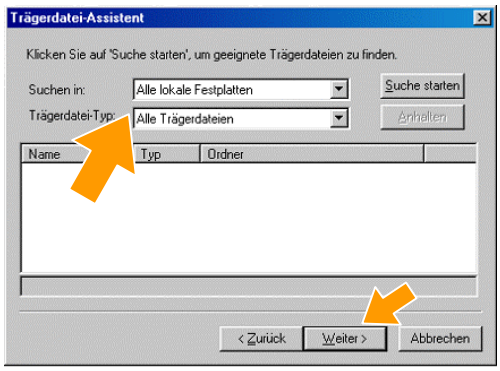
Man befindet sich nun im Trägerdatei-Assistenten. Er führt in 3 Schritten durch die Trägerdateisuche.



Schritt 1:

Das Programm benötigt ein geeignetes Trägermaterial. Dieses kann manuell zugewiesen, direkt erstellt oder gesucht werden. Zur Demonstration entscheiden wir uns für die Suchfunktion.

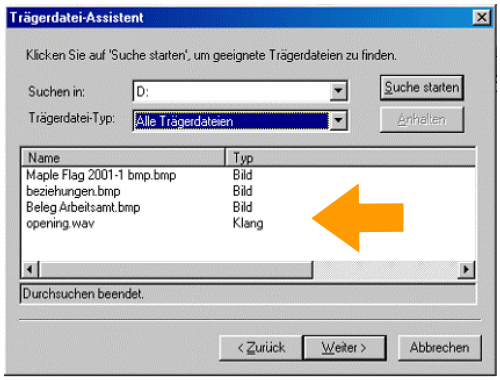
Abbildung 29 Trägerdatei-Assistent Schritt 1



Schritt 2:

In der Maske können die zu durchsuchenden Medien angegeben werden. Außerdem kann hier festgelegt werden, ob eine Ton- oder/und Bild-Datei gesucht werden soll.

Abbildung 30 Trägerdatei-Assistent Schritt 2



Schritt 3:

Wie man an den Sucheinstellungen sehen kann, sucht Steganos hauptsächlich wav- und bmp-Dateien. Mit der Entscheidung für eine Trägerdatei beendet man den Trägerdatei-Assistenten.

Abbildung 31 Trägerdatei-Assistent Schritt 3

Nachdem die Trägerdatei ausgewählt und bestätigt wurde, muss ein Passwort eingegeben werden.

Abbildung 32 Passworteingabe

Im Bildschirmfenster wird die Qualität des gewählten Passwortes angezeigt. Mit Bestätigung des Passwortes wird die Datei nun als Steganogramm gespeichert. Dieses Passwort muß dem Empfänger der Daten mitgeteilt werden, damit dieser die Informationen wieder herstellen kann.

Das entstandene Steganogramm kann nun als z.B. Email verschickt werden.

Zur Wiederherstellung der Informationen startet man den Datei-Manager der Steganos Security Suite und betätigt den Button DATEI ÖFFNEN und wählt das entsprechende Steganogramm durch anklicken aus. Jetzt muss das vereinbarte Passwort eingegeben werden.

Abbildung 33 Passworteingabe

Nach der korrekten Eingabe wird das Steganogramm „entpackt" und man erhält die herausgefilterte Word-Datei.

Abbildung 34 „entpackte" Datei

Diese Datei kann der Empfänger nun lesen, abspeichern oder wieder als Steganogramm verschlüsseln und weitersenden.

5.2. TextHide

Wir haben als zweites Produkt ein Text-Tool ausgewählt, um auch diese Funktionalitäten einmal vorzustellen.

Mit TextHide lassen sich beliebige Daten in Text verbergen. Es ist ein Produkt der Compris.com GmbH Strother Str. 13, D-34477 Twistetal-Berndorf. Die 12 Monate lauffähige CD-ROM Fassung von TextHide kostete 179,- DM (III. Quartal 2000). Das Update für weitere 12 Monate gibt es für 99,- DM. Außerdem gibt es mit SubiText ein kostenloses Dekodierprogramm.

Auf der Internetseite www.TextHide.de kann eine Shareware-Version heruntergeladen werden. Diese Demo-Version enthält nur ein stark reduziertes Lexikon und keine Funktionalität zur Satzumstellung. In der Vollversion erhält man ein korrektes, aufwendig manuell nachkorrigiertes Lexikon.

5.2.1. Technische Daten

„TextHide gibt es für die Betriebssysteme: MS-Windows 95/98/NT, Unix (Linux, Solaris). Es ist in den Sprachen Deutsch, Englisch und Französisch verfügbar. Der Speicherbedarf beträgt ca. 16 MB für das Wörterbuch. Bei Verwendung eines Pentium II – Prozessors mit 400 MHz verarbeitet TextHide ca . 100 kB Text pro Minute. Dabei beträgt das Verhältnis von Daten zu Text - 1:10 bis 1:20 (schlechter bei Fachwörtern).

Das Programm ist mit Verschlüsselungsprogrammen wie PGP kombinierbar. SubiText hat eine eigene Public-Key-Kryptographie integriert (basiert auf RSA [4096 Bit Schlüssellänge] und dem neuem Twofish-Verfahren [256 Bit Schlüssellänge]).[24]"

TextHide verwendet Texte als Trägermedium, die durch den Gebrauch von Synonymen steganographisch verändert werden.

Die Daten werden durch automatisches Umformulieren mit TextHide/SubiText im Text verborgen. Der Sinn bleibt dabei vollständig erhalten. Die Information kann vorher auch verschlüsselt werden.

[24] Herstellerangaben, http://www.compris.com/TextHide/de

Beispiel der Funktionsweise beim Verbergen von Informationen:

„Es sei der geheime Text "Treffen um 9 Uhr bei mir" zu verbergen. Ein nichtgeheimer Text - etwa aus der mitgelieferten Sammlung von Texten - ist "das Auto fährt schnell bei glatter Straße über den Hügel". Der geheime Text steuert die Umformulierung dieses Textes und liefert etwa: "Über die Anhöhe rast der Pkw blitzschnell auf eisglatter Fahrbahn." Der Sinn des Satzes ist erhalten geblieben. Dadurch, welches Synonym oder welche Wortstellung von sehr vielen möglichen gewählt wird, ist die geheime Information gespeichert.[25]"

Codierung

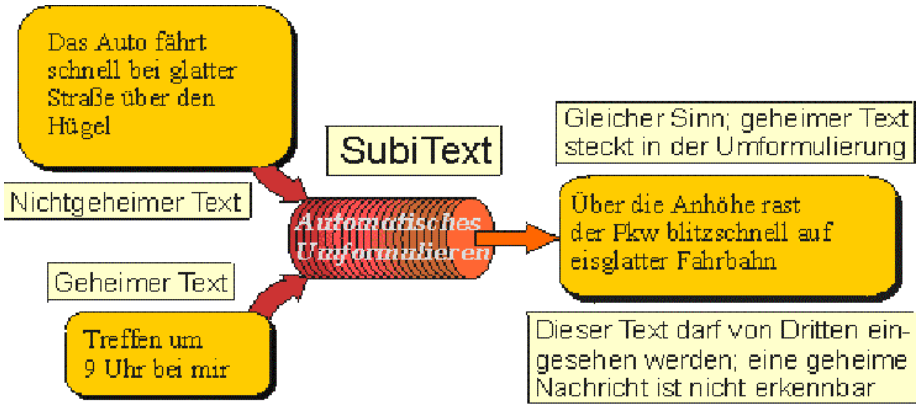


Abbildung 35 Codierung Quelle: Produktinfo compris.com GmbH, www.compris.com

Decodierung

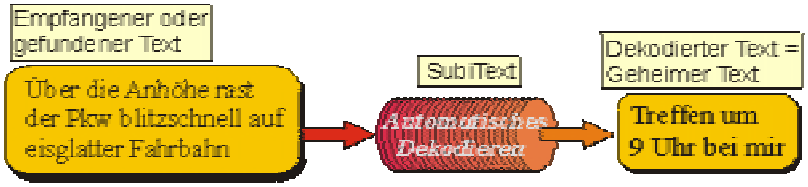


Abbildung 36 Decodierung Quelle: Produktinfo compris.com GmbH, www.compris.com

Umformulieren

„TextHide kann automatisch mehrere Texteigenschaften, wie z.B. Erzählzeit und Erzählperspektive, ändern und einzelne Wörter durch Synonyme ersetzen. Besonders interessant ist dies für:

- Journalisten, Pressereferenten, Schüler, Studenten, Autoren
- alle, die einen Text in verschiedenen Versionen brauchen oder einen schon existierenden Text für ihre Zwecke anpassen wollen[26]"

[25] Produktinfo compris.com GmbH, www.compris.com

[26] http://www.compris.com/TextHide/de/

Verbergen

„Durch gezieltes Umformulieren können Daten (Text, binäre Daten, ...) in einem Text verborgen werden. Es kann zwar jeder aus dem Text etwas dekodieren, aber wenn die Daten vorher verschlüsselt wurden, ist nicht zu erkennen, ob es sich um sinnvolle Daten handelt oder nicht.

Alle, denen einfaches Verschlüsseln der Daten nicht ausreicht, haben jetzt eine Möglichkeit, ihre Daten/Nachrichten echt zu verbergen. Es ist nicht mehr auf den ersten Blick erkennbar, ob verschlüsselte Daten vorliegen; somit kann kein gezielter Entschlüsselungsangriff auf die Daten erfolgen.[27]"

Die offensichtliche Verwendung verschlüsselter Daten ermöglicht es Lauschern, das Verschlüsselungsverfahren zu ermitteln, um dann gezielt dieses Verfahren anzugreifen. TextHide ermöglicht es, wichtige Informationen unverschlüsselt aussehen zu lassen.

Mit TextHide können aus jedem Text „geheime" Daten herausgelesen werden, unabhängig davon, ob zuvor sinnvolle Daten verborgen wurden. Durch gleichzeitige Verwendung von Verschlüsselung und dem Verbergen der Daten im Text wird noch zusätzliche Sicherheit erreicht.

Kryptographische Anwendung

„RSA- Verschlüsselungsverfahrens

TextHide verwendet des RSA-Verschlüsselungsverfahrens mit öffentlichem und geheimem Schlüssel. Dies gehört zu den sichersten Systemen überhaupt und wird auch im berühmten PGP eingesetzt.

Twofish

Ein weiteres Verschlüsselungsverfahren ist Twofish. Dies ist ein Kandidat für den amerikanischen AES-Standard (Advanced Encryption Standard), d.h. zur DES-Nachfolge (DES – Data Encryption Standard). Es wird als Blockchiffrierer genutzt.

Eine Eigenschaft dieses Verfahrens sind lange Schlüssel von ca. 50 - 100 kB zum Umstellen des Synonymwörterbuchs. Ein Knacken dieses Schrittes durch systematisches Testen mit 100 000 Großrechnern würde noch ca. 10^{2000} mal länger dauern als die Erde bisher existiert. [28]"

[27] http://www.compris.com/TextHide/de/

[28] Produktinfo compris.com GmbH, www.compris.com

Steganographische Anwendung

Das Verstecken der Nachricht erfolgt durch das patentierte Verfahren zum Umformulieren von Text. Dieses Modul heißt SubiText. Es ist sehr unauffällig und ein sehr breit anwendbares Verschleierungsverfahren zum Verbergen von Informationen. Die Anwendungsgebiete erstrecken sich von E-Mails und Netzwerkdatenkommunikation über Zeitungsanzeigen und Webseiten bis hin zu normalen Gesprächen.

- Mehrere Umformulierungsaspekte wie Satzstellung und Synonymersetzung können so ausgenutzt werden, dass geheime Informationen auch bei Umformulierungen durch Angreifer noch erhalten bleiben
- Textsammlung enthalten aus den Bereichen: Urlaub, Politik, Wirtschaft, Witze, Anekdoten, Glossen, Nachrichten/Zeitungsartikel (diese Texte können direkt als banale Texte ausgewählt und verwendet werden)

5.2.2. Steganographie mit TextHide

Die nachfolgende Bildschirmmaske stellt das Register „Verbergen", der TextHide-Anwendung dar. Über den ÖFFNEN -Button wird der zu verschlüsselnde Text über eine Datei im TXT-Format in das obere Feld eingefügt.

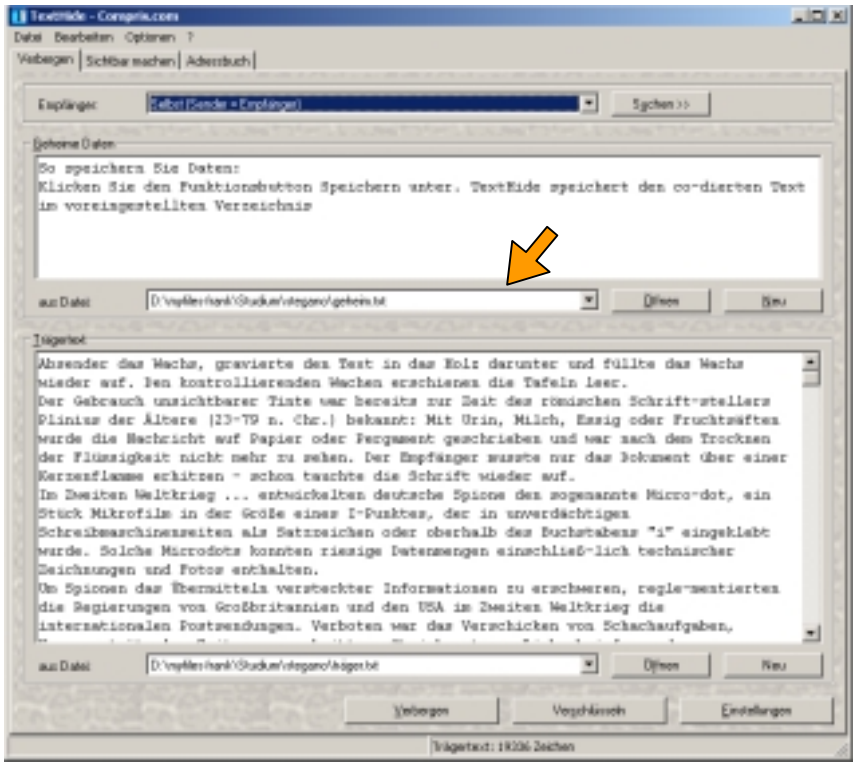


Abbildung 37 TextHide, Verbergen-Register

Im unteren Feld wird die Trägerdatei dargestellt. Auch diese sollte im TXT-Format vorliegen. Zum Verbergen der geheimen Daten muss nun über das Adressbuch (Adressbuch-Register) ein entsprechender Empfänger ausgewählt werden. Dieser besitzt einen öffentlichen Schlüssel, da er sonst die Daten nicht decodieren kann. Wird jetzt der Button Verbergen betätigt, verschlüsselt TextHide den Text und schaltet in die nächste Ansicht, den Sichtbar-Machen-Register.

In dieser Ansicht kann der codierte Text wieder über den ÖFFNEN-Button eingelesen werden. Die Veränderung des Textes erkennt man an den hinzugefügten Absätzen, die sich durch das gesamte neue Dokument ziehen.

Zum Wiederherstellen der Informationen kann nun der „Sichtbar machen" Button betätigt werden und der Originaltext erscheint im unteren Teil des Bildschirms im Feld decodierte Daten.

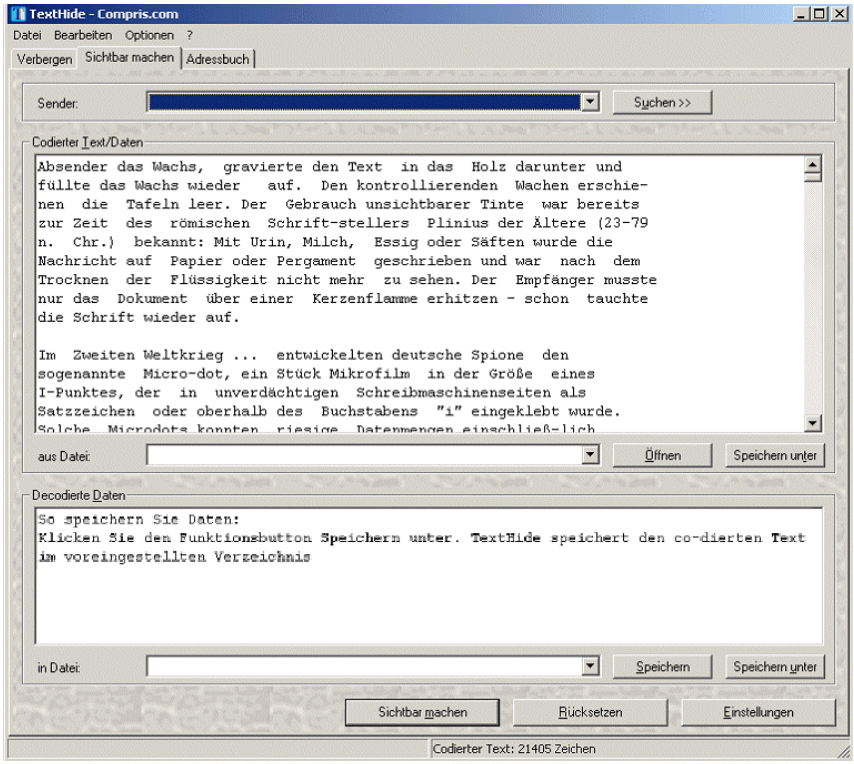


Abbildung 38 TextHide, „sichtbar machen"- Register

Alle benutzten und erstellten Texte können im TXT-Format abgespeichert werden. Diese Daten kann man dann weitersenden.

6. Erklärung

Die Unterzeichner erklären hiermit, vorliegende Belegarbeit: „Steganographie" eigenhändig und selbständig erstellt zu haben. Zitate und Quellen wurden entsprechend gekennzeichnet.

Die Hausarbeit ist eine Gemeinschaftsleistung. Auf Grund diverser Überarbeitungen und Ergänzungen ist keine Kennzeichnung der jeweils erbrachten Einzelleistungen möglich.

Datum, Unterschrift
Ulrike Nehls, Matrikelnummer:

Datum, Unterschrift
Frank Gleichmann, Matrikelnummer:

7. Abbildungsverzeichnis