bmpPacker

Benutzer Handbuch

Autor : Jens Gödeke
Programm Version : 1.2

Email:
Security@goedeke.net

Homepage :
http://www.goedeke.net/bmppacker.html

Inhalt :

1 Haftungsausschusserklärung:

**bmpPacker benutzt extrem harte Kompressionsalgorithmen.
Auch wenn das Programm in liberalen europäischen Ländern erstellt, verwaltet und verteilt wird, so kann es in anderen Teilen der Welt diversen Import/Export Beschränkungen unterliegen.**

Bitte beachten Sie, dass die Verwendung von harten Verschlüsselungsalgorithmen in einigen Teilen der Welt illegal ist.

Wenn Sie also bmpPacker in Ihr Land einführen, es dort benutzen oder vertreiben, sind Sie angehalten sich über Ihre Export/Import und/oder Benutzungs-Bestimmungen zu informieren.

**Ferner benutzt bmpPacker durch Verstecken von beliebigen Dateien in BMP-Kontainern eine Art „Hintertür“.
Bei nicht privater Nutzung von bmpPacker sollten Sie auf jeden Fall die Erlaubnis Ihres Systemadministrators einholen, bevor Sie das Programm benutzen.**

Der Autor von bmpPacker ist nicht verantwortlich für irgendwelche illegalen Aktionen die durch den Import/Export oder durch die Nutzung von bmpPacker entstehen.

##

**bmpPacker uses strong cryptography,
so even if it is created, maintained and distributed from liberal countries in Europe (where it is legal to do this), it falls under certain export/import and/or use restrictions in some other parts of the world.**

Please remember that export/import and/or use of strong cryptography software is illegal in some parts of the world.

So when you import bmpPacker to your country or re-distribute it from there you are strongly advised to pay close attention to any export/import and/or use laws that apply to you.

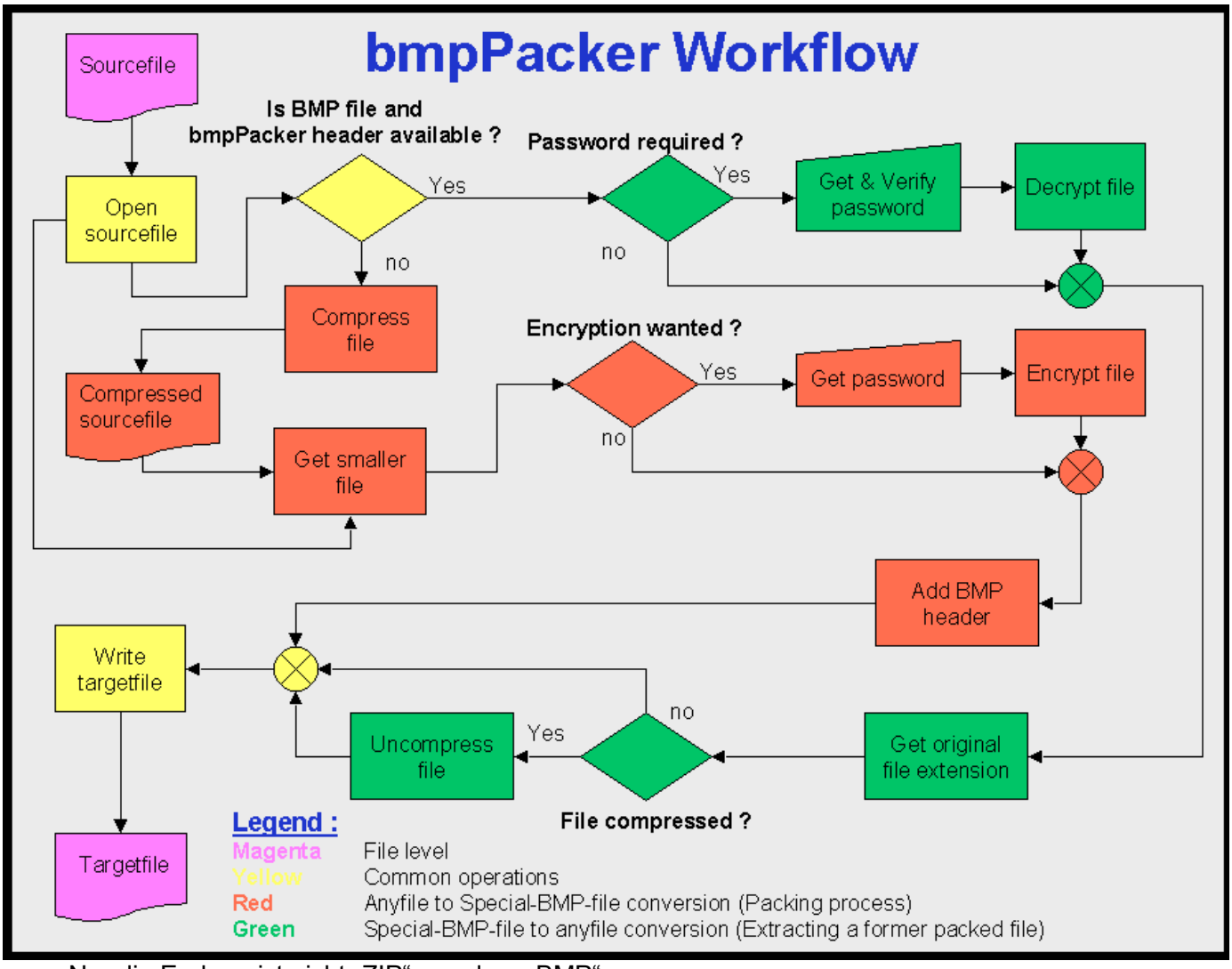
Further bmpPacker provides a kind of „backdoor“ by hiding any kind of file in BMP-containers. You have to ensure that you have the system adminstrators permission if you want to use bmpPacker in your company or in any other non-private environment.

The author of bmpPacker is not liable for any illegal acts you may perform by exporting/importing or using bmpPacker.

2 Was ist bmpPacker

bmpPacker ist ein Programm, mit dem eine einzelne Quell-Datei in eine Bitmap-Grafik konvertiert werden kann. Optionale Kompression und Verschlüsselung sind ebenfalls möglich.

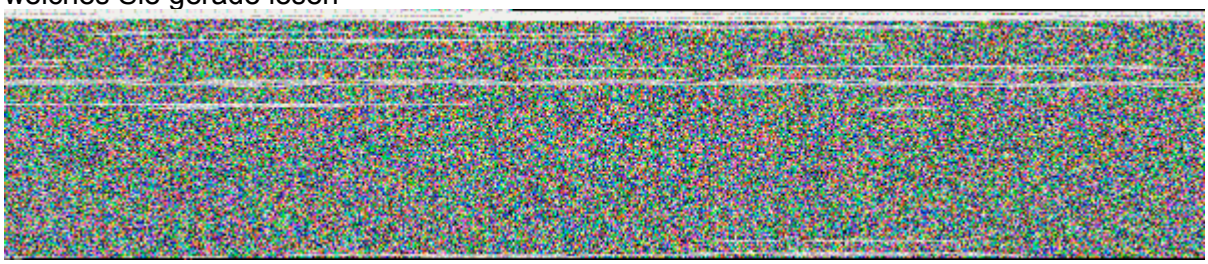
Die resultierende Datei ist eine 100% kompatible Windows BMP Datei.
Sie können sich dies als eine Art ZIP Datei vorstellen, welche optional verschlüsselt ist.



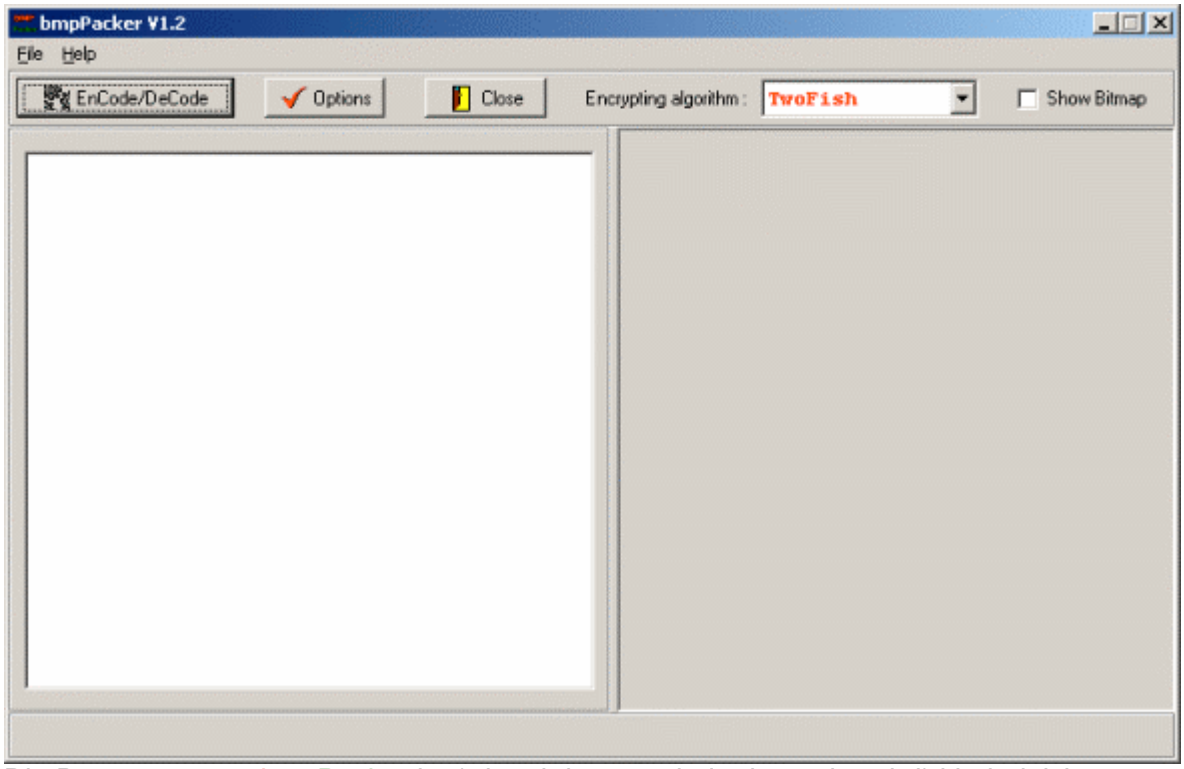
Nur die Endung ist nicht „ZIP“, sondern „BMP“

Sie können sich das Ergebnis einer Konvertierung als normale Windows BMP Datei betrachten. Jedoch ist **bmpPacker** die einzige Möglichkeit, die Originaldatei wieder aus dem Bild zu extrahieren.

Ein kleines Beispiel :
Das folgende Bild, ist die konvertierte Version des Benutzerhandbuchs „**bmpPacker_ger.pdf**“, welches Sie gerade lesen



3 Wie benutzt man bmpPacker

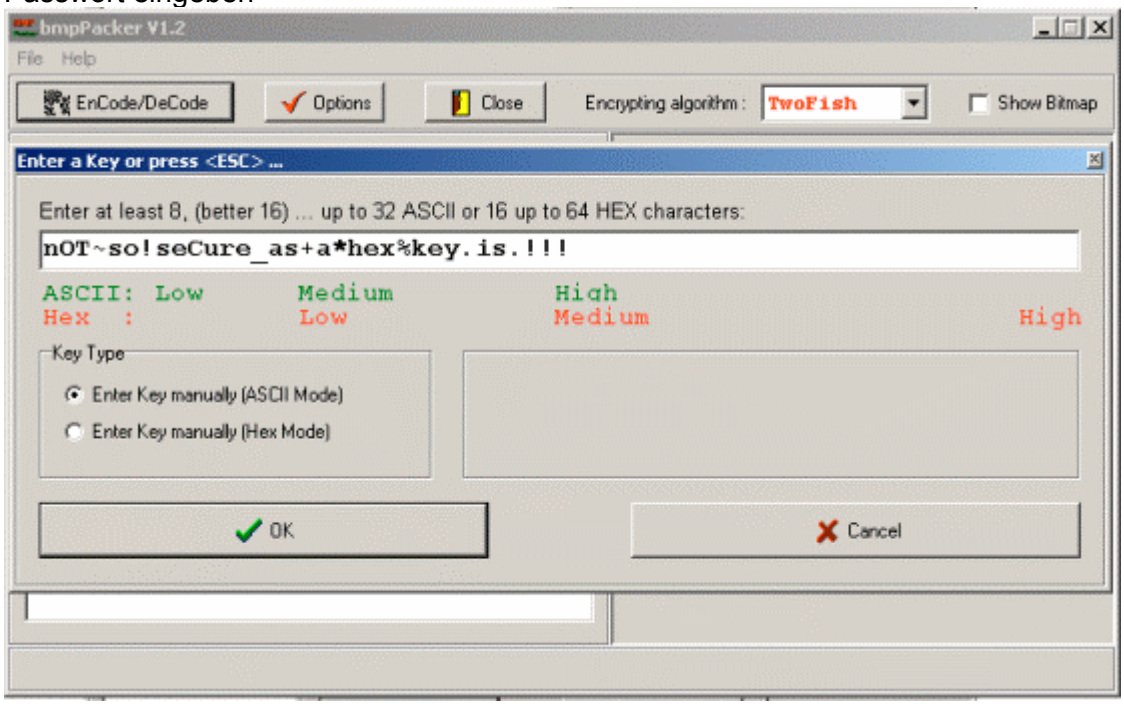


Die Benutzung von bmpPacker ist (wie mir immer wieder bestätigt wird) kinderleicht...

3.1 Interaktive Kodierung/Dekodierung

3.1.1 Kodierung einer Datei

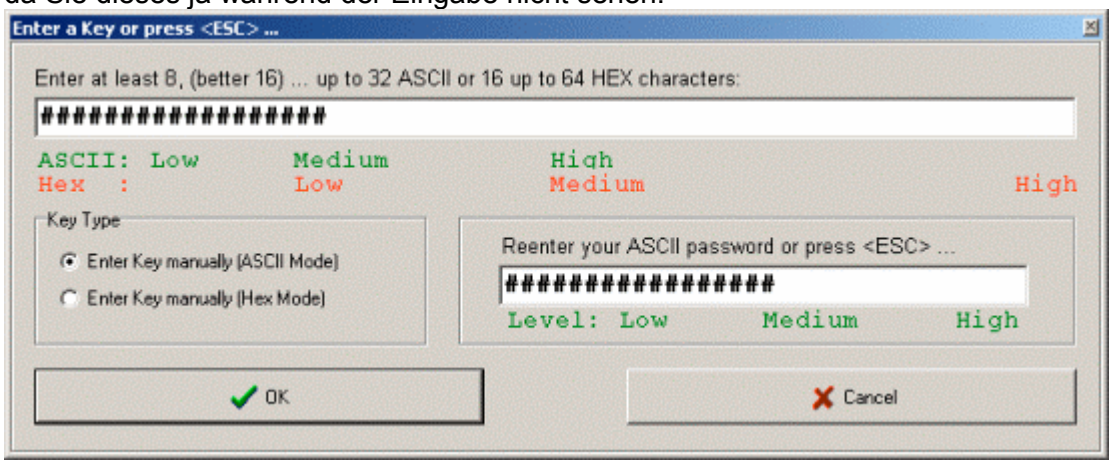
Drücken Sie den "EnCode/Decode" Knopf und wählen Sie eine Datei aus. Je nachdem, ob Sie einen Verschlüsselungsalgorithmus ausgewählt haben, müssen Sie anschließend ein Passwort eingeben



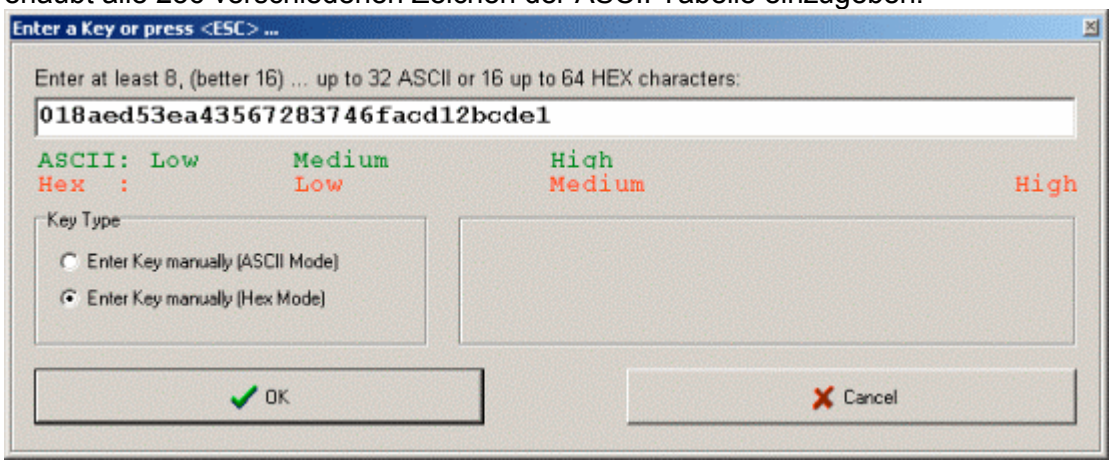
Tipp: Benutzen Sie dabei so viel wie möglich auch Sonderzeichen anstelle von normalen Buchstaben ...

Wenn Sie die Option „hide password“ (siehe 3.3.3) gewählt haben,

müssen Sie das Passwort zur Sicherheit zweimal eingeben,
da Sie dieses ja während der Eingabe nicht sehen.



Der sicherste Schlüssel ist hingegen ein Hex-Schlüssel, welcher es Ihnen
erlaubt alle 256 verschiedenen Zeichen der ASCII Tabelle einzugeben:



Wenn Sie zu irgendeiner Zeit abbrechen wollen, müssen Sie nur die Escapetaste betätigen.

Anschließend beginnt die Konvertierung:
(Beispiel mit „bmpPacker.pdf“):

```
Try to encode file:
-> "bmpPacker.pdf"

Open source file
-> File size        : 231.219 bytes

Try to compress sourcefile
-> Compr. file size : 216.921 bytes
-> Savings          : 6.2 %

Calculating Checksum(s)
-> Source file CRC32  : 0x0DAA7AA4
-> Compr. file CRC32  : 0xD0BEDCDB

Open target file
-> "bmpPacker.bmp"

Encoding process starts
-> Writing BMP file with
-> 608 x 120 Pixel

Enabling password encryption
-> CRC Key : 0xCBCC
```

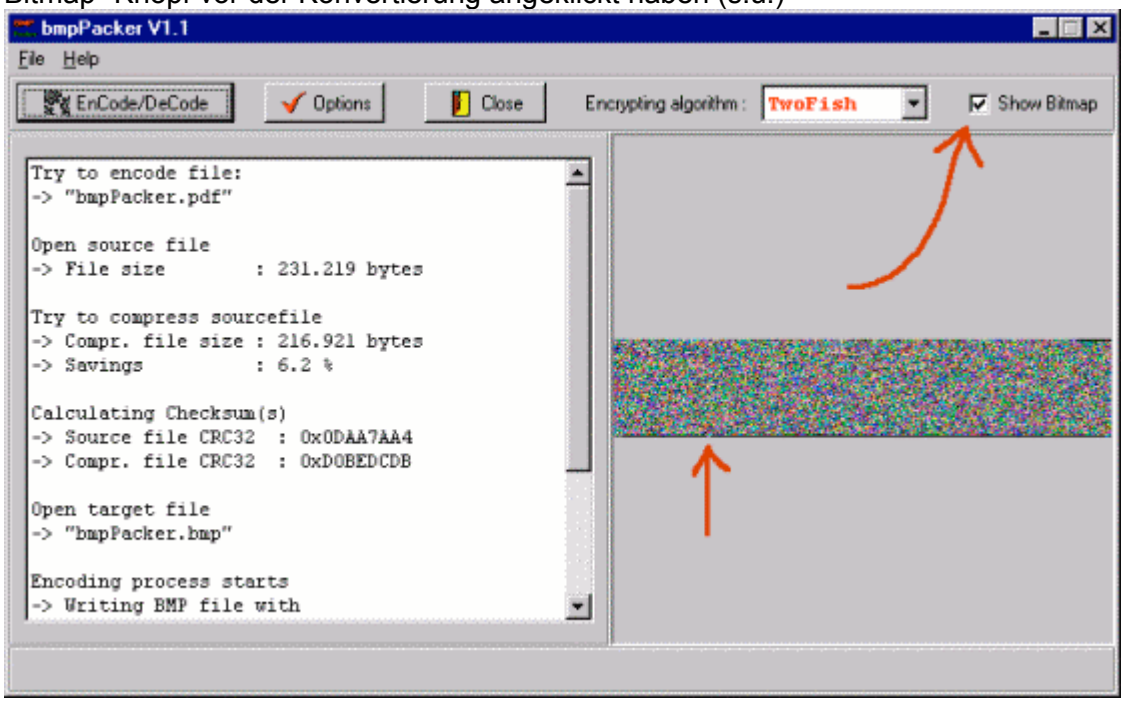
```
-> TwoFish encryption method
-> Block cipher mode = 192 bit

Ready in (hh:mm:ss) = 00:00:00
```

Das war's schon.

Tipp: Wenn Sie auf das Ablauffenster doppelt mit der linken Maustaste klicken, wird der Inhalt des Fensters in die Zwischenablage kopiert.

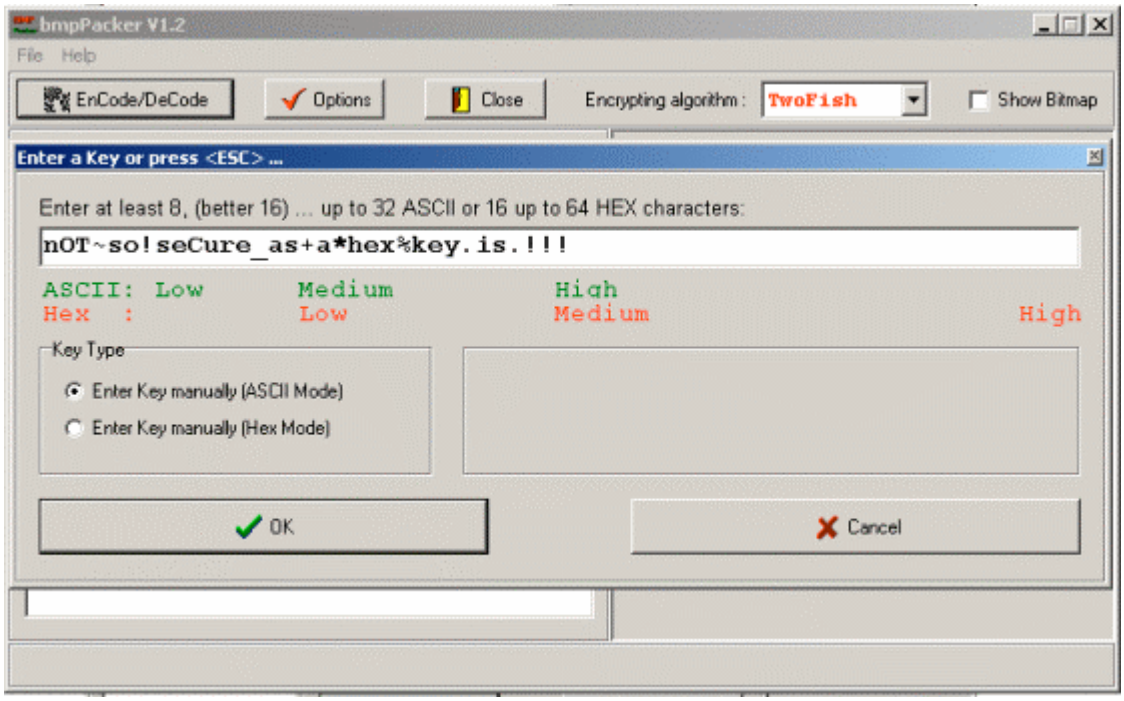
Wenn Sie sich das Resultat Ihrer Konvertierung als Grafik ansehen wollen, müssen Sie den "Show Bitmap" Knopf vor der Konvertierung angeklickt haben (s.u.)



Die betreffende Datei wird, wenn Sie kleiner als 10MB Speicher benötigt auf der rechten Fensterhälfte angezeigt.

3.1.2 Dekodierung einer Datei

Dies funktioniert genauso wie die Kodierung:
Drücken Sie den "EnCode/Decode" Knopf und wählen Sie eine BMP Datei aus.
Wenn die BMP Datei mit einem Passwort verschlüsselt war,
werden Sie aufgefordert das korrekte Passwort einzugeben



Wenn das Passwort nicht richtig war, erhalten Sie die folgende Meldung:

Bei richtigem Passwort oder wenn kein Passwort notwendig war, startet der Dekodierungsprozess...

Der aktuelle Status des Decodierungsprozesses wird wieder im Ablauffenster angezeigt:

```
Try to decode file:
-> "NVIDIA_nforce-1.bmp"

Open source file
-> File size         : 98.550 bytes
-> Searching for bmpPacker header

Decoding process starts
-> Password is required...

Writing target file :
-> "NVIDIA_nforce-1.0-0241.suse80.i386.rpm"
-> Original   file size : 97.646 bytes
-> Compressed file size : 95.863 bytes

Enabling password decryption
-> CRC Key : 0xE910
-> TwoFish decryption method
-> Block cipher mode = 192 bit (medium)
```

```
-> Calculation CRC 32 checksum ...

Checking compressed file consistency...
-> Stored CRC32 checksum     : 0xF23D1148
-> Calculated CRC32 checksum : 0xF23D1148
-> Compressed File should be OK :-)

Decompressing file ...

-> Calculating CRC 32 checksum ...

Checking file consistency...
-> Stored CRC32 checksum     : 0x1E9B3D38
-> Calculated CRC32 checksum : 0x1E9B3D38
-> File should be OK :-)

Ready in (hh:mm:ss) = 00:00:00
```

Fertig...

3.2 Automatisches Kodieren/Dekodieren

bmpPacker kann einen Dateinamen auch per Kommandozeilenoption empfangen.

Tipp:
Erzeugen Sie eine Verknüpfung zu bmpPacker und legen diese im Sendtod Ordner Ihres Windowsprofils ab.

Für XP:

```
c:\Dokumente & Einstellungen\<Benutzer Profil>\SendTo
```

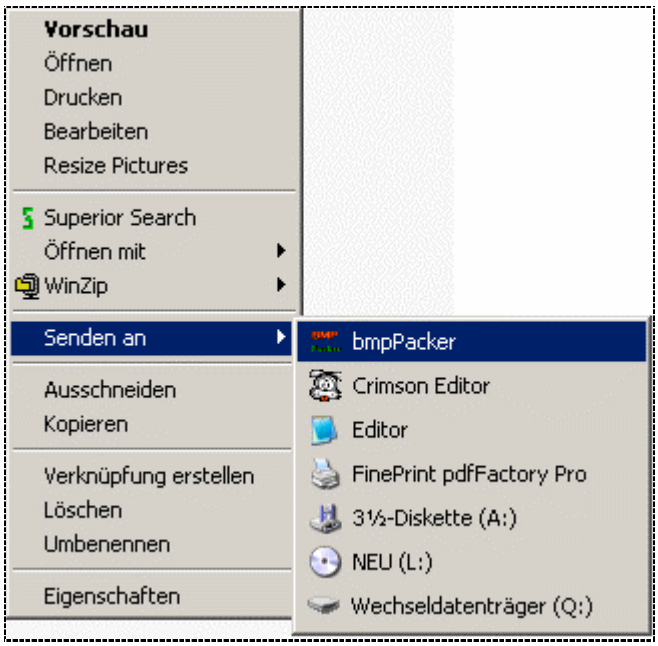
For Win95/98/Me Users:

```
c:\<Windows-Installationsverzeichnis >\SendTo
```

Für NT & W2K:

```
c:\<Windows-Installationsverzeichnis>\profiles\<Benutzer Profil>\SendTo
```

Wenn Sie eine betreffende Datei kodieren oder dekodieren wollen, so müssen Sie nur noch mit der rechten Maustaste auf die betreffende Datei klicken. Anschließend können Sie unter dem „Senden an“ Sub-Menü bmpPacker auswählen:



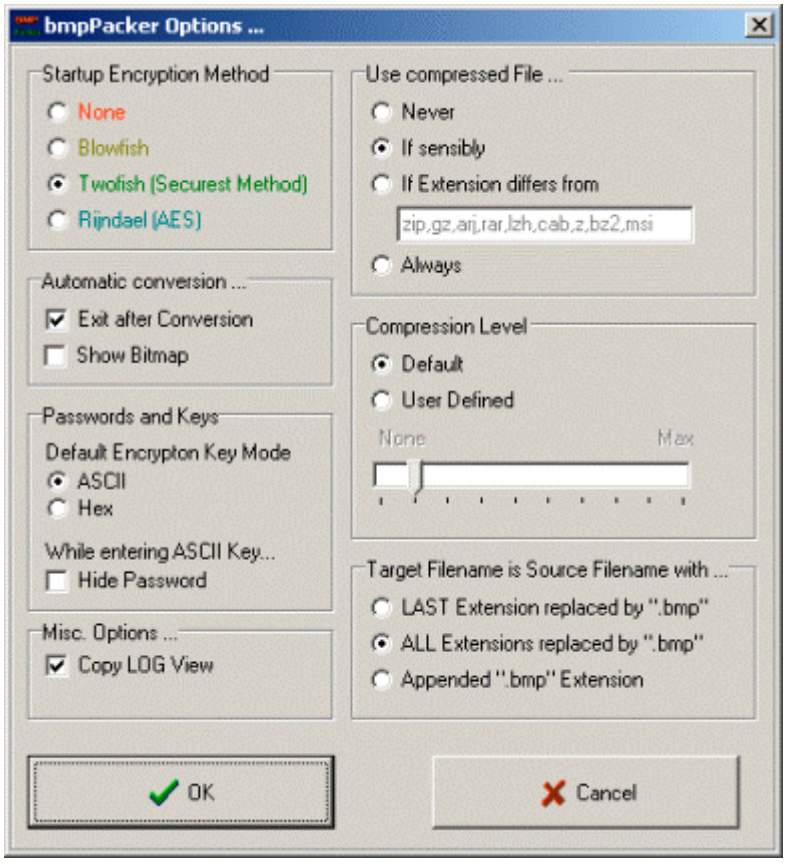
bmpPacker wird dann die Kodierung/Dekodierung gemäß Ihren Einstellungen vornehmen.

3.3 Optionen

Seit Version 1.1 können diverse Optionen eingestellt werden. Diese werden bei Verlassen des Programms automatisch gespeichert.

Das Optionsmenü erreichen Sie über den Knopf „Options“ oder den Menüpunkt „Options“ aus dem „File“ Menü.

Einige Optionen werden sofort umgesetzt, andere erst beim nächsten Start.



3.3.1 Startup Encryption Method

Dies ist die Verschlüsselungs-Methode die nach dem Start des Programms eingestellt ist.

3.3.2 Automatic Conversion

- **Exit after Conversion**
  Bei einer Konvertierung die durch einen Aufruf aus der Kommandozeile erfolgte, legt diese Option fest, ob bmpPacker nach der Kodierung/Dekodierung automatisch beendet wird.

- **Show Bitmap**
  Hier können Sie festlegen, ob nach einem Start die BMP Dateien angezeigt werden sollen oder nicht.

3.3.3 Passwords & Keys

- Default Encryption Key Method
  ASCII : Passwörter standardmäßig als ASCII Zeichenkette eingegeben.
  Hex : Passwörter standardmäßig als hexadezimale Zeichenkette eingegeben.

- **Hide Password**
  Wenn Sie diese Option gewählt haben, wird das Passwort (im ASCII Modus) während der Eingabe nur durch „#“ Zeichen dargestellt, dafür müssen Sie dieses dann aus Sicherheitsgründen zweimal eingeben.

3.3.4 Use compressed file ...

Seit V1.1 unterstützt bmpPacker auch die Kompression der Quelldatei.
Die folgenden Optionen legen fest, wie die Kompression erfolgen soll.

- **Never**
  Keine Kompression.

- **If sensibly (default)**
  bmpPacker komprimiert die Quelldatei und schaut anschließend nach, ob es sinnvoll ist, die komprimierte oder die Originaldatei zu nehmen.
  => Wenn die komprimierte Datei kleiner als die Originaldatei, so wird die komprimierte Datei kodiert
  => Wenn die komprimierte Datei größer als die Originaldatei, so wird die Originaldatei kodiert

- **if Extension differs from...**
  Sie können die Kompression auch abhängig von der Dateinamens-Erweiterung der Quelldatei machen. Führen Sie in der Liste alle Erweiterungen auf, bei denen bmpPacker keine Kompression vornehmen soll, da diese höchstwahrscheinlich keinen Erfolg bringt.

  Die Erweiterungen müssen durch Kommata oder Semikola getrennt sein. Die Groß/Kleinschreibung wird dabei nicht unterschieden. Die voreingestellten Erweiterungen sind: zip,gz,arj,rar,lzh,cab,z,bz2,msi

  zip Komprimierte Datei, verwaltet durch **PKZIP** oder **WINZIP**
  gz Komprimierte Datei, verwaltet durch **GNU ZIP** (gzip)
  arj Komprimierte Datei, verwaltet durch **ARJ**
  rar Komprimierte Datei, verwaltet durch **RAR**
  lzh Komprimierte Datei, verwaltet durch **Lharc**
  cab Komprimierte Datei, verwaltet durch Windows
  z Komprimierte Datei, verwaltet durch das UNIX/LINUX/BSD **compress**
  bz2 Komprimierte Datei, wird häufig von UNIX/LINUX/BSD Systemen verwandt
  msi Microsoft eigenes Kompressionsformat

- **Always**
  Es wird immer komprimiert und auch immer die komprimierte Datei kodiert.

3.3.5 Compression Level

Dies ist der Grad an Komprimierung, der auf die betreffenden Quelldateien angewandt werden soll.

- **Default**
  Standard Grad wird verwandt.

- **User Defined**
  Benutzerabhängige Kompression wird genutzt. Sie können die Kompression in 10 Schritten einstellen:
  Ganz-links = keine Kompression
  Ganz-rechts = Maximale Kompression

3.3.6 Misc. Options

- Copy LOG View
  Nachdem eine Kodierung/Dekodierung durchlief, wird bei gesetzter Option der Inhalt des Ablauffensters automatisch in die Zwischenablage kopiert.

3.3.7 Target Filename is Source Filename with...

- Last Extension replaced by „.bmp“
  Der Zieldateiname ist der gleiche wie der Quellname mit der Ausnahme, dass die letzte Dateierweiterung durch „.bmp“ ersetzt wird. Dies ist die Standardeinstellung der bisherigen bmpPacker Versionen.
  Beispiel:
  Quelldateiname = xyz.txt.tar.**gz** => Zieldateiname = xyz.txt.tar.**bmp**

- All Extensions replaced by „.bmp“
  Der Zieldateiname ist der gleiche wie der Quellname mit der Ausnahme, dass alle Extensionen durch „.bmp“ ersetzt werden.
  Beispiel:
  Quelldateiname = xyz.**txt.tar.gz** => Zieldateiname = xyz.**bmp**

- Appended „.bmp“ Extension
  Der Zieldateiname ist der gleiche wie der Quellname mit der Ausnahme, dass die Extension „.bmp“ lediglich nur angefügt wird.
  Beispiel:
  Quelldateiname = xyz.txt.tar.gz => Zieldateiname = xyz.txt.tar.gz.**bmp**

3.4 Konfigurations-Dateiformat

Die Optionen werden in einer Datei namens "**bmpPacker.cfg**" gespeichert.
Sie müssen sicherstellen, dass das Verzeichnis in dem sich bmpPacker befindet nicht schreibgeschützt ist, da sonst die Konfigurationsdatei nicht abgelegt werden kann.

3.5 Kommandozeilen Optionen

Die Kommandozeilen Syntax:

```
bmpPacker [ -K:Key | -k:Key | -B | -b | -T | -t | -R | -r | -N | -n | -S | -s ]
[filename]
```

Bedeutung:

`-K: oder –k:`	„Key“ im Hexformat, welcher für die Kompression genutzt werden soll. (Zur automatischen Verschlüsselung per Kommandozeile sollten Sie die Option „**Exit after Conversion**“ aktivieren)
`-B oder -b`	Blowfish Verschlüsselung
`-T oder -t`	Twofish Verschlüsselung
`-R oder -r`	Rijndael Verschlüsselung
`-N oder -n`	Keine Verschlüsselung
`-S oder -s`	Zeigt die BMP Datei nach der Konvertierung an.
`filename`	Der Dateiname einer zu kodierenden/dekodierenden Datei

Die Kommandozeilen Optionen haben eine höhere Priorität als die eingestellten Optionen, jedoch werden die eingestellten Optionen nicht überschrieben.

Beispiel:
Wenn Sie „Rijndael“ als Startverschlüsselungsmethode eingestellt haben und folgendes eingeben:

```
bmpPacker -T myfile.doc
```

wird die Datei mit der Twofish Verschlüsselungsmethode verschlüsselt. Wenn Sie das nächste Mal bmpPacker ohne Kommandozeilenoptionen starten, ist wieder „Rijndael“ eingestellt.

4 Module

Das Programm bmpPacker wurde in C++ geschrieben.
Ich benutzte den Borland C++ Builder 5 um das Projekt zu erstellen.

http://www.borland.com/cbuilder/index.html

Das Benutzerhandbuch wurde mit MS-Word 97 erstellt,
die Konvertierung ins PDF Format wurde mit dem Adobe Acrobat 4.05 durchgeführt.

4.1 Verschlüsselungsmethoden

Alle verwendeten Verschlüsselungsmethoden sind bis heute nicht nicht geknackt worden.
Sie sind ferner wesentlich sicherer als DES oder triple DES.

4.1.1 BlowFish

Ein sehr sicherer 64bit Block-Cipher-Verschlüsselungsalgorithmus mit variabler Schlüssellänge.
Blowfish wurde 1993 von Bruce Schneier entwickelt
Zahlreiche Applikationen benutzen BlowFish.
U.a. werden bei Open BSD die Passworte mit diesem Algorithmus verschlüsselt.

Der Sourcecode in C stammt von Bruce Schneier.

http://www.counterpane.com/blowfish.html

Der C++ Code den ich für bmpPacker verwende wurde von Jim Conger geschrieben.
(Der Code ist ebenfalls auf der Counterpane Homepage verfügbar)

4.1.2 TwoFish

TwoFish wurde ebenfalls von Bruce Schneier entwickelt.
TwoFish ist nicht so weit verbreitet wie Blowfish, jedoch noch einiges sicherer.
Die Block-Cipher-Größe ist abhängig von der Passwortlänge und umfasst 128 bis 256bit

Link zur Counterpane Homepage

http://www.counterpane.com/twofish.html

Der C/C++ Code den ich in bmpPacker verwende stammt von Dr B. R. Gladman

Link zur Homepage von Dr B. R. Gladman

http://fp.gladman.plus.com/cryptography_technology/aes2/index.htm

4.1.3 Rijndael

Rijndael ist der AES (American Encryption Standard) Gewinner des Jahres 2002.
Rijndael wurde von Vincent Rijmen und Joan Daemen entwickelt.
Die Block-Cipher-Größe ist abhängig von der Passwortlänge und umfasst 128 bis 256bit

Link zur Homepage von Vincent Rijmen und Joan Daemen

http://www.esat.kuleuven.ac.be/~rijmen/rijndael/

Der C/C++ Code den ich in bmpPacker verwende stammt von Dr B. R. Gladman

Link zur Homepage von Dr B. R. Gladman

http://fp.gladman.plus.com/cryptography_technology/aes2/index.htm

4.1.4 NIST National Institute of Standards & Technology

http://csrc.nist.gov

Link zum American Encryption Standard

http://csrc.nist.gov/encryption/aes/

4.2 Checksummen Methoden

Ich benutze eine **CRC 16** Routine zur Passwort Checksummenbildung und eine **CRC 32** Routine zur Berechnung der Quell/Kompressionsdatei Checksumme.

Link zum CRC32 Sourcecode:

http://www.createwindow.com/programming/crc32/crcfile.htm

4.3 Kompression Methode

Zur Kompression verwende ich die *gnu* **zlib compression library** von Jean-Loup Gailly und Mark Adler welche ebenfalls auch u.a. für ***gnu zip*** (gzip & gunzip etc.) verwendet wird.
Link zur zlib Homepage:

http://www.gzip.org/zlib

4.4 History

Version	Build – ID	Contents
1.0	2002-11-12	• Erste Veröffentlichung
1.1	2002-11-27	• Dateikompression nun möglich • Options-Maske implementiert • Kommandozeilen Optionen wurden implementiert • Eine neue Password Eingaberoutine wurde implementiert
1.1a	2003-06-13	• Ein paar Syntax Korrekturen im Dokument und im Programm von Andrew Miller
1.2	2003-11-30	• Die Ablauf-Listbox zeigt nun immer die aktuelle Zeile an • Es können nun auch hexadezimale Passwörter eingegeben werden. (Eine neue Password Eingaberoutine wurde implementiert) • Wie in der Doku beschrieben lässt sich mit -n/-N die Verschlüsselung nun per Kommandozeilenoption ausschalten. • Das ConfigFile wird nun bei Nicht-Vorhandensein im gesamten Suchpfad gesucht. • Neue Option -K ermöglicht die Übergabe eines Hexschlüssels per Kommandozeilenoption. • Neue Optionen erlauben verschiedene Ausprägungen der .bmp Extension