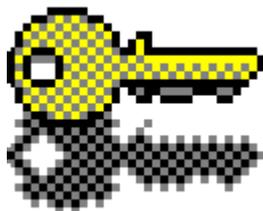


Enigma for Windows



Version 3.2

User's Manual

**© Copyright 1992, 1993, 1994, 1995 by SWS
All Rights Reserved
Software - Made in Germany**

**Stefan Wolf Software
GartenStr. 22
D-61449 Steinbach/Ts.
GERMANY
FON/FAX: +49 (0) 6171 980483
Compu\$erve: 100111,140**

Table of Contents

- 1. What is Enigma for Windows ?**
 - 1.1. History of Enigma for Windows**
 - 1.2. Specifications**
 - 1.3. Technical Support**
- 2. Installation**
 - 2.1. System Requirements**
 - 2.2. Installation Procedure**
 - 2.2.1. The Setup Program**
 - 2.2.2. Installing Enigma for Windows**
 - 2.3. Updating Enigma for Windows**
 - 2.4. Starting Enigma for Windows**
 - 2.5. Accessing Help**
- 3. The User Interface**
 - 3.1. The Enigma for Windows Setup**
 - 3.2. Default User Passwords**
 - 3.2.1. Generating good keys**
 - 3.2.2. Key in hexadecimal notation**
 - 3.3. Screen Locking**
- 4. Getting Started**
 - 4.1. Selecting Files**
 - 4.1.1. Expert Adjustments**
 - 4.2. EnCrypting File(s)**
 - 4.3. DeCrypting File**
 - 4.4. Wiping File(s)**
- 5. Algorithms**
 - 5.1. Data Encryption Standard (DES)**
 - 5.1.1. The Security of DES**
 - 5.2. XOR**
 - 5.3. Regular Expression**
- A. Appendix**
 - A.1. Footnotes**
 - A.2. Distribution**
 - A.3. Program Files**

1. What is Enigma for Windows ?

Enigma for Windows is a powerful program for encrypting¹ and decrypting files and directories of any type. Besides being able to conceal the contents of files it can be used as an electronic paper shredder. This program is named after the legendary encoding machine **Enigma 4** that was used by the Germans in the Second World War.

Everyone has files that should not be seen by others. Be it a patent or something as important as a love letter. Everyday many employees handle data that is not meant for the eyes of others, for example company statistics, personnel records, payrolls and others. This type of data is only "safe" after it has been locked away with a lock and key.

In this day and age of massive computer use by banks, doctors, officials and a multitude of other offices it has become necessary to find alternatives to the traditional methods of securing data. Computer networks and the free exchange of data across these networks have added a whole new dimension to this problem.

Even though it is a good idea to lock away diskettes which contain sensitive data, encoding the data on those diskettes and using your own personal password as the key gives you a higher level of security. You should always encrypt sensitive or secret documents that you have received so they can under no circumstances be read without your permission.

Encrypted files cannot be read or decrypted by any other users. The only way to make the file readable and usable again is to decrypt it with the same password that was used to encrypt it.

The ability to keep your data safe from unauthorized access depends on the encryption method that you use. Two methods have gained widespread acceptance; the RSA² - Public Key Encryption method and the Data Encryption Standard (DES). The DES is used by many US. Government agencies and is a de facto standard. This method was also implemented in **Enigma for Windows** because of its safety and proven workability in everyday use. One can be sure that data encrypted with (Triple) DES cannot be decrypted in a reasonable amount of time with the help of today's technology.

Many offices and government agencies use paper shredders to destroy their sensitive documents. The function **Wipe** is the electronic equivalent of this. Many computer users do not know that files deleted with the DOS command **del** can often be recovered from their hard disks without much trouble even after a longer period of time. After using the **Wipe** function on a file you can be sure that no trace of it can be found on your harddisk any more.

1.1. History of Enigma for Windows

Version 1.0 - 09.01.1992

* Initial release.

Version 1.1 - 05.01.1993

* Error in file handling fixed.

- * Option to compress files before encrypting.

Version 2.0 - 05.01.1994

- * Comfortable installation program added.
- * Context sensitive help by pressing the **F1** key.
- * Encrypting, decrypting and deleting of several files or whole directories in one step.
- * Dialog controlled choice of the target directory.
- * Stopping the encryption process.
- * 15 % performance gain for DES based operations.
- * Command line interface added.
- * English version is now available.
- * Enhanced setup.
- * The option of compressing a file before it is encrypted was removed.
- * Files created with Version 1.x are incompatible with version 2.0. This was necessary in order to permit the simultaneous handling of several files. The product of this work is a modern archive structure on which future versions will be oriented.

Version 3.0 - 12.01.1994

- * New double and triple DES algorithms added.
- * DES routines has been isolated in a separate DLL to support their implementation in third party DES applications.
- * Triple DES developer kit added.
- * Screen distortion error by use of custom screen fonts fixed.
- * Support of a default user specific password for each algorithm.
- * Screen locking feature added.

Version 3.1 - 06.01.1995

- * Runs in protected mode only (386+ required).
- * Error in decrypting big files fixed.
- * Professional key input dialog added.
- * More than 300 % performance gain for DES based operations.

Version 3.2 - 09.01.1995

- * Some minor bugfixes
- * Output to self-extracting EXE file added.
- * Professional search criteria in file selection dialog added.
- * Enigma compatible DOS utility added

1.2. Specifications

The DES-Algorithm used in this program conforms to the following standards (as far as this is possible for a software implementation).

FIPS³ PUB 46-1 - Data Encryption Standard (1988)

Contains the specification for the Data Encryption Standard (DES) algorithm, which can be implemented hardware to protect sensitive unclassified information.

FIPS PUB 74 - Guidelines for Implementing and Using the NBS DES (1981)

Companion to FIPS PUB 46-1. Contains guidance for the use of cryptographic techniques.

FIPS PUB 81 - DES Modes of Operation (1980)

Companion to FIPS-PUB 46-1. Contains descriptions of the four modes of operation for the DES: Electronic Code book (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), and Output Feedback (OFB).

ANSI⁴ X3.92 - Data Encryption Algorithm (DEA)

ANSI X3.106 - DEA Modes of Operation

In 1986 the ISO⁵ published the "DEA-1" specification, where it is recommended that DES be used for encoding data. The Wipe algorithm conforms to specification CSC-STD-005-85 of the National Computer Security Center, it is described in the department of Defense Magnetic Remanence Security Guideline, 15 Nov. 85, Section 5.3.1 .

1.3. Technical Support

All questions regarding technical support, discount and wholesale prices should be addressed to:

**Stefan Wolf Software
GartenStr. 22
D-61449 Steinbach/Ts.
GERMANY
FON/FAX: +49 (0) 6171 980483
Compu\$erve: 100111,140**

2. Installation

2.1. System Requirements

The minimum requirements for running **Enigma for Windows** Version 3.2 are:

Software:

* **MS Windows 3.1+** or **MS Windows NT 3.1+** or **MS Windows 95+**.

Note: If you use on-line compressors such as **Stacker** or **DoubleSpace** we **cannot** guarantee that data which has been deleted with **Wipe** cannot be recovered again.

Hardware:

* 386+, VGA, 2.5 MB free space on harddisk

Enigma for Windows does not require any special hardware to other than the computers ability to run one of the above mentioned operating systems.

Note: Even though **Enigma for Windows** uses very fast algorithms their complexity (mainly the DES modes) make encrypting and decrypting data a time-consuming operation. It is therefor recommended that you use an i486 or Pentium based PC.

2.2. Installation Procedure

2.2.1. The Setup Program

The setup program performs the following tasks:

- * Expand and copies the **Enigma for Windows** program to your hard disk to the directory you specify (default is **C:\EFW32U**). Once installed **Enigma for Windows** Version 3.2 will occupy approximately 2500 KB of hard disk space.
- * It modifies the MS-Windows initialization file **WIN.INI** by adding the following line **EN3=C:\EFW32U\EFW32.EXE ^.EN3**.
- * Creates the MS-Windows Program Manager group **Enigma for Windows**.
- * Creates the **EFW32.INI** file in the Windows directory.

2.2.2. Installing Enigma for Windows

1. Start MS-Windows !
2. Start the Program Manager (starts automatically) !
3. Choose the **Run** command in the **File** menu in the Program Manager !
4. Type **A:\INSTALL** or **B:\INSTALL** depending on which drive you are installing from !
5. A dialog box will appear and the recommended directory for the installation of **Enigma for Windows** will be shown.



Choose the directory in which you want to install the program. If the chosen directory doesn't exist it will be created. Click the button **OK** to start the installation.

6. The installation program will now begin to copy the **Enigma for Windows** files to the target directory.

Note: If you want to install **Enigma for Windows** in a network environment make sure that you have read/write privileges in the directory specified.

2.3. Updating Enigma for Windows

The versions 1.1 and 3.2 are **not** compatible. It is therefore necessary to decrypt the data with the version that it was encrypted with. Files encrypted with version 2.x can be decrypted by the current version. Older or existing versions are not removed by the installation program. You will have to remove your old installation manually.

2.4. Starting Enigma for Windows

There are several ways to start **Enigma for Windows** :

Starting from the MS-Windows Program Manager:

1. Open or activate the Program Manager window !
2. Open the group window which contains **Enigma for Windows** !
3. Double-click the **Enigma for Windows** symbol or use the cursor and press **Enter** !

Starting from the Program Managers **File** menu:

1. Open the **File** menu in the Program Managers menu bar !
2. Choose **Run** !
 - > If the program is in your path enter **EFW32** !
 - > If the program is not in your path enter the complete path to where it is located, for example **C:\EFW32\EFW32.EXE** !
3. Click **OK**

Starting from the MS-DOS-prompt:

1. At the DOS-Prompt type the command **WIN EFW32** !
2. Press **Enter** !

Note: If you receive a message that the file could not be found this means that the directory containing **Enigma for Windows** is not in the path. Change to the directory which contains **EFW32.EXE** and try to start it again.

Starting from a MS-Windows command line interface:

1. Once **WinCLI, WinCLI Pro, 4Win** ... is running change the directory to where the program is located enter **EFW32**.

When you start **Enigma for Windows** for the first time you will see a dialog box which will ask you to identify yourself as a legal user of the program. Enter your serial number here. You will find it written on the label of your program diskette.

2.5. Accessing Help

Enigma for Windows has a Context Sensitive Help System to guide you through commands and procedures.

To view the Help Contents:

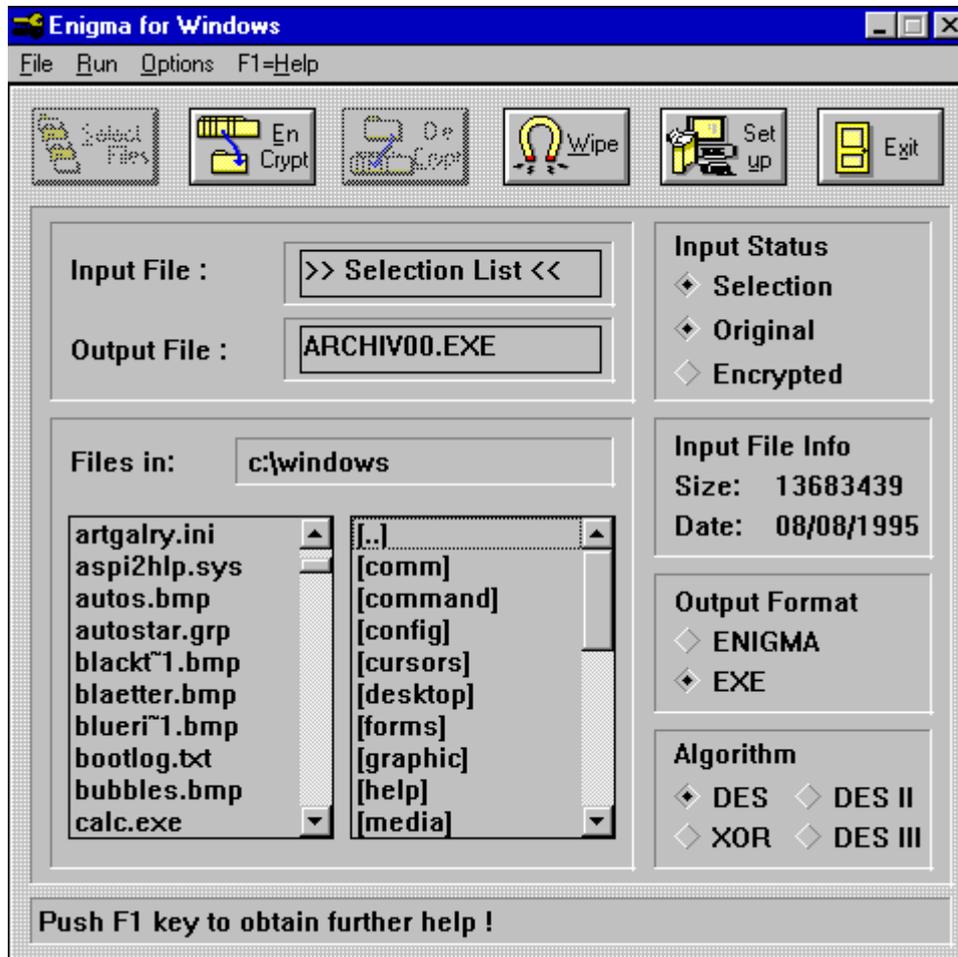
- * From the **Help** menu, choose **Contents**.

To get Context Sensitive Help where you are on the screen:

* Press **F1** or within a dialog box, press the **Help** button if available.

3. The User Interface

In the main window you will find two listboxes and several text fields, 6 dialog buttons, a few status elements and the menu bar.



Dialog Buttons: Each of these 6 buttons (from left to right) has an equivalent in the menu bar. It is also possible to activate each button by a key stroke.

Select Files: Opens a dialog in which files or whole directories can be selected. [(ALT-S),(ALT-F,S)]

EnCrypt: Encrypts the selected files. [(ALT-E),(ALT-R,E)]

DeCrypt: Decrypts the selected file. [(ALT-C),(ALT-R,C)]

Wipe: Wipes out the selected files. [(ALT-W),(ALT-R,W)]

Setup: Opens the window in which various program parameters can be changed. [(ALT-U),(ALT-O,S)]

Exit: Exits *Enigma for Windows*. [(ALT-X),(ALT-F,X),(ALT-F4)]

List Boxes:

Left: Shows the files which are in the current directory.

Right: Shows all directories and drives.

Text Fields:

Input file: This text field shows the name of the input file. It is automatically filled by clicking a file in the left listbox.

Output file: Text field for the name of the output file. You can manually enter the name of the output file here before encrypting or use the given default name.

Files in: Shows the current directory.

Size: Shows the size of the selected input file(s).

Date: Gives the date on which the selected file was created or will be created.

Bottom border: A text field in which help texts are displayed dependent on the mouse cursor position.

Status Elements:

Selection: This status element is marked if a valid input file list has been selected.

Original: This status element is on if the file shown in the input file field has not yet been encrypted with ***Enigma for Windows***. If you want to encrypt a file that has already been encrypted you must click by hand this element.

Encrypted: This status element is marked when the file in the text field **Input file** has been encrypted with this program.

ENIGMA: Creates an Enigma compatible output file format.

EXE: Creates a self extracting EXE file. Use this format to send encrypted files to people who do not have a copy of Enigma.

DES: Status element shows that the DES algorithm is being used. It is automatically marked when a DES encrypted input file has been chosen for decoding.

DES II: Status element shows that the double DES algorithm is being used. It is automatically marked when a double DES encrypted input file has been chosen for decoding.

DES III: Status element shows that the triple DES algorithm is being used. It is automatically marked when a triple DES encrypted input file has been chosen for decoding.

XOR: Shows that the XOR algorithm is being used for encryption and decryption. It is set automatically when the input file has been encrypted with XOR.

3.1. The Enigma for Windows Setup

This dialog is opened by clicking the **Setup** Button in the Main Dialog Box or by pressing the key combination **[(Alt-U)]**.

Save the last directory as default working directory

Mark this status element if you want to change into the last working directory by default. This can be helpful if you want to backup selected files regularly. **(default: not marked) [(ALT-L)]**

Remove files with a simple delete instead of using Wipe

The files are simply deleted and can possibly be restored. **(default: not marked) [(ALT-R)]**

Remove empty directories when deleting directory trees

Removes empty directories when deleting files and whole directory trees with Wipe. **(default: marked) [(ALT-V)]**

Create necessary directories while decrypting

Creates the necessary directory structure while decryption. If this button is not marked the filenames containing a path name will be written into the current directory. For example, *tmp\dir1\file.txt* will be decrypted and written into the current directory with the name *file.txt*. (default: marked) [(ALT-C)]

Encrypt all selected files without further questions

If this status element is marked the selected files will be encrypted without further questions, otherwise you would be able to modify your selection in a dialog box. (default: not marked) [(ALT-E)]

Wipe out all selected files without further questions

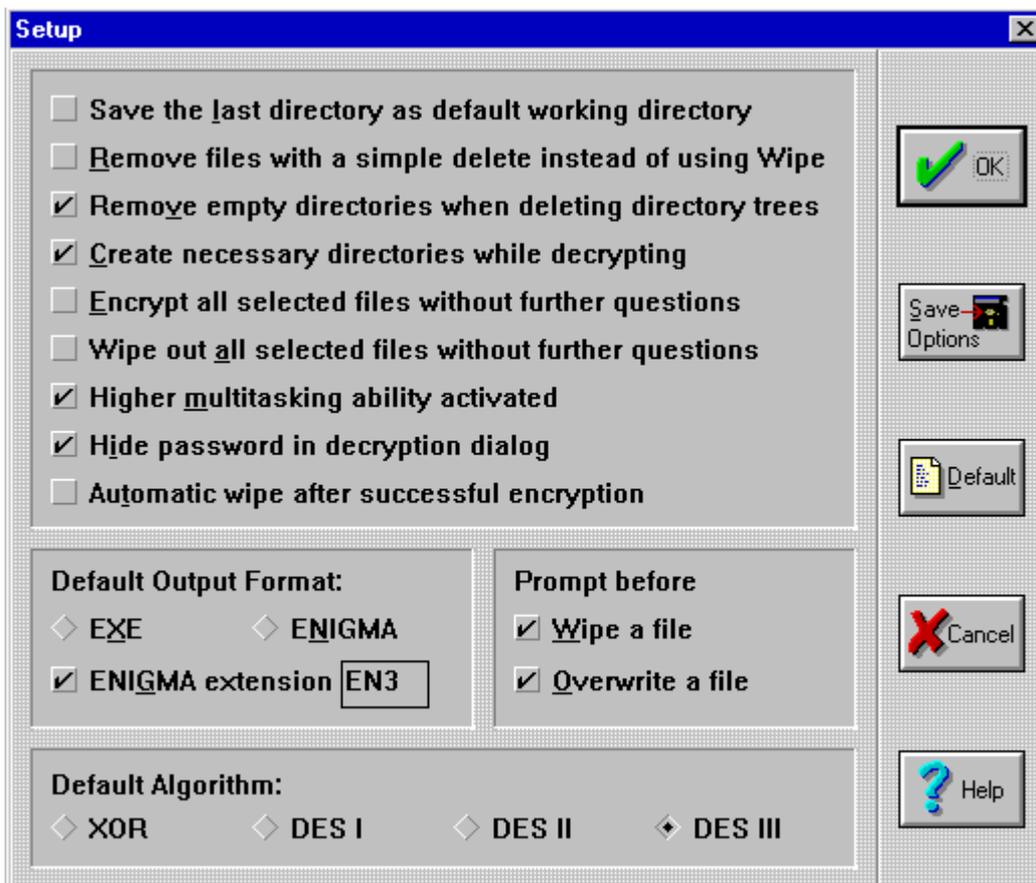
If this status element is marked the selected files will be wiped without further question, otherwise you would be able to modify your selection in a dialog box. (default: not marked) [(ALT-A)]

Higher multitasking ability activated

If this status element is marked, MS-Windows has more time to process the internal message queue and it uses more CPU time for other applications which are running. (default: marked) [(ALT-M)]

Hide password in decryption dialog

If this status element is marked the password will not be showed in the decryption dialog, otherwise the entered password will be showed. (default: marked) [(ALT-I)]



Automatic wipe after successful encryption

If this status element is marked your selected files will be wiped after a successful encryption. You should be very careful with this option. (default: not marked) [(ALT-T)]

Default Output Format EXE

If status element is marked Enigma will produce a self-extracting executable file with an EXE extension. **(default: marked) [(ALT-X)]**

Default Output Format ENIGMA

If status element is marked Enigma will produce a Enigma compatible archiv file with an EN3 extension. You can decrypt this kind of files directly in Enigma. **(default: not marked) [(ALT-N)]**

ENIGMA extension [EN3]

If you don't add an extension to the name of the Enigma output file, the program will automatically add the extension of the textbox. The use of a systematic extension can be helpful in locating encrypted files.**(default: marked) [(ALT-G)]**

Prompt before Wipe out a file

Asks for confirmation before deleting a file with **Wipe**. **(default: marked) [(ALT-W)]**

Prompt before Overwriting a file

Asks for confirmation before overwriting an existing file. This option should always stay marked and you should always make sure that the file was decrypted with the right password otherwise rubbish might be written over the input file.

(default: marked) [(ALT-O)]

Default Algorithm

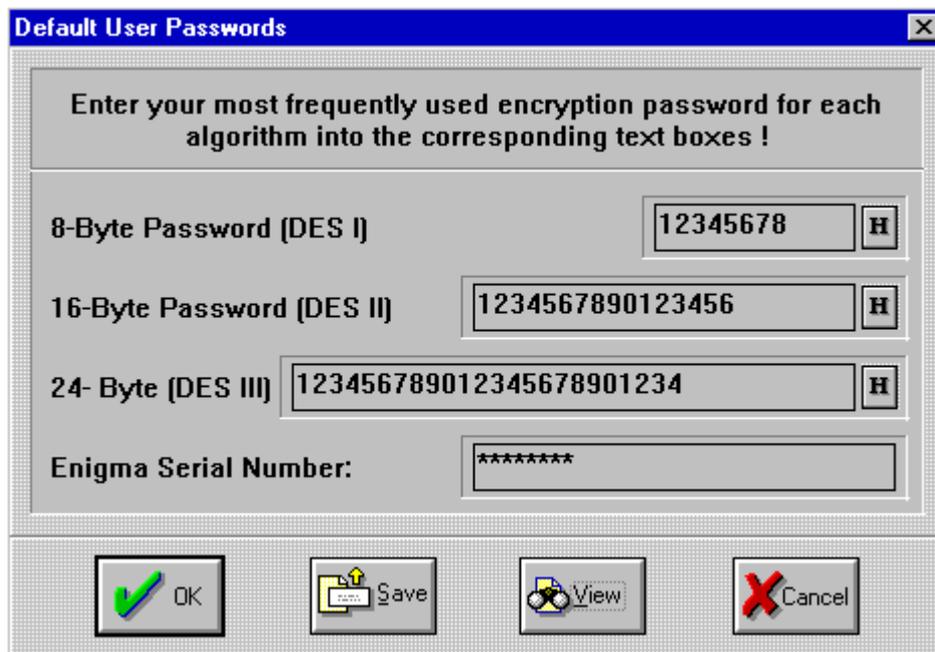
Mark here the algorithm which you want to use for encryption by default.

(default: DES III)

Changes in this menu are only active for the current session. If you want to change the option's permanently you must click the button **Save Options**. **[(ALT-S)]**. To restore the defaults values you can click the button **Default** **[(ALT-D)]**.

3.2. Default User Passwords

With the help of this dialog you can set a fixed private passwords for each algorithm which can be used for encryption. Enter your favorite passwords and your serial number into the corresponding text boxes. For the DES and XOR algorithm you should choose an eight character word - 16 byte and 24 byte length keys are essential for double DES and triple DES to work properly. To enter your keys in hexadecimal notation press the **H** button. You must click the **Save** button if you want use these keys in later sessions. To show your current passwords enter your serial number and click the **View** button !



You should keep your installation diskette in a safe place so that no one can find out the serial number.

3.2.1. Generating good keys

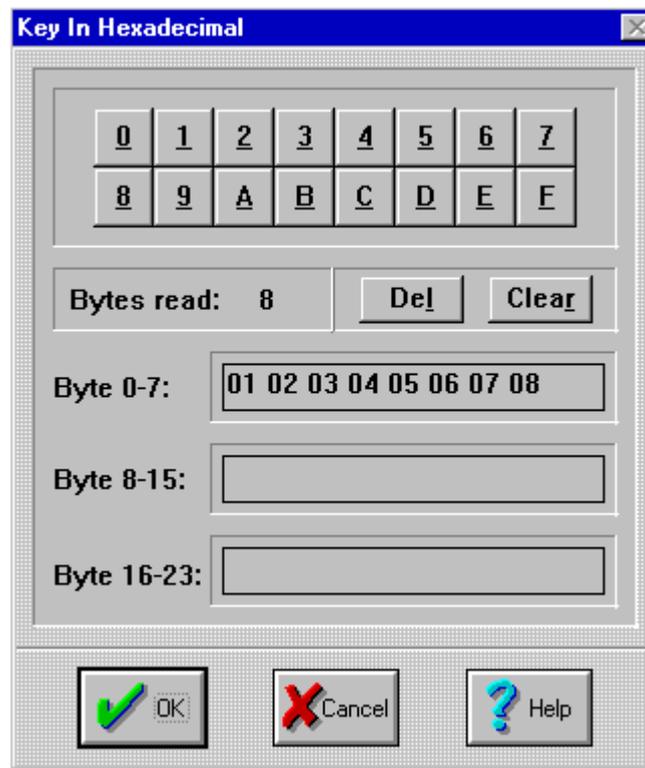
The security of an algorithm rests in the key. No one would try to break the algorithm if they knew that preferably you use the names of your children or your date of birth as your password. It doesn't matter that the program uses DES encryption - it would be simple to decrypt your encrypted data.

If you want to encrypt your data for a short time period it is completely sufficient to encrypt your data with single DES and an alphanumeric key. A good key should be easy to remember, but difficult to guess. Never use parts of your name, dictionary words of common languages, words from databases, names of famous people and places etc. Instead combine several word with special characters, for example "Hund&Katze", "Ja||Nein" or "69You2me". A better method is to use a string of letters which are the initials of a longer phrase, for example "Who is John Galt ?" generates the key "WhisJoGa?" !

If you need to encrypt more valuable data or the data has to be encrypted for a longer time period you should use the double- or triple DES algorithm and correspondingly long keys. The program allows to generate keys made up of the entire ASCII character set. It is worth the trouble to enter a full 24 byte key in hexadecimal notation. Finally you should write down this key and lock it in a safe.

3.2.2. Key in hexadecimal notation

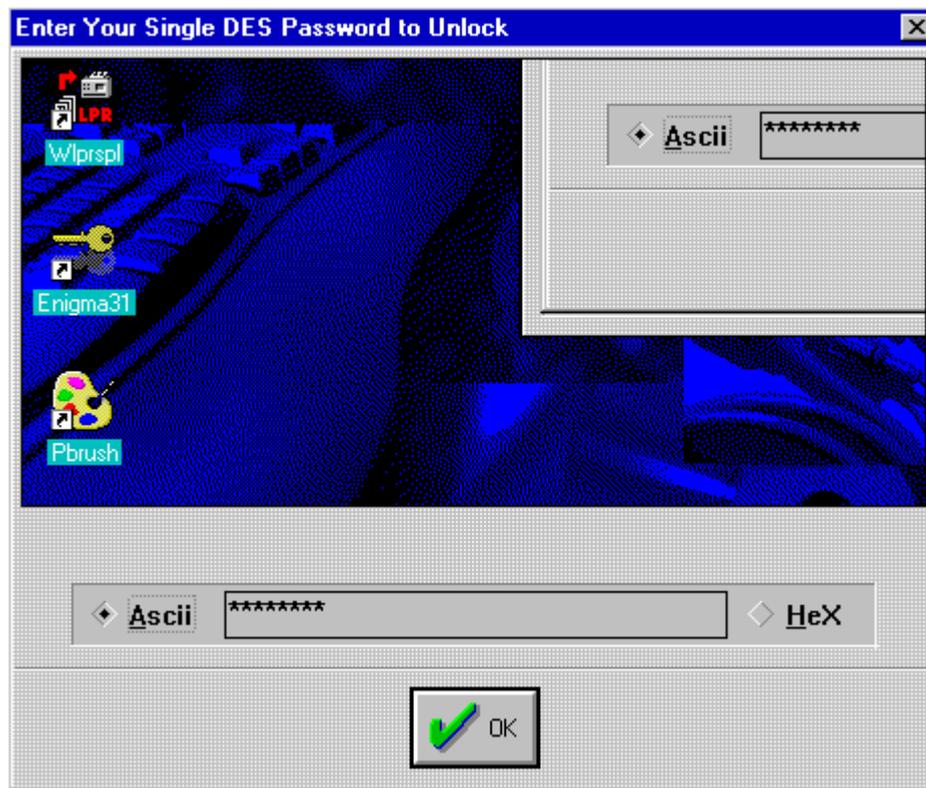
By selecting this dialog you are been able to enter a key in hexadecimal notation and use characters in your password which can't directly be entered with your keyboard



Using characters from the entire 8-bit ASCII character set increases the security of your encrypted data gigantically. Even at a key-length of 8 characters there are 20000 times more possible keys than if you are using only alphanumeric characters [A-Za-z0-9].

3.3. Screen Locking

By selecting this menu item you are been able to lock your current MS-Windows session. You computer will ignore any input by mouse or keyboard outside of this window. It is not possible to start an another program or to switch the session.



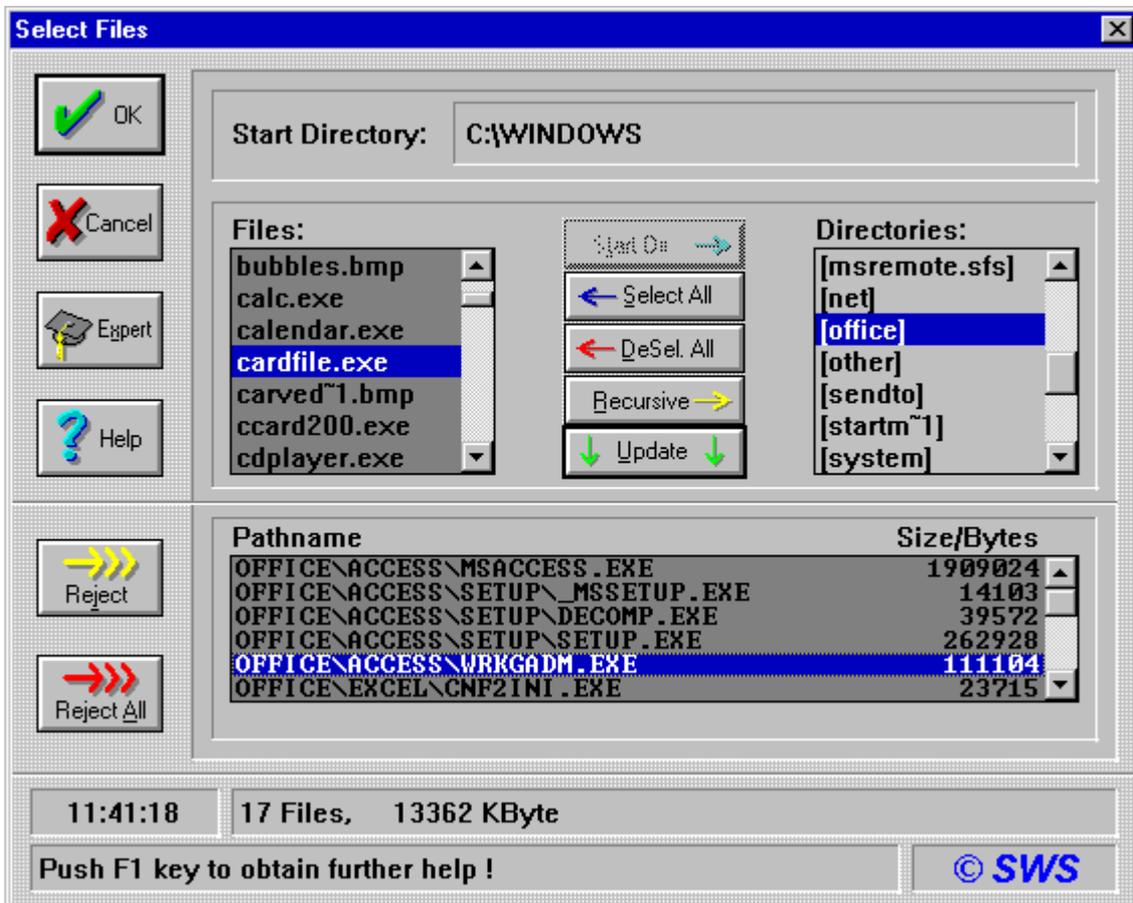
You can unlock your screen by typing in your default DES I password in the text box. This feature protects your computer from attacks by unauthorized persons and little green men and women and vogons and klingons while you are absent from your desk. If you haven't a private DES-I password you can unlock the screen by typing "12345678".

4. Getting Started

Before you can encrypt or delete several files in one step you must collect them in a file list. This section will lead you through the selection process.

4.1. Selecting Files

This dialog box contains three listboxes which are used to select and collect the files that are to be encrypted or deleted. With the help of various buttons you can select individual files or whole directory trees. The selected files are listed in the bottom listbox. Marked files in the other two listboxes can be moved to the bottom window by clicking the **Update** button. After you are done selecting files confirm your selection by clicking the **OK** button.



Start Dir: This button selects the starting directory for the encryption. When you are working with several files in different directories a defined starting point must be set in order to restore the directory structure when the files are decrypted. At first this button is grayed and the current directory is set as the starting directory. This button becomes available when you change to a directory which is higher up in the directory hierarchy ([..]) than the current starting directory or when you change to another drive. **[(ALT-T)]**

Select All: Tags all files in the left listbox. This listbox allows a so called multiple choice selection, this means that you can select files by simply holding down the left mouse button and pulling the mouse cursor downward. If you press the **CTRL** key at the same time you can also select files which do not immediately follow each other. **[(ALT-S)]**

DeSel. All: Untags all files in the left listbox. **[(ALT-D)]**

Recursive: Clicking this button causes the highlighted directory in the right listbox to be tagged. Clicking the **Update** button will copy all files in that directory or those of its sub directories into the bottom listbox in accordance with the file search criteria. **[(ALT-R)]**

Update: This button causes all tagged files to be copied into the bottom listbox. It must be activated again to copy each subsequently tagged file into the bottom listbox. **[(ALT-U)]**

Expert: Opens a new dialog in which you can declare the criteria by which the files will be selected. **[(ALT-X)]**

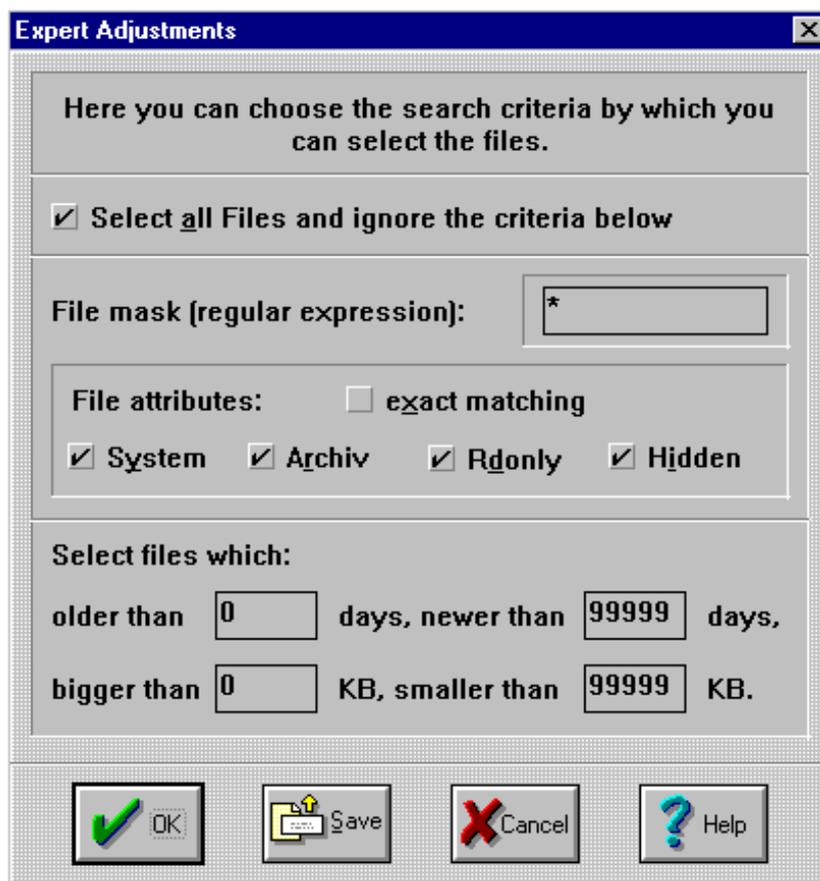
Reject: Removes marked files from the bottom listbox. **[(ALT-J)]**

Reject All: Removes all files from the bottom listbox. **[(ALT-A)]**

Further information about a file can be obtained by double clicking it in the bottom listbox.

4.1.1. Expert Adjustments

In this dialog you can choose the search criteria to select the files which you want to encrypt. If you wish to regularly encrypt your latest files, you would enter a certain date here and all files newer than this date will be selected. Furthermore you can use regular expressions in the search mask to find files in a more professional manner. It is also possible to search for files with a certain size or file attributes. The chosen settings can be saved permanently by clicking the **Save** button. By default all files will be selected.

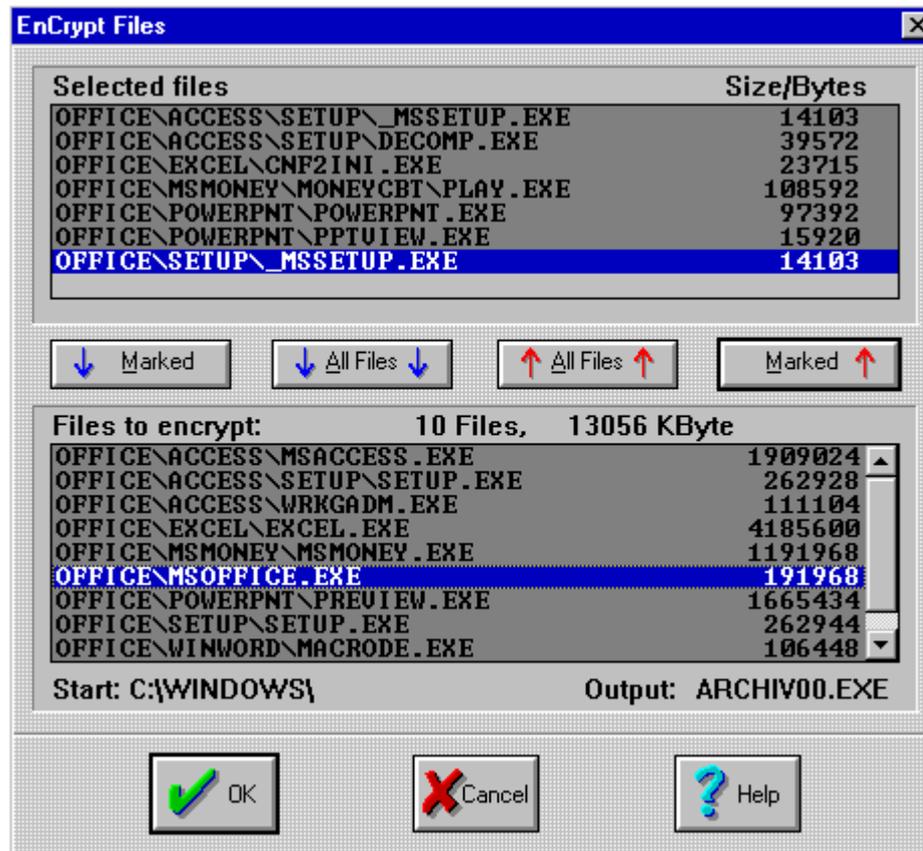


4.2. Encrypting Files

It is possible to either encrypt a single file or several files at once. If several files are to be encrypted they must be marked in the dialog **Select Files**. A permissible selection of files can be recognized by the status of the status element **Selection**. In this case the words **>> Selection List <<** will appear in the text field **Input File**, if only one file is selected this text field will contain the file name. Now you must select the output file format. By default a self-extracting **EXE** file with the output name **ARCHIVxx.EXE** will be created. Use this format to send encrypted files to people who do not have a copy of Enigma. If you want to create an encrypted archive which can be decrypted again directly from Enigma click the **ENIGMA** status button. Then a file with an **EN3** extension will be created.

After selecting the input file(s) and the output file, the encryption algorithm must be chosen. In order to do this click either the status element **XOR**, **DES**, **DES II** or **DES III**, then confirm the choice by clicking the button **EnCrypt** or by choosing the command **EnCrypt File(s)...** in the **Run** menu. If you have selected several files a new dialog will appear which prompts you to confirm your selection.

Note: The status element **Original** must be marked by hand if an encrypted files is to be encrypted again.



Use the 4 buttons in the middle of the dialog box to move the files around between the listboxes. All files shown in the bottom listbox will be encrypted. When you are ready to encrypt click the **OK** button.

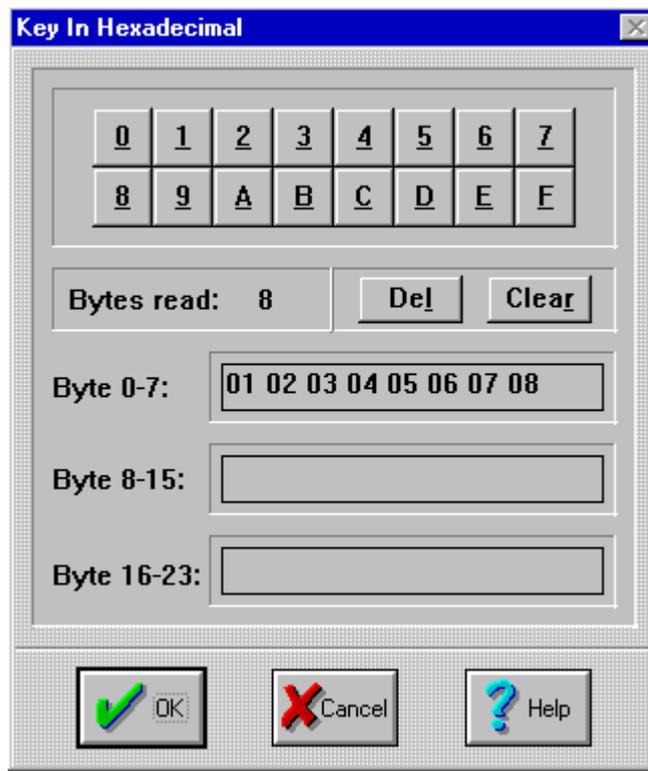
A new dialog will appear in which you can chose in what directory the output file will be copied into. Compare the file size with the directory size in order to ensure that there is enough space to hold the output file.



Before the selected files are encrypted you must enter your personal password. No one can decrypt the encrypted file without knowledge of this password. The password is not shown on the screen when you enter it for protection against unwanted observers. For safety reasons it has to be entered twice (Fields **Password:** and **Confirmation:**).

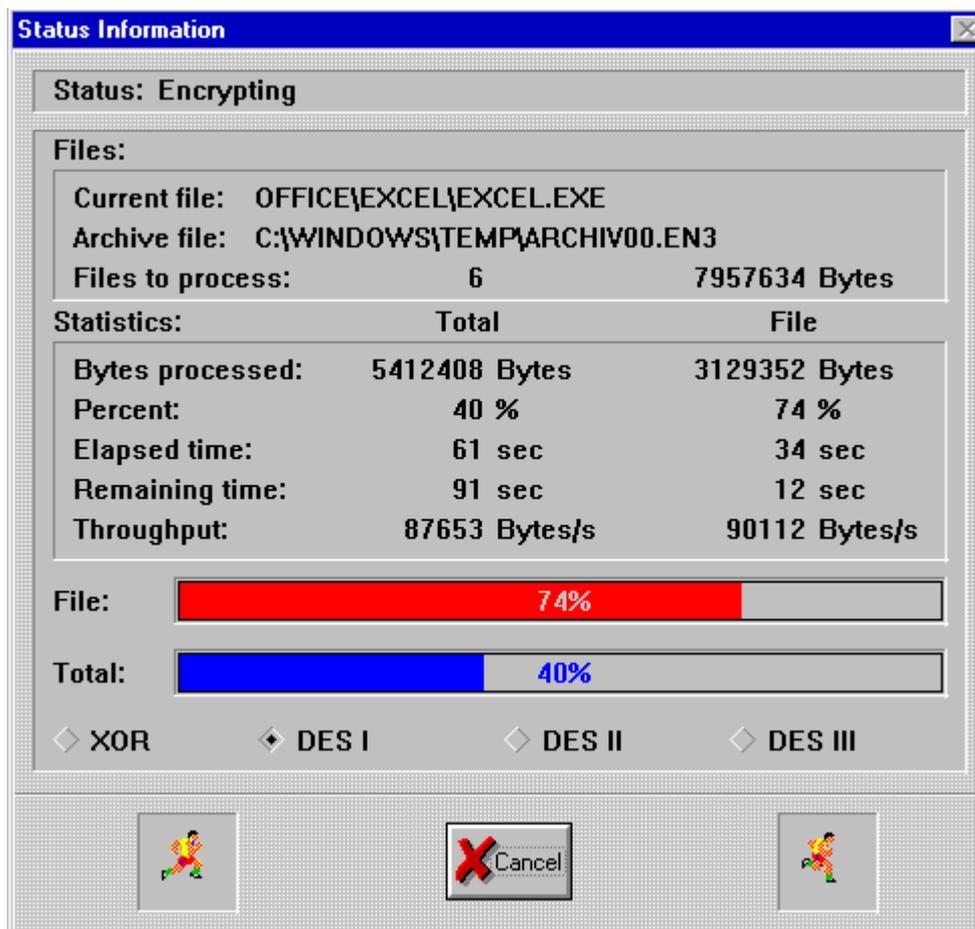


Clicking the **Make Key** switch causes an algorithm specific password to generated by a random character generator, it can be seen in the field **Automatic:**. After clicking the **Hex** button a new dialog appears and you can enter or show your key in hexadecimal notation.



You should write this password down before clicking **OK**. If you click the **Default** button your default user password is used for encryption.

Now the encryption process can be started; a new window will appear which informs you about the encryption process and from here you can interrupt the encryption process at any time.



After successful encryption of the tagged files, you can immediately wipe out these tagged files from your hard disk with **Wipe**.

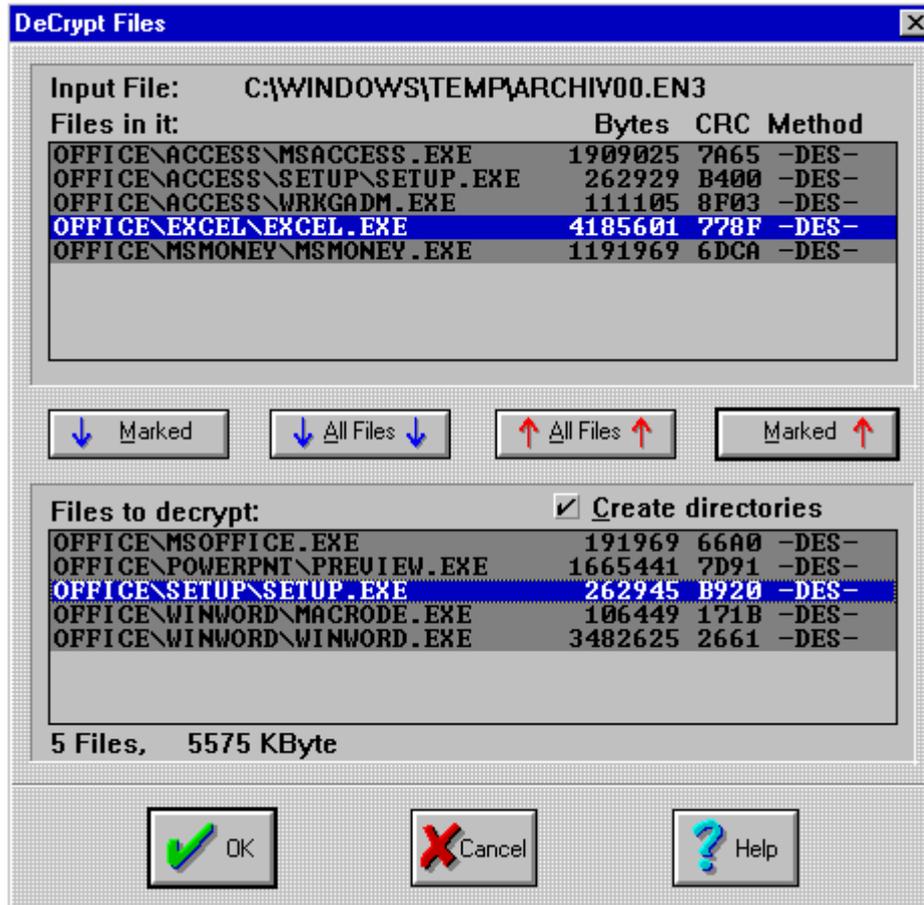


If you have selected an encrypted Enigma compatible file as the output file all files will be appended to it which are not already contained in it. If files have been selected with the same name(s) as those already in the encrypted file the latter will be replaced. If you use a different password from that in the existing encrypted file you must ensure that you use the appropriate password for decrypting each encrypted file. We do NOT recommend this procedure ! Please be careful when wiping out the selected files. Make sure that your output file is not in this list, or else your data would be lost forever.

4.3. Decrypting File

To decrypt a self-extracting EXE file, simply run it from the DOS command line, or pick File/Run in Program Manager, or double click it from File Manager. The program will prompt the user for the password and decrypt itself. To decrypt a file which was encrypted by Enigma to a Enigma compatible output format, tag this file in the left listbox in the main dialog. The status element **Encrypted** will automatically be marked. The file name will appear in the text field **Input File**. After the file has been selected click the switch **DeCrypt** or activate the **DeCrypt File** command in the **Run** menu.

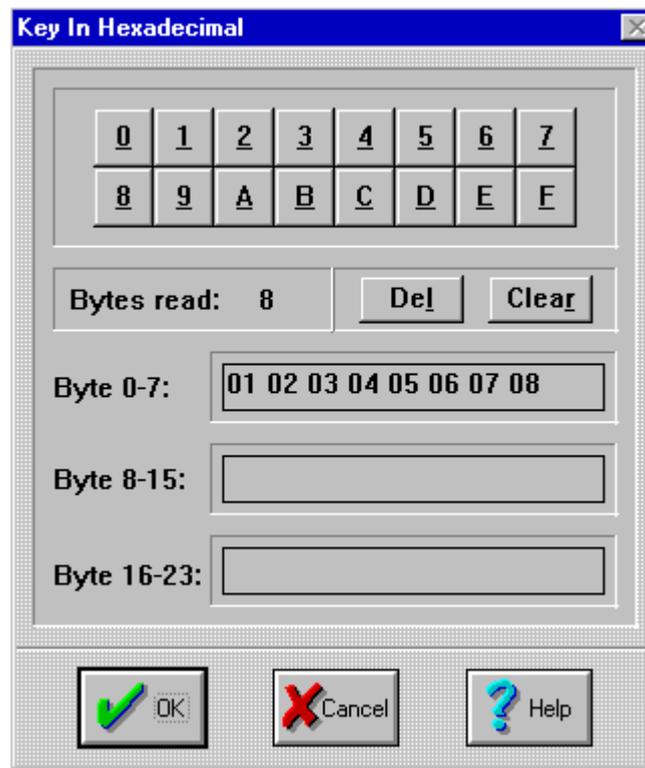
After this a dialog will appear which shows what files are present in the input file. Here it is possible to select the files which should be decrypted. Confirm the selection by clicking **OK**.



Now a new dialog will appear in which you can select in which directory the decrypted files should be copied into. Afterwards a new dialog will prompt you for the password that was used to encrypt the file(s).



By clicking the **Hex** button you are able to enter or show your key in hexadecimal notation.



The decryption process can now be started. Once started a new window will appear which informs you about the decryption process. Here the decryption process can be interrupted at any time.

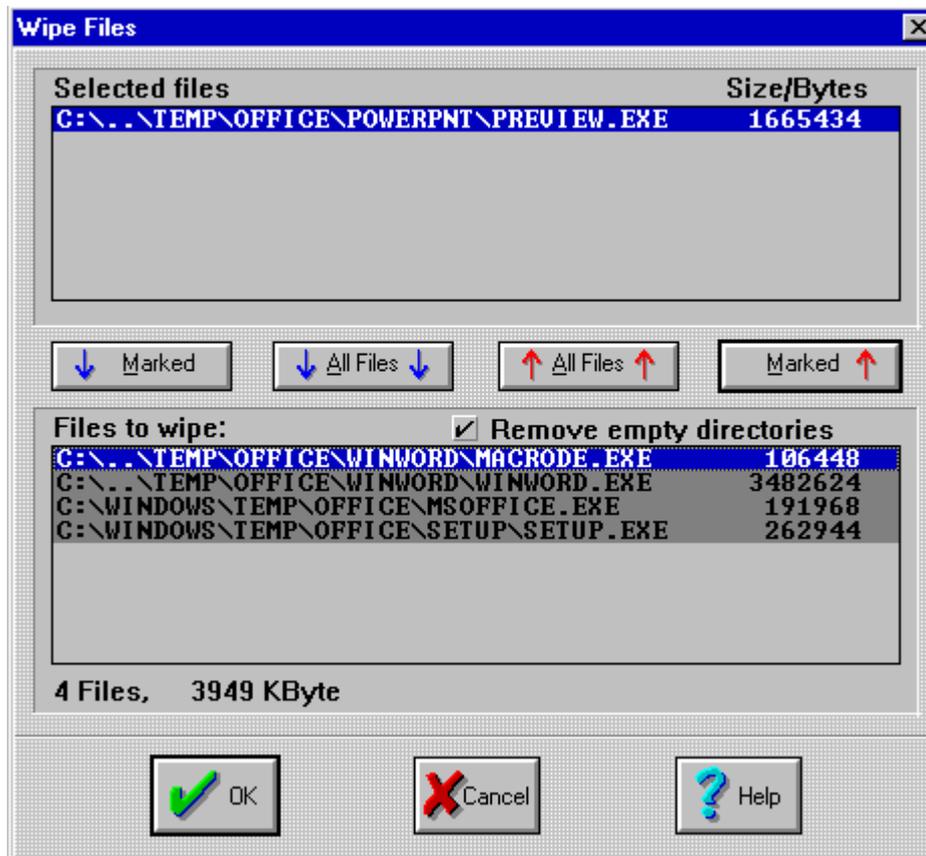


After all the files in your input file have been decrypted make sure that your data has been decrypted correctly before you delete the input file. The program has no way of testing whether the correct password was used to encrypt the file(s) and there is no sure way of testing whether the result is meaningful. If you use the wrong password to decrypt the input file the output file will contain rubbish and you must decrypt the input file again with the correct password.

4.4 Wiping Files

It is possible to delete a single file or several files at once. If several files are to be deleted they must be selected in the dialog **Select Files**. A permissible selection of files can be recognized by the status of the status element **Selection**. In this case the words **>> Selection List <<** will appear in the text field **Input File**, if only one file is selected this text field will contain the file name.

After this is done you can click the **Wipe** button or activate the **Wipe File(s)...** command in the **Run** menu. If you have selected several files a new dialog will appear which prompts you to confirm your selection.



When you are ready to delete click the **OK** button and a new window will appear which informs you about the progress of the deleting operation and here it is possible to interrupt the operation at any time.



After this operation the data is lost for ever, so please be careful when selecting the files which you want to delete.

5. Algorithms

5.1. Data Encryption Standard (DES)

In 1972 the National Bureau of Standards (hereafter: NBS) made a public invitation to tender for the development of a program which would allow files (unclassified computer data) of any type to be encrypted. The low response prompted the NBS to ask the National Security Agency (NSA) for help. Here they had some experience in the development of simple encoding and encryption algorithms. After long discussions the NBS decided to use the Data Encryption Standard (DES) as a standard. The DES had been developed at IBM.

The DES has its roots in an encoding method which was developed in Germany during WW I by the electrical engineer Arthur Scherbius. In the second World War the Germans developed an electromechanical encoding device called **Enigma 4** which was based on the work of Arthur Scherbius. Like the **Enigma 4** the DES uses a series of permutations which for themselves are individually rather simple but when used repeatedly they are extremely complicated. In the **Enigma 4** encoding machine

the permutations are generated by mechanical wheels while in the DES they are produced by program code or by hard wired chips.

DES in ECB mode handles data blocks of 64 bits at one time. DES is basically a bit permutation, substitution, and recombination function performed on blocks of 64 bits of data and 56 bits of key (eight 7-bit characters).

First, the 64 bit input block is subjected to a fixed initial permutation IP and split into two 32 bit blocks L_0 and R_0 . Then each block is scrambled up in 16 iterations. The resulting 32 bit blocks L_{16} and R_{16} are then permuted back to a 64 bit block by a permutation table IP' which is the inverse of IP . The resulting 64 bit encrypted block is then written to the output block. In each iteration I_i , the block L_{i-1} is coupled with the 32 bit output of the function $f(R_{i-1}, k_i)$ by a XOR operation. The iteration I_{16} is an exception, here the blocks are swapped. The function $f()$ receives the block R_{i-1} and the 48 bit output of the function $K(i)$ as its arguments. $f()$ permutes the 32 bits of R_{i-1} into 48 bits by using the permutation table E . The result is exclusive ORed with the 48 bit output from the function $K(i)$. The 48 bit result is then split into eight 6 bit values. Then the function S realizes a 4 bit value for each 6 bit value by a non-linear substitution. The eight 4 bit values are then combined to a 32 bit value, which is then coupled with the permutation table P . The resulting 32 bit is the output of the $f()$. The function S composed of eight substitution modules s_1, s_2, \dots, s_8 (the mysterious S-boxes) which are used on the eight 6 bit values from above. In this 16x4 matrix each of the 64 elements has a value between 0 and 15, a 4 bit value which substitutes a 6 bit value. The matrix co-ordinates of a 6 bit value are obtained in the following manner: bits 1 and 6 as binary give column 0..3, with bits 2 through 5 the row 0..15 is calculated. The function S returns the 4 bit value of the so addressed matrix element. The function $K(i)$ returns the 48 bit value k_i based on the key. There are two further permutation tables for the key. In the first iteration the key is permuted with the first table and then split into two halves. Each of these halves is shifted to the left once ($i = 1, 2, 9, 16$) or twice depending on the iteration number i . Each subsequent iteration I_i after the first uses the shifted value of the preceding iteration as input, shifts the value again and finally permutes it with the second permutation table. The decryption process uses the same algorithm, except that the decryption reverses the half exchanges during the iterations and uses the permuted key values returned by $K(i)$ in the reverse order. In double and triple DES mode the key (16 byte or 24 byte) is split into 8 byte subkeys. Then the described algorithm is executed 2 or 3 times consecutively, each time with an another subkey.

5.1.1. The Security of DES

>> The best that can be expected is that the degree of security be great enough to delay solution by the enemy for such a length of time that when the solution is finally reached, the information thus obtained has lost all its value.
<< William F. Friedman

It is best to trust algorithms because professional cryptographers have scrutinized them for years without cracking them, not because the algorithm's designer makes grandiose claims about its security. In the last 20 years hundreds of experts tried to find an efficient attack to break DES.

In the process it turned out that after a few iteration steps each bit in the output block is dependent upon every bit of the input block and the key. A minimal change in the input block or in the key causes more than half of the bits in the output block to change, this is the so-called avalanche effect. To date there is no better way known to break DES other than to try every possible key; this is called a brute-force attack. This means that theoretically 72 quadrillion (2^{56}) keys have to be tried.

On a Sun SparcStation-2 , a key can be tested in 0.00005 seconds, an average of 20000 keys per second - results in a time of maximal 114168 years to find a key for a given encrypted text. With the help of a custom chip which is able to test a million keys a second it would take about 2284 years to try all possible combinations. 10000 of these chips in a parallel array would get the same result in about 80 days. A test for the plausibility of the decrypted text which has to be done after each test is not included in these calculations.

In August 1993 the Canadian Michael J. Wiener described how to build an exhaustive DES key search machine for \$ 1 million that can find a key in 3.5 hours in average. Each of the used key search frames has the equivalent power of 14 million Sun workstations.

To greatly improve the security of DES the double- and triple DES algorithms have been implemented. With triple DES encryption, each input block is processed three times using independent keys to produce the encrypted block. Almost all attacks concentrate on exhaustive or brute force methods as well as Wiener's approach. In the age of colossally increased computer power and parallel systems these methods have become alarmingly practical - certainly with an unreasonable expenditure for private persons. A fairly painless way to improve security dramatically is to switch to triple-DES - which for the next decades is a sure-fire method to protect against these attacks.

Industry estimates suggest that by 1996 there will be 200M computers in use worldwide. If every one of those computers worked together on a brute-force attack, and each computer performed a million encryption's per second, it would still take a million times the age of the universe to recover a 24 byte triple DES key.

The weakest link in DES are the users themselves whom exchange their passwords or keep their passwords insufficiently secure or choose a poor key.

5.2. XOR

XOR uses a substitution algorithm, this means that every character of the text is coupled with a character in the password by a XOR operation to produce a character in the output file. This means that in contrast to other encoding algorithms where the characters of the password are coupled with the text characters one after another the XOR uses a procedure that randomly selects a character of the password to couple with a character of the text.

The randomizing procedure is dependent upon the length of the password. By filling the output buffer with random numbers the degree of disorder is further increased.

You should thoroughly memorize your password. If a file is accidentally encrypted more than once it can be decrypted by entering the passwords in the opposite order. A text that has been encrypted twice with the same password does NOT yield the original text.

THERE IS NO REAL SECURITY HERE. THIS KIND OF ENCRYPTION IS TRIVIAL TO BREAK BY A CRYPTOANALYST, EVEN WITHOUT COMPUTERS. IT WILL ONLY TAKE A FEW MINUTES WITH A COMPUTER !

5.3. Regular expression

In the **Select Files** dialog a (limited) regular expression can be entered to create a file mask. The following characters have been implemented:

EFW32U.HLP	Enigma for Windows 3.2 help file (English)
EFW32U.WRI	Enigma for Windows 3.2 user's manual
DES3.DLL	16 Bit Triple DES DLL
EFWSFX.DLL	Enigma for Windows 3.2 SFX DLL
EFD.EXE	Enigma compatible DOS utility
INSTALL.EXE	Installation utility
EFW32U.INF	Configuration file for the installation
README.1ST	You should read this first
BWCC.DLL	Borland's Custom Controls DLL

BWCC.DLL is copyrighted (c) by Borland International.