Table of Contents

1 Introduction

1.1 What is SecExFile?

SecExFile is a data archiving and encryption tool to help you protect the integrity of confidential information contained in files and folders on your computer.

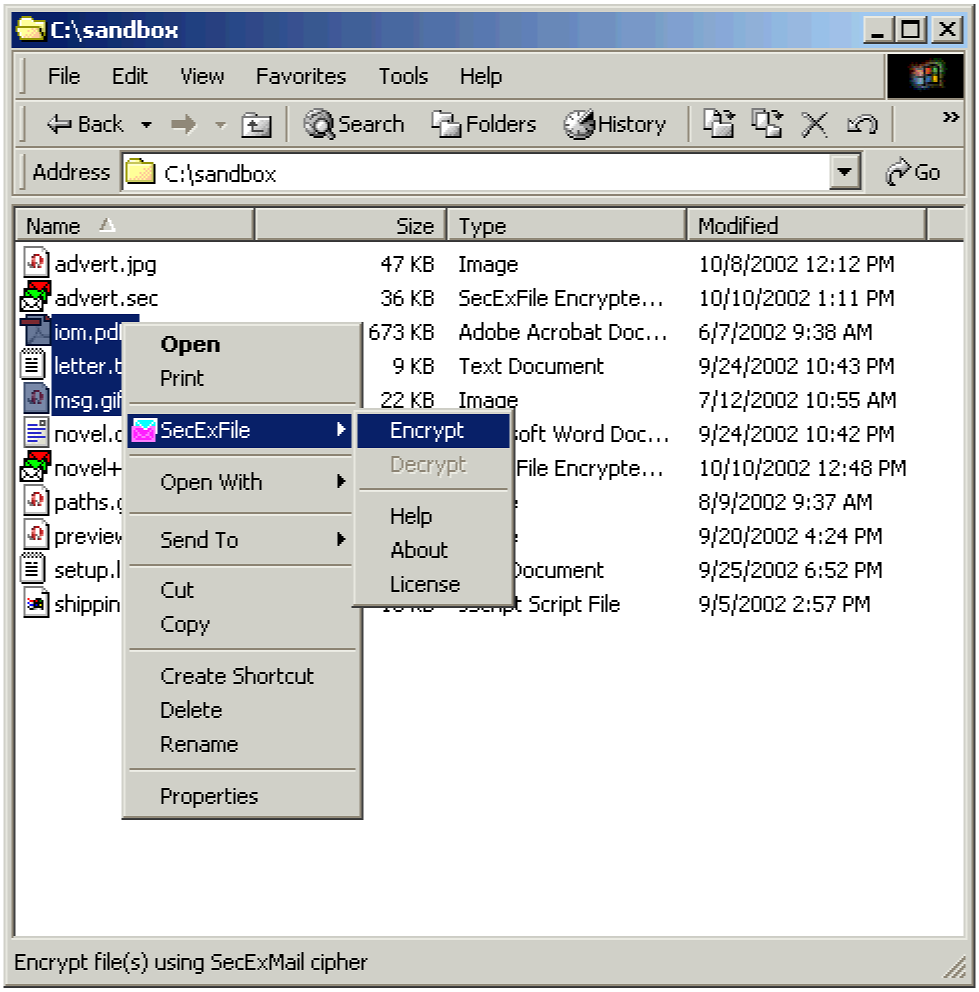
General features :

- Integrates seamlessly with Windows explorer
- Easy to use, password based, intuitive
- Strong 128/384 bit SecExMail cipher, incorporating 256 bit Twofish
- Encrypt multiple files and folders into a single encrypted archive
- Known plain text attack and frequency analysis protection

2 Encrypting Files 1,2,3

2.1 Step 1: Select Files to Encrypt

To create encrypted archives of files or folders on your hard drive, simply right-click on the files or folders you wish to encrypt and select ***SecExFile*** & ***Encrypt*** from the drop down menu as shown below. This will invoke the SecExFile Encrypt Screen. Alternatively, you may invoke ***SecExFile - Encrypt*** from the Window ***START & Programs*** menu.

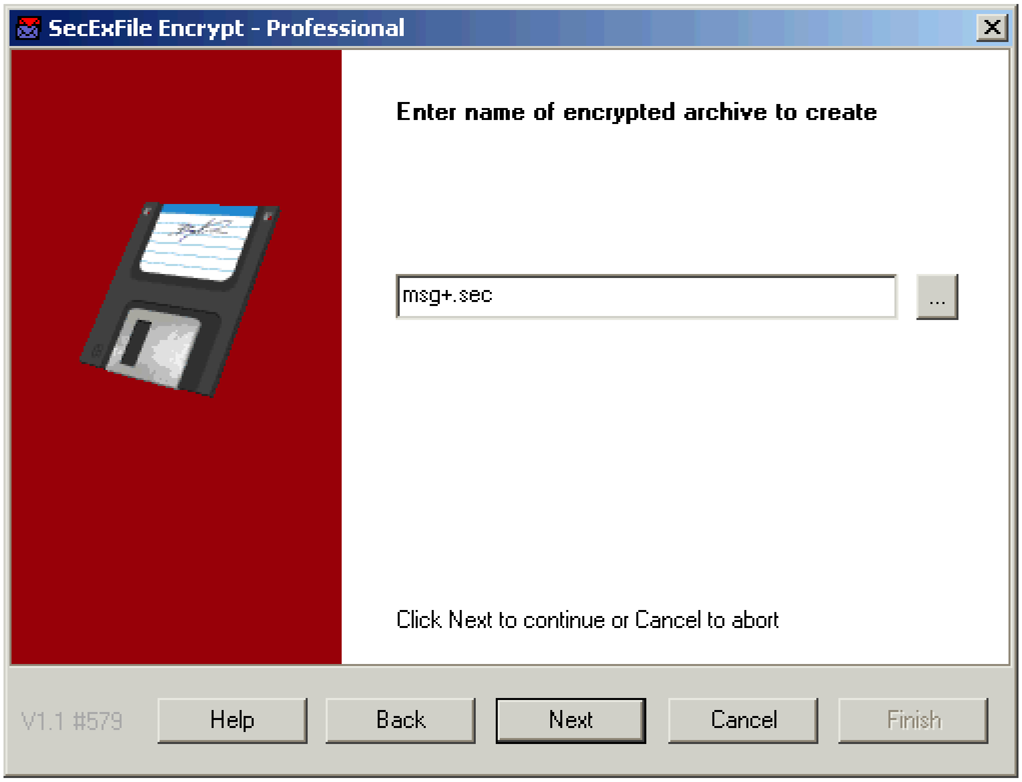


Name	Size	Type	Modified
advert.jpg	47 KB	Image	10/8/2002 12:12 PM
advert.sec	36 KB	SecExFile Encrypte...	10/10/2002 1:11 PM
iom.pd	673 KB	Adobe Acrobat Doc...	6/7/2002 9:38 AM
letter.t	9 KB	Text Document	9/24/2002 10:43 PM
msg.gif	22 KB	Image	7/12/2002 10:55 AM
novel.c		oft Word Doc...	9/24/2002 10:42 PM
novel+		File Encrypte...	10/10/2002 12:48 PM
paths.(			8/9/2002 9:37 AM
previev			9/20/2002 4:24 PM
setup.l		ocument	9/25/2002 6:52 PM
shippin		t Script File	9/5/2002 2:57 PM

Encrypt file(s) using SecExMail cipher

2.2 Step 2: Confirm Name of Archive

This screen lets you confirm or change the name of encrypted archive to be created. By default, SecExCipher files use the extension ".sec". When encrypting multiple files or folders, the name of the first item is chosen as the name of the encrypted archive and a plus sign is appended to the name. Click ***Next*** to continue.



SecExFile Encrypt - Professional

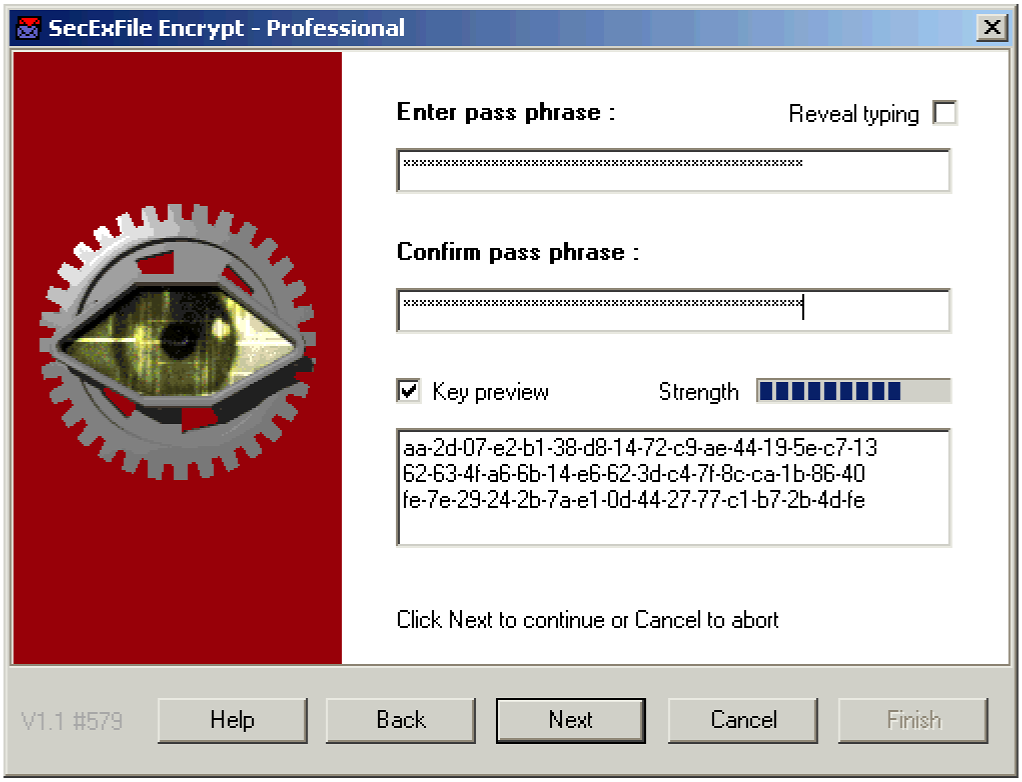
Enter name of encrypted archive to create

msg+.sec

Click Next to continue or Cancel to abort

2.3 Step 3: Define Pass Phrase

This screen lets you chose a passphrase to protect the encrypted archive. The suggested pass phrase length is between 12 and 25 characters for the 128 bit SecExMail cipher version and between 12 and 70 characters for the 384 bit SecExMail cipher version and should contain both letters and numbers. Always remember : pass phrases are case sensitive. Click ***Next*** to continue.



Enter pass phrase : Reveal typing ☐

****

Confirm pass phrase :

****

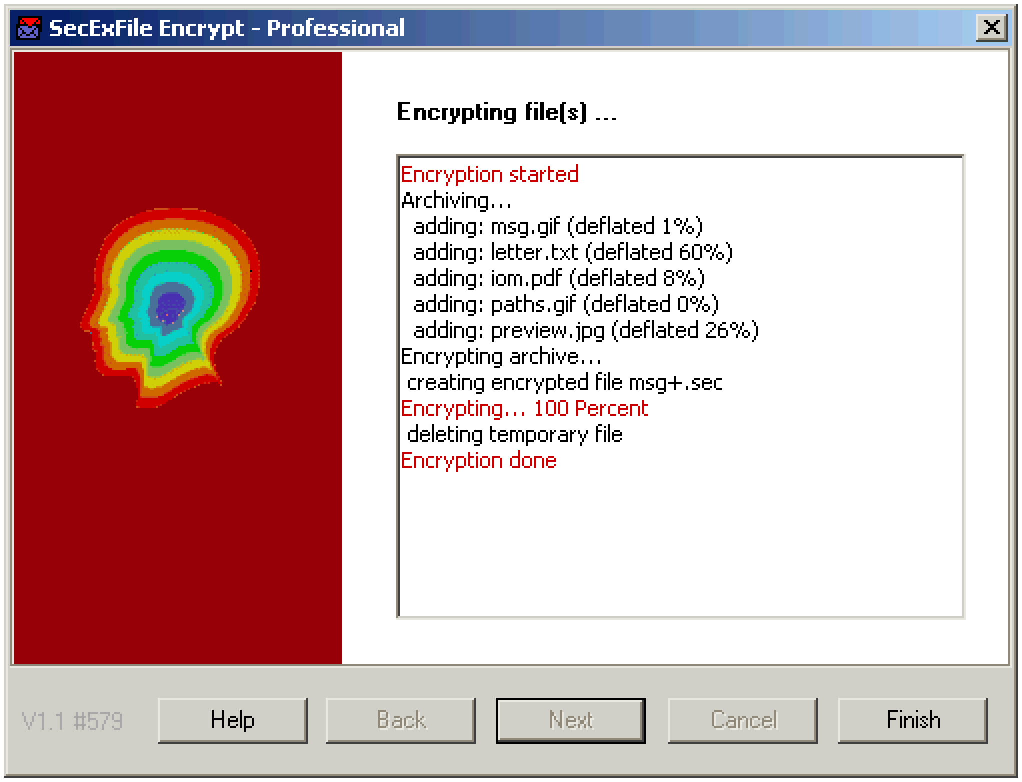
☑ Key preview Strength

aa-2d-07-e2-b1-38-d8-14-72-c9-ae-44-19-5e-c7-13
62-63-4f-a6-6b-14-e6-62-3d-c4-7f-8c-ca-1b-86-40
fe-7e-29-24-2b-7a-e1-0d-44-27-77-c1-b7-2b-4d-fe

Click Next to continue or Cancel to abort

2.4 Summary

The summary screen shows the progress of archiving and encryption in real-time. Click ***Finish***.



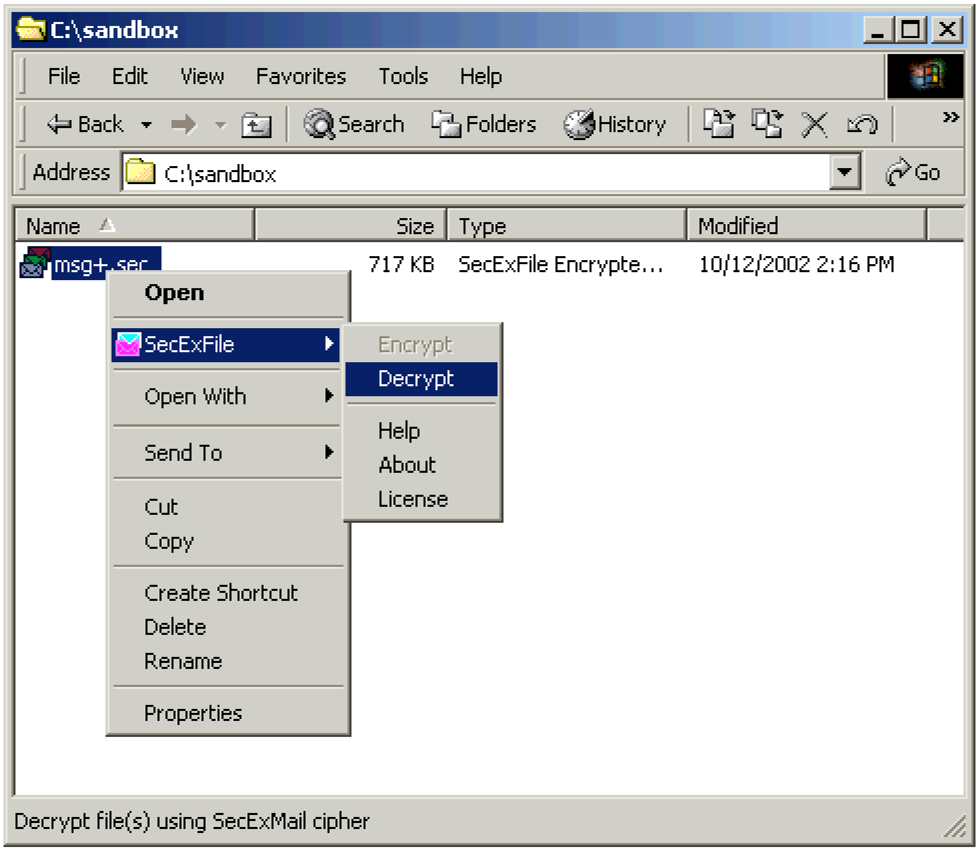
Encrypting file(s) ...

```
Encryption started
Archiving...
  adding: msg.gif (deflated 1%)
  adding: letter.txt (deflated 60%)
  adding: iom.pdf (deflated 8%)
  adding: paths.gif (deflated 0%)
  adding: preview.jpg (deflated 26%)
Encrypting archive...
 creating encrypted file msg+.sec
Encrypting... 100 Percent
 deleting temporary file
Encryption done
```

3 Decrypting Files 1,2,3

3.1 Step 1: Select Archive to Decrypt

To decrypt SecExFile encrypted archives, simply right-click the archive you wish to decrypt in Windows explorer and select ***SecExFile*** & ***Decrypt*** from the drop down menu as shown below. This will invoke the SecExFile Decrypt Screen. Alternatively, you may invoke ***SecExFile*** **-** ***Decrypt*** from the Window ***START & Programs*** menu.

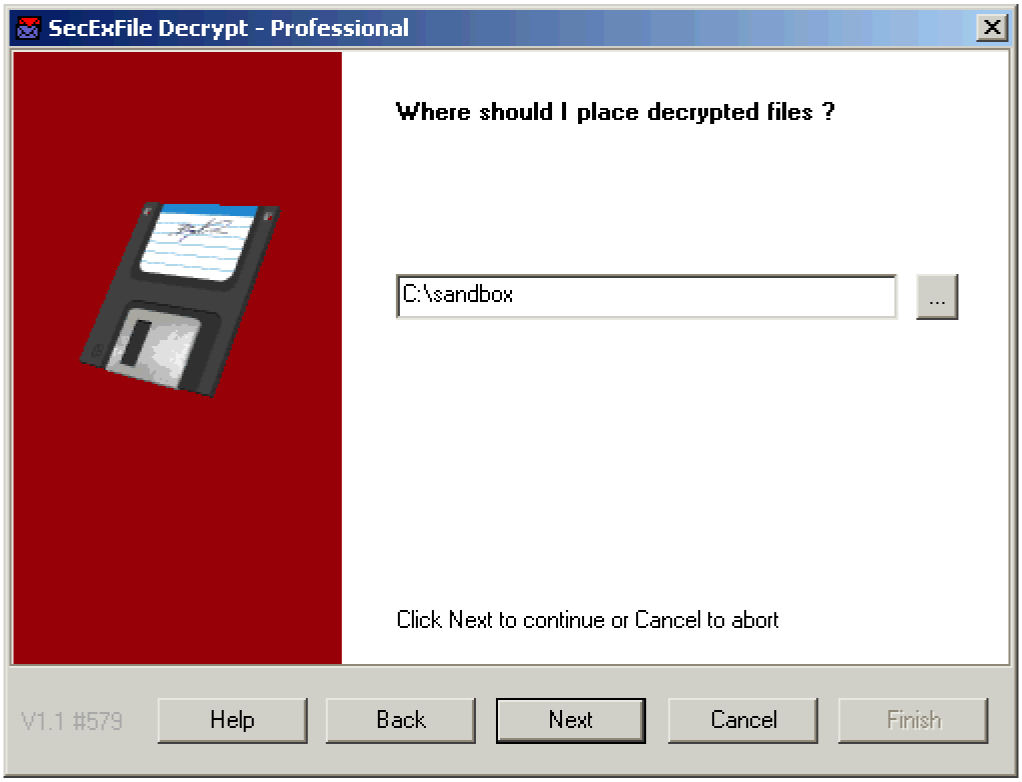


Name	Size	Type	Modified
msg+.sec	717 KB	SecExFile Encrypte...	10/12/2002 2:16 PM

Decrypt file(s) using SecExMail cipher

3.2 Step 2: Confirm Target Folder

This screen lets you select the folder to which decrypted files will be extracted from the SecExFile archive following their successful decryption. Click ***Next*** to continue.



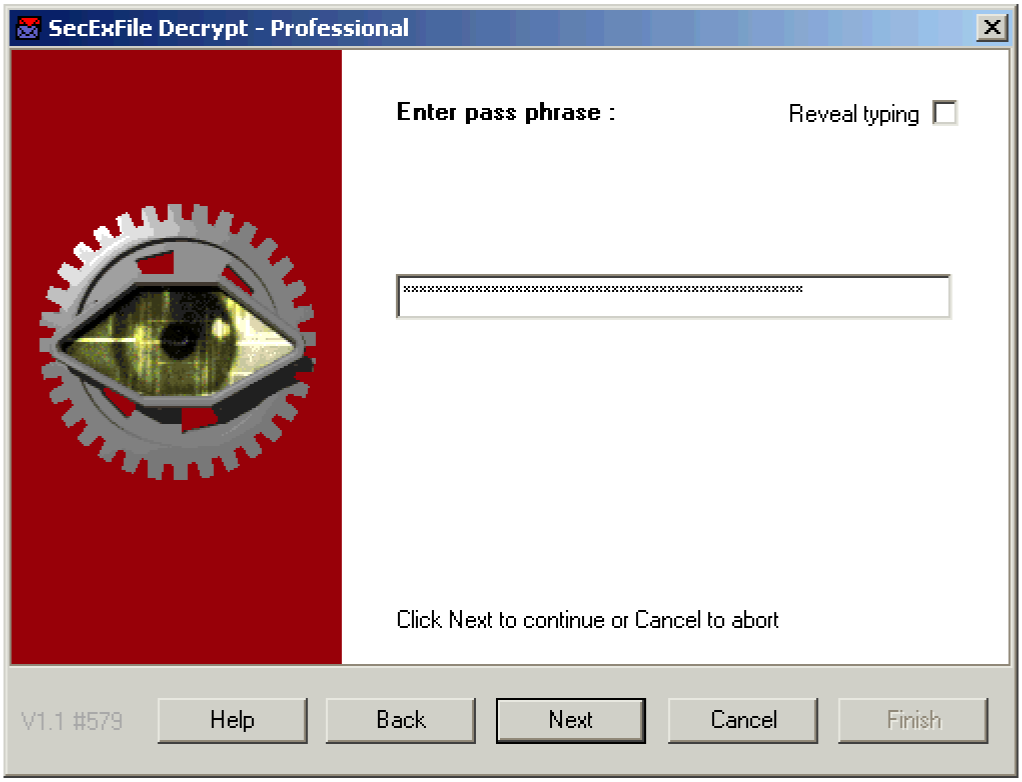
Where should I place decrypted files ?

C:\sandbox

Click Next to continue or Cancel to abort

3.3 Step 3: Enter Pass Phrase

This screen lets you enter the pass phase which unlocks the encrypted archive. Please remember that pass phrases are case sensitive. Click ***Next*** to continue.



SecExFile Decrypt - Professional

Enter pass phrase :

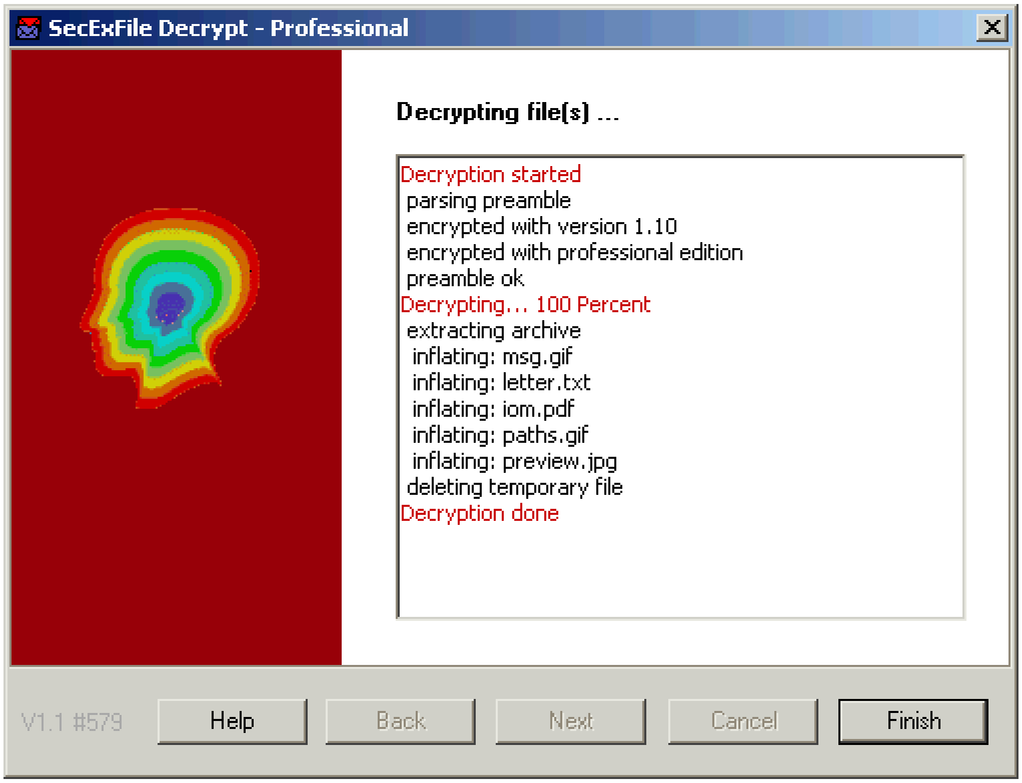
Reveal typing □

```
*****************************************************
```

Click Next to continue or Cancel to abort

3.4 Summary

The summary screen shows the progress of decryption and file extraction in real-time. Click ***Finish***.



SecExFile Decrypt - Professional

Decrypting file(s) ...

```
Decryption started
 parsing preamble
 encrypted with version 1.10
 encrypted with professional edition
 preamble ok
Decrypting... 100 Percent
 extracting archive
  inflating: msg.gif
  inflating: letter.txt
  inflating: iom.pdf
  inflating: paths.gif
  inflating: preview.jpg
 deleting temporary file
Decryption done
```

4 Technical

4.1 Proactive Security

- Pass Phrase Security

Pass words and pass phrases tend to be the weakest elements of secure systems. Unfortunately encryption keys and certificates can be complicated to manage and are impossible to remember, so many people prefer the use of passwords to protect access to computers and files. On the other hand, long pass phrases which contain both letters and numbers can provide good security if they are chosen wisely. Therefore we have tried to mitigate the consequences of poor pass phrase selection and ensure the use of string pass phrases in the design of SecExFile.

The pass phrase provided by the user is never employed directly in encrypting and archiving files. Firstly, the passphrase as entered by the user is stretched to a fixed length using secure hash functions. By default, the passphrase is stretched to 20 characters in the 128 bit SecExCipher version and to 60 characters in the 384 bit SecExCipher version. Please note that when entering the suggested maximum pass phrase lengths of 25 characters for the 128 bit SecExCipher version and 70 characters for the 384 bit SecExCipher version, the pass phrase will actually be shrunk. This excess margin is designed to account for the fact that users are limited to entering printable characters at their keyboards rather than the full set of ASCII characters. The key which is derived from the user passphrase is then used to encrypt the session key which is employed to encrypt the actual data. See Known Plain Text Protection.

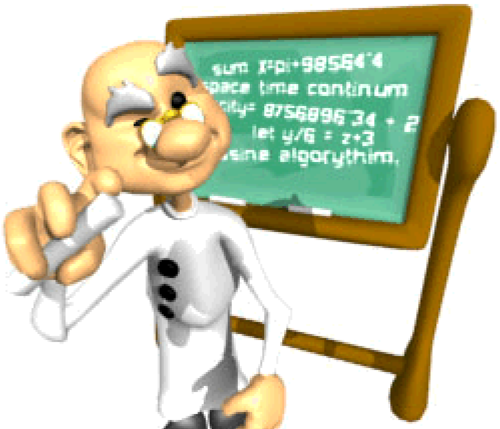
- Known Plain Text Protection

When encrypting files with pass phrases, there is a danger that users will select the same or similar pass phrases to protect multiple archives. A savvy cryptanalyst will be able to exploit this by comparing the cipher text of encrypted archives especially when some or all of the plain text input is known or may be reasonably "suspected". In some cases it may be sufficient to know the format in which the data is stored even if the actual content of the documents is unknown. This is because many file formats employ fixed header sections to store attributes of the document in question. For example, image files may contain palette information in their headers which are identified by specific byte sequences, etc. SecExFile takes comprehensive steps to protect against this kind of cryptanalysis.

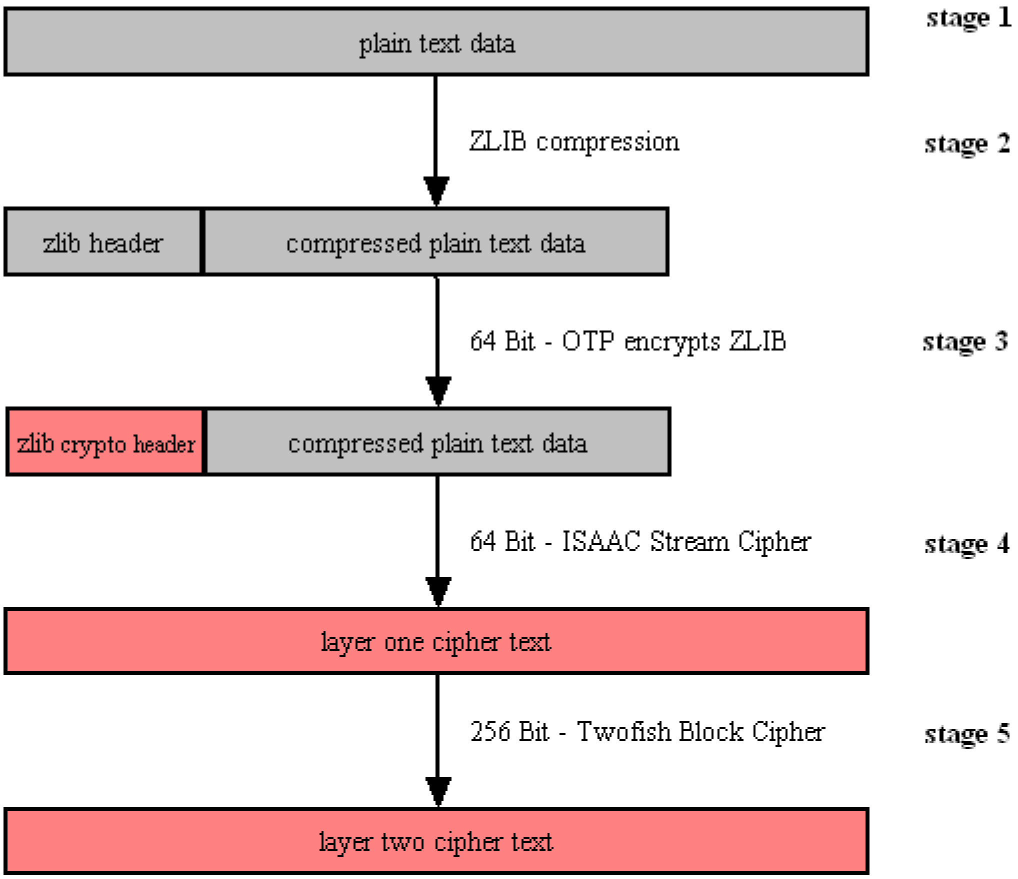
Firstly, even where the user enters the same pass phrase to protect multiple archives, this pass phrase and its derived key material is only used to encrypt the random session key which in turn is employed to encrypt the actual data. This means each file is encrypted with a different key. Further, the SecExMail cipher protects against known plain text attack by compressing all input and obscuring the header of the compression layer via a one time pad.

4.2 The SecExMail Cipher

The SecExMail cipher is a composite cipher originally designed to operate on real-time email streams. It uses cryptographic primitives which are available to the general public and have been subject to extensive peer review. Message encryption is performed via the Twofish block cipher and the ISAAC stream cipher. The SecExMail cipher is warranted to be free from spyware, key escrow or key recovery features of any kind. The email encryption process is described in detail below. See diagram.



SecExMail Composite Cipher



- **Stage 1**

  Email data is received in variable length data blocks. SecExMail parses SMTP header info, mail and data bodies.

- **Stage 2**

  Because email messages frequently contain known plain text, such as salutation and or tag lines, which gives rise to known plain text attacks on the encrypted message and in order to minimize overall message expansion, the plain text is first compressed using the ZLIB compression algorithm. The net effect of deflating large amounts of data, containing both tidbits of known plain text such as greeting or tag lines as well as unknown message text into a compressed data stream is that any known plain text is effectively obscured.

- **Stage 3**

The ZLIB stream has a fixed header format which in itself might be exploited as known plain text by a savvy cryptanalyst. For this reason, the first 64 bits of the steam are enciphered by way of a One Time Pad, using standard XOR masking. This approach acknowledges that email messages will contain portions of known plain text and proactively manages this problem.

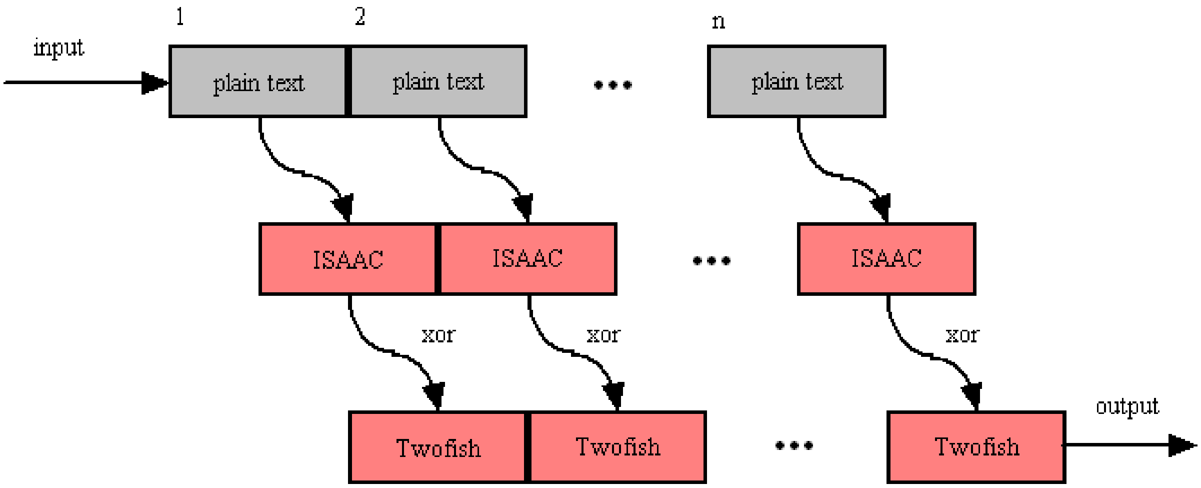
- **Stage 4**

  At this point the compressed data is encoded using the 64 bit ISAAC stream cipher creating the layer one cipher text.

- **Stage 5**

  The next step in the encryption process is to encrypt the layer one cipher text using the 256 bit Twofish block cipher. Twofish is used in chained block mode, but instead of XOR'ing the previous block's cipher text into the plain text of the current block, the output from the ISAAC layer is "chained in". This chaining process is illustrated below.

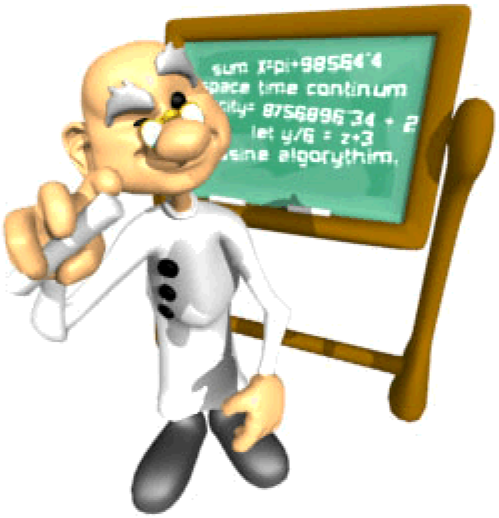
ISAAC Twofish Block Chaining



4.3 Known Plain Text Attack

A known plain text attack is the attempt by a cryptanalyst to break a cipher based on knowledge about the plain text of a message prior to its encryption. Simply put, if the cryptanalyst knows the method of encryption, any encryption, part or all of the plain text input to the cipher, and is able to observe the encrypted message text, he / she will likely be able to infer the key used to encrypt the message. This in turn can compromise the security of future messages sent with that key. In greatly simplified terms :

Plain Text + Key = Cipher Text
Cipher Text - Plain Text = Key

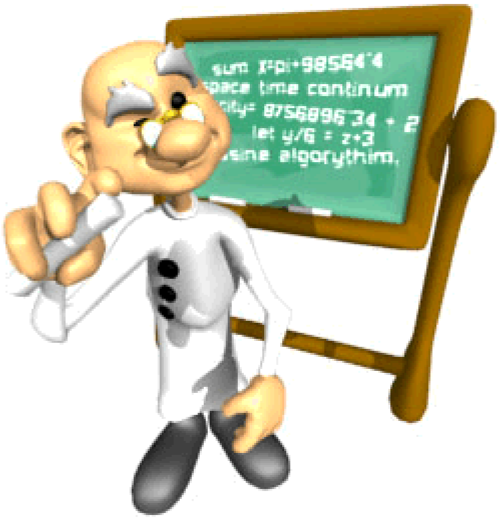


SecExFile includes comprehensive protection against known plain text attacks.

4.4 One-Time Pads

A one-time pad is a block of random data used to encrypt a block of equal length plain text data. Encryption is usually by way of XOR'ing the one-time pad with the message text. This process may be thought of as a 100% noise source used to mask the message. The one-time pad is secure if it is comprised of random data and is never reused. Because of this, one-time pads have limited application in modern ciphers, but are commonly acknowledged as the holy grail of cryptography.

SecExMail uses one-time pads to encrypt the ZLIB compression header in SecExMail messages.

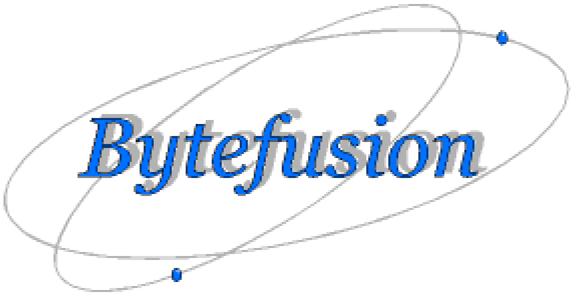


5 About

5.1 About SecExFile

SecExFile
Version 1.1
Copyright © 2002, Bytefusion Ltd.
All Rights Reserved

5.2 About Bytefusion Ltd.



Bytefusion Ltd.
22 Duke Street
Douglas, IOM
IM1 2AY
British Isles

Inquiries: sales@bytefusion.com