CenoCipher 4.0

CenoCipher is a free, open-source, easy-to-use tool for exchanging secure encrypted communications over the internet. It uses strong cryptography to convert messages and files into encrypted *cipher-data*, which can then be sent to the recipient via regular email or any other channel available, such as instant messaging or shared cloud storage.

CenoCipher utilizes *symmetric* encryption involving pass-phrases. Communicating parties need to agree on a pass-phrase ahead of time which will be used to protect their communications, and must establish this pass-phrase by a secure means first (such as meeting in person) before using it within CenoCipher to correspond.

CenoCipher is an *end-to-end* encryption tool. It does not perform any direct network communication itself, or depend on any specific intermediary service provider or transmission conduit. Instead it works in conjunction with any available communication conduit of choice.

To compose and send an outgoing message

Step 1. Output format

Choose the preferred format for outputting the encrypted *cipher-data* once it is ready:

- **Normal File**: The encrypted data will be created as a generic computer file, to be sent as an email attachment or by any other method. The name for this file can be generated automatically at random, or with a custom name as specified.

- **Modified Jpeg:** The encrypted data will be embedded within a Jpeg image file of choice, using a technique known as ***steganography,*** which subtly modifies the Jpeg image in a way that isn't noticable to anyone else, but can be decoded by a recipient in possession of the correct passphrase. A specific image can be chosen for use, or a folder can be specified from which a suitable image will be selected at random. *See additional notes on steganography further below.*

- **Text:** The encrypted data will be delivered as a stream of garbled text (base64 ascii), to be sent as the main body of an email, instant message, or in any other suitable container.

Step 2. Output destination

Choose the desired destination for the *cipher-data* once it is ready:

- **Screen:** The cipher-data will be placed on screen for easy drag-drop operations. Normal files and Modified Jpegs will appear in a dedicated area near the bottom of the program interface, while text will appear in the main message box.

- **Clipboard:** The cipher-data will be copied onto the system clipboard for easy paste operations elsewhere.

- **Folder:** The cipher-data (file or jpeg) will be created in a specifically chosen folder for later usage. This can be useful when planning to share or upload the file by using another program's 'Browse/Locate' dialog.

Step 3. Message text

Enter the desired message text into the large main message box in the center of the program interface.

Step 4. File attachments

Attach any number of files you may wish to include by adding them to the Files box immediately to the right of the Message box. You can do this either by clicking on the "Add Files" button below the Files box, or by dragging and dropping files directly into the box from wherever you like.

Step 5. Passphrase

Type a passphrase of your choice into the Passphrase box below the main Message box. This should be a shared secret phrase known to both you and the intended recipient, but not to anyone else. It will be used along with multiple cryptographic algorithims to create the encrypted cipher-data, so that only you and the recipient are able to decrypt it.

Step 6. Encrypt

Once you have typed your chosen passphrase, click the Encrypt button immediately below the Passphrase box to begin the encryption process. Alternatively you can simply hit Enter on the keyboard if you prefer. Depending on the size of the file attachments you may have opted to include, encryption may occur instananeously or take a few seconds to complete. The initially-grey notification bar immediately below the Message box will inform you when the process is finished.

~

Once the encrypted cipher-data is ready, you can send it to the intended recipient by whatever method is most convenient, depending on the output format and destination you have chosen. Regular email, instant messaging, shared cloud storage, or physical transfer via USB flash drive are just a few of the many possible options.

Note that it is not possible to actually transmit the data using CenoCipher alone, as the program performs no direct network activities whatsoever. It must therefore be used in conjunction with another normal communication channel as described above.

~

Additional notes on steganography

Those utilizing the option to conceal encrypted data within Jpeg images should keep the following in mind:

- The original image is never altered. Instead a new copy is created and the encrypted data is embedded in this copy.

- Storage capacity of Jpegs for holding hidden data is quite small, typically only 1-2% of total Jpeg file size. Thus the method is more suitable for transmitting text than other files.

- Images from a public source like the internet should not be used for holding hidden data, since comparing the public original and the modified copy would allow a third party to observe a difference and become suspicious. Instead, use a unique private image not accessible elsewhere.

- **Although using steganography can help to provide some degree of extra camouflage against an observer who is not looking too closely, it may still be possible for someone examining the image carefully to detect unusual statistical anomalies which can suggest deliberate modification. It is therefore not advisable to depend completely upon this technique for security, but instead to consider it only an additional layer of obfuscation.**

To receive and decrypt an incoming message

Step 1. Load the cipher-data

Cipher-data received from another party can be quickly loaded into the program by any of three convenient methods:

- **Drag-and-drop** the data from elsewhere directly into the main Message box of the CenoCipher proram interface. This can be done with either a file or a block of text.

- **Load clipboard contents** if you have already *copied* the cipher-data to the system clipboard from elsewhere. This can be done with a copied file or block of text.

- **Load file from folder** to browse and locate the cipher-file if you have previously saved it somewhere on a local or network device.

Step 2. Enter passphrase

Once the cipher-data has been successfully loaded into the program as described above, you will be prompted to enter the passphrase and proceed with decryption. Type the passphrase (which the sender will need to have communicated to you by some other means) into the passphrase box just below the main Message box.

Step 3. Decrypt

Click the button immediately below the Passphrase box to proceed with decryption. Alternatively, you can simply hit the Enter key instead. Depending on the size of the cipher-data received, decryption may occur instantaneously or take several seconds to complete. The initially-grey notification bar immediately below the Message box will inform you when the process is finished.

If the passphrase entered is correct and decryption is successful, the contents of the original message will be displayed in the main Message box of the program interface. If any files were included by the sender along with the message, they will appear in the "Files" box to the right of the Message box.

Auto-Decrypt on load: If this option is enabled, the program will attempt to immediately decrypt any cipher-data as soon as it is loaded, without waiting for the user to click Decrypt. This only occurs if a passphrase is already present in the passphrase box. This can be useful when exchanging an ongoing series of messages with the same party during a given session, for more quick and convenient operation.

Other options

Save program settings: Enable this option if you wish remember and restore the program's configuration on next launch. All of the on-screen selections and entries will be preserved, with the exception of message text, passphrase and file attachments. Note that enabling this feature will cause a small config file to be saved on the host computer.

On decrypted disk-write attempt: Normally CenoCipher keeps decrypted data in temporary memory only, erasing it once it is no longer needed. However there are two scenarios in which it may be necessary to write un-encrypted data to disk.

> *Huge data volume:* If the received cipher-data is so gigantic in size (multiple gigabytes) that its decrypted content cannot be stored entirely in RAM, it must be written to a file instead in order to proceed.

Opening file attachments: If you receive some cipher-data which upon decryption turns out to have some file attachments contained within it, these will normally still only be held within temporary memory and not written to disk (except in huge data situation mentioned above). However in the event you choose to open/view those attached files, they must be written to disk first so that the operating system can load them into the associated software handler (text editor, picture viewer etc).

In either of the two events described above, CenoCipher will proceed as directed according to the following available options:

Prompt: The program will ask in each encountered instance how to proceed.

Use system default location: The decrypted data will be written to a standard temporary storage location on the host computer, typically within the 'Application Data' folder.

Use specified location: The decrypted data will be written to a custom location indicated by the user for this purpose.

Normally all data written to the temporary system default location by CenoCipher is wiped and deleted when the program closes, as long as it is not in use (open in another program) at the time. Data saved or written to custom locations is not automatically deleted.

Technical specifications

CenoCipher is fully open-source, written in C++, and employs the following well-established cryptographic algorithms and hashing functions:

- AES/Rijndael, Twofish and Serpent ciphers (256-bit keysize variants), cascaded together in CTR mode for triple-encryption of messages and files
- HMAC-SHA-256 for construction of message authentication code
- PBKDF2-HMAC-SHA256 for derivation of separate AES, Twofish and Serpent keys from user-chosen passphrase
- Cryptographically safe pseudo-random number generator ISAAC for production of Initialization Vectors (AES/Twofish/Serpent) and Salts (PBKDF2)

License

CenoCipher is free software, and is released under the GNU General Public License, version 2 or any later version, as published by the Free Software Foundation, Inc.