Ladhe's Encryption Utility (LE)

Experience the next generation of data security with **Ladhe's Encryption Utility (LE)**—a cutting-edge, **quantum-safe** encryption tool designed for Mac terminals. Built with advanced features and unparalleled encryption power, LE ensures your sensitive data is protected from modern and future threats.

Key Features:

1. **Symmetric Key Generation**: Effortlessly generate secure keys for symmetric encryption, similar to AES. (Only in Licensed version)
2. **Asymmetric Key Generation**: Create powerful encryption and decryption keys, comparable to RSA. (Only in Licensed version)
3. **Versatile Encryption Options**: Encrypt plaintext, pipelined text from files, or entire files seamlessly. (Only in Licensed version)
4. **Flexible Decryption**: Decrypt ciphertexts created using symmetric or asymmetric keys with ease. (Only in Licensed version)
5. **File/Folder Encrypt/Decrypt**: Protect your file/folder with robust encryption/ decryption. For folder you can use **recursive** option.
6. **Force/Clean Mode**: With force mode it will overwrite encrypted file if exists. Clean will move .leXXX file to original file on successful decryption.
7. **Password-Protected Encryption**: Encrypt files and safeguard them with a password.
8. **Time-Lock Encryption**: Secure files with encryption that follows specific time constraints:
    - Decrypt only **before a certain date**.
    - Decrypt only **after a certain date**.
    - Decrypt **within a specified date range**.
9. **Password Protection**: Encrypted files require the original password for decryption, ensuring confidentiality.
10. **New File Extensions**: Depending on type of file extension (XXX), the new encrypted files will have leXXX extension.

    For e.g. If you encrypt Test.txt with LE, the encrypted file will be Test.letxt

11. **File Info Insights**: Retrieve metadata about encrypted files using the `Get Info` feature.
12. **Integrated Comments**: LE encryption automatically adds a comment attribute detailing the encryption, visible in file properties.
13. **Comment Cleanup**: Upon decryption, LE removes the extended comment for a seamless experience.
14. **Date Tampering**: In licensed version, online date is checked against system date.
15. **File Size**: In beta, we support up to 4 MB, in licensed version, we support up to 100 MB.

16. **File Extensions**: As of now we support these extensions, if you need support for new type, please let us know at spalgorithm@gmail.com

    “java,txt,rtf,c,vb,cs,csv,log,asp,class, m,h,php,aspx,js,html,htm,js,xhtml,storyboard,json, xml,py,sql,swift,xib”

Why Choose LE?

LE is currently in Beta and represents the forefront of post-quantum cryptography, making it the ideal solution for safeguarding your sensitive data in the quantum computing era. If you find it meets your needs and expectations, you can request a licensed copy to continue benefiting from its powerful features.

Enjoy the confidence of knowing your data is secure with **Ladhe’s Encryption Utility (LE)**—the quantum-safe tool to keep your information safe, today and tomorrow!

The LE command on terminal:

```
OVERVIEW: LE encrypts or decrypts data from plaintext or file, it works with pipeline. If you want to get plaintext or ciphertext from file.

© Ladhe 2025! This is in BETA, if you find it meets your needs and expectations, you can request a licensed copy to continue benefiting from its powerful features

USAGE: le <options>

ARGUMENTS:
  <strings>               -

OPTIONS:
  -h, --force             Override encrypted file
  -z, --clean             Remove encrypted/decrypted file
  -s, --setup             Setup LE
  -n, --recursive         Recursive/nested for folder
  -e, --encrypt-file <encrypt-file>
                          name of file/folder to encrypt(the command only writes to current directory
                          ./LE -e file/folder --force --clean
                           ./LE -e file/folder -w pass.txt.
                          Please try with -w password file name. Use -n for recursive
  -d, --decrypt-file <decrypt-file>
                          name of file/folder to decrypt(the command only writes to current directory)
                          ./LE -d file/folder --clean
                           ./LE -d file/folder -w actual password or password file. For Password encrypted files -w password. Use -n for recursive
  -i, --get-info-file <get-info-file>
                          Get info on encrypted file/folder with future date.
                          ./LE -i file/folder -n for recursive
  -w, --pass-word <pass-word>
                          Password

                          ./LE -f file/folder -w Password
  -t, --encrypt-till-date <encrypt-till-date>
                          Date till you want to allow to decrypt file/folder. Works with only -e
                          ./LE -e file/folder -t encryptTillDate e.g. ./LE -e a.txt -t '2024/12/13 14:22' for 2.22 PM , for 2.22 AM -> '2024/12/13 2:22'
  -l, --decrypt-from-date <decrypt-from-date>
                          Date from to allow to decrypt file/folder. Works only with -f
                          ./LE -e file/folder -l decryptFromDate   e.g. ./LE -e a.txt -l '2024/12/13 14:22' for 2.22 PM , for 2.22 AM -> '2024/12/13 2:22'
  -r, --decrypt-to-date <decrypt-to-date>
                          To Date range to allow to decrypt file/folder.Works only with -e
                          ./LE -f file/folder -l decryptFromDate -r decryptToDate e.g. ./LE -e a.txt -l '2024/12/13 14:22' for 2.22 PM , for 2.22 AM -> '2024/12/13 2:22' -r  '2024/12/13 14:22' for 2.22
                          PM , for 2.22 AM -> '2024/12/13 2:22'
  -k, --encrypt_type <encrypt_type>
                          Type of Encryption -- asymmetrical (a) or symmetrical (s)(Not available in Beta)
  -f, --ladhe-pubkey-en <ladhe-pubkey-en>
                          ./LE -f Key (Not available in Beta)
  -u, --ladhe-pubkey-de <ladhe-pubkey-de>
                          ./LE -u Key (Not available in Beta)
  -a, --decrypt_-key <decrypt_-key>
                          ./LE -u Key -a Decrypt-Key (Not available in Beta)
  -p, --plain-text <plain-text>
                          PlainText

                          ./LE -f Key -p PlainText (Not available in Beta)
  -o, --output-file <output-file>
                          name of output file(the command only writes to current directory)
  --version               Show the version.
```