# POLYALPHABETIC CIPHER PROGRAM
## User's Manual

24 December 2021

Lucid Technologies
http://www.lucidtechnologies.info/
Email: info@lucidtechnologies.info

## LEGAL STUFF

The information in this manual has been carefully checked and is believed to be accurate. However, Lucid Technologies makes no warranty for the use of its products and assumes no responsibility for any errors which may appear in this document. Lucid Technologies reserves the right to make changes in the products contained in this manual to improve design or performance and to supply the best possible product.

Lucid Technologies assumes no liability arising out of the application or use of any product. Lucid Technologies' products should NEVER be used in, or to support, safety critical applications such as medical, industry automation, automotive, or transport.

Polyalphabetic Cipher Software by Lucid Technologies is distributed as FREEWARE. FREEWARE is covered by copyright and subject to the conditions defined by the holder of the copyright. Lucid Technologies retains the Copyright for Polyalphabetic Cipher Software. Users may not modify the software or sell copies to others. FREEWARE software may not be modified or extended and then sold as COMMERCIAL or SHAREWARE software.

## CONTENTS

## 1.0 Introduction

Lucid Technologies' Polyalphabetic Cipher Software was inspired by a reading of the novel *800 Leagues on the Amazon* by Jules Verne. The cipher, and the breaking thereof, are a major theme and source of drama in the novel.

## 2.0 System Requirements

Lucid Technologies' Polyalphabetic Cipher Software was written for Windows systems. It has been tested on WindowsXP and Windows10. It is distributed as a ZIP archive. Create a new folder under Program Files, such as *C:\Program Files\Polyalphabetic Cipher*, and extract the contents of the ZIP archive to this directory. Click on *Cipher tool 2021.10.18.exe* to run the program. The program occupies less than 1 MB of hard disk space.

## 3.0 Polyalphabetic Ciphers

A polyalphabetic cipher is a form of substitution cipher. A simple substitution cipher is illustrated by the code wheel shown in Figure 1. An 'F' in the text (outer ring) would be encoded as an 'A' in the coded message (inner ring). A 'G' would be replaced by 'B', a 'H' by 'C', etc. If we number the letters in the alphabet from 1 to 27 it is easy to see that what this code does is take the numeric value for each letter and subtract five. Note the code wheel is set to CODE 'E' which if the fifth letter of the alphabet. 'F' is the sixth letter of the alphabet, thus 6-5 = 1, or 'A'. Wrap around is easily handled by the code wheel; in a computer program it is done via modular arithmetic. Thus 'A', letter 1, is encoded as 1-5 = -4 where Mod(26) of -4 equals 22, or the letter 'V'. Decoding is accomplished by the reverse process, adding five to each encoded letter.


Figure 1. Simple substitution code.

This simple substitution cipher is not very secure - there are only 26 possibilities to test and, because any letter in the plain text is always encoded to the same letter, it is particularly subject to "frequency analysis". See the references in Appendix A or search the web for "polyalphabetic substitution cipher" for further information.

There are several ways to make a substitution more secure. The first is to use more than a single substitution key. In the code wheel example above, the key was 'E' or -5. We could use a longer key, like ECHO (-5, -3, -8, -15) and repeat it until the plain text message ends. This means a plain text character would not always be represented by the same encoded character. The second way is to use a long key so the encoding pattern repeats fewer times. If the plain text is 100 characters and the key is 5 characters, it would take 20 repetitions of the key to encode the plain text. But if the key were 25 characters, it would take only 4 repetitions of the key to encode the plain text. The third way is to allow more characters in plain text and key. Note the code wheel in Figure 1 does not work for punctuation, lower case or spaces; this hinders the understandability of the decoded message. Expanding the list of allowable characters to include all printable ASCII characters (95 total) greatly increases the randomness of the encoded message, thus decreasing the chances of breaking the code.

Lucid Technologies' Polyalphabetic Cipher Software uses all these methods. The Plain Text, Cipher Text and Cipher Key may each be up to 2000 characters in length. The Plain Text and Cipher Key may include any of the ASCII printable character; see Appendix B.

## 4.0 Program Operation

Click on the executable file (*.exe) to start the program. Two windows will open, the Control window (Figure 2) and the Data window (Figure 3).

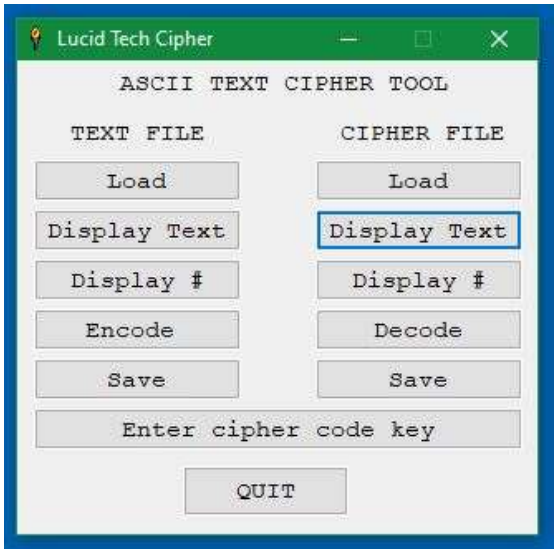


Figure 2. Control window.


Figure 3. Data window.

Initially only the Load, Enter cipher code key, and QUIT buttons will be active; the others will be grayed out until required data is loaded.

## 4.1 TEXT FILE options

TEXT FILE refers to the file held in the TEXT FILE buffer. This will either be a file loaded with the TEXT FILE/Load option or a decoded file resulting from the CIPHER FILE/Decode option.

### 4.1.1 Load

Before loading a text message, it must be created with an editor that can store ASCII text files as [filename].TXT. Clear text messages can't be created in the Polyalphabetic Cipher program. The text message can be no more than 2000 characters and composed of printable ASCII characters. Clicking on the Load button will allow you to select your text message file via a standard file selection window. While loading, the program will ignore non-printable ASCII characters in the file. A Load will overwrite any existing data in the TEXT FILE buffer.

### 4.1.2 Display Text

This option will display, in the data window, the ASCII text of the data in the TEXT FILE buffer.

### 4.1.3 Display #

This option will display, in the data window, the decimal ASCII codes for the data in the TEXT FILE buffer.

### 4.1.4 Encode

This option will use the previously entered cipher CODE KEY to encode (encrypt) the data in the TEXT FILE buffer and store it in the CIPHER FILE buffer.

#### 4.1.5 Save

Clicking on the Save button will allow you to save the data in the TEXT FILE buffer via a standard file selection window. This option is most useful for storing a decoded (unencrypted) file.

### 4.2 CIPHER FILE options

CIPHER FILE refers to the file held in the CIPHER FILE buffer. This will either be an encoded file loaded with the CIPHER FILE/Load option or an encoded file resulting from the TEXT FILE/Encode option.

#### 4.2.1 Load

Cipher files can be loaded from local storage. Cipher files are ASCII text and may be viewed with a file editor - but should not be altered. Cipher files can be no more than 2000 characters. Clicking on the Load button will allow you to select your cipher file via a standard file selection window. A Load will overwrite any existing data in the CIPHER FILE buffer.

#### 4.2.2 Display Text

This option will display, in the data window, the ASCII text of the data in the CIPHER FILE buffer.

#### 4.2.3 Display #

This option will display, in the data window, the decimal ASCII codes for the data in the CIPHER FILE buffer.

#### 4.2.4 Decode

This option will use the previously entered cipher CODE KEY to decode (unencrypt) the data in the CIPHER FILE buffer and store it in the TEXT FILE buffer.

#### 4.2.5 Save

Clicking on the Save button will allow you to save the cipher file via a standard file selection window. Cipher files should be stored with a 'PAC' file extension. This option is most useful for storing an encoded (encrypted) message for future reference or to send to a friend as an email attachment.

### 4.3 Enter cipher code key

The cipher CODE KEY is entered in the data window. The following prompt will appear in the data window - “Enter the Cipher Key String: “. Place your cursor after this prompt and type in your cipher CODE KEY. The cipher CODE KEY may composed of as many as 2000 printable ASCII characters. After it is entered, the program will display the length of the cipher key. The cipher CODE KEY may be used to encode the TEXT FILE or decode the CIPHER FILE; it may be shorter or longer than either of these files.

## 5.0 Bugs, Suggestions and Donations

The most current version of Polyalphabetic Cipher Software should always be available at www.lucid.technologies.info. If you discover any bugs or have suggestions for improving the program please send them to info@lucidtechnologies.info.

If you find the software useful and would like to send a donation to Lucid Technologies you can do so via PayPal (http://www.lucidtechnologies.info/pay_card.htm).

## APPENDIX A

## References

Frederick Gass, Solving a Jules Verne Cryptogram, Mathematics Magazine, Vol. 59, No. 1 (Feb., 1986), pp. 3-11 (9 pages).
https://www.jstor.org/stable/2690010?refreqid=excelsior%3A4e970e712f75b4c3a98d6d3b942d8cf6

Klaus Pommerening, Commentary on the cryptologic episode in Jules Verne: La Jangada (Eight Hundred Leagues on the Amazon), Johannes Gutenburg University
https://www.staff.uni-mainz.de/pommeren/Kryptologie/Klassisch/0_Unterhaltung/Jangada/Comment_J.html

Polyalphabetic Substitution Ciphers (March 18, 2004), Cornell University.
http://pi.math.cornell.edu/~mec/2003-2004/cryptography/polyalpha/polyalpha.html

https://people.wou.edu/~beaverc/440/W15/3%20Polyalphabetic%20ciphers.pdf

## APPENDIX B

Printable ASCII characters

| Dec | Hex | Char | Description |
|---|---|---|---|
| 32 | 20 | space | Space |
| 33 | 21 | ! | Exclamation mark |
| 34 | 22 | " | Double quote |
| 35 | 23 | # | Number |
| 36 | 24 | $ | Dollar sign |
| 37 | 25 | % | Percent |
| 38 | 26 | & | Ampersand |
| 39 | 27 | ' | Apostrophe |
| 40 | 28 | ( | Left parenthesis |
| 41 | 29 | ) | Right parenthesis |
| 42 | 2A | * | Asterisk |
| 43 | 2B | + | Plus sign |
| 44 | 2C | , | Comma |
| 45 | 2D | - | Minus sign |
| 46 | 2E | . | Period |
| 47 | 2F | / | Forward Slash |
| 48 | 30 | 0 | Zero |
| 49 | 31 | 1 | One |
| 50 | 32 | 2 | Two |
| 51 | 33 | 3 | Three |
| 52 | 34 | 4 | Four |
| 53 | 35 | 5 | Five |
| 54 | 36 | 6 | Six |
| 55 | 37 | 7 | Seven |
| 56 | 38 | 8 | Eight |
| 57 | 39 | 9 | Nine |
| 58 | 3A | : | Colon |
| 59 | 3B | ; | Semicolon |
| 60 | 3C | < | Less than |
| 61 | 3D | = | Equality sign |
| 62 | 3E | > | Greater than |
| 63 | 3F | ? | Question mark |
| 64 | 40 | @ | At sign |
| 65 | 41 | A | Capital A |
| 66 | 42 | B | Capital B |
| 67 | 43 | C | Capital C |
| 68 | 44 | D | Capital D |
| 69 | 45 | E | Capital E |
| 70 | 46 | F | Capital F |
| 71 | 47 | G | Capital G |
| 72 | 48 | H | Capital H |
| 73 | 49 | I | Capital I |
| 74 | 4A | J | Capital J |
| 75 | 4B | K | Capital K |
| 76 | 4C | L | Capital L |
| 77 | 4D | M | Capital M |
| 78 | 4E | N | Capital N |

| 79 | 4F | O | Capital O |
|---|---|---|---|
| 80 | 50 | P | Capital P |
| 81 | 51 | Q | Capital Q |
| 82 | 52 | R | Capital R |
| 83 | 53 | S | Capital S |
| 84 | 54 | T | Capital T |
| 85 | 55 | U | Capital U |
| 86 | 56 | V | Capital V |
| 87 | 57 | W | Capital W |
| 88 | 58 | X | Capital X |
| 89 | 59 | Y | Capital Y |
| 90 | 5A | Z | Capital Z |
| 91 | 5B | [ | Left square bracket |
| 92 | 5C | \ | Backslash |
| 93 | 5D | ] | Right square bracket |
| 94 | 5E | ^ | Caret / circumflex |
| 95 | 5F | _ | Underscore |
| 96 | 60 | ` | Grave / accent |
| 97 | 61 | a | Small a |
| 98 | 62 | b | Small b |
| 99 | 63 | c | Small c |
| 100 | 64 | d | Small d |
| 101 | 65 | e | Small e |
| 102 | 66 | f | Small f |
| 103 | 67 | g | Small g |
| 104 | 68 | h | Small h |
| 105 | 69 | I | Small I |
| 106 | 6A | j | Small j |
| 107 | 6B | k | Small k |
| 108 | 6C | l | Small l |
| 109 | 6D | m | Small m |
| 110 | 6E | n | Small n |
| 111 | 6F | o | Small o |
| 112 | 70 | p | Small p |
| 113 | 71 | q | Small q |
| 114 | 72 | r | Small r |
| 115 | 73 | s | Small s |
| 116 | 74 | t | Small t |
| 117 | 75 | u | Small u |
| 118 | 76 | v | Small v |
| 119 | 77 | w | Small w |
| 120 | 78 | x | Small x |
| 121 | 79 | y | Small y |
| 122 | 7A | z | Small z |
| 123 | 7B | { | Left curly bracket |
| 124 | 7C | \| | Vertical bar |
| 125 | 7D | } | Right curly bracket |
| 126 | 7E | ~ | Tilde |