Methods of encryption using Enigma cubes

august 2021

Stefan Berinde (sberinde@math.ubbcluj.ro)

1. Introduction

This material has educational purposes only. It demonstrates various concepts of classic cryptography using one of the most popular toys around, the *Rubik's Cube*.

Rubik's Cube and twisty puzzles in general are mechanical puzzles that exploit their 3-dimensional structure in order to generate a large number of distinct scrambled states. By applying specific image layouts onto these cubes, this number can be converted into a large number of letter substitutions. Moreover, these substitutions can be easily changed in a complex manner by performing specific cube rotations.

Every device that is able to generate letter substitutions can be used as a cryptographic device. One classic example is the historic *Enigma machine*.

Before going further, let's get acquainted with some general terminology.

2. Terminology

cryptography = the science of writing secret messages that are processed in some manner to make them difficult or impossible for unauthorized persons to read; *classic cryptography* depends on a secret key, and this must be securely exchanged in advance between communicants; *modern cryptography* does not rely on secret key exchange.

encryption = the process of converting the original message to a new unreadable message.

decryption = the opposite of encryption.

plaintext = unencrypted (or decrypted) message text.

ciphertext = encrypted message text.

alphabet = the set of characters used in both plaintext and ciphertext (e.g. the 26 English letters).

cipher = a pair of algorithms that create the encryption and the decryption, based on a (secret) key.

key = a piece of information that controls (initializes) the cipher method, the only thing that should be kept secret.

self-reciprocal cipher = a cipher that uses the same algorithm to decrypt a message as the one used to encrypt it.

substitution cipher = a cipher that replaces characters of the alphabet with a permutation of itself; if the cipher replaces characters of the plaintext with a permutation of itself, it is called a *transposition cipher*.

monoalphabetic substitution cipher = a substitution cipher based on a one-to-one letter substitution for the entire message.

polyalphabetic substitution cipher = a substitution cipher based on multiple substitutions (one for each successive plaintext letter, or group of letters).

repeated key cipher = a cipher that uses a piece of information repeatedly, usually a fixed key, in order to perform polyalphabetic substitution; it is also known in literature as *periodic key cipher*, because the generated substitutions are periodic; if the period length is considerably larger than the whole plaintext, the cipher is named *quasi-nonperiodic key cipher*.

autokey cipher = a cipher that uses plaintext letters in order to perform polyalphabetic substitution; these substitutions are truly non-periodic.

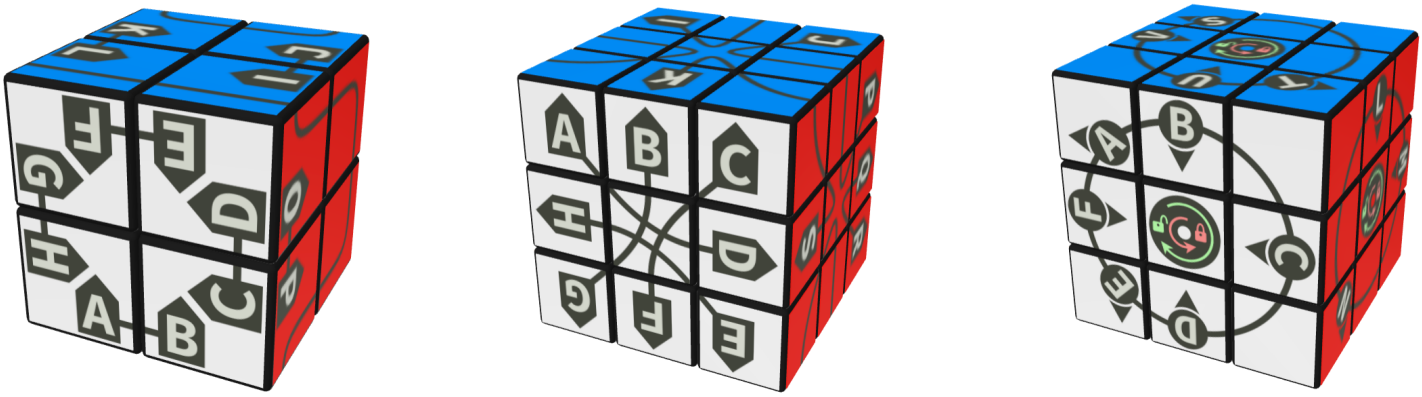
cryptographic device = a physical device that implements a cipher.

Enigma machine = a cryptographic device in the form of an electromechanical rotor machine, implementing a polyalphabetic substitution cipher with quasi-nonperiodic key.

cryptanalysis = the science/art of deciphering encrypted messages by unauthorized persons, by exploiting the weaknesses of the system.

3. Enigma cubes

We introduce here 3 cryptographic devices: *Enigma Pocket Cube*, *Enigma Rubik's Cube* and *Non-reciprocal Enigma Rubik's Cube* (see figure).



All these cubes have 26 letters spread over the faces and various paths used to create a correspondence between a plaintext and a ciphertext character; in the case of Non-reciprocal Enigma Rubik's Cube, this correspondence is not reciprocal and should be performed in the manner described by the padlock pictograms shown on the cube's centers.

Each letter has an associated orientation given by the tip of the surrounding pentagon, respectively, by an arrow nearby. At various stages during encryption/decryption such a letter, called the *reference letter*, is used to set up an overall orientation of the entire cube, and also to perform one or more layer rotations, called the *incremental scramble*.

We define the *default incremental scramble* to be a quarter turn rotation of the layer containing the reference letter, upwards, in the direction indicated by its orientation. We denote this particular scramble with the symbol ^.

Please visit cubes web pages for more details

Enigma Pocket Cube
http://www.randelshofer.ch/rubik/virtual_cubes/pocket/picture_cubes/2x_enigma_cube.html

Enigma Rubik's Cube
http://www.randelshofer.ch/rubik/virtual_cubes/rubik/picture_cubes/enigma_cube.html

Non-reciprocal Enigma Rubik's Cube
http://www.randelshofer.ch/rubik/virtual_cubes/rubik/picture_cubes/enigma_cube_nr.html

These cubes are not (yet) implemented in physical form, but they can be played in a virtual environment at the links above.

4. Methods of encryption

We describe here several methods of encryption, in order, from less secure to more secure. Less secure ones are also easier to implement and less prone to human errors. Every method applies to every Enigma cube.

Always start from the cube's solved state, then use the secret key in order to scramble the cube to a specific state. This key could be conceived as a cube formula (algorithm) in standard notation, for example $RU^2D'BD'$. We just have to establish the front face (F) to be white and up face (U) blue, and the rest will follow. The final cube state generates a monoalphabetic substitution that can be used for encryption/decryption of the entire message. This is fast but least secure, especially for longer messages, since analysis of letters' frequency can reveal the substitution.

Traditionally in plaintext and ciphertext the spaces and punctuation are discarded. Messages are shown in groups of 5 letters, like `ENIGM ACUBE S`.

4.1. Better use words than formulas

Since we have letters all over the cube, a natural way to describe the key is by using a string of letters, particularly a word. Apply the key, that is, choose sequentially each letter in the key to be the reference letter and perform the default incremental scramble (^) with it. The effect is more chaotic than using formulas, since letters are further displaced during scramble.

Let's introduce *method 1* (named monoalphabetic) with an example, using Enigma Pocket Cube.

---

key: UNKNOWN
ciphertext: **QYRVU FHYNV GRKWU GLZNU GLZNU KQNEQ FAWLC TLAQL FOVNY NAFKJ NWWTZ VUKFA W**

---

method: monoalphabetic
device: Enigma Pocket Cube
incremental scramble: ^

From the solved state apply the key by performing a sequence of 7 layer rotations, using default incremental scramble for each letter. This is equivalent with performing the following moves in standard cube notation BUB'RU'D'F. Now we can read the correspondence between letters, which is reciprocal, therefore we list just half of them `A-N, B-X, C-Q, D-J, E-L, F-O, G-H, I-K, M-Z, P-V, R-Y, S-W, T-U`. The ciphertext decodes into **CRYPT OGRAP HYIST HEMAT HEMAT ICALC ONSEQ UENCE OFPAR ANOID ASSUM PTION S**, which reads "Cryptography is the mathematical consequence of paranoid assumptions".

4.2. Encrypt stronger: change substitution

Polyalphabetic substitution is more secure, but it needs a way to specify *when* and *how* substitutions are changed during message encryption/decryption. We can think of *group substitution* (when the substitution is changed after each 5-letters group), or *letter substitution* (when the substitution is changed after each letter). First one has the advantage of being faster, but we have to keep in mind that each group is monoalphabetically encrypted. This might reveal a partial correspondence among letters, especially when using self-reciprocal devices, like Enigma Pocket Cube or Enigma Rubik's Cube.

How to change substitutions? By using a second key. We call it *permutation key*. An interesting way to specify the permutation key is to send it along the message. It could be the first group of 5 letters in the plaintext. Of course, this is encoded with the secret key, so it is also part of the ciphertext. After decoding it, the permutation key is applied at once, repeatedly, after each group of 5 letters. The nice thing about the permutation key is that the sender can choose it at random, making the polyalphabetic substitution more unpredictable. Better than random is to choose the key such that the generated cube moves do not cancel out, nor act on the same layer. This could happen mainly with Enigma Pocket Cube, where letters are clustered together.

By terminology, this is a polyalphabetic substitution with repeated key. But even if permutation key is repeated, the sequence of cube moves is not, at least for a while. The actual periodicity of the substitutions might be very long, the permutation key being in fact quasi-nonperiodic.

We should mention that at some point in history the Enigma machine was used with a second key of 3 letters (called indicator), sent twice for redundancy at the beginning of each message. The operator used this key to make further settings to the machine, before decoding the actual message.

Let's describe now *method 2* (named polyalphabetic group substitution with repeated key) with an example.

---

key: PETERNEUMANN
ciphertext: **YDICV HKSQT DUHZY IVFAD PRVGA OKYLX DTCLH ALWUG AGRHO HYETV VHOLQ HVCWV KHDKB CCZJH UGRCT OBRTQ KXE**

---

method: polyalphabetic group substitution with repeated key
device: Non-reciprocal Enigma Rubik's Cube
incremental scramble: ^

From the solved state apply first the secret key as in method 1. Look now at the first group of letters and mark the correspondences: `Y-N, D-O, I-B, C-L, V-E`. Thus, the permutation key is `NOBLE`. In the current state of the cube apply this key at once and start decoding the next group of letters, `HKSQT-IFYOU`. Apply the permutation key again and continue as before with the next group of letters `DUHZY-THINK`, and so on. We get the rest of them

```
IVFAD-CRYPT, PRVGA-OGRAP, OKYLX-HYIST, DTCLH-HEANS, ALWUG-WERTO, AGRHO-YOURP,
HYETV-ROBLE, VHOLQ-MTHEN, HVCWV-YOUDO, KHDKB-NTKNO, CCZJH-WWHAT, UGRCT-YOURP,
OBRTQ-ROBLE, KXE-MIS.
```

The ciphertext decodes into **IFYOU THINK CRYPT OGRAP HYIST HEANS WERTO YOURP ROBLE MTHEN YOUDO NTKNO WWHAT YOURP ROBLE MIS**, which reads "If you think cryptography is the answer to your problem, then you don't know what your problem is" (Peter G. Neumann).

4.3. Be more involved: forget group substitutions

To avoid monoalphabetic substitutions entirely we switch now to letter substitutions. The permutation key is provided as before, but it is encrypted/decrypted and used differently, that is, letter by letter. After the secret key is applied to the cube, start encrypt/decrypt the first letter of the permutation key on the spot. Next, perform an incremental scramble with this decrypted letter, then encrypt/decrypt the second one. Continue in this way until the entire 5-letters permutation key is encrypted/decrypted. Don't forget at the end to apply the incremental scramble associated with the last letter in the decrypted permutation key.

Now encrypt/decrypt the first letter in the second group of letters. Use again the first letter in the permutation key and perform the required incremental scramble. Encrypt/decrypt the second letter in the group, then use the second letter in the permutation key and decode the next letter. Continue in this way by reusing the permutation key letters cyclically, as long as necessary. This method is more involved, but offers better security. The main drawback is that someone must keep track of which letter in the permutation key follows.

Let's describe *method 3* (named polyalphabetic letter substitution with repeated key) with an example.

---

key: DAVIDKAHN
ciphertext: **OFRXK FSBUT SJPPS MBTEX LIGCA GCVKE UQWIZ IHQVA HGCZP HDIKC OKYVF XDZWT BKVIX LGRTA AFHBT RQXMB SFQZI JJC**

---

method: polyalphabetic letter substitution with repeated key
device: Enigma Rubik's Cube
incremental scramble: ^

From the solved state apply first the secret key. Look now at the first letter, mark the correspondence O-W and perform an incremental scramble indicated by letter W. Next letter correspondence is F-A, so perform an incremental scramble indicated by letter A. Next three correspondences are R-T, X-E, K-R. The permutation key is WATER. Continue with the second group of letters, which marks the beginning of the message itself. Start decrypt the first letter F-N, then apply an incremental scramble associated with the first letter in the permutation key (W), and so on. Second group translates to

```
F-N (W)
S-E (A)
B-A (T)
U-R (E)
T-L (R)
```

We marked the permutation key along the letters, to keep track of it more easily. The other groups are

```
S-Y (W)   M-Y (W)   L-N (W)   G-F (W)   U-H (W)   I-S (W)   H-A (W)   H-N (W)
J-E (A)   B-I (A)   I-T (A)   C-A (A)   Q-E (A)   H-T (A)   G-S (A)   D-C (A)
P-V (T)   T-N (T)   G-O (T)   V-C (T)   W-R (T)   Q-E (T)   C-B (T)   I-O (T)
P-E (E)   E-V (E)   C-R (E)   K-I (E)   I-S (E)   V-M (E)   Z-E (E)   K-N (E)
S-R (R)   X-E (R)   A-O (R)   E-P (R)   Z-Y (R)   A-H (R)   P-E (R)   C-V (R)

O-I (W)   X-O (W)   B-U (W)   L-V (W)   A-I (W)   R-H (W)   S-A (W)   J-I (W)
K-N (A)   D-F (A)   K-N (A)   G-A (A)   F-T (A)   Q-I (A)   F-I (A)   J-L (A)
Y-C (T)   Z-T (T)   V-S (T)   R-B (T)   H-Y (T)   X-S (T)   Q-N (T)   C-D (T)
V-E (E)   W-H (E)   I-O (E)   T-I (E)   B-O (E)   M-B (E)   Z-C (E)
F-D (R)   T-E (R)   X-L (R)   A-L (R)   T-F (R)   B-R (R)   I-H (R)
```

The ciphertext decodes into **NEARL YEVER YINVE NTORO FACIP HERSY STEMH ASBEE NCONV INCED OFTHE UNSOL VABIL ITYOF HISBR AINCH ILD**, which reads "Nearly every inventor of a cipher system has been convinced of the unsolvability of his brainchild" (David Kahn).

4.4. Good scrambles: return of formulas!

It's obvious that default incremental scramble (^) will change the substitution from one letter to another quite superficially. This can be seen in the previous example, where several pairs of letters remain unchanged for quite long. For example, the pair K-N appears several times in the middle of the decryption. This could be a major vulnerability of the cipher.

As stated before, a reference letter can define an overall cube orientation. We set this orientation to be the one with the front face (F) in which the letter resides, and the up face (U) towards where letter's tip/arrow points. We call this the *incremental cube orientation*.

The solution here is to provide a more complex incremental scramble. This could be formed by the default incremental scramble followed by one or more cube moves, for example ^FD. The convention is, those cube moves are applied relative to the incremental cube orientation defined at the beginning of the incremental scramble.

The new incremental scramble must be used for the entire process of encryption/decryption, including the scramble generated by the secret key and the recovery of permutation key.

The incremental scramble is part of the cryptographic device, so it is known information. It should be agreed by the communicants for the whole transmission, and specified together with the device in use.

Here is an example using previous method of encryption. Note that the incremental scramble is indicated below the device name.

---

key: `GIVIERGE`
ciphertext: **`MFNFX GTRLU TVBVQ KKWSC ULRMQ HCJYQ LZRUQ UUTHK AO`**

---

method: polyalphabetic letter substitution with repeated key
device: Non-reciprocal Enigma Rubik's Cube
incremental scramble: ^FD

From the solved state apply first the secret key using the new incremental scramble. Let's give an example for the first letter (`G`). Orient the cube such that the letter resides on the front face, with its arrow pointing upward. Notice that the letter is turned upside down. Apply a middle layer rotation MR (in extended cube notation), which corresponds to ^, then perform the moves F and D. Locate now the second letter (`I`), which happens to be on the right spot, and continue the process. At the end you get into a really good scrambled state.

Continue with the ciphertext and decode each letter in the first group, `M-A, F-B, N-C, F-D, X-E`. Don't forget to apply the new incremental scramble after each decoded letter. The permutation key is `ABCDE`. Using this key and the new incremental scramble, continue as in the previous example, decoding letter by letter the entire message

```
G-S (A)   T-F (A)   K-L (A)   U-L (A)   H-I (A)   L-A (A)   U-L (A)   A-R (A)
T-U (B)   V-I (B)   K-C (B)   L-I (B)   C-O (B)   Z-N (B)   U-L (B)   O-Y (B)
R-P (C)   B-C (C)   W-O (C)   R-C (C)   J-N (C)   R-B (C)   T-U (C)
L-E (D)   V-I (D)   S-M (D)   M-A (D)   Y-S (D)   U-E (D)   H-S (D)
U-R (E)   Q-A (E)   C-P (E)   Q-T (E)   Q-C (E)   Q-I (E)   K-O (E)
```

The ciphertext decodes into **`SUPER FICIA LCOMP LICAT IONSC ANBEI LLUSO RY`**, which reads "Superficial complications can be illusory" (M. Givierge).

4.5. Simplify things: give up some secrecy

The permutation key introduced before is a secret piece of information, because it is encoded along the message with the secret initialization key. Is this additional secrecy necessary?

We can devise a method of encryption in which the permutation key is known (public). This was in fact the case with Enigma machine, where the permutation key was physically implemented in the form of electric wires configuration for each rotor. By the principle "The enemy knows the system" (C. Shannon), this was a known permutation key.

For our method the known permutation key is set to be the sequence of letters

ABCDEFGHIJKLMNOPQRSTUVWXYZ

which is quite long, so more secure. Moreover, since it is known, the key should not be sent along the message.

Let's describe *method 4* (named polyalphabetic letter substitution with known repeated key) with an example.

---

key: OTTOHORAK
ciphertext: **MNMTP NPAUX NCPDQ LYRKV SNFLM DOBRA YZNCY KQBIF KMWLJ MBCYZ NSSSM QFMHH WGG**

---

method: polyalphabetic letter substitution with known repeated key
device: Enigma Pocket Cube
incremental scramble: ^U

From the solved state apply the secret key. First group of letters is already the message. Decrypt the first letter M-S, then use the first letter from the known permutation key (A) and perform the incremental scramble. Continue with the second letter from the message N-E, then use the second letter from the known permutation key (B), and so on. Letters from the permutation key will be reused cyclically if necessary. We list below the entire decoding

```
M-S (A)   N-I (F)   N-A (K)   L-C (P)   S-R (U)   D-O (Z)   Y-O (E)   K-R (J)
N-E (B)   P-T (G)   C-W (L)   Y-I (Q)   N-M (V)   O-D (A)   Z-T (F)   Q-E (K)
M-C (C)   A-Y (H)   P-E (M)   R-P (R)   F-E (W)   B-I (B)   N-I (G)   B-A (L)
T-U (D)   U-O (I)   D-A (N)   K-H (S)   L-T (X)   R-S (C)   C-N (H)   I-S (M)
P-R (E)   X-F (J)   Q-K (O)   V-E (T)   M-H (Y)   A-N (D)   Y-C (I)   F-E (N)

K-D (O)   M-Y (T)   N-O (Y)   Q-I (D)   W-R (I)
M-B (P)   B-I (U)   S-K (Z)   F-T (E)   G-E (J)
W-Y (Q)   C-N (V)   S-E (A)   M-S (F)   G-T (K)
L-T (R)   Y-G (W)   S-E (B)   H-E (G)
J-R (S)   Z-T (X)   M-P (C)   H-C (H)
```

The ciphertext decodes into **SECUR ITYOF AWEAK CIPHE RMETH ODISN OTINC REASE DBYTR YINGT OKEEP ITSEC RET**, which reads "Security of a weak cipher method is not increased by trying to keep it secret" (Otto J. Horak).

4.6. Avoid repetitions

A polyalphabetic substitution with repeated key is ultimately periodic, even if the period is very long. This was also the case with Enigma machine, where the 3-rotor version had a period around 26^3. This periodicity is an inherent weakness of the cipher. To remove it, it's necessary to use substitutions in a nonrepetitive (aperiodic) manner. One way to accomplish this is to change substitution based on letters from the plaintext. So, we will speak here about polyalphabetic substitution with autokey. This is the default encryption/decryption method used in the description pages of Enigma cubes.

The method is similar to previous one, but the permutation key is the plaintext itself. This simplifies a bit the encryption/decryption, but the method is prone to human errors. If a single letter is wrongly encrypted/decrypted, the rest of the message is compromised.

Let's describe *method 5* (named polyalphabetic letter substitution with autokey) with an example.

---

key: KERCKHOFFS
ciphertext: **HSFWE LXQCO EYEXW GFEFN SPXNA SEWGP ZOGCJ SJMQQ JKAXF UQNDH DZIAL SVCBB FMHCH VKHFT WANXR ZXRBZ LSGGP OSAJX WBSX**

---

method: polyalphabetic letter substitution with autokey
device: Enigma Rubik's Cube
incremental scramble: ^U^

Notice first that the incremental scramble contains 3 moves, involving twice the reference letter (^). From the solved state apply the secret key, then decrypt the first letter H-A, use the decrypted letter (A) and perform the incremental scramble. Continue with the second letter from the message S-N, then use the newly decrypted letter (N) to make the scramble, and so on. In this case it's easy to keep track on which reference letter must be used: always the last decoded letter. We list below the entire decoding

```
H-A   L-R   E-O   G-O   S-M   S-L   Z-E   S-E   J-F   U-Y   D-G   S-T   F-Y   V-E
S-N   X-Y   Y-N   F-R   P-S   E-D   O-C   J-V   K-E   Q-T   Z-A   V-T   M-S   K-X
F-E   Q-P   E-A   E-I   X-H   W-B   G-U   M-E   A-V   N-H   I-B   C-H   H-T   H-C
W-N   C-T   X-L   F-T   N-O   G-E   C-R   Q-N   X-E   D-I   A-O   B-E   C-E   F-E
E-C   O-I   W-G   N-H   A-U   P-S   J-E   Q-I   F-R   H-N   L-U   B-S   H-M   T-P

W-T   Z-E   L-U   O-K   W-E
A-T   X-Y   S-B   S-N   B-D
N-H   R-I   G-L   A-O   S-G
X-E   B-S   G-I   J-W   X-E
R-K   Z-P   P-C   X-L
```

The ciphertext decodes into **ANENC RYPTI ONALG ORITH MSHOU LDBES ECURE EVENI FEVER YTHIN GABOU TTHES YSTEM EXCEP TTHEK EYISP UBLIC KNOWL EDGE**, which reads "An encryption algorithm should be secure even if everything about the system, except the key, is public knowledge" (Kerckhoffs's principle).

4.7. Security through obscurity: unspecified method of encryption

A nice feature of previous methods of encryption is that the communicants involved in the transmission can infer which method is used, without knowing it in advance. There are some requirements for this to work: the decrypted message should make sense and exactly one of the above methods should be used.

After applying the initial key to the cube, someone can easily check if the monoalphabetic substitution is used. If not, try to decipher using known permutation key, then autokey. If none of the above works, it means the permutation key is sent along the message. Then try to decipher with repeated key in both group and letter substitutions. One of the above should work. In this case it might be possible that permutation key is sent twice, to offer some confidence to the decoder. Of course, the cryptographic device and incremental scramble are always specified.

4.8. Don't use a secret key twice!

Secret key repetition is a major sin in cryptography. How to avoid key repetition? One way is by using incremental key changes between messages. That is, change the key after each message using an agreed rule. For example, start with a secret key for the first message, let's say STRONG. For next messages increment letters in the secret key from left to right, as follows TTRONG, UTRONG and so on. After ZTRONG will come AURONG, BURONG and so on. In this way we get almost 26^6 unique keys. If you feel insecure, the number of letters could be increased. Of course, no message should be missed among communicants.

4.9. Got really paranoid? Use double encryption

Double encryption, or superencryption, is the process of encrypting a message twice, usually by different methods and/or devices. Basically, after the first encipherment of the plaintext, the obtained ciphertext is considered the new plaintext and encoded again with the second method/device. Before the second encipherment, the cube must be initialized and the secret key applied again. If the methods or devices are different between encipherments, they must be used in reverse when decrypting.

5. Challenge

---

key: `ENIGMA`
ciphertext: **`CPFHD JWKPS YDLWG PUQTT MWYLY BSLBV JIJFM EUCCX JIAJD FNJCG DUJZD MAVUQ OLYDX RCHZ`**

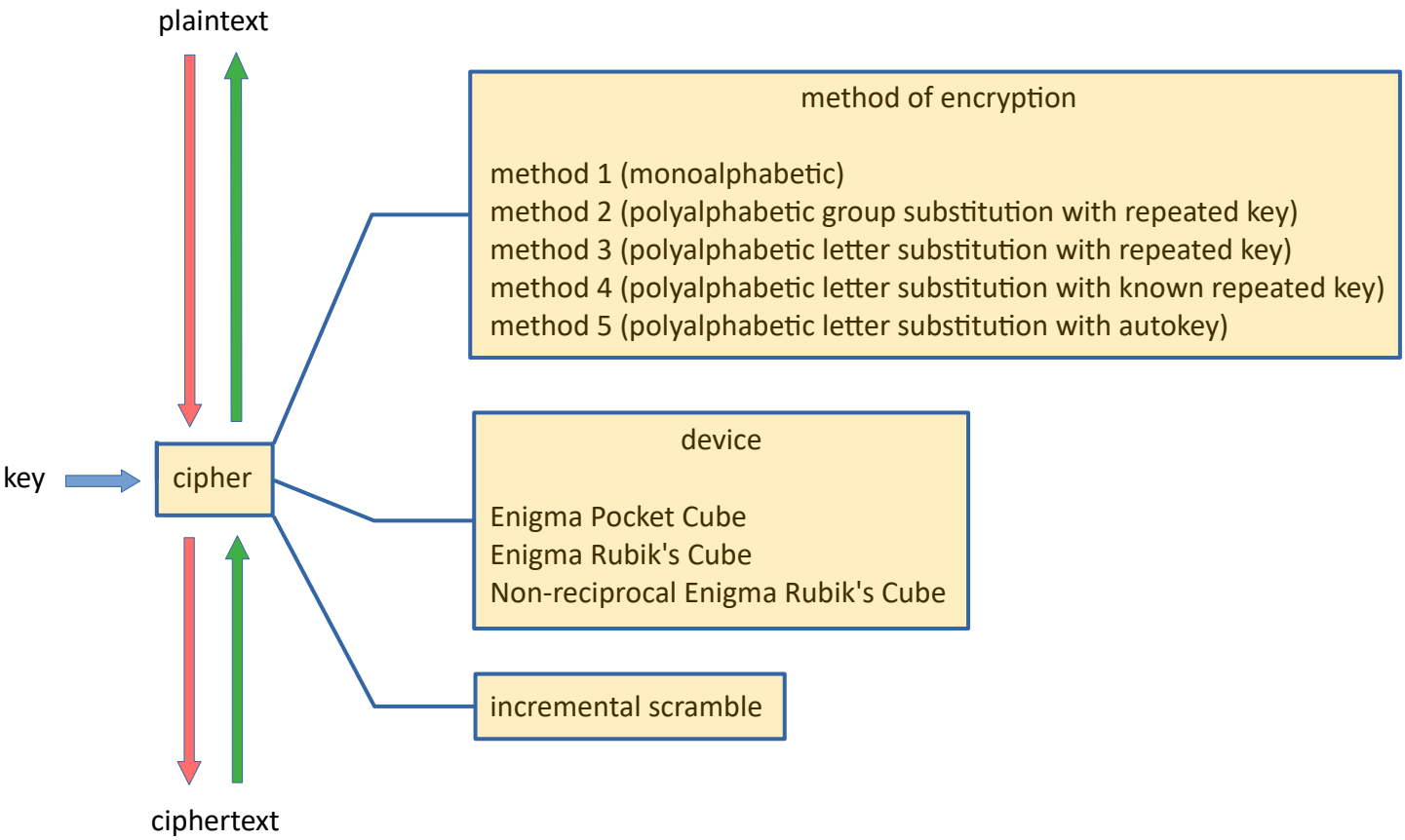
---

method: unspecified (see above)
device: Non-reciprocal Enigma Rubik's Cube
incremental scramble: ^U^

---

6. Summary

Below is a diagram of the entire process of encryption using devices and methods introduced above.



References

[1] F.L. Bauer, Decrypted Secrets – Methods and Maxims of Cryptology, 4th edition, Springer, 2007.

[2] A.R. Miller, The Cryptographic Mathematics of Enigma, Center for Cryptologic History, National Security Agency, 2019, retrieved 3.08.2021, https://www.nsa.gov/Portals/70/documents/about/cryptologic-heritage/historical-figures-publications/publications/wwii/CryptoMathEnigma_Miller.pdf

[3] W. Randelshofer, Virtual Cubes, retrieved 3.08.2021, https://www.randelshofer.ch/rubik/virtual_cubes/rubik/instructions/instructions.html

[4] D. Rijmenants, Technical Details of the Enigma Machine, retrieved 3.08.2021, http://users.telenet.be/d.rijmenants/en/enigmatech.htm

[5] Superset ENG 3x3 Notation, retrieved 3.08.2021, https://www.randelshofer.ch/rubik/patterns/doc/supersetENG_3x3.html

Annexes

Below are some printable sheets to help with the methods.

method: generic

inc. scramble: [ ]

key:

ciphertext: _____ _____ _____ _____ _____ _____ _____ _____
plaintext : _____ _____ _____ _____ _____ _____ _____ _____

ciphertext: _____ _____ _____ _____ _____ _____ _____ _____
plaintext : _____ _____ _____ _____ _____ _____ _____ _____

ciphertext: _____ _____ _____ _____ _____ _____ _____ _____
plaintext : _____ _____ _____ _____ _____ _____ _____ _____

ciphertext: _____ _____ _____ _____ _____ _____ _____ _____
plaintext : _____ _____ _____ _____ _____ _____ _____ _____

ciphertext: _____ _____ _____ _____ _____ _____ _____ _____
plaintext : _____ _____ _____ _____ _____ _____ _____ _____

ciphertext: _____ _____ _____ _____ _____ _____ _____ _____
plaintext : _____ _____ _____ _____ _____ _____ _____ _____

ciphertext: _____ _____ _____ _____ _____ _____ _____ _____
plaintext : _____ _____ _____ _____ _____ _____ _____ _____

ciphertext: _____ _____ _____ _____ _____ _____ _____ _____
plaintext : _____ _____ _____ _____ _____ _____ _____ _____

method: polyalphabetic letter substitution with repeated key

inc. scramble: [ ]

key:

ciphertext: _____ _____ _____ _____ _____ _____ _____ _____
plaintext : _____ _____ _____ _____ _____ _____ _____ _____
rep. key : _____ _____ _____ _____ _____ _____ _____ _____

ciphertext: _____ _____ _____ _____ _____ _____ _____ _____
plaintext : _____ _____ _____ _____ _____ _____ _____ _____
rep. key : _____ _____ _____ _____ _____ _____ _____ _____

ciphertext: _____ _____ _____ _____ _____ _____ _____ _____
plaintext : _____ _____ _____ _____ _____ _____ _____ _____
rep. key : _____ _____ _____ _____ _____ _____ _____ _____

ciphertext: _____ _____ _____ _____ _____ _____ _____ _____
plaintext : _____ _____ _____ _____ _____ _____ _____ _____
rep. key : _____ _____ _____ _____ _____ _____ _____ _____

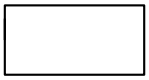
ciphertext: _____ _____ _____ _____ _____ _____ _____ _____
plaintext : _____ _____ _____ _____ _____ _____ _____ _____
rep. key : _____ _____ _____ _____ _____ _____ _____ _____

ciphertext: _____ _____ _____ _____ _____ _____ _____ _____
plaintext : _____ _____ _____ _____ _____ _____ _____ _____
rep. key : _____ _____ _____ _____ _____ _____ _____ _____

ciphertext: _____ _____ _____ _____ _____ _____ _____ _____
plaintext : _____ _____ _____ _____ _____ _____ _____ _____
rep. key : _____ _____ _____ _____ _____ _____ _____ _____

ciphertext: _____ _____ _____ _____ _____ _____ _____ _____
plaintext : _____ _____ _____ _____ _____ _____ _____ _____
rep. key : _____ _____ _____ _____ _____ _____ _____ _____

method: polyalphabetic letter substitution with known repeated key

inc. scramble: 

key:

ciphertext : _____ _____ _____ _____ _____ _____ _____ _____
plaintext : _____ _____ _____ _____ _____ _____ _____ _____
known key: ABCDE FGHIJ KLMNO PQRST UVWXY ZABCD EFGHI JKLMN

ciphertext : _____ _____ _____ _____ _____ _____ _____ _____
plaintext : _____ _____ _____ _____ _____ _____ _____ _____
known key: OPQRS TUVWX YZABC DEFGH IJKLM NOPQR STUVW XYZAB

ciphertext : _____ _____ _____ _____ _____ _____ _____ _____
plaintext : _____ _____ _____ _____ _____ _____ _____ _____
known key: CDEFG HIJKL MNOPQ RSTUV WXYZA BCDEF GHIJK LMNOP

ciphertext : _____ _____ _____ _____ _____ _____ _____ _____
plaintext : _____ _____ _____ _____ _____ _____ _____ _____
known key: QRSTU VWXYZ ABCDE FGHIJ KLMNO PQRST UVWXY ZABCD

ciphertext : _____ _____ _____ _____ _____ _____ _____ _____
plaintext : _____ _____ _____ _____ _____ _____ _____ _____
known key: EFGHI JKLMN OPQRS TUVWX YZABC DEFGH IJKLM NOPQR

ciphertext : _____ _____ _____ _____ _____ _____ _____ _____
plaintext : _____ _____ _____ _____ _____ _____ _____ _____
known key: STUVW XYZAB CDEFG HIJKL MNOPQ RSTUV WXYZA BCDEF

ciphertext : _____ _____ _____ _____ _____ _____ _____ _____
plaintext : _____ _____ _____ _____ _____ _____ _____ _____
known key: GHIJK LMNOP QRSTU VWXYZ ABCDE FGHIJ KLMNO PQRST

ciphertext : _____ _____ _____ _____ _____ _____ _____ _____
plaintext : _____ _____ _____ _____ _____ _____ _____ _____
known key: UVWXY ZABCD EFGHI JKLMN OPQRS TUVWX YZABC DEFGH