**CryptoGRaphy**

This is the category in which we encode a message into another language or any of the ciphers or codes.

**Some of the basic ciphers or challenges in cryptography are,,,,,,**

**1. Base 64, Base 32, Base 85 (Website to decode: www.dcode.fr/Dragoman)**

**2. Caeser Cipher (www.dcode.fr/www.cryptii.com/Dragoman)**

**3. Vigenere Cipher (www.dcode.fr /guballe.de/ Dragoman)**

**4. Malbolge Language (Malbolge interpreter online)**

**5. Rot47/Rot13 (dcode.fr/ Dragoman {Mera tool hai GitHub pe})**

**6. Alien Languages |**

**!-- Pigpen cipher |**

**!-- Hylian Cipher |**

**!--Alienises Futurama cipher | (www.dcode.fr)**

**!--Daggers alphabets |**

**!--Dancing men Ciphers |**

**7. RSA cryptography**

**8. Morse code (dcode.fr/crytpii.com/Dragoman)**

**[+] CAESER CIPHER IS OFTEN HINTED TO YOU USING THESE KEYWORD:--**

Ancient Rome, Ancient Roman, Julius ceaser, or may not be hinted as this the basic cipher and does not carry much points if it is given directly.

**Caeser Cipher** is encoding a works as follows:

Plaintext + key(digits)=cipher text

example:

Shivu + 3 = Vklyx

S->t->u->V

h+3=k so on....

it adds the no. of alphabets = key.

It is the most common and carries less points in CTF's.

**[+]ASCII TO TEXT –**

In this the the plaintext is encoded to the ascii codes of the respective characters of the plain text.

**Example:** Shivu 🡪 83 104 105 118 117

**[+]BINARY TO TEXT-**

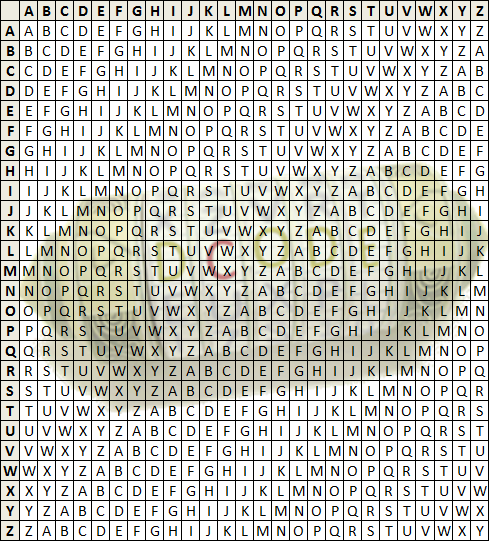
The text is converted to the binary digits of the respective alphabet.

Example: Shivu 🡪 01010011 01101000 01101001 01110110 01110101

***[+]VIGNERE CIPHER*=**

**Vignere cipher** is at the one level above the cease cipher, It is also a common Cipher which comes in the CTF competitions,

ENCODING PROCESS:

It may be difficult to be decoded at times, Vignere cipher decoding get different results on different sites. use the sites stated in the cipher table above. 

Example: The key is KEY, and the plaintext is DCODE.

Locate the first letter of the plaintext message in the first line of the table and the first letter of the key on the left column. The cipher letter is at the intersection.

Example: Locate the letter D on the first row, and the letter K on the first column, the ciphered letter is the intersection cell N.

Continue with the next letter of the plaintext, and the next letter of the key. When arrived at the end of the key, go back to the first letter of the key.

**It may be difficult to be decoded at times, Vignere cipher decoding get different results on different sites. use the sites stated in the cipher table above.**

**[+] Morse Code:**

The morse code is easily recognizable .

It is represented by dots (.) and dashes(-).

Any cipher text may look like :

Example : Shivu 🡪 ... .... .. ...- ..-

**[+] Base 64/32/85:**

At no. 1 in the table. the challenges of bases are usually basic challenges warmup challenges for the participants.

The working can be studied online.

the challenges of bases are easy to recognize....

Majority of the times the base ciphers end with "=" symbols.

Value Encoding Value Encoding Value Encoding Value Encoding

0 A 17 R 34 i 51 z

1 B 18 S 35 j 52 0

2 C 19 T 36 k 53 1

3 D 20 U 37 l 54 2

4 E 21 V 38 m 55 3

5 F 22 W 39 n 56 4

6 G 23 X 40 o 57 5

7 H 24 Y 41 p 58 6

8 I 25 Z 42 q 59 7

9 J 26 a 43 r 60 8

10 K 27 b 44 s 61 9

11 L 28 c 45 t 62 +

12 M 29 d 46 u 63 /

13 N 30 e 47 v

14 O 31 f 48 w (pad) =

15 P 32 g 49 x

16 Q 33 h 50 y

**ENCODING BASE64**

**Input:** a@bc

**Binary Representation of input (8-bit bytes):**

01100001 01000000 01100010 01100011

**Step 1:** Organize the input into 24-bit groups (having four 6-bit groups each). Pad with zero bits at the end to form an integral no of 6-bit groups.

011000 010100 000001 100010 011000 110000

# (padded with four zeros at the end)

**Step 2:** Convert the 6-bit sequences to Base64 alphabets by indexing into the Base64 index table. Add pad character if zero bits are added at the end of the input.

The above 6-bit groups equate to the following indexes:

24 20 1 34 24 48

Indexing into the Base64 alphabet table gives the following output:

YUBiYw== # (padded with two `=` characters)

**[+] TAP code:**

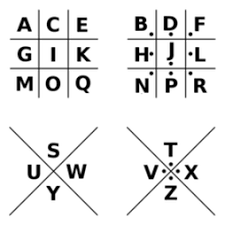
Tap code is represented just by dots (.).

**Example:** Shivu 🡪 .... ... .. ... .. .... ..... . .... .....

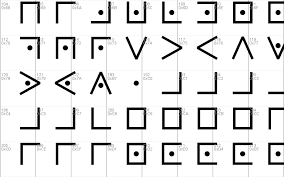
**[+] ROT13/ROT(n) 🡪 n=digits/key**

**NOTE:** Rotation/Rot cipher works same as caeser cipher.

**[+] PIGPEN CIPHER:**

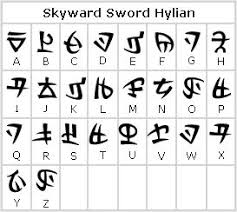
****

The position the alphabet is placed that is the cipher for that alphabet.



The above image explains how the cipher looks in encoded text.

**[+] SKYWARD HYLIAN CIPHER**



The above image about how hylian cipher looks.

**THERE ARE SEVERAL CATEGORIES IN HYLIAN CIPHERS. (Zelda/skyward/twilight princess)**

**THE ABOVE IS ONE OF THEM.**

**[+] SHEIKAH LANGUAGE**

****

**[+] DNA cipher –**

DNA cipher is recognized by ‘A’, ‘T’, ‘G’, ‘C’.

The plaintext is encoded like:

**Example:** Shivu 🡪 GTAACTAGACCCCCA

**As there is no online decoder for this. Either we need to write your own script or do manually.**

**To decode this cipher, I would strongly suggest DRAGOMAN TOOL.**

**[+] BACON CIPHER:**

Bacon cipher is represented by grouping of ‘A’ and ‘B’.

**NOTE:** The cipher an be represented in CAPITAL OR SMALL Alphabets.

**For example**: Shivu 🡪 baaab aabbb abaaa baabb baabb **OR** BAAAB AABBB ABAAA BAABB BAABB

**[+] A1Z26 Cipher:**

A1Z26 cipher is just encoding the alphabets to their respective numbers in English Alphabets.

A=1 , B=2 , C=3, D=4, … , Z=26.

**Example:** S h i v u 🡪 19 8 9 22 21

19 8 9 22 21

**[+] Railfence/ZigZag Cipher:**

**Rail Fence**Encryption uses an integer for the number of levels of the **zigzag**.

The encoded message is written in **zig-zag** (like a **rail fence/**sawtooth) along a path with N levels/floors.

**EXAMPLE:**

**With Key = 2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rail Fence (2↕ ↘↗ ) | | | | |
| S |  | I |  | U |
|  | H |  | V |  |

**With Key = 3**

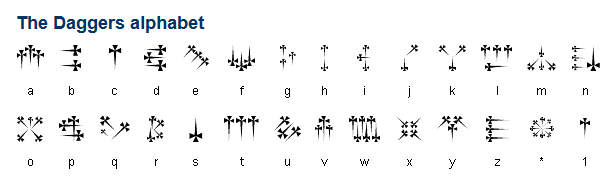
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Rail Fence (3↕ ↘↗ )** | | | | |
| S |  |  |  | U |
|  | **H** |  | **V** |  |
|  |  | **I** |  |  |
|  |  |  |  |  |

**With Key = 4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rail Fence (4↕ ↘↗ ) | | | | |
| S |  |  |  |  |
|  | H |  |  |  |
|  |  | I |  | U |
|  |  |  | V |  |

**[+] Daggers Cipher**

The daggers cipher is another silly text-to-image encoder. This is the key, and you can find a decoder on [www.dcode.fr](http://www.dcode.fr)



**[+] TRANSPOSITION CIPHER**

**[+] SUBSTITUTION CIPHER**

**[+] MALBOLGE CIPHER:**

Malbolge is an old programming language.

It looks like some gibberish/rubbish text.

**Example:**

Plaintext: Hello World!

Ciphertext: ('&%:9]!~}|z2Vxwv-,POqponl$Hjig%eB@@>}=<M:9wv6WsU2T|nm-,jcL(I&%$#"`CB]V?Tx<uVtT`Rpo3NlF.Jh++FdbCBA@?]!~|4XzyTT43Qsqq(Lnmkj"Fhg${z@>

**To Decode search for Malbolge interpreter online.**

**[+] BRAINFUCK LAUNGUAGE**

Brain Fuck is not a proper encryption system, but rather a programming language that has been obfuscated. Encoding consists in writing machine code that returns text as output.

Machine operations are:

> : increment the pointer (+1),

< : decrement the pointer (-1),

+ : increment the byte in the memory cell where the pointer is located,

- : decrement the byte in the memory cell where the pointer is located,

. : send the value of the pointed byte as output (treated as an [**ASCII**](https://www.dcode.fr/ascii-code) value),

, : insert an input byte (user input) in the memory cell where the pointer is located ([**ASCII**](https://www.dcode.fr/ascii-code) value),

[ : if the pointed byte is 0 then jump to instruction after the corresponding ],

] : if the pointed byte is not 0 then jump to the instruction after the corresponding [

**Example :**

Plaintext : **Shivu**

Ciphertext:++++++++++[>+>+++>+++++++>++++++++++<<<<]>>>+++++++++++++.>++++.+.+++++++++++++.-.