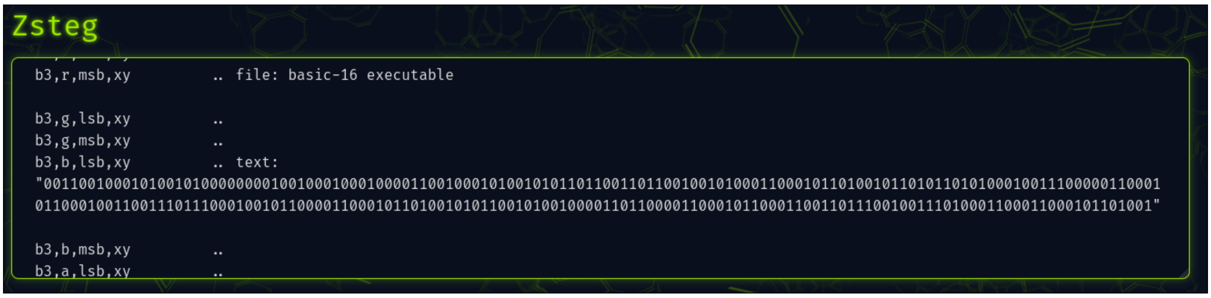
# When we got the archive, we can see it’s protected with a password. We can go on **https://www.lostmypass.com/try/** and upload the archive to recover the password. We will find out that the password is ***password*** 😐

# Now we got an image with WinRAR beer. After I uploaded the image on **https://www.aperisolve.com** I got this enormous binary with **zsteg** :



Zsteg

```
b3,r,msb,xy         .. file: basic-16 executable

b3,g,lsb,xy         ..
b3,g,msb,xy         ..
b3,b,lsb,xy         .. text:
"001100100010100101000000000100100010001000011001000101001010110110011011001001010001100010110100101101011010100010011100000110001
0110001001100111011100010010110000110001011010010101100101001000011011000011000101100011001101110010011101000110001100010110100l"

b3,b,msb,xy         ..
b3,a,lsb,xy         ..
```

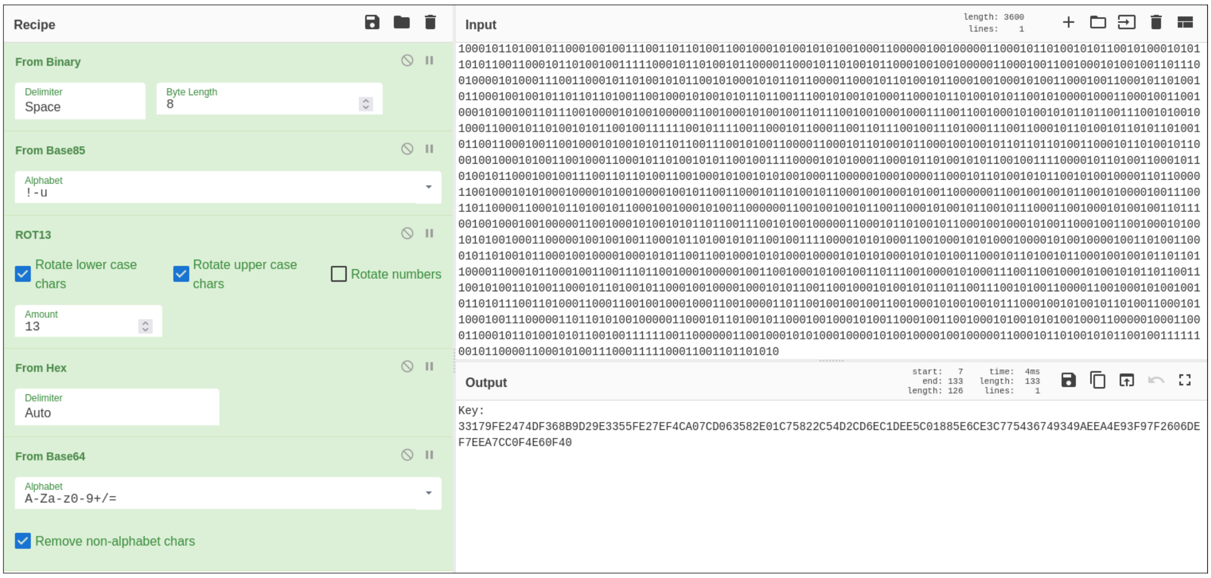
# So I understood it’s an LSB/MSB exercise. I ran zsteg local :

zsteg flag.png

# And I extracted the whole text :

zsteg -E "b3,b,lsb,xy" flag.png > binary.txt

# Now, inspecting the file **binary.txt,** there is a lot of text. I opened the file with an hex editor (**https://hexed.it/**) and I took the binary. We can go now on CyberChef. With the help of **https://www.dcode.fr/cipher-identifier** also, I found the message from the binary with the following ciphers : **from Binary -> from Base85 -> ROT13 -> from HEX -> from Base64**



Recipe

From Binary
Delimiter: Space
Byte Length: 8

From Base85
Alphabet: !-u

ROT13
☑ Rotate lower case chars
☑ Rotate upper case chars
□ Rotate numbers
Amount: 13

From Hex
Delimiter: Auto

From Base64
Alphabet: A-Za-z0-9+/=
☑ Remove non-alphabet chars

Input — length: 3600, lines: 1

```
1000101101001011000100100111001101101001100100010100101010100100011000001001000001100010110100101011001010001010
1101011001100010110100100111110001011010010110000110001011010010110001001001000001100010011001000101001001101110
0100001010001110011000101101001010110010100010101101100001100010110100101100010010001010011000100110001011010010
0110001001001011011011010011001000101001010110110011100101001010001100010110100101011001010000100011000100110010
0001010010011011100100001010010000011001000101001001101110010010001000111001100100010100101011011001110010100101
0100011000101101001010110010011111100101111001100010110001100110111001001110100011100110001011010010110101101001
0110011000100110010001010010101101100111001010011000011000101101001011000100100101101101101001100010110100101101
1000100100010100110010001100010110100101011001001111000010101000110001011010010101100100111100001011010011000101
1010010110001001001110011011010011001000101001010101001000110000010001000011000101101001010110010100100001101100
0011001000101010001000010100100001001011001100010110100101100010010001010011000000110010010010110010100001001110
0110110000110001011010010110001001000101001100000011001001001011001100010100101100101110001100100010100100110111
1001001000100100000011001000101001010110110011100101001000001100010110100101100010010001010011000100110010001010
0101010010001100000100100100110001011010010101100100111100001010100011001000101010001000010100100001001101001100
0101101001011000100100001000101011001100100010101000100001010101000101010100110001011010010110001001001011011010
1100000110001011000100110011101100100010000001001100100010100100110111001000010100011100110010001010010101101100
1110010100110100110001011010010110001001000010001010110011001000101001010110110011100101001100001100100010100100
1011010111001101000110001100100100010001100100001101100100100100110010001010010010111000100101001011010011000101
1000100111000000110110101001000000110001011010010110001001000101001100010011001000101001010100100011000000100011
0001100010110100101011001001111111001100000011001000101010001000010100100001001000001100010110100101011001001111
11001011000011000101001110001111100011001101101010
```

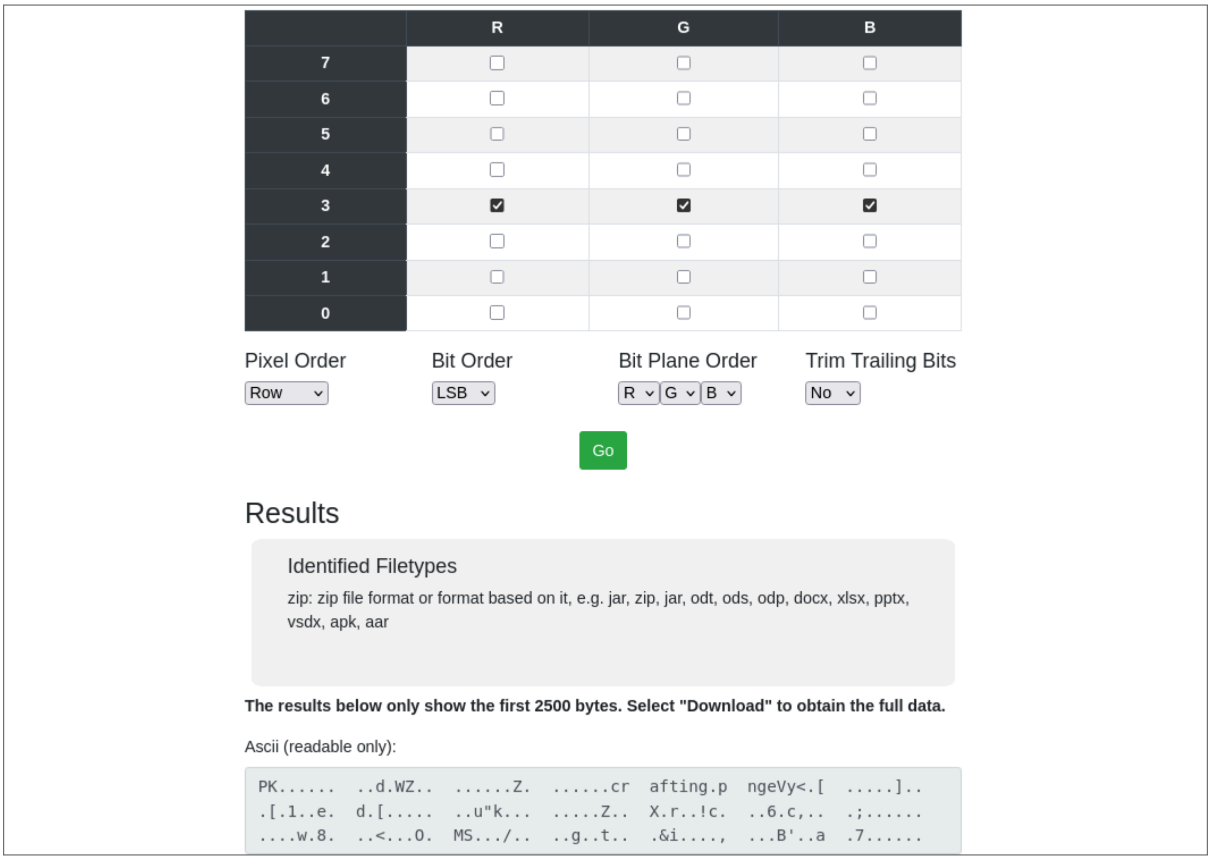
Output — start: 7, end: 133, length: 126; time: 4ms, length: 133, lines: 1

```
Key:
33179FE2474DF368B9D29E3355FE27EF4CA07CD063582E01C75822C54D2CD6EC1DEE5C01885E6CE3C775436749349AEEA4E93F97F2606DE
F7EEA7CC0F4E60F40
```

# **Key:**
33179FE2474DF368B9D29E3355FE27EF4CA07CD063582E01C75822C54D2CD6EC1DE E5C01885E6CE3C775436749349AEEA4E93F97F2606DEF7EEA7CC0F4E60F40

# Now we have a key. What should we do now? Because there is nothing with **binwalk, exiftool, steghide, foremost** and I found ONLY 1 message with zsteg (I also **used zsteg -a flag.png**), I continued with LSB/MSB experiments on https://www.georgeom.net/StegOnline because there is no MSB on CyberChef

# After some tries with LSB/MSB, I found an archive hidden in the image :



	R	G	B
7	☐	☐	☐
6	☐	☐	☐
5	☐	☐	☐
4	☐	☐	☐
3	☑	☑	☑
2	☐	☐	☐
1	☐	☐	☐
0	☐	☐	☐

Pixel Order: Row
Bit Order: LSB
Bit Plane Order: R G B
Trim Trailing Bits: No

Results

Identified Filetypes

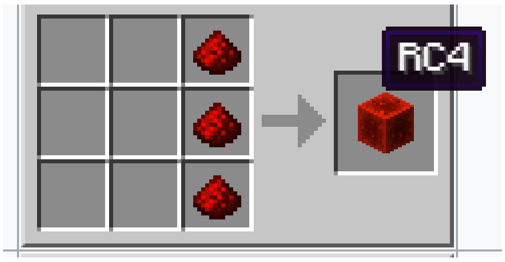
zip: zip file format or format based on it, e.g. jar, zip, jar, odt, ods, odp, docx, xlsx, pptx, vsdx, apk, aar

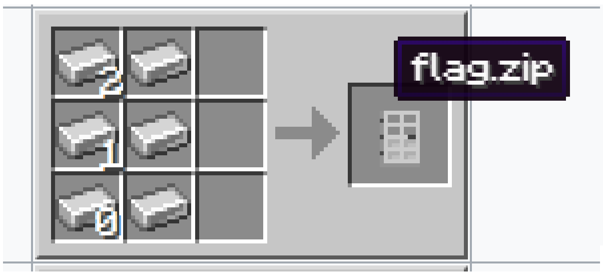
The results below only show the first 2500 bytes. Select "Download" to obtain the full data.

Ascii (readable only):

```
PK......  ..d.WZ..  ......Z.  ......cr  afting.p  ngeVy<.[  .....]..
.[.1..e.  d.[.....  ..u"k...  .....Z..  X.r..!c.  ..6.c,..  .;......
....w.8.  ..<...O.  MS.../..  ..g..t..  .&i....,  ...B'..a  .7......
```

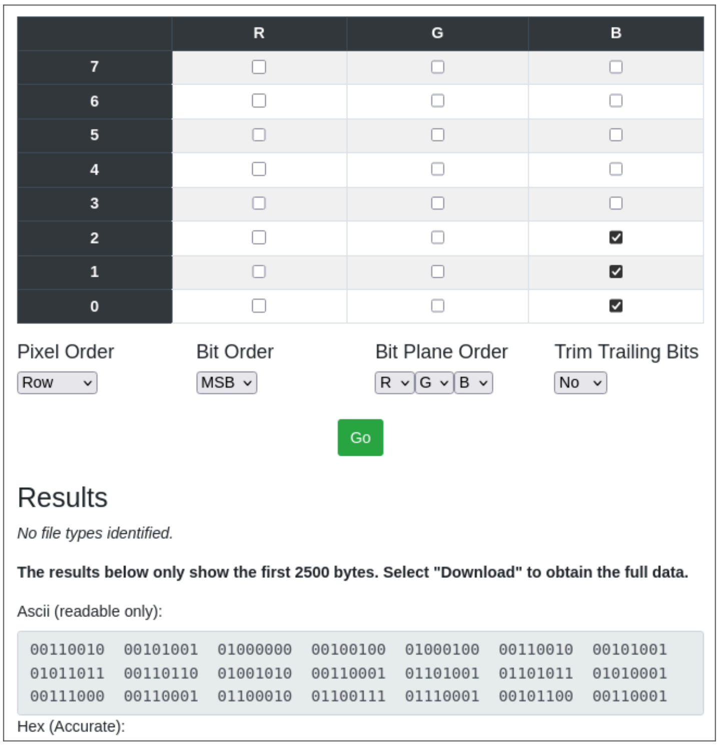
# After I extracted the archive I found these 2 images :





# Now it's clear : we have to use LSB/MSB on the **main** image like there is in these images. Let's go with the first one :

# And there is :



	R	G	B
7	□	□	□
6	□	□	□
5	□	□	□
4	□	□	□
3	□	□	□
2	□	□	☑
1	□	□	☑
0	□	□	☑

Pixel Order: Row

Bit Order: MSB

Bit Plane Order: R G B

Trim Trailing Bits: No

Results

No file types identified.

The results below only show the first 2500 bytes. Select "Download" to obtain the full data.

Ascii (readable only):

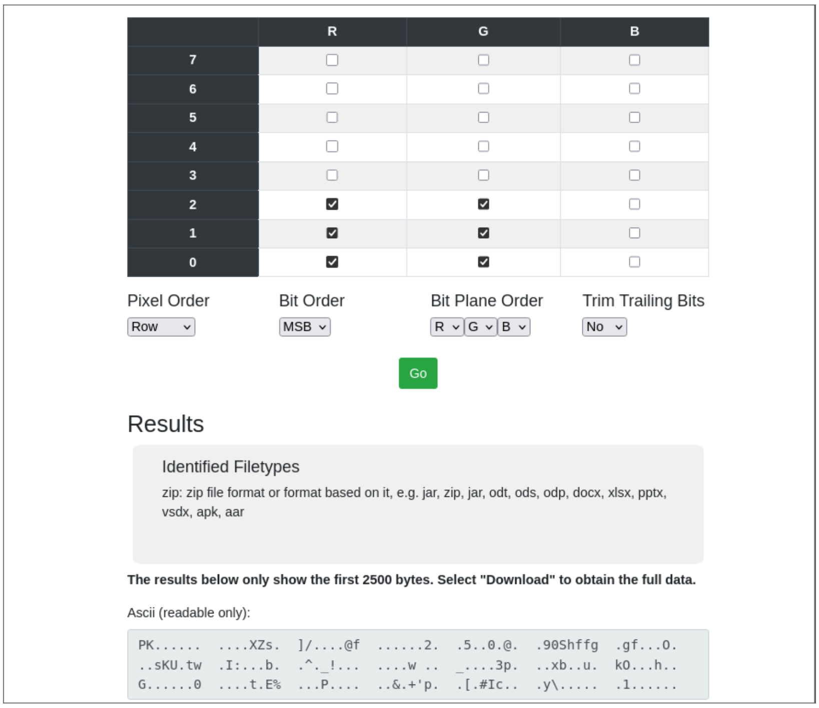
```
00110010  00101001  01000000  00100100  01000100  00110010  00101001
01011011  00110110  01001010  00110001  01101001  01101011  01010001
00111000  00110001  01100010  01100111  01110001  00101100  00110001
```

Hex (Accurate):

# It’s the same binary as the earlier! I think the challenge was not supposed to be solved with **zsteg**. But I did this writeup at the same time with the challenge because it was difficult and I kept the hints here.

# But the redstone image says : RC4. I decrypted the key with no RC4, so this is a hint. Let’s extract the **flag.zip** from the image with the iron door :

# And there is :



	R	G	B
7	□	□	□
6	□	□	□
5	□	□	□
4	□	□	□
3	□	□	□
2	☑	☑	□
1	☑	☑	□
0	☑	☑	□

Pixel Order: Row

Bit Order: MSB

Bit Plane Order: R G B

Trim Trailing Bits: No

Results

Identified Filetypes

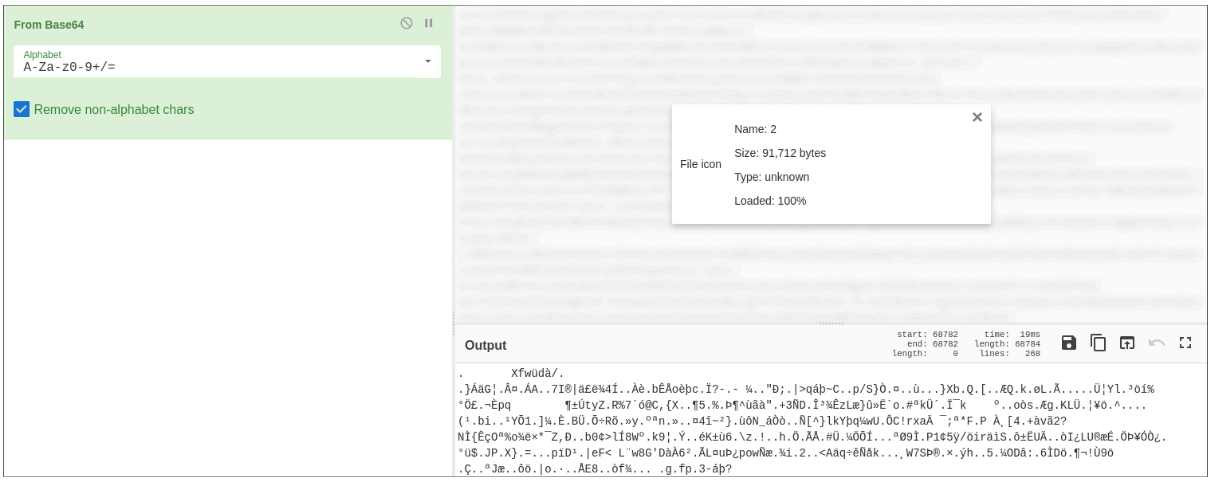
zip: zip file format or format based on it, e.g. jar, zip, jar, odt, ods, odp, docx, xlsx, pptx, vsdx, apk, aar

The results below only show the first 2500 bytes. Select "Download" to obtain the full data.

Ascii (readable only):

```
PK......  ....XZs.  ]/....@f  ......2.  .5..0.@.  .90Shffg  .gf...O.
..sKU.tw  .I:...b.  .^._!...  ....w ..  _....3p.  ..xb..u.  kO...h..
G......0  ....t.E%  ...P....  ..&.+'p.  .[.#Ic..  .y\.....  .1......
```

# After I extracted the archive, we got a file named **2**. We need to use CyberChef again :



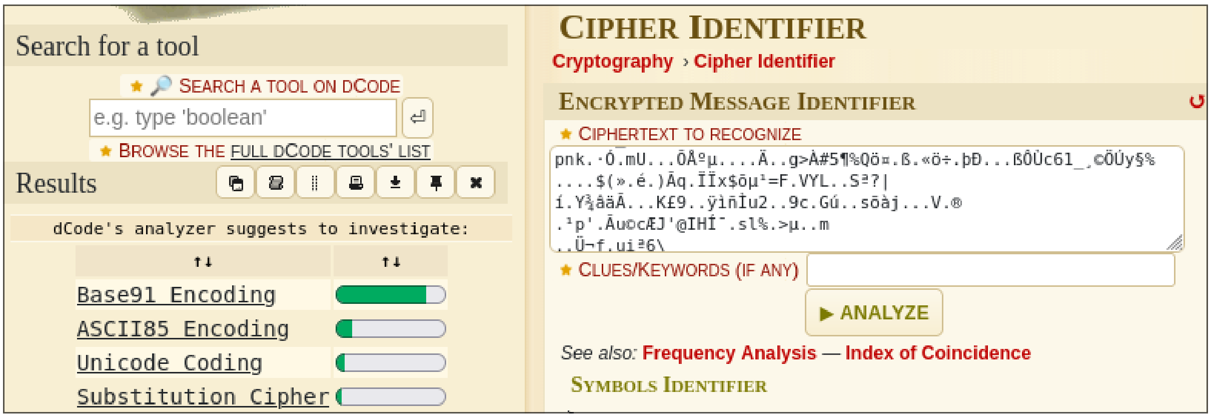
Alphabet: A-Za-z0-9+/=

Name: 2
Size: 91,712 bytes
Type: unknown
Loaded: 100%

```
start: 68782    time:  19ms
  end: 68782  length: 68784
length:    0   lines:   268
```

```
.       Xfwüdà/.
.}ÁäG¦.Â¤.ÁA..7I®|ä£ë¾4Í..Àè.bÊÅoèþc.Ï?-.- ¼.."Ð;.|>qáþ~C..p/S}Ò.¤..ù...}Xb.Q.[..ÆQ.k.øL.Ã.....Ü¦Yl.³öí%
°Ö£.¬Èpq        ¶±ÚtyZ.R%7`ó@C,{X..¶5.%.Þ¶^ùãà".+3ÑD.Î³¾ÊzLæ}û»Ë`o.#ªkÜ´.Ï¯k    º..oòs.Æg.KLÜ.¦¥ö.^....
(¹.bi..¹YÕ1.]¼.È.BÜ.Õ÷Rõ.»y.ºªn.»..¤4î~²}.ùôN_áÒò..Ñ[^}lkYþq¼wU.ÔC!rxaÄ ¯;ª*F.P À¸[4.+àvã2?
NÌ{ÊçOª%o¾ë×*¯Z,Ð..b0¢>lÍ8Wº.k9¦.Ý..éK±ù6.\z.!..h.Ö.ÃÅ.#Ü.¼ÖÕÍ...ªØ9Ì.P1¢5ÿ/öiräìS.ô±ËUÄ..òI¿LU®æÉ.ÖÞ¥ÓÒ¿.
°ü$.JP.X}.=...pïD¹.|eF< L¨w8G'DàÀ6².ÃL¤uÞ¿powÑæ.¾i.2..<Aäq÷êÑåk...¸W7SÞ®.×.ýh..5.¼ODâ:.6ÌDö.¶¬!Ù9ö
.Ç..ªJæ..ôö.|o.·..ÅE8..òf¾... .g.fp.3-áþ?
```

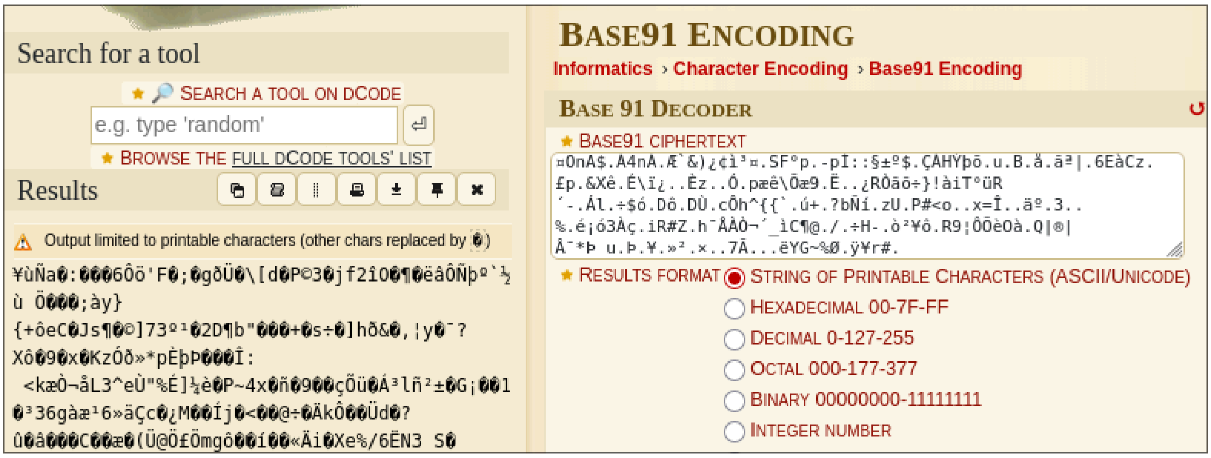
# After I got this strange text (cipher : from Base64) from the file, I uploaded the result on **https://www.dcode.fr/cipher-identifier**. I got Base91 encode, but I don't get an result :



dCode's analyzer suggests to investigate:

↑↓	↑↓
Base91 Encoding	🟩🟩🟩🟩🟩🟩🟩🟩
ASCII85 Encoding	🟩
Unicode Coding	🟩
Substitution Cipher	

```
pnk.·Ō.mU...ŌÅºµ....Ä..g>À#5¶%Qö¤.ß.«ö÷.þÐ...ßÔÙc61_¸©ÖÚy§%
....$(».é.)Āq.ĪÏx$õµ¹=F.VYL..Sª?|
í.Y¾âäĀ...K£9..ÿìñÌu2..9c.Gú..sõàj...V.®
.¹p'.Āu©cÆJ'@IHÍ¯.sl%.>µ..m
..Ü¬f.ui²6\
```



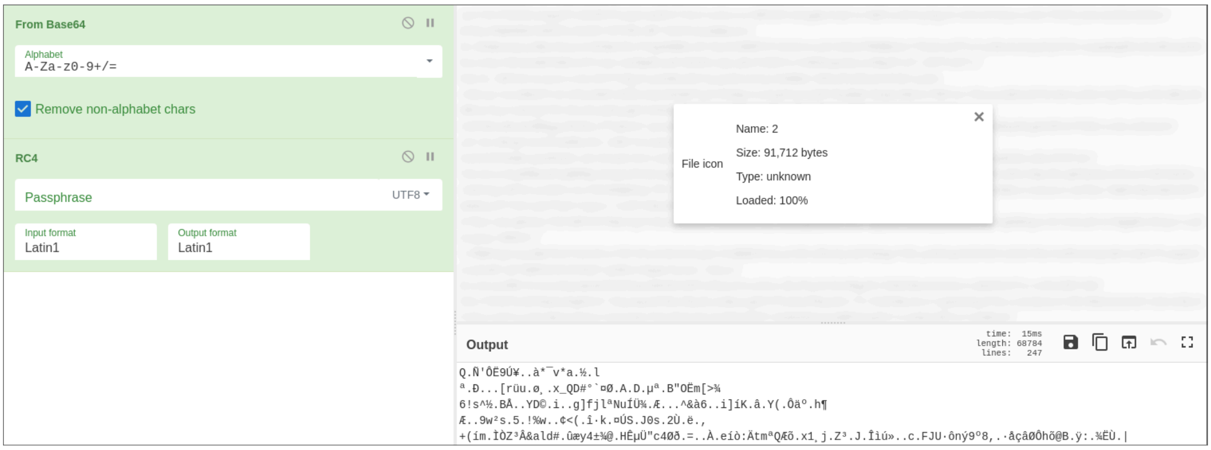
⚠ Output limited to printable characters (other chars replaced by �)

```
¥ùÑa�:���6Ôö'F�;�gðÜ�\[d�P©3�jf2îO�¶�ëâÔÑþº`½
ù Ö���;ày}
{+ôeC�Js¶�©]73º¹�2D¶b"���+�s÷�]hð&�,¦y�¯?
Xô�9�x�KzÓð»*pÈþÞ���Î:
 <kæÒ¬åL3^eÙ"%É]¾è�P~4x�ñ�9���çÕü�Á³lñ²±�G¡��1
�³36gàæ¹6»ãÇc�¿M��Íj�<��@÷�ÄkÔ��Üd�?
û�â���C��æ�(Ü@Ö£Ömgô��í��«Äi�Xe%/6ÊN3 S�
```

```
¤OnA$.Á4nÁ.Æ`&)¿¢ì³¤.SF°p.-pÏ::§±º$.ÇAHÝþõ.u.B.å.ãª|.6EàCz.
£p.&Xê.É\ï¿..Èz..Ó.pæê\Õæ9.Ë..¿RÒãõ÷}!àiT°üR
´-.Ál.÷$ó.Dô.DÙ.cÔh^{{`.ú+.?bÑí.zU.P#<o..x=Î..äº.3..
%.é¡ó3Àç.iR#Z.h¯ÅÀÒ¬´_ìC¶@./.÷H-.ò²¥ô.R9¦ÔÕèOà.Q|®|
Ā¯*Þ u.Þ.¥.»².×..7Ā...ëYG~%Ø.ÿ¥r#.
```

RESULTS FORMAT
● STRING OF PRINTABLE CHARACTERS (ASCII/UNICODE)
○ HEXADECIMAL 00-7F-FF
○ DECIMAL 0-127-255
○ OCTAL 000-177-377
○ BINARY 00000000-11111111
○ INTEGER NUMBER

# So I used RC4 from early :



From Base64

Alphabet
A-Za-z0-9+/=

☑ Remove non-alphabet chars

RC4

Passphrase UTF8

Input format: Latin1

Output format: Latin1

File icon

Name: 2

Size: 91,712 bytes

Type: unknown

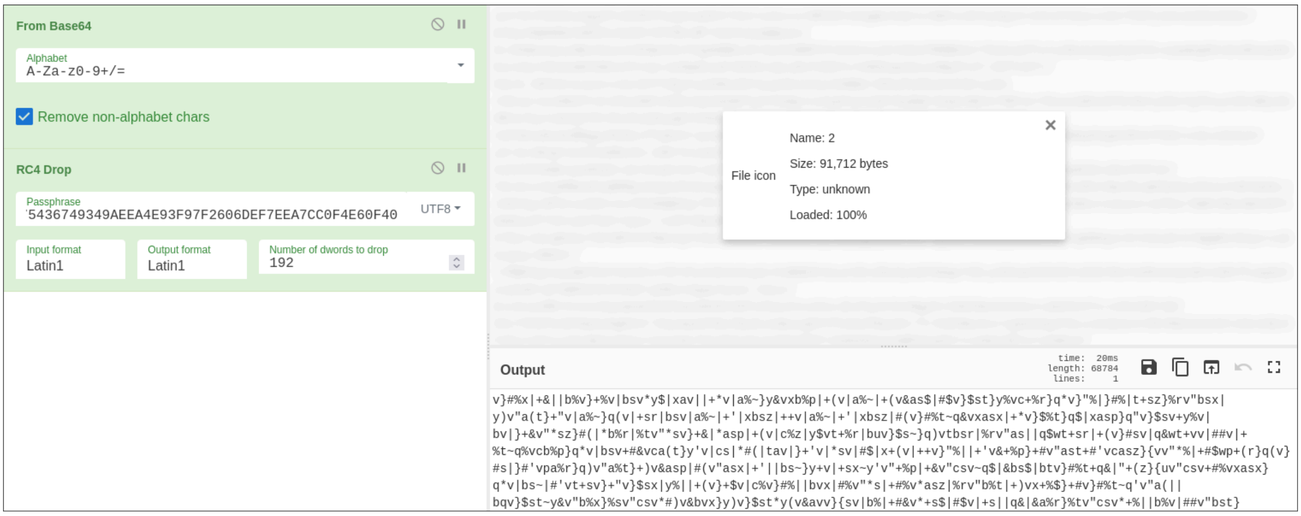
Loaded: 100%

Output

```
time:   15ms
length: 68784
lines:   247
```

```
Q.Ñ'ÔË9Ú¥..à*¯v*a.½.l
ª.Ð...[rüu.ø¸.x_QD#°`¤Ø.A.D.µª.B"OËm[>¾
6!s^½.BÅ..YD©.i..g]fjlªNuÍÜ¾.Æ...^&à6..i]íK.â.Y(.Ôäº.h¶
Æ..9w²s.5.!%w..¢<(.î·k.¤ÚS.J0s.2Ù.ë.,
+(ím.ÌÒZ³Â&ald#.ûæy4±¾@.HÊµÜ"c4Øð.=..À.eíò:ÄtmªQÆõ.x1¸j.Z³.J.Îìú»..c.FJU·ônýý9º8,.·åçâØÔhõ@B.ÿ:.¾ËÙ.|
```

# Still nothing! But I found on CyberChef a cipher named **RC4 Drop** with a passphrase and I put the Key from early. And I got something more clear, with way more normal characters :



From Base64

Alphabet
A-Za-z0-9+/=

☑ Remove non-alphabet chars

RC4 Drop

Passphrase
5436749349AEEA4E93F97F2606DEF7EEA7CC0F4E60F40 UTF8

Input format: Latin1

Output format: Latin1

Number of dwords to drop: 192

File icon

Name: 2

Size: 91,712 bytes

Type: unknown

Loaded: 100%

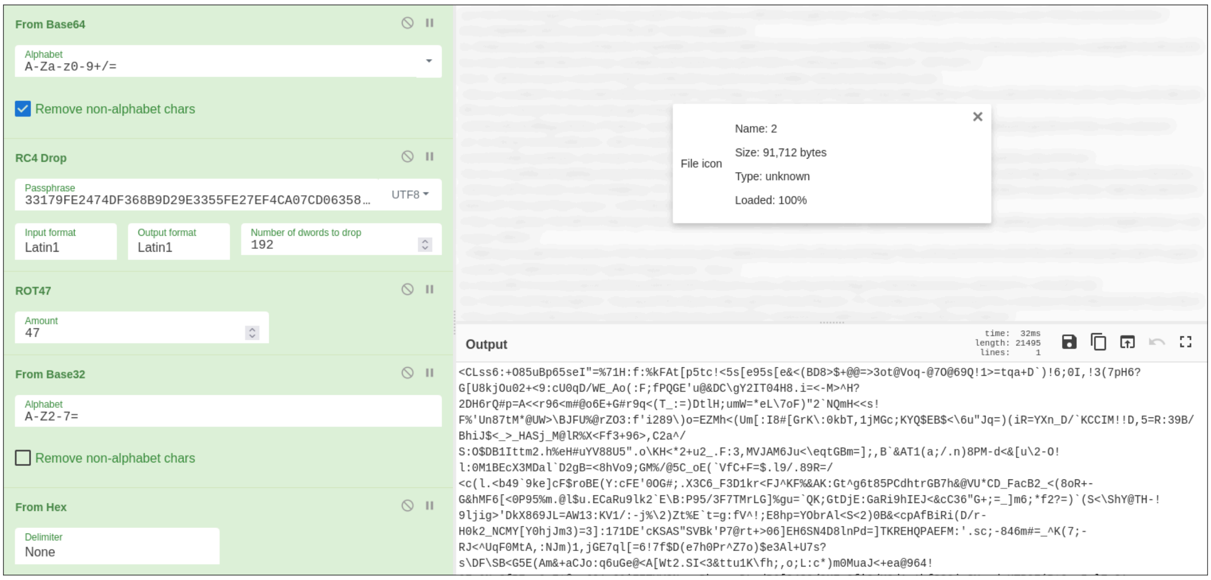
Output

```
time:   20ms
length: 68784
lines:     1
```

```
v}#%x|+&||b%v}+%v|bsv*y$|xav||+*v|a%~}y&vxb%p|+(v|a%~|+(v&as$|#$v}$st}y%vc+%r}q*v}"%|}#%|t+sz}%rv"bsx|
y)v"a(t}+"v|a%~}q(v|+sr|bsv|a%~|+'|xbsz|++v|a%~|+'|xbsz|#(v}#%t~q&vxasx|+*v}$%t}q$|xasp}q"v}$sv+y%v|
bv|}+&v"*sz}#(|*b%r|%tv"*sv}+&|*asp|+(v|c%z|y$vt+%r|buv}$s~}q)vtbsr|%rv"as||q$wt+sr|+(v}#sv|q&wt+vv|##v|+
%t~q%vcb%p}q*v|bsv+#&vca(t}y'v|cs|*#(|tav|}+'v|*sv|#$|x+(v|++v}"%||+'v&+%p}+#v"ast+#'vcasz}{vv"*%|+#$wp+(r}q(v}
#s|}#'vpa%r}q)v"a%t}+)v&asp|#(v"asx|+'||bs~}y+v|+sx~y'v"+%p|+&v"csv~q$|&bs$|btv}#%t+q&|"+(z}{uv"csv+#%vxasx}
q*v|bs~|#'vt+sv}+"v}$sx|y%||+(v}+$v|c%v}#%||bvx|#%v"*s|+#%v*asz|%rv"b%t|+)vx+%$}+#v}#%t~q'v"a(||
bqv}$st~y&v"b%x}%sv"csv*#)v&bvx}y)v}$st*y(v&avv}{sv|b%|+#&v*+s$|#$v|+s||q&|&a%r}%tv"csv*+%||b%v|##v"bst}
```

# Still confusing. In these situations when you got nothing valid on **https://www.dcode.fr/cipher-identifier (** or if you got a valid cipher the text it's corrupt **)**, you can use ROT13 or ROT47 when you get clearer text, because you can be on the right way, but there are rotated characters like early. Earlier I realized that Rot13 was used because the text obtained was very similar to the one for Hex and I saw some "q" character for example, which made me think that such a Cipher can be used.

# So, found out that the next cipher it’s ROT47 and CyberChef auto-completed until I got this result :



From Base64
Alphabet: A-Za-z0-9+/=
☑ Remove non-alphabet chars

RC4 Drop
Passphrase: 33179FE2474DF368B9D29E3355FE27EF4CA07CD06358… UTF8
Input format: Latin1
Output format: Latin1
Number of dwords to drop: 192

ROT47
Amount: 47

From Base32
Alphabet: A-Z2-7=
□ Remove non-alphabet chars

From Hex
Delimiter: None

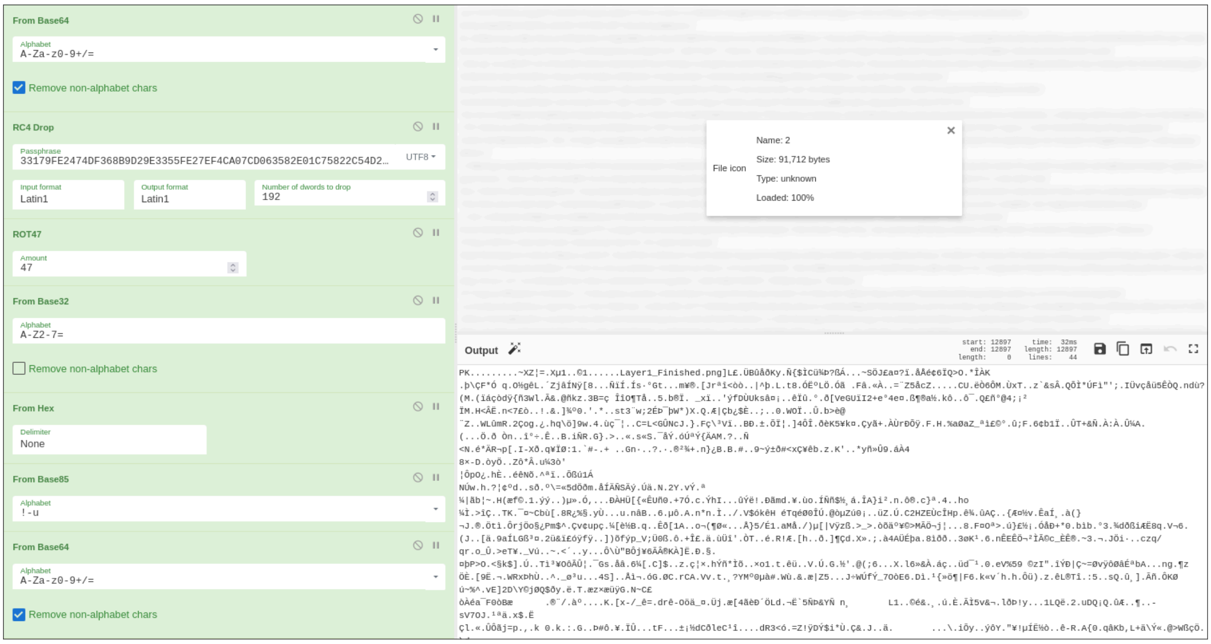
File icon
Name: 2
Size: 91,712 bytes
Type: unknown
Loaded: 100%

Output

```
time:   32ms
length: 21495
lines:      1
```

```
<CLss6:+O85uBp65seI"=%71H:f:%kFAt[p5tc!<5s[e95s[e&<(BD8>$+@@=>3ot@Voq-@7O@69Q!1>=tqa+D`)!6;0I,!3(7pH6?
G[U8kjOu02+<9:cU0qD/WE_Ao(:F;fPQGE'u@&DC\gY2IT04H8.i=<-M>^H?
2DH6rQ#p=A<<r96<m#@o6E+G#r9q<(T_:=)DtlH;umW=*eL\7oF)"2`NQmH<<s!
F%'Un87tM*@UW>\BJFU%@rZO3:f'i289\)o=EZMh<(Um[:I8#[GrK\:0kbT,1jMGc;KYQ$EB$<\6u"Jq=)(iR=YXn_D/`KCCIM!!D,5=R:39B/
BhiJ$<_>_HASj_M@lR%X<Ff3+96>,C2a^/
S:O$DB1Ittm2.h%eH#uYV88U5".o\KH<*2+u2_.F:3,MVJAM6Ju<\eqtGBm=];,B`&AT1(a;/.n)8PM-d<&[u\2-O!
l:0M1BEcX3MDal`D2gB=<8hVo9;GM%/@5C_oE(`VfC+F=$.l9/.89R=/
<c(l.<b49`9ke]cF$roBE(Y:cFE'0OG#;.X3C6_F3D1kr<FJ^KF%&AK:Gt^g6t85PCdhtrGB7h&@VU*CD_FacB2_<(8oR+-
G&hMF6[<0P95%m.@l$u.ECaRu9lk2`E\B:P95/3F7TMrLG]%gu=`QK;GtDjE:GaRi9hIEJ<&cC36"G+;=_]m6;*f2?=)`(S<\ShY@TH-!
9ljig>'DkX869JL=AW13:KV1/:-j%\2)Zt%E`t=g:fV^!;E8hp=YObrAl<S<2)0B&<cpAfBiRi(D/r-
H0k2_NCMY[Y0hjJm3)=3]:171DE'cKSAS"SVBk'P7@rt+>06]EH6SN4D8lnPd=]TKREHQPAEFM:'.sc;-846m#=_^K(7;-
RJ<^UqF0MtA,:NJm)1,jGE7ql[=6!7f$D(e7h0Pr^Z7o)$e3Al+U7s?
s\DF\SB<G5E(Am&+aCJo:q6uGe@<A[Wt2.SI<3&ttu1K\fh;,o;L:c*)m0MuaJ<+ea@964!
```

# I uploaded the new result on **https://www.dcode.fr/cipher-identifier** and I found the next cipher : Base85. Then it was clear Base64 and I got the archive :



From Base64
Alphabet: A-Za-z0-9+/=
☑ Remove non-alphabet chars

RC4 Drop
Passphrase: 33179FE2474DF368B9D29E3355FE27EF4CA07CD063582E01C75822C54D2… UTF8
Input format: Latin1
Output format: Latin1
Number of dwords to drop: 192

ROT47
Amount: 47

From Base32
Alphabet: A-Z2-7=
□ Remove non-alphabet chars

From Hex
Delimiter: None

From Base85
Alphabet: !-u

From Base64
Alphabet: A-Za-z0-9+/=
☑ Remove non-alphabet chars

File icon
Name: 2
Size: 91,712 bytes
Type: unknown
Loaded: 100%

Output

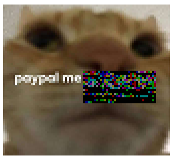
```
 start: 12897     time:  32ms
   end: 12897   length: 12897
length:     0    lines:    44
```

```
PK.........~XZ¦=.Xµ1..©1......Layer1_Finished.png]L£.ÜBûåðKy.Ñ{$ÌCü¾Þ?ßÁ...~SÖJ£a¤?ï.åÅé¢6ÏQ>O.*ÎÀK
.þ\ÇF*Ó q.O½gêL.´ZjâÍNÿ[8...ÑïÍ.Ís·ºGt...m¥®.[Jrªî<òò..|^þ.L.t8.ÓËºLÖ.Óã .Fâ.«À..=¨Z5åcZ.....CU.ëÒ6ÔM.ÙxT..z`&sÂ.QÕÌ*ÚFì"';.IÜvçåü5ÊÒQ.ndù?
(M.(ïáçòdÿ{ñ3Wl.Ã&.@ñkz.3B=ç ÎîO¶Tå..5.b®Ï. _xï..'ýfDÙUksâ¤¡..êÏû.°.ð[VeGUïI2+eº4e¤.ß¶®a½.kô..ô¯.Q£ñº@4;¡²
ÏM.H<ÂË.n<7£ò..!.&.]¾º0.'.*..st3¨w;2ÉÞ¯þW*)X.Q.Æ|Çb¿$È..;..0.WOÏ..Û.b>è@
¨Z..WLûmR.2Çog.¿.hq\ö]9w.4.ùç¯¦..C=L<GÛNcJ.}.Fç\³Vï..BÐ.±.ÕÏ¦.]4ÔÎ.ðèK5¥k¤.Çyã+.ÀÙrÐÕÿ.F.H.%aØaZ_ªi£©°.û;F.6¢b1Ï..ÛT+&Ñ.À:À.Û¼A.
(...Ö.ð Òn..îº÷.Ê..B.iÑR.G}.>..«.s«S.¯åÝ.óÚªÝ{ÄAM.?..Ñ
<N.é*ÄR¬p[.I-Xð.q¥ÏØ:1.`#-.+ ..Gn·..?.·.®²¾+.n}¿B.B.#..9~ý±ð#<xÇ¥êb.z.K'..*yñ»Û9.áÀ4
8×-D.òyÖ..Zô*Â.u¾3ò'
¦ÔpO¿.hÈ..éêNõ.^ªï..ÕßúlÁ
NÚw.h.?¦¢ºd..sð.º\=«5dÕðm.åÍÄÑSÄý.Úä.N.2Y.vÝ.ª
¼|ãb¦~.H(æf©.1.ýý..)µ».ó,...ÐÀHÜ[{«ÊUñ0.+7Ó.c.ÝhI...ûÝë!.Ðãmd.¥.ùo.ÍÑñ$½¸á.ÎA}i².n.ô®.c}ª.4..ho
¼Ì.>îÇ..TK.¯¤~Cbù[.8R¿%§.yÙ...u.nâB..6.µô.A.n*n.Ì../.V$ókêH éTqéØ0ÎÚ.@òµZú0¡..üZ.Ú.C2HZEÙcÎHp.ê¾.ûAÇ..{Æ¤½v.ÊaÍ¸.à(}
¬J.®.Õti.ÔrjÖo§¿Pm$^.Çv¢upç.¼[è½B.q..Êð[1A..o¬(¶Ø«...Å}5/É1.aMå./)µ[|Vÿzß.>_>.òõäº¥©>MÃÖ¬j¦...8.F¤Oª>.ú}£½¡.ÓåÐ+*0.bìb.°3.%dõßiÆË8q.V¬6.
(J..[ä.9aÍLGß³¤.2ü&ï£óÿfÿ..])õfýp_V;Ü0ß.ô.+Î£.ä.ùÜï'.ÒT..é.R!Æ.[h..ð.]¶Çd.X».;.à4AÜÉþa.8iðð..3øK¹.6.nÊÉÊÖ¬²ÌÃ©c_ÈÊ®.~3.¬.JÕi·..czq/
qr.o_Û.>eT¥._Vú..~.<´..y...Ô\Ù"BÔj¥6ÃÂ®KÀ]Ë.Ð.§.
¤þP>O.<§k$].Ú..Ti³¥OôÂÛ¦.¯Gs.åâ.6¼[.C]$..z.ç¦×.hÝñ*Ìõ..×o1.t.êü..V.Ú.G.½'.@(;6...X.l6»&À.áç..üd¯¹.0.eV%59 ©zI".ïÝÐ|Ç~=ØvÿôØâÉªbA...ng.¶z
ÖÈ.[9Ë.¬.WRxÞhÙ..^._ø³u...4S]..Åi¬.óG.ØC.rCA.Vv.t.¸?YMº0µà#.Wù.&.æ|Z5...J÷WÚfÝ_7OòE6.Dì.¹{»ö¶|F6.k«v´h.h.Ôü).z.êL®Tî.:5..sQ.û¸].Ãñ.ÔKØ
ú~%^.vE]2D\Y©jØQ$ðy.ë.T.æz×æüÿG.N~C£
òÀéa¯F0òBæ      .®¨/.àº....K.[x-/_ê=.drê-Oöä_¤.Üj.æ[4ãèÐ´ÖLd.¬Ë`5ÑÞ&YÑ n¸        L1..©é&.¸.ú.È.ÃÌ5v&¬.lðÞ!y...1LQë.2.uDQ¡Q.ûÆ..¶..-
sV7OJ.¹ªä.x$.Ë
Çl.«.ÛÔãj=p.,.k 0.k.:.G..Þ#ô.¥.ÏÛ...tF...±¡½dCðleC¹î....dR3<ó.=Z!ÿDÝ$i*Ù.Ç&.J..ä.       ...\.iõy..ýôY."¥!µÍË½ò..ê-R.A{0.qâKb,L+ä\Ý«.@>WßçÖ.
```

# The ciphers used : from Base64 -> from RC4-Drop (passphrase the Key from early), from ROT47 -> from Base32 -> from Hex -> from Base85 -> from Base64

# Now we got an archive. If we try to decompress it, we can see it’s protected with a password. I randomly tried the password from RC4 and it worked 😂 from the first try

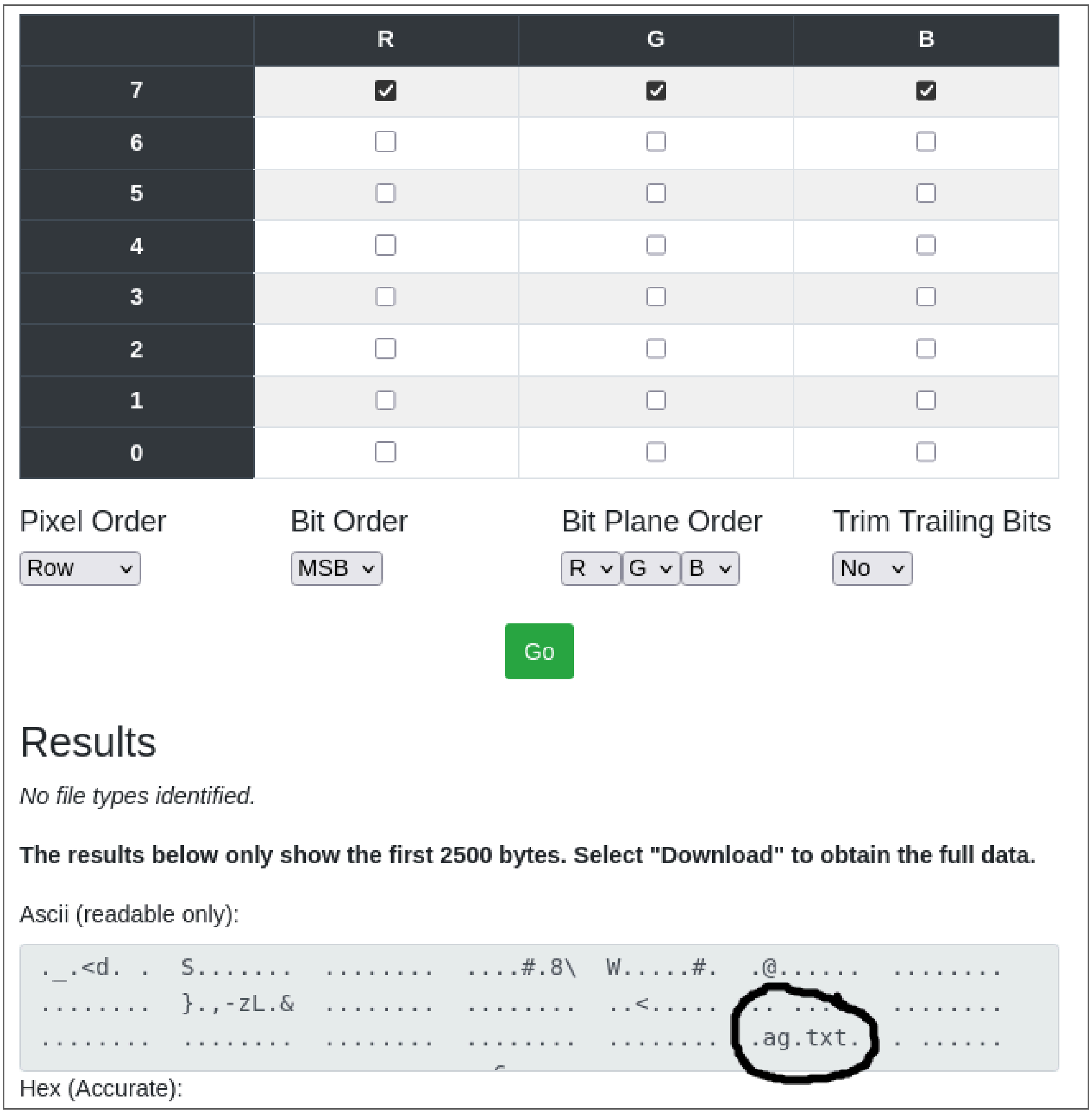
# Now we got this image :



# Instantly I uploaded it on **https://aperisolve.com** and I found nothing. The thing is, there is an output at zsteg so we need to use again LSB/MSB 😒 cmon MettleSphee

# Again, I uploaded the image on **https://www.georgeom.net/StegOnline/upload** and I did LSB/MSB

# AGAIN, from the first try, I got this hint (from **Layer1_Finished.png** image, the one with the cat) : there is an archive hidden in this image :



	R	G	B
7	☑	☑	☑
6	□	□	□
5	□	□	□
4	□	□	□
3	□	□	□
2	□	□	□
1	□	□	□
0	□	□	□

Pixel Order: Row

Bit Order: MSB

Bit Plane Order: R G B

Trim Trailing Bits: No

Results

No file types identified.

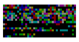
The results below only show the first 2500 bytes. Select "Download" to obtain the full data.

Ascii (readable only):

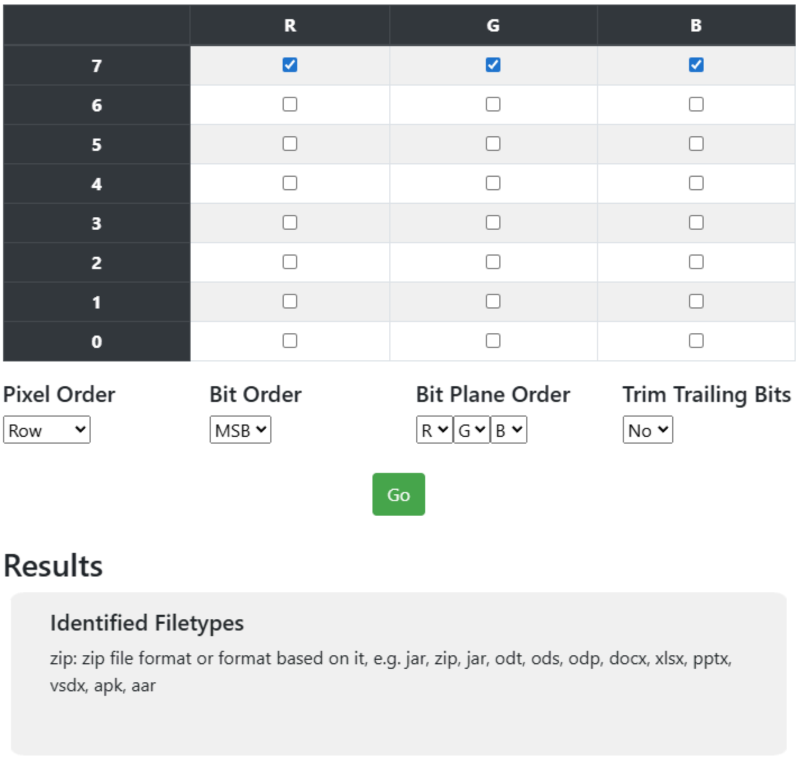
```
._.<d. .  S.......  ........  ....#.8\  W.....#.  .@......  ........
........  }.,-zL.&  ........  ........  ..<.....  ..  ...   ........
........  ........  ........  ........  ........  .ag.txt.  . ......
```

Hex (Accurate):

# Now, what? so there is a **flag.txt** file. I tried every possible combination with LSB/MSB, also on CyberChef. So I decided to cut perfectly the chunk of corrupt pixels on Windows, paint and to use LSB/MSB on it, because **https://aperisolve.com** it's useless :



# I uploaded the trimmed image on **https://georgeom.net/StegOnline**



	R	G	B
7	☑	☑	☑
6	□	□	□
5	□	□	□
4	□	□	□
3	□	□	□
2	□	□	□
1	□	□	□
0	□	□	□

Pixel Order: Row

Bit Order: MSB

Bit Plane Order: R G B

Trim Trailing Bits: No

Results

Identified Filetypes

zip: zip file format or format based on it, e.g. jar, zip, jar, odt, ods, odp, docx, xlsx, pptx, vsdx, apk, aar

# We found the archive 🥳🥳🥳🥳🥳🥳🥳 if we decompress this, we got the flag

THE FLAG :
CTF{1d97814524da2a33d9d709a04250c679dc1429cbbd6362023f8b3b130fbae28a}
~Z4que