

ECE 471 Lab 1

Secret-Key Encryption Lab

Mitchell Dzurick

2/3/2020

Github with all documentation - <https://www.github.com/mitchdz/ECE471>

Contents

1	Overview	2
2	Lab Tasks	3
2.1	Task 1: Frequency c Against Monoalphabetic Substitution Cipher	3
2.1.1	Task 1: solution	3
2.2	Task 2: Encryption using Different Ciphers and Modes	21
2.2.1	Task2: solution	21
2.3	Task 3: Encryption Mode – ECB vs. CBC	26

Secret Key Encryption Lab

Copyright © 2018 Wenliang Du, Syracuse University. The development of this document was partially funded by the National Science Foundation under Award No. 1303306 and 1718086. This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. A human-readable summary of (and not a substitute for) the license is the following: You are free to copy and redistribute the material in any medium or format. You must give appropriate credit. If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original. You may not use the material for commercial purposes.

1 Overview

The learning objective of this lab is for students to get familiar with the concepts in the secret-key encryption. After finishing the lab, students should be able to gain a first-hand experience on encryption algorithms, encryption modes, paddings, and initial vector (IV). Moreover, students will be able to use tools and write programs to encrypt/decrypt messages. This lab covers the following topics:

- Secret-key encryption
- Substitution cipher and frequency analysis
- Encryption modes and paddings
- Programming using the crypto library

Lab Environment. This lab has been tested on our pre-built Ubuntu 12.04 VM and Ubuntu 16.04 VM, both of which can be downloaded from the SEED website.

2 Lab Tasks

2.1 Task 1: Frequency c Against Monoalphabetic Substitution Cipher

It is well-known that monoalphabetic substitution cipher (also known as monoalphabetic cipher) is not secure, because it can be subjected to frequency analysis. In this lab, you are given a cipher-text that is encrypted using a monoalphabetic cipher; namely, each letter in the original text is replaced by another letter, where the replacement does not vary (i.e., a letter is always replaced by the same letter during the encryption). Your job is to find out the original text using frequency analysis. It is known that the original text is an English article.

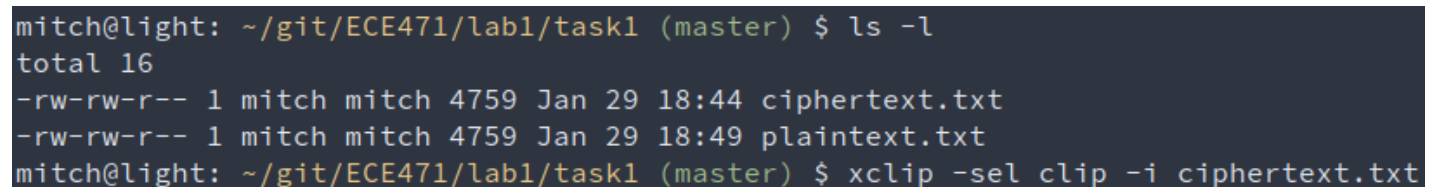
2.1.1 Task 1: solution

Shown below is the linux command

```
$ ls -l
```

in the lab1/task1 directory of my github repository. Inside that repository, the following command is used to copy the contents of the ciphertext into my X clipboard:

```
$ xclip -sel clip -i ciphertext.txt
```



```
mitch@light: ~/git/ECE471/lab1/task1 (master) $ ls -l
total 16
-rw-rw-r-- 1 mitch mitch 4759 Jan 29 18:44 ciphertext.txt
-rw-rw-r-- 1 mitch mitch 4759 Jan 29 18:49 plaintext.txt
mitch@light: ~/git/ECE471/lab1/task1 (master) $ xclip -sel clip -i ciphertext.txt
```

Figure 1: listing of lab1/task1 directory

The contents of ciphertext.txt is copied below:

```
ytn xqavhq yzhu xu qzupvd ltmat qnncq vgxzy hmrtv vbynh ytmq ixur qyhvurn
vlvhpq yhme ytn gvrrnh bnniq imsn v uxuvrnuvhmvu yxx
```

```
ytn vlvhpq hvan lvq gxxsnupnp gd ytn pncmqn xb tvhfnd lnmuqynmu vy myq xzyqny
vup ytn veevhnuv mceixqmxu xb tmq bmie axcevud vy ytn nup vup my lvq qtvenp gd
ytn ncnhrnuan xb cnyxx ymcnq ze givasrxlu eximymaq vhcavupd vaymfmqc vup
v uvymxuvi axufnhqvymxu vq ghmbn vup cvp vq v bnfnh phnvc vgxzy ltnytnh ytnhn
xzrty yx gn v ehmqmpnuv lmubhnd ytn qnvqxu pmpuy ozqy qnnc nkyhv ixur my lvq
nkyhv ixur gnavzqn ytn xqavhq lhn cxfnp yx ytn bmhgy lnnsnup mu cvhat yx
vfxmp axubimaymur lmyt ytn aixqmur anhnxcud xb ytn lmuynh xidcemaq ytvusq
ednxuratvur
```

```
xun gmr jznqymxu qzhhxzupmur ytmq dnvhq vavpncd vlvhpq mq txl xh mb ytn
```

anhncxud lmii vpphnqq cnyxx nqenamviid vbynh ytn rxipnu rixgnq ltmat gnavcn
v ozgmivuy axcmurxzy evhyd bxh ymcnq ze ytn cxfnucny qenvhtnvpnp gd
exlnhbzi txiidlxxp lxcnu ltx tnienp hvmqn cmiimxuq xb pxiihvq yx bmrty qnkzvi
tvhvqqcnuy vhxzup ytn axzuyhd

qmruvimur ytnmh qzeexhy rxipnu rixgnq vyynupnnq qlvytnp ytnqcniifnq mu givas
qexhynp iveni emuq vup qxzupnp xbb vgxzy qnkmqy exlnh mcgvivuanq bhxc ytn hnp
avheny vup ytn qyvrn xu ytn vmh n lvq aviinp xzy vgxzy evd munjzmyd vbynh
myq bxhcnh vuatxh avyy qvpinh jzmy xuan qtn invhunp ytvty qtn lvq cvsmur bvh
inqq ytvu v cvin axtxqy vup pzhmur ytn anhncxud uvyvimn exhycvu yxxs v gizuy
vup qvymqbdmur pmr vy ytn viicvin hxqynh xb uxcmuvynp pmhnayxhq txl axzip
ytvy gn yxeenp

vq my yzhuq xzy vy invqy mu ynhcq xb ytn xqavhq my ehxgvgid lxuy gn

lxcnu mufxifnp mu ymcnq ze qvmp ytvty viytxzrt ytn rixgnq qmrumbmnp ytn
mumymvymfnq ivzuat ytnu unfnh muynupnp my yx gn ozqy vu vlvhpq qnvqxu
avcevmru xh xun ytvty gnavcn vqxxamvynp xuid lmyt hnpavheny vaymxuq muqynvp
v qexsnqlxcvu qvmp ytn rhxze mq lxhsmur gntmup aixqnp pxxhq vup tvq qmuan
vcvqqnp cmiimxu bxh myq inrvi pnbnuqn bzip ltmat vbynh ytn rixgnq lvq
bixxnp lmyt ytxzqvupq xb pxuvymxuq xb xh inqq bhxc enxein mu qxcn
axzuyhmnq

ux avii yx lnhv givas rxluq lnuy xzy mu vpfvuan xb ytn xqavhq ytxzrt ytn
cxfnucny lmii vicxqy anhyvmuid gn hnbhnuanp gnbxhn vup pzhmur ytn anhncxud
nqenamviid qmuan fxavi cnyxx qzeexhynhq imsn vqtind ozpp ivzhv pnhu vup
umaxin smpcvu vhn qatnpzinp ehqnquynhq

vuxytnh bnvyzhn xb ytmq qnvqxu ux xun hnviid suxlq ltx mq rxmur yx lmu gnqy
emayzhn vhrzvqid ytmq tveenuq v ixy xb ytn ymcn muvhrzvqid ytn uvmigmyh
uvhhvymfn xuid qnhfnq ytn vlvhpq tden cvatmun gzy xbynu ytn enxein bxhnavqymur
ytn hvan qxaviinp xqavhxixrmqyq avu cvsn xuid npzavynp rznqqnq

ytn lvd ytn vavpncd yvgzivynq ytn gmr lmuunh pxnquy tnie mu nfnhd xytnh
avynrxhd ytn uxcmunn lmyt ytn cxqy fxynq lmuq gzy mu ytn gnqy emayzhn
avynrxhd fxynhq vhn vqsnq yx imqy ytnmh yxe cxfmnq mu ehnbhnuymvi xhpnh mb v
cxfmn rnyq cxhn ytvu enhanuy xb ytn bmhqeivan fxynq my lmuq ltnu ux
cxfmn cvuvrnq ytvty ytn xun lmyt ytn bnlqy bmhqeivan fxynq mq nimcmuvynp vup
myq fxynq vhn hnpmqyhmgyznp yx ytn cxfmnq ytvty rvhunhnp ytn nimcmuvynp gviixyq
qnaxupeivan fxynq vup ytmq axuymuznq zuymi v lmuunh ncnhrnq

my mq vii ynhhmgid axubzqmur gzy veevhnuyid ytn axuqnuqzq bvfxhmyn axcnq xzy
vtntp mu ytn nup ytmq cnvuq ytvty nupxbqnvqxu vlvhpq atvyynh mufvhmvgid
mufxifnq yxhyzhnp qenazivymxu vgxzy ltmat bmic lxzip cxqy imsnid gn fxynhq

qnaxup xh ytmhp bvfxhmy n vup ytnu njzviid yxhyzhnp axuaizqmxuq vgxzy ltmat
bmhc cmrty ehmfvmi

mu my lvq v yxqqze gnylnnu gxdtxxp vup ytn nfnuyzvi lmuunh gmhpcvu
mu lmyt ixyq xb nkenhyq gnyymur xu ytn hnfuvuy xh ytn gmr qtxhy ytn
ehmwn lnuy yx qexyimrty ivqy dnvh unvhid vii ytn bxhnavqynhq pnaivhnp iv
iv ivup ytn ehmqzceymfn lmuunh vup bxh ylx vup v tvib cmuzynq ytn d lnhn
axhhnay gnbxhn vu nufnixon quvzb lvq hnfvinp vup ytn hmrtzbzi lmuunh
cxuimrty lvq ahxlunp

ytmq dnvh vlvhpq lvyatnhq vhn zunjzviid pmfmpnp gnylnnu ythnn gmiigxvhpq
xzyqmpn nggmur cmqxxzhm ytn bvfxhmy n vup ytn qtven xb lvynh ltmat mq
ytn gvrnrhq ehnpmaymxu lmyt v bnl bxhnavqymur v tvmi cvhd lmu bxh rny xzy

gzy vii xb ytxqn bmhcq tvfn tmqyxhmavi xqavhfxymur evyynhuq vrvmuqy ytn c ytn
qtven xb lvynh tvq uxcmuvymxuq cxhn ytvu vud xytnh bmhc vup lvq viqx
uvcnp ytn dnvhq gnqy gd ytn ehxpzanhq vup pmhnayxhq rzmipq dny my lvq uxy
uxcmuvynp bxh v qahnnu vayxhq rzmip vlvhp bxh gnqy nuqncgin vup ux bmhc tvq
lxu gnqy emayzhn lmytxzy ehnmfxzqid ivupmur vy invqy ytn vayxhq uxcmuvymxu
qmuan ghvftntvhy mu ytmq dnvh ytn gnqy nuqncgin qvr nupnp ze rxmur yx
ythnn gmiigxvhpq ltmat mq qmrumbmavuy gnavzqn vayxhq cvsn ze ytn vavpncdq
ivhrnqy ghvuat ytv y bmhc ltmin pmfmqmf n viqx lxu ytn gnqy phvcv rxipnu rixgn
vup ytn gvbyv gzy myq bmiccvsnh cvhymu capxuvrt lvq uxy uxcmuvynp bxh gnqy
pmhnayxh vup vevhy bhxc vhrx cxfmnq ytv y ivup gnqy emayzhn lmytxzy viqx
nvhumur gnqy pmhnayxh uxcmuvymxuq vhn bnl vup bv h gnylnnu

The related plaintext is shown below using the final (substitution) key ‘abcdefghijklmnopqrstuvwxyz’ to ‘cfmypvbrlqxwiejdsgkhnazotu’. The process to derive this key is outlined below with accompanying pictures.

```
mitch@light: ~/git/ECE471/lab1/task1 (master) $ tr 'abcdefghijklmnopqrstuvwxyz' 'cfmypvbrlqxwiejdsgkhnazotu' < ciphertext.txt > plaintext.txt
```

Figure 2: Command to convert ciphertext to plaintext

Figure 2 shows the linux command used to convert the ciphertext that was shown into the accompanying plaintext.

the oscars turn on sunday which seems about right after this long strange
awards trip the bagger feels like a nonagenarian too

the awards race was bookended by the demise of harvey weinstein at its outset
and the apparent implosion of his film company at the end and it was shaped by
the emergence of metoo times up blackgown politics armcandy activism and
a national conversation as brief and mad as a fever dream about whether there
ought to be a president winfrey the season didnt just seem extra long it was

extra long because the oscars were moved to the first weekend in march to avoid conflicting with the closing ceremony of the winter olympics thanks pyeongchang

one big question surrounding this years academy awards is how or if the ceremony will address metoo especially after the golden globes which became a jubilant comingout party for times up the movement spearheaded by powerful hollywood women who helped raise millions of dollars to fight sexual harassment around the country

signaling their support golden globes attendees swathed themselves in black sported lapel pins and sounded off about sexist power imbalances from the red carpet and the stage on the air e was called out about pay inequity after its former anchor catt sadler quit once she learned that she was making far less than a male cohost and during the ceremony natalie portman took a blunt and satisfying dig at the allmale roster of nominated directors how could that be topped

as it turns out at least in terms of the oscars it probably wont be

women involved in times up said that although the globes signified the initiatives launch they never intended it to be just an awards season campaign or one that became associated only with redcarpet actions instead a spokeswoman said the group is working behind closed doors and has since amassed million for its legal defense fund which after the globes was flooded with thousands of donations of or less from people in some countries

no call to wear black gowns went out in advance of the oscars though the movement will almost certainly be referenced before and during the ceremony especially since vocal metoo supporters like ashley judd laura dern and nicole kidman are scheduled presenters

another feature of this season no one really knows who is going to win best picture arguably this happens a lot of the time inarguably the nailbiter narrative only serves the awards hype machine but often the people forecasting the race socalled oscarologists can make only educated guesses

the way the academy tabulates the big winner doesnt help in every other category the nominee with the most votes wins but in the best picture category voters are asked to list their top movies in preferential order if a movie gets more than percent of the firstplace votes it wins when no movie manages that the one with the fewest firstplace votes is eliminated and its votes are redistributed to the movies that garnered the eliminated ballots

secondplace votes and this continues until a winner emerges

it is all terribly confusing but apparently the consensus favorite comes out ahead in the end this means that endofseason awards chatter invariably involves tortured speculation about which film would most likely be voters second or third favorite and then equally tortured conclusions about which film might prevail

in it was a tossup between boyhood and the eventual winner birdman in with lots of experts betting on the revenant or the big short the prize went to spotlight last year nearly all the forecasters declared la la land the presumptive winner and for two and a half minutes they were correct before an envelope snafu was revealed and the rightful winner moonlight was crowned

this year awards watchers are unequally divided between three billboards outside ebbing missouri the favorite and the shape of water which is the baggers prediction with a few forecasting a hail mary win for get out

but all of those films have historical oscarvoting patterns against them the shape of water has nominations more than any other film and was also named the years best by the producers and directors guilds yet it was not nominated for a screen actors guild award for best ensemble and no film has won best picture without previously landing at least the actors nomination since braveheart in this year the best ensemble sag ended up going to three billboards which is significant because actors make up the academys largest branch that film while divisive also won the best drama golden globe and the bafta but its filmmaker martin mcdonagh was not nominated for best director and apart from argo movies that land best picture without also earning best director nominations are few and far between

After the ciphertext is copied, a program is utilized that is supplied by cryptoclub.org. The domain is shown below.

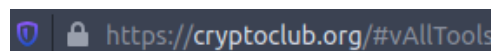


Figure 3: cryptoclub website

Below is the page you should see upon opening the URL in Figure 3



Figure 4: cryptoclub main page

The contents of the ciphertext are copied into the program as shown in Figure 5. On this page, the ciphertext shows the frequency of letters and relates the frequency of letters in the ciphertext to the frequency of letters in the English alphabet.



Figure 5: Crack Substitution Tool main page

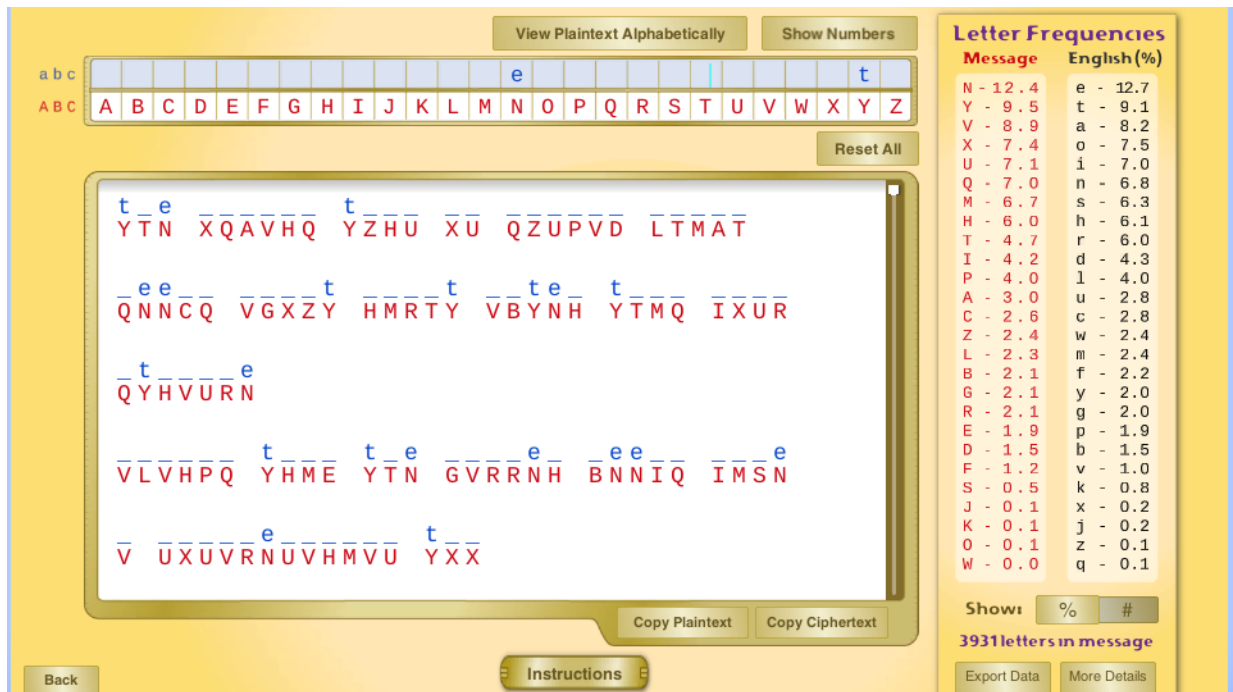


Figure 6: Mapping N to e and Y to t

In Figure 6 N is mapped to 'e' and Y is mapped to 't' due to frequency analysis (You can see that N matches e and Y matches t in frequency on the right). It is very easy then to notice that T maps to 'h' shown in Figure 7 to create the Trigram 'the'.

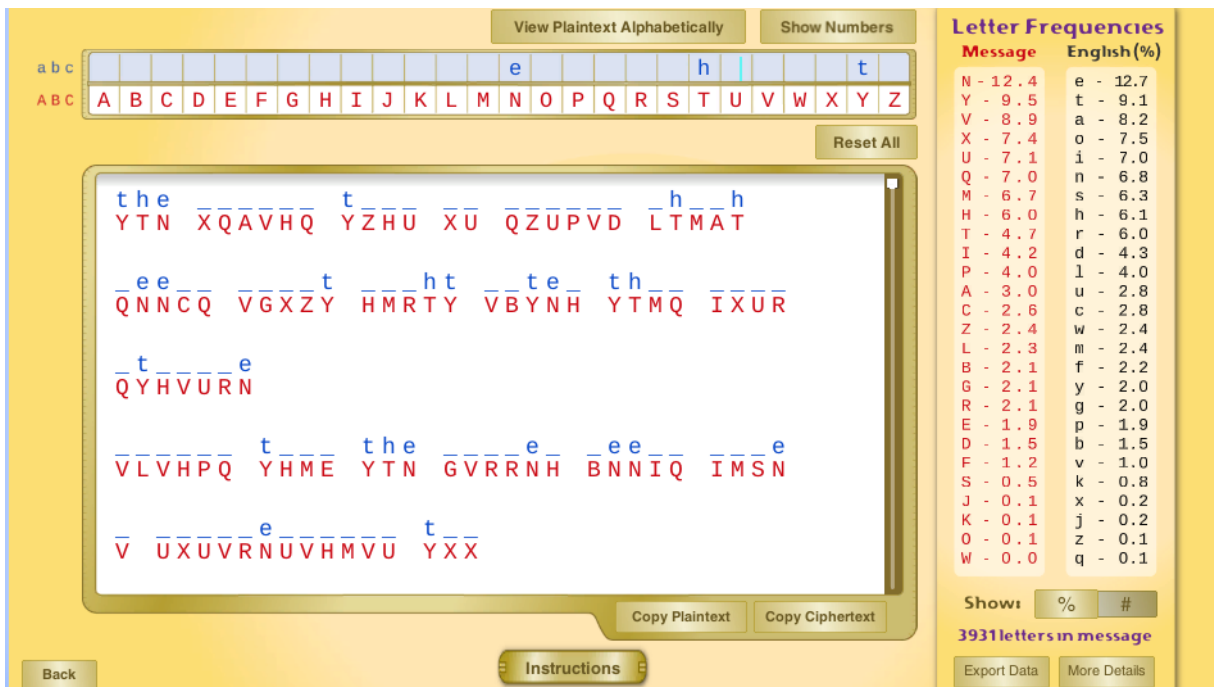


Figure 7: Mapping T to h

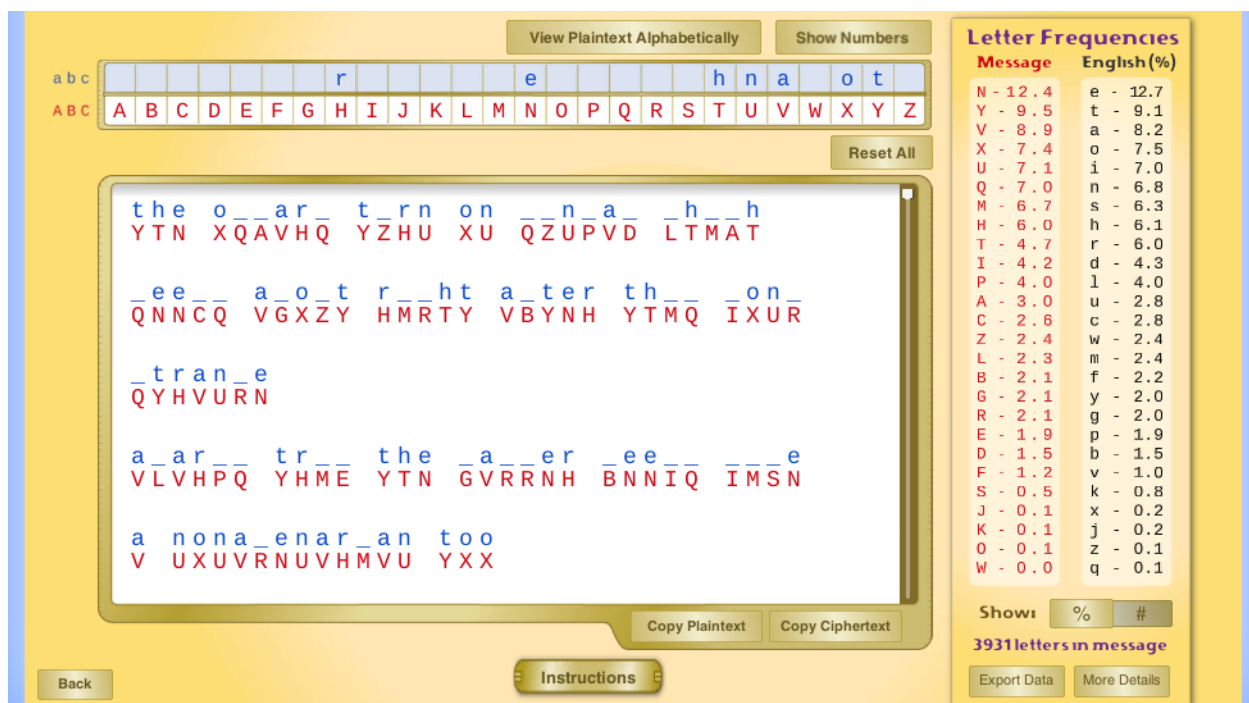


Figure 8: Mapping U to n, V to a, X to o, and H to r

Figure 8 shows how the ciphertext letters 'UVXH' are mapped to 'naor'. This mapping was done by looking at the ciphertext frequency and mapping the frequency to the closest related

frequency in the english alphabet. Words do not particularly make sense yet, so it is just assumed for now that this is the correct mapping.

The screenshot shows a web-based cryptanalysis tool. At the top, there are buttons for 'View Plaintext Alphabetically' and 'Show Numbers'. Below these is a grid for mapping the alphabet (A-Z) to another set of letters. The current mapping shows 'r' for 'A', 'w' for 'B', 'e' for 'C', 'h' for 'D', 'n' for 'E', 'a' for 'F', and 'o' for 'G'. Below the grid is a 'Reset All' button. The main area displays a ciphertext: 'VUP CVP VQ V BNFNH PHNV C VGXZY'. Below this, the ciphertext is decrypted into a readable form: 'whether there'. The next line of ciphertext is 'XZRTY YX GN V EHNQMPNUY LMUBHND', which decrypts to 'o _ht to _e a _re _ent w _n _re _'. The next line is 'the _ea _on _ _nt _ _t _ee _ e _tra', which decrypts to 'the _ea _on _ _nt _ _t _ee _ e _tra'. The next line is '_on _ _t wa _', which decrypts to '_on _ _t wa _'. The next line is 'IXUR MY LVQ', which decrypts to 'IXUR MY LVQ'. At the bottom, there are buttons for 'Copy Plaintext' and 'Copy Ciphertext'. On the right side, there is a 'Letter Frequencies' table comparing the message frequencies to English frequencies. The message frequencies are: N - 12.4, Y - 9.5, V - 8.9, X - 7.4, U - 7.1, Q - 7.0, M - 6.7, H - 6.0, T - 4.7, I - 4.2, P - 4.0, A - 3.0, C - 2.6, Z - 2.4, L - 2.3, B - 2.1, G - 2.1, R - 2.1, E - 1.9, D - 1.5, F - 1.2, S - 0.5, J - 0.1, K - 0.1, O - 0.1, W - 0.0. The English frequencies are: e - 12.7, t - 9.1, a - 8.2, o - 7.5, i - 7.0, n - 6.8, s - 6.3, h - 6.1, r - 6.0, d - 4.3, l - 4.0, u - 2.8, c - 2.8, w - 2.4, m - 2.4, f - 2.2, y - 2.0, g - 2.0, p - 1.9, b - 1.5, v - 1.0, k - 0.8, x - 0.2, j - 0.2, z - 0.1, q - 0.1. At the bottom right, there are buttons for 'Show: % #', '3931 letters in message', 'Export Data', and 'More Details'.

Figure 9: mapping L to w

In Figure 9 the string ‘hether’ was shown, which clearly meant to spell ‘whether’. Therefore, L maps to w.

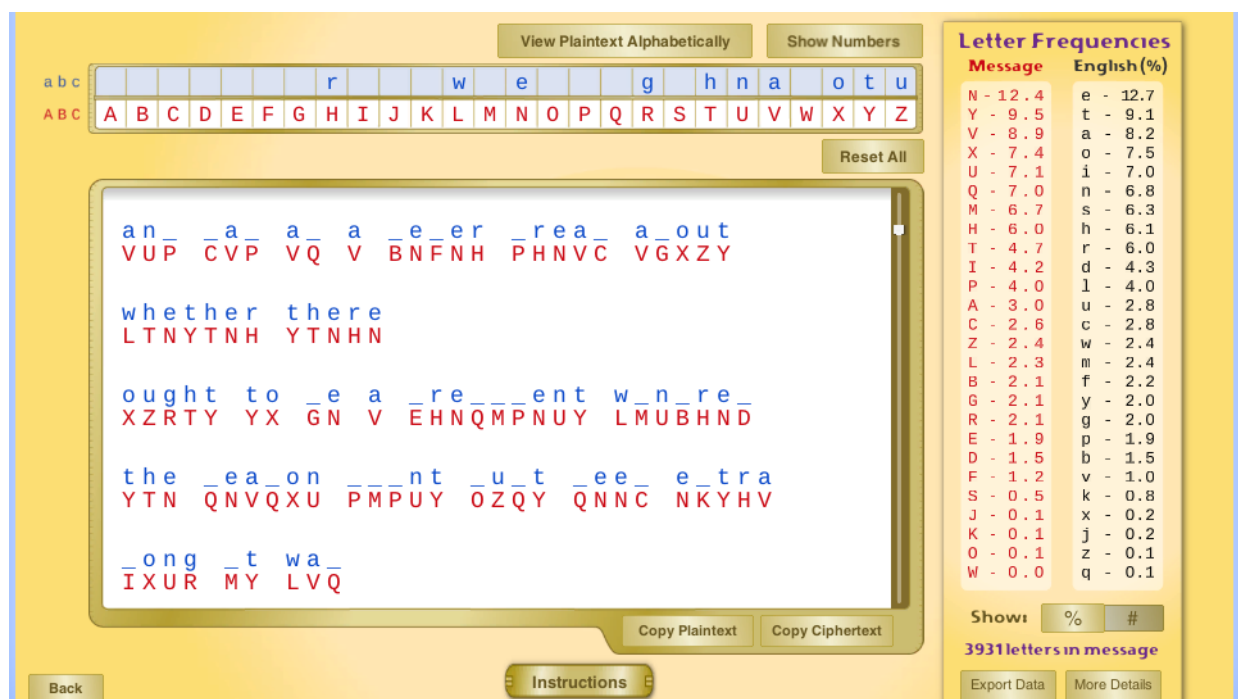


Figure 10: Mapping R to g

In Figure 10 the string 'ou-ht' is shown, which is meant to say 'ought'. Therefore, R maps to g.

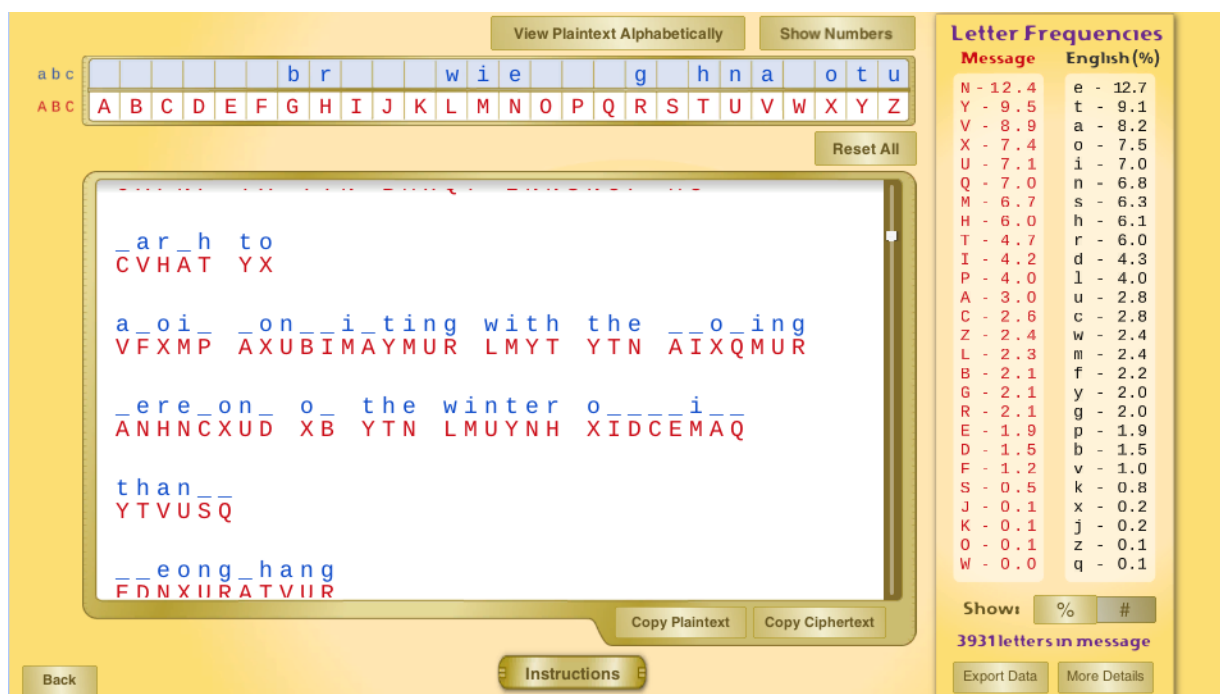


Figure 11: Mapping G to b and M to i

In Figure 11 The string 'w-nter' was shown, which should mean 'winter', Thus M maps to i. The mapping for G to b is not shown in this picture, but was done in a very similar fashion with another word.

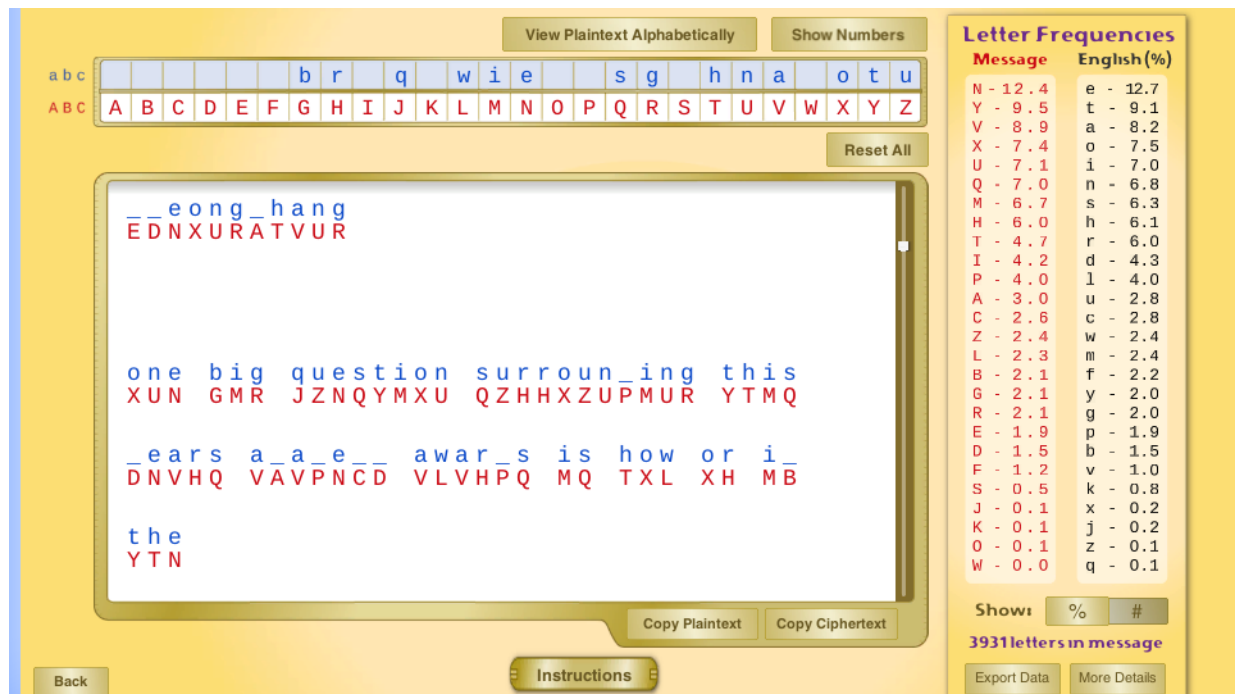


Figure 12: Mapping J to q and Q to s

In Figure 12 The string '-ue-tion' was present, which obviously meant 'question', therefore J maps to q Q maps to s.

The screenshot shows a cryptanalysis tool interface. At the top, there are buttons for "View Plaintext Alphabetically" and "Show Numbers". Below these is a keyboard layout with letters A-Z. The main text area displays the following ciphertext and its corresponding plaintext:

```

_ess than a _a_e _ohost and during
INQQ YTVU V CVIN AXTXQY VUP PZHMUR

the _ere_on_ nata_ie _ort_an too_
YTN ANHNCXUD UVYVIMN EXHYCVU YXXS

a b_unt
V GIZUY

and satis__ing dig at the a__a_e
VUP QVYMQBDMUR PMR VY YTN VIICVIN

roster o_ no_inated dire_tors how
HXQYNH XB UXCMUVYNP PMHNAYXHQ TXL

```

At the bottom of the text area are buttons for "Copy Plaintext" and "Copy Ciphertext". Below the text area is a "Back" button and an "Instructions" button. On the right side, there is a "Letter Frequencies" table comparing the message frequencies to English frequencies.

Message	English (%)
N - 12.4	e - 12.7
Y - 9.5	t - 9.1
V - 8.9	a - 8.2
X - 7.4	o - 7.5
U - 7.1	i - 7.0
Q - 7.0	n - 6.8
M - 6.7	s - 6.3
H - 6.0	h - 6.1
T - 4.7	r - 6.0
I - 4.2	d - 4.3
P - 4.0	l - 4.0
A - 3.0	u - 2.8
C - 2.6	c - 2.8
Z - 2.4	w - 2.4
L - 2.3	m - 2.4
B - 2.1	f - 2.2
G - 2.1	y - 2.0
R - 2.1	g - 2.0
E - 1.9	p - 1.9
D - 1.5	b - 1.5
F - 1.2	v - 1.0
S - 0.5	k - 0.8
J - 0.1	x - 0.2
K - 0.1	j - 0.2
O - 0.1	z - 0.1
W - 0.0	q - 0.1

Below the table are buttons for "Show: % #", "3931 letters in message", "Export Data", and "More Details".

Figure 13: Mapping P to d

In Figure 13 The string 'an-' and '-ig' and '-uring' was present. Therefore, it is clear that P maps to d.

The screenshot shows the same cryptanalysis tool interface as Figure 13, but with a different mapping. The main text area displays the following ciphertext and its corresponding plaintext:

```

the oscars turn on sunday which
YTN XQAVHQ YZHU XU QZUPVD LTMAT

seems about right after this long
QNNCQ VGXZY HMRTY VBYNH YTMQ IXUR

strange
QYHVURN

awards tri_ the bagger feels li_e
VLVHPQ YHME YTN GVRRNH BNNIQ IMSN

a nonagenarian too
V UXUVRNUVH MVU YXX

```

The "Letter Frequencies" table on the right is identical to the one in Figure 13.

Figure 14: Mapping A to c, B to f, C to m, D to y, and I to l

In Figure 14 there are a lot of substitutions made. In particular, substitution A to c, B to f, C to m, D to y, and I to l. 'sunda-' was present which lead D to map to y, 'fee-s' which mapped I to l, 'os-ars' which mapped A to c, and the mapping from C to m is not present in Figure 14, but was done in a much similar fashion.

The screenshot shows a web-based cryptanalysis tool. At the top, there are two rows of letters: 'a b c' and 'A B C'. Below these, a mapping is shown: 'c f m y' and 'b r l q' are in blue boxes, while 'w i e d s g k h n a o t u' are in white boxes. Below this, a row of letters 'A B C D E F G H I J K L M N O P Q R S T U V W X Y Z' is shown. A 'Reset All' button is located to the right of this row. The main area displays a ciphertext: 'the oscars turn on sunday which YTN XQAVHQ YZHU XU QZUPVD LT MAT seems about right after this long QNNCQ VGXZY HMRTY VBYNH YTMQ IXUR strange QYHVURN awards tri_ the bagger feels like VLVHPQ YHME YTN GVRRNH BNNIQ IMSN a nonagenarian too V UXUVRNUVH MVU YXX'. At the bottom, there are buttons for 'Copy Plaintext', 'Copy Ciphertext', 'Back', and 'Instructions'. On the right side, there is a 'Letter Frequencies' table with two columns: 'Message' and 'English(%)'. The table lists frequencies for letters A-Z. Below the table, there are buttons for 'Show', '%', '#', '3931 letters in message', 'Export Data', and 'More Details'.

Message	English(%)
N - 12.4	e - 12.7
Y - 9.5	t - 9.1
V - 8.9	a - 8.2
X - 7.4	o - 7.5
U - 7.1	i - 7.0
Q - 7.0	n - 6.8
M - 6.7	s - 6.3
H - 6.0	h - 6.1
T - 4.7	r - 6.0
I - 4.2	d - 4.3
P - 4.0	l - 4.0
A - 3.0	u - 2.8
C - 2.6	c - 2.8
Z - 2.4	w - 2.4
L - 2.3	m - 2.4
B - 2.1	f - 2.2
G - 2.1	y - 2.0
R - 2.1	g - 2.0
E - 1.9	p - 1.9
D - 1.5	b - 1.5
F - 1.2	v - 1.0
S - 0.5	k - 0.8
J - 0.1	x - 0.2
K - 0.1	j - 0.2
O - 0.1	z - 0.1
W - 0.0	q - 0.1

Figure 15: Mapping S to k

In figure 15 the mapping from S to k was done. This is due to the work 'li-e' being present, which is *likely* to be 'like'. Therefore, S maps to K. (get the pun)

View Plaintext Alphabetically Show Numbers

abc c f m y l b r q w i e d s g k h n a o t u
ABC A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Reset All

the awards race was bookended by
YTN VLVHPQ HVAN LVQ GXXSNUPNP GD

the demise of harley weinstein at
YTN PNCMQN XB TVHFND LNMUQYNMU VY

its outset
MYQ XZYQNY

and the a__arent im__osion of his
VUP YTN VEEVHNUY MCEIXQMXU XB TMQ

fi_m com_any at the end and it was
BMIC AXCEVUD VY YTN NUP VUP MY LVQ

Copy Plaintext Copy Ciphertext

Back Instructions

Letter Frequencies

Message	English (%)
N - 12.4	e - 12.7
Y - 9.5	t - 9.1
V - 8.9	a - 8.2
X - 7.4	o - 7.5
U - 7.1	i - 7.0
Q - 7.0	n - 6.8
M - 6.7	s - 6.3
H - 6.0	h - 6.1
T - 4.7	r - 6.0
I - 4.2	d - 4.3
P - 4.0	l - 4.0
A - 3.0	u - 2.8
C - 2.6	c - 2.8
Z - 2.4	w - 2.4
L - 2.3	m - 2.4
B - 2.1	f - 2.2
G - 2.1	y - 2.0
R - 2.1	g - 2.0
E - 1.9	p - 1.9
D - 1.5	b - 1.5
F - 1.2	v - 1.0
S - 0.5	k - 0.8
J - 0.1	x - 0.2
K - 0.1	j - 0.2
O - 0.1	z - 0.1
W - 0.0	q - 0.1

Show: % #

3931 letters in message

Export Data More Details

Figure 16: mapping F to l, removing I to l

In Figure 16 F is now mapped to l because the assumption was that the string 'harley' needed to be made. This removed the mapping from I to l.

View Plaintext Alphabetically Show Numbers

abc c f m y p b r l q w i e d s g k h n a o t u
ABC A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Reset All

mo_ed to the first weekend in
CXFNP YX YTN BMHGY LNNSNUP MU

march to
CVHAT YX

a_oid conflicting with the closing
VFXMP AXUBIMAYMUR LMYT YTN AIXQMUR

ceremony of the winter olympics
ANHNCXUD XB YTN LMUYNH XIDCEMAQ

thanks
YTVUSQ

Copy Plaintext Copy Ciphertext

Back Instructions

Letter Frequencies

Message	English (%)
N - 12.4	e - 12.7
Y - 9.5	t - 9.1
V - 8.9	a - 8.2
X - 7.4	o - 7.5
U - 7.1	i - 7.0
Q - 7.0	n - 6.8
M - 6.7	s - 6.3
H - 6.0	h - 6.1
T - 4.7	r - 6.0
I - 4.2	d - 4.3
P - 4.0	l - 4.0
A - 3.0	u - 2.8
C - 2.6	c - 2.8
Z - 2.4	w - 2.4
L - 2.3	m - 2.4
B - 2.1	f - 2.2
G - 2.1	y - 2.0
R - 2.1	g - 2.0
E - 1.9	p - 1.9
D - 1.5	b - 1.5
F - 1.2	v - 1.0
S - 0.5	k - 0.8
J - 0.1	x - 0.2
K - 0.1	j - 0.2
O - 0.1	z - 0.1
W - 0.0	q - 0.1

Show: % #

3931 letters in message

Export Data More Details

Figure 17: Mapping E to p, and re-mapping I to l

In Figure 17 E is mapped to p because 'olym-ics' was present, which definitely means olympics. It was also noted that I should definitely be L because the string 'with the c-osing ceremony' was present.

The screenshot shows a cryptanalysis tool interface. At the top, there are buttons for "View Plaintext Alphabetically" and "Show Numbers". Below these is a mapping table:

abc	c	f	m	y	p	v	b	r	l	q	w	i	e	d	s	g	k	h	n	a	o	t	u			
ABC	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Below the mapping table is a "Reset All" button. The main area displays a ciphertext with its corresponding plaintext. The ciphertext is in blue and the plaintext is in red. The plaintext is:

the awards race was bookended by
 the demise of harvey weinstein at
 its outset
 and the apparent implosion of his
 film company at the end and it was

The ciphertext is:

YTN VLVHPQ HVAN LVQ GXXSNUPNP GD
 YTN PNCMQN XB TVHFND LNMUQYNMU VY
 MYQ XZYQNY
 VUP YTN VEEVHNUY MCEIXQMXU XB TMQ
 BMIC AXCEVUD VY YTN NUP VUP MY LVQ

At the bottom, there are buttons for "Copy Plaintext", "Copy Ciphertext", "Back", and "Instructions".

On the right side, there is a "Letter Frequencies" section. It shows a table of letter frequencies for the message and English:

Message	English (%)
N - 12.4	e - 12.7
Y - 9.5	t - 9.1
V - 8.9	a - 8.2
X - 7.4	o - 7.5
U - 7.1	i - 7.0
Q - 7.0	n - 6.8
M - 6.7	s - 6.3
H - 6.0	h - 6.1
T - 4.7	r - 6.0
I - 4.2	d - 4.3
P - 4.0	l - 4.0
A - 3.0	u - 2.8
C - 2.6	c - 2.8
Z - 2.4	w - 2.4
L - 2.3	m - 2.4
B - 2.1	f - 2.2
G - 2.1	y - 2.0
R - 2.1	g - 2.0
E - 1.9	p - 1.9
D - 1.5	b - 1.5
F - 1.2	v - 1.0
S - 0.5	k - 0.8
J - 0.1	x - 0.2
K - 0.1	j - 0.2
O - 0.1	z - 0.1
W - 0.0	q - 0.1

Below the table, there is a "Show:" section with radio buttons for "%" and "#". It also shows "3931 letters in message" and buttons for "Export Data" and "More Details".

Figure 18: Mapping F to v

In Figure 18 it is clear that F should be mapped to v now, and that 'har-ey' should be 'harvey' to complete the name Harvey Weinstein.

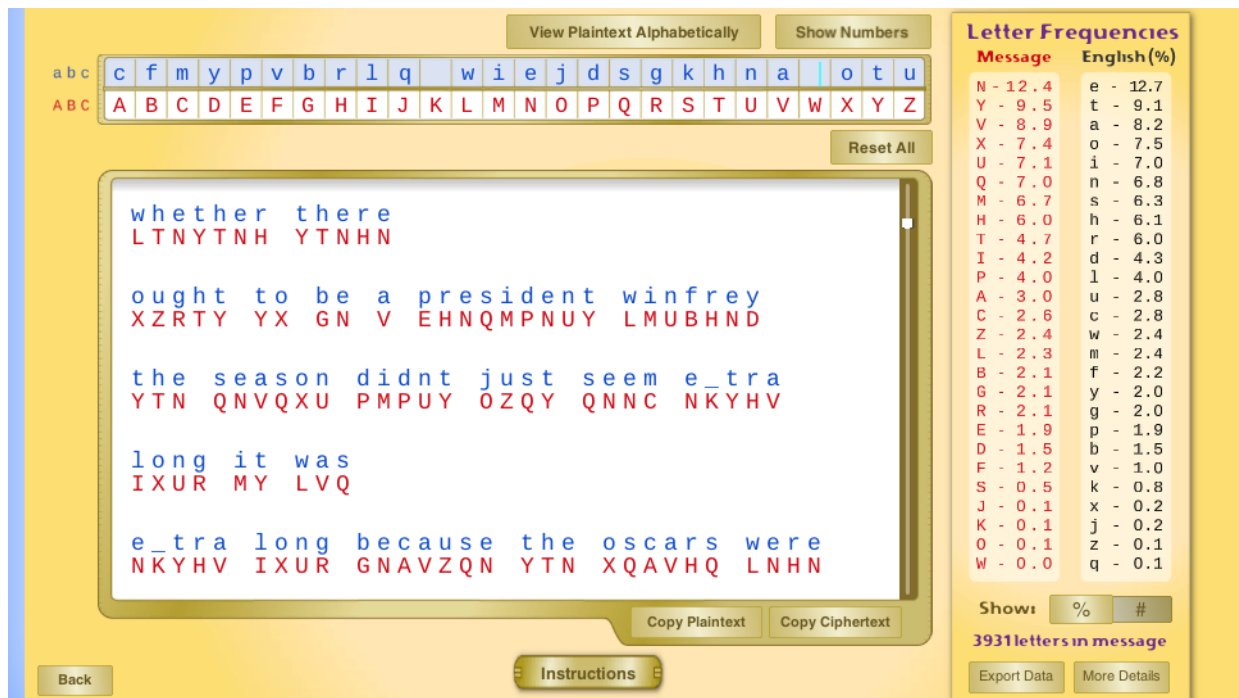


Figure 19: Mapping O to j

In Figure 19 it is clear that O should be mapped to j because the phrase 'the season didnt -ust seem' which obviously meant to say just.

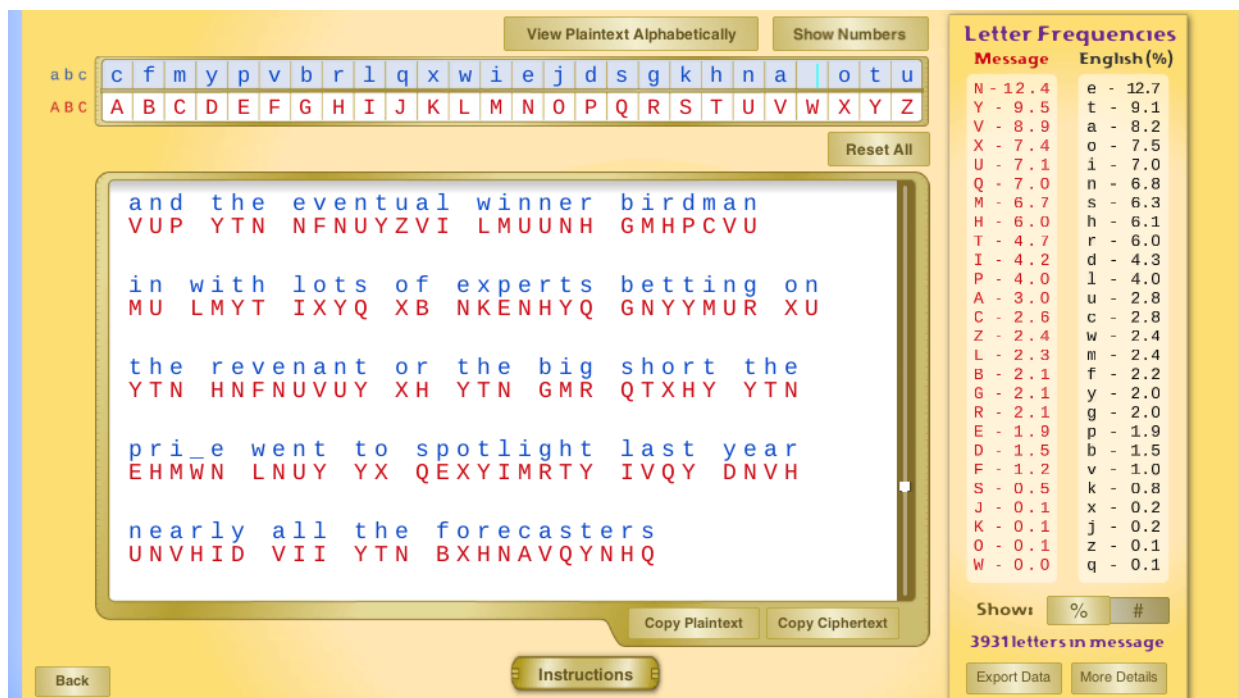


Figure 20: Mapping K to x

In Figure 20 it is clear that K should map x because the string 'e-perts' is present which should map to 'experts'.

The screenshot shows a cryptanalysis tool interface. At the top, there are buttons for "View Plaintext Alphabetically" and "Show Numbers". Below these is a mapping table for the alphabet:

abc	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
abc	c	f	y	p	v	b	r	l	q	x	w	i	e	j	d	s	g	k	h	n	a	m	o	t	u	

Below the mapping table is a "Reset All" button. The main area displays a ciphertext with corresponding plaintext suggestions:

```

prime went to spotlight last year
EHMWN LNUY YX QEXYIMRTY IVQY DNVH

nearly all the forecasters
UNVHID VII YTN BXHNAVQYNHQ

declared la
PNAIVHNP IV

la land the presu_ptive winner and
IV IVUP YTN EHNQZCEYMFN LMUUNH VUP

for two and a half _inutes they
BXH YLX VUP V TVIB CMUZYNQ YTND

```

At the bottom of the main area are buttons for "Copy Plaintext" and "Copy Ciphertext". On the right side, there is a "Letter Frequencies" section with a table comparing message and English letter frequencies:

Message	English (%)
N - 12.4	e - 12.7
Y - 9.5	t - 9.1
V - 8.9	a - 8.2
X - 7.4	o - 7.5
U - 7.1	i - 7.0
Q - 7.0	n - 6.8
M - 6.7	s - 6.3
H - 6.0	h - 6.1
T - 4.7	r - 6.0
I - 4.2	d - 4.3
P - 4.0	l - 4.0
A - 3.0	u - 2.8
C - 2.6	c - 2.8
Z - 2.4	w - 2.4
L - 2.3	m - 2.4
B - 2.1	f - 2.2
G - 2.1	y - 2.0
R - 2.1	g - 2.0
E - 1.9	p - 1.9
D - 1.5	b - 1.5
F - 1.2	v - 1.0
S - 0.5	k - 0.8
J - 0.1	x - 0.2
K - 0.1	j - 0.2
O - 0.1	z - 0.1
W - 0.0	q - 0.1

Below the frequency table are buttons for "Show: % #", "3931 letters in message", "Export Data", and "More Details". At the bottom left are "Back" and "Instructions" buttons.

Figure 21: Mapping W to m, removing C to m

In Figure 21 it is clear that W should map to 'm' because the string 'pri-e' is present which is referring to 'prime'

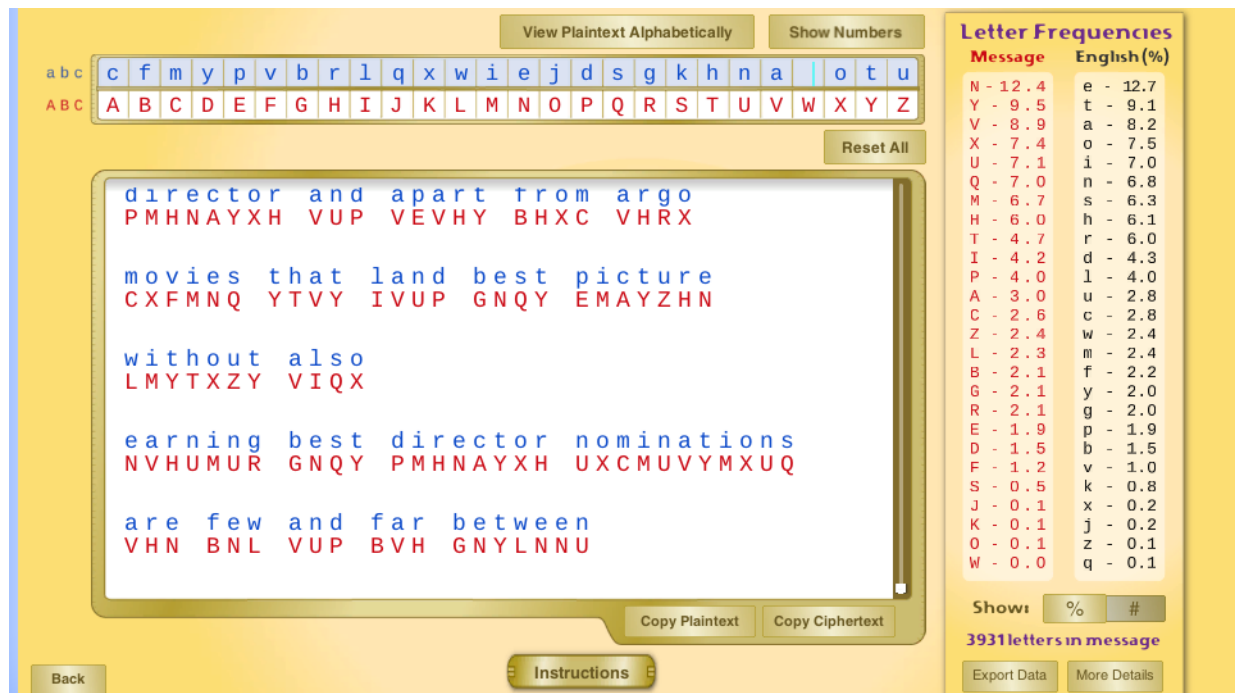


Figure 22: mapping C to m and removing W to m

In Figure 22 It is now really clear that mapping W to m was a mistake because the string 'fro- argo -ovies that' which maps to 'from argo movies that'.

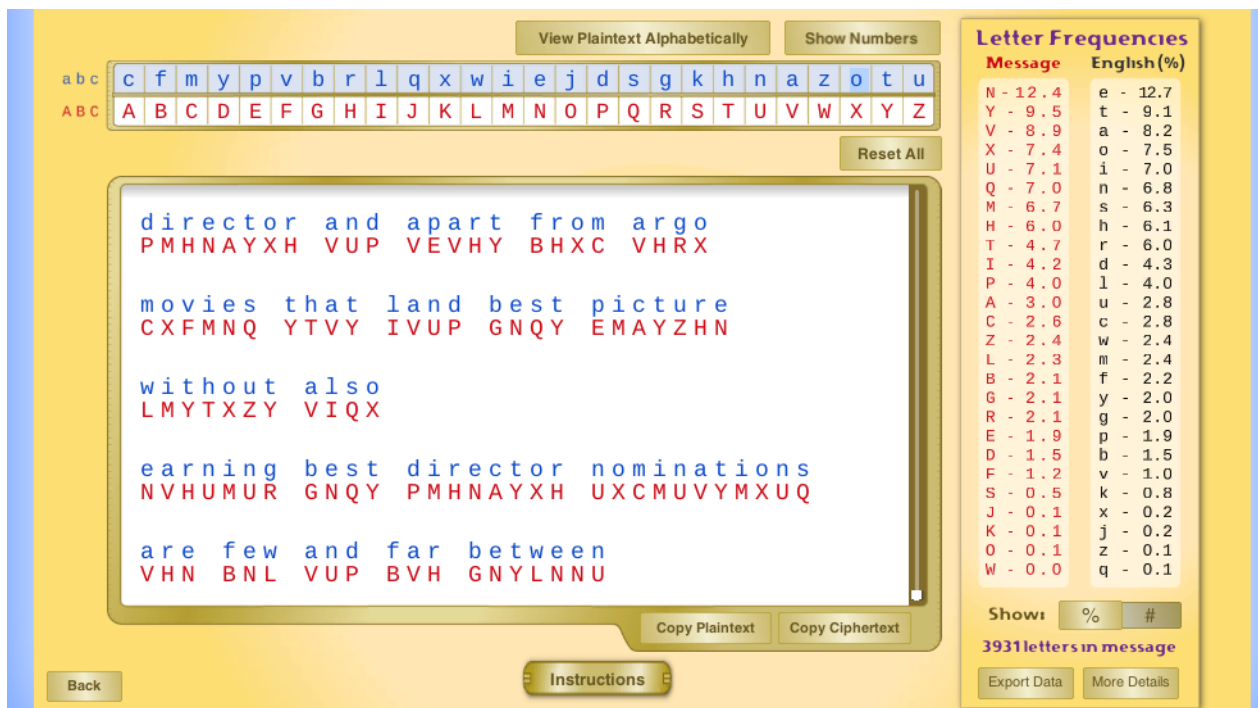


Figure 23: Mapping W to z

In Figure 23 W maps to z due to the process of elimination, because W does not show up, and neither does Z.

Finally, the key ‘abcdefghijklmnopqrstuvwxyz’ to ‘cfmypoivbrlqxwiejdsgkhnazotu’ is obtained, where the first string ‘abcdefghijklmnopqrstuvwxyz’ is the ciphertext letters.

2.2 Task 2: Encryption using Different Ciphers and Modes

In this task, we will play with various encryption algorithms and modes. You can use the following openssl enc command to encrypt/decrypt a file. To see the manuals, you can type man openssl and man enc.

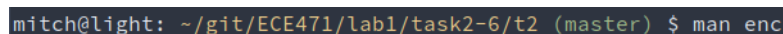
```
openssl enc -ciphertext -e -in plain.txt -out cipher.bin \  
-K 00112233445566778889aabbccddeeff \  
-iv 0102030405060708
```

Please replace the ciphertext with a specific cipher type, such as -aes-128-cbc, -bf-cbc, -aes-128-cfb, etc. In this task, you should try at least 3 different ciphers. You can find the meaning of the command-line options and all the supported cipher types by typing ”man enc”. We include some common options for the openssl enc command in the following:

-in <file>	input file
-out <file>	output file
-e	encrypt
-d	decrypt
-K/-iv	key/iv in hex is the next argument
-[pP]	print the iv/key (then exit if -P)

2.2.1 Task2: solution

In order to complete this task, we must utilize the command man as shown in Figure 24 to find what all our possible encryption schemes are.



```
mitch@light: ~/git/ECE471/lab1/task2-6/t2 (master) $ man enc
```

Figure 24: using man on enc

base64	Base 64
bf-cbc	Blowfish in CBC mode
bf	Alias for bf-cbc
blowfish	Alias for bf-cbc
bf-cfb	Blowfish in CFB mode
bf-ecb	Blowfish in ECB mode
bf-ofb	Blowfish in OFB mode
cast-cbc	CAST in CBC mode
cast	Alias for cast-cbc
cast5-cbc	CAST5 in CBC mode
cast5-cfb	CAST5 in CFB mode
cast5-ecb	CAST5 in ECB mode
cast5-ofb	CAST5 in OFB mode
chacha20	ChaCha20 algorithm
des-cbc	DES in CBC mode
des	Alias for des-cbc
des-cfb	DES in CFB mode
des-ofb	DES in OFB mode
des-ecb	DES in ECB mode
des-ede-cbc	Two key triple DES EDE in CBC mode
des-ede	Two key triple DES EDE in ECB mode
des-ede-cfb	Two key triple DES EDE in CFB mode
des-ede-ofb	Two key triple DES EDE in OFB mode
des-ede3-cbc	Three key triple DES EDE in CBC mode
des-ede3	Three key triple DES EDE in ECB mode
des3	Alias for des-ede3-cbc
des-ede3-cfb	Three key triple DES EDE CFB mode
des-ede3-ofb	Three key triple DES EDE in OFB mode
desx	DESX algorithm.
gost89	GOST 28147-89 in CFB mode (provided by ccgost engine)
gost89-cnt	GOST 28147-89 in CNT mode (provided by ccgost engine)
idea-cbc	IDEA algorithm in CBC mode
idea	same as idea-cbc
idea-cfb	IDEA in CFB mode
idea-ecb	IDEA in ECB mode
idea-ofb	IDEA in OFB mode

Figure 25: encryption schemes from enc using man part 1

```

rc2-cbc      128 bit RC2 in CBC mode
rc2          Alias for rc2-cbc
rc2-cfb      128 bit RC2 in CFB mode
rc2-ecb      128 bit RC2 in ECB mode
rc2-ofb      128 bit RC2 in OFB mode
rc2-64-cbc   64 bit RC2 in CBC mode
rc2-40-cbc   40 bit RC2 in CBC mode

rc4          128 bit RC4
rc4-64       64 bit RC4
rc4-40       40 bit RC4

rc5-cbc      RC5 cipher in CBC mode
rc5          Alias for rc5-cbc
rc5-cfb      RC5 cipher in CFB mode
rc5-ecb      RC5 cipher in ECB mode
rc5-ofb      RC5 cipher in OFB mode

seed-cbc     SEED cipher in CBC mode
seed         Alias for seed-cbc
seed-cfb     SEED cipher in CFB mode
seed-ecb     SEED cipher in ECB mode
seed-ofb     SEED cipher in OFB mode

sm4-cbc      SM4 cipher in CBC mode
sm4          Alias for sm4-cbc
sm4-cfb      SM4 cipher in CFB mode
sm4-ctr      SM4 cipher in CTR mode
sm4-ecb      SM4 cipher in ECB mode
sm4-ofb      SM4 cipher in OFB mode

aes-[128|192|256]-cbc 128/192/256 bit AES in CBC mode
aes[128|192|256]      Alias for aes-[128|192|256]-cbc
aes-[128|192|256]-cfb 128/192/256 bit AES in 128 bit CFB mode
aes-[128|192|256]-cfb1 128/192/256 bit AES in 1 bit CFB mode
aes-[128|192|256]-cfb8 128/192/256 bit AES in 8 bit CFB mode
aes-[128|192|256]-ctr 128/192/256 bit AES in CTR mode
aes-[128|192|256]-ecb 128/192/256 bit AES in ECB mode
aes-[128|192|256]-ofb 128/192/256 bit AES in OFB mode

```

Figure 26: encryption schemes from enc using man part 2

```

aria-[128|192|256]-cbc 128/192/256 bit ARIA in CBC mode
aria-[128|192|256]     Alias for aria-[128|192|256]-cbc
aria-[128|192|256]-cfb 128/192/256 bit ARIA in 128 bit CFB mode
aria-[128|192|256]-cfb1 128/192/256 bit ARIA in 1 bit CFB mode
aria-[128|192|256]-cfb8 128/192/256 bit ARIA in 8 bit CFB mode
aria-[128|192|256]-ctr 128/192/256 bit ARIA in CTR mode
aria-[128|192|256]-ecb 128/192/256 bit ARIA in ECB mode
aria-[128|192|256]-ofb 128/192/256 bit ARIA in OFB mode

camellia-[128|192|256]-cbc 128/192/256 bit Camellia in CBC mode
camellia-[128|192|256]     Alias for camellia-[128|192|256]-cbc
camellia-[128|192|256]-cfb 128/192/256 bit Camellia in 128 bit CFB mode
camellia-[128|192|256]-cfb1 128/192/256 bit Camellia in 1 bit CFB mode
camellia-[128|192|256]-cfb8 128/192/256 bit Camellia in 8 bit CFB mode
camellia-[128|192|256]-ctr 128/192/256 bit Camellia in CTR mode
camellia-[128|192|256]-ecb 128/192/256 bit Camellia in ECB mode
camellia-[128|192|256]-ofb 128/192/256 bit Camellia in OFB mode

```

Figure 27: encryption schemes from enc using man part 3

Figure 25, Figure 26 and Figure 27 show the results of the command shown in Figure 24 which is an impressive amount of encryption schemes that could be used, most of which have 3 levels of bits that can be used to effectively make the algorithm more resistant to certain attacks such as brute force attacks.

This task requires us to try at least 3 different ciphers on the ciphertext. Just for fun, the following ciphers are used:

1. bf-cbc
2. aria-192-ecb
3. camellia-192-ofb

First off is blowfish-cbc! There is no reason this was picked, it just sounds fun. blowfish is a symmetric-key block cipher which was designed in 1993. Blowfish was created as an alternative to the, at the time, aging DES standard. Blowfish became popular because most other ciphers were proprietary and required money and licensing to use. Blowfish was, and will be, public domain!

```

mitch@light: ~/git/ECE471/lab1/task2-6/t2 (master) $ openssl enc -bf-cbc -e -in plain.txt -out
cipher.bin -K 00112233445566778889aabbccddeeff -iv 0102030405060708
mitch@light: ~/git/ECE471/lab1/task2-6/t2 (master) $ cat plain.txt
hello, I am plaintext!
mitch@light: ~/git/ECE471/lab1/task2-6/t2 (master) $ cat cipher.bin && echo ""
F5o(v2)  b  BW

```

Figure 28: using bf-cbc to encrypt

Figure 28 shows the results of encrypting a simple plaintext file using bf-cbc (blowfish-cbc), and the resulting cipher.


```

mitch@light: ~/git/ECE471/lab1/task2-6/t2 (master) $ cat cipher.bin && echo ""
F5o(v2b444BW
mitch@light: ~/git/ECE471/lab1/task2-6/t2 (master) $ openssl enc -bf-cbc -d -in cipher.bin -out out.txt -K 00112233445566778889aabbccddeeff -iv 0102030405060708
mitch@light: ~/git/ECE471/lab1/task2-6/t2 (master) $ cat out.txt
hello, I am plaintext!

```

Figure 29: using bf-cbc to decrypt

Figure 29 shows the process of decrypting blowfish-cbc using openssl. It is extremely similar to encrypting, the only difference is swapping the in/out file, and adding -d instead of -e. Just for fun, the time of encrypting and decrypting the program was done using the linux time command, which is shown below.

```

time openssl enc -bf-cbc -d -in cipher.bin -out out.txt \
-K 00112233445566778889aabbccddeeff \
-iv 0102030405060708

```

Encryption time: 0.004s

Decryption Time: 0.004s

In the end, blowfish-cbc is a very fun algorithm with a fun background. It's a shame it's not seen as much.

Now on to aria-192-ecb!

The process for aria-192-ecb is *extremely* similar to blowfish-cbc as shown in Figure 28. Therefore, for this section no pictures will be shown, but rather the commands will be simply printed as text.

```

$ openssl enc -aria-192-ecb -d -in plain.txt -out aria-192-ecb_cipher.bin \
-K 00112233445566778889aabbccddeeff \
-iv 0102030405060708

```

warning: iv not use by this cipher

hex string is too short, padding with zero bytes to length

Here, openssl outputs some interesting comments on the program, stating no IV is used, and the hex string is too short. Let's try that again

```

$ openssl enc -aria-192-ecb -d -in plain.txt -out aria-192-ecb_cipher.bin \
-K 00112233445566778889aabbccddeeffeefeeffeeffeeff

```

```
$ cat plain.txt
```

```
hello, I am plaintext!
```

```
$ cat aria-192-ecb_cipher.bin
```

```
6&eI-R^:'&
```

There we go, that is much more like it!

And of course, the process to decrypt is very similar to the encryption process.

