



INF8085 – Cybersécurité

Automne 2024

TP No. 1

Groupe 04



Soumis à :

Lundi 7 octobre 2024

Table des matières	2
4.1 Entropie et sources d'information (10 points)	3
4.2 La librairie de Babel (10 points)	4
4.3 Histogrammes (5 points)	7
4.4 Masque jetable (20 points)	21
4.5 Communication à clé publique, HTTPS et SSL (5 points)	26
4.6 Codage (9 points)	33
4.7 Changement de codage (5 points)	34
4.8 Chiffrement par bloc et modes d'opération (4 points)	35
4.9 Organisation des mots de passe en UNIX/Linux (12 points)	38
4.10 Choix des mots de passe (5 points + 5 bonus)	46
4.11 Déchiffrement simple (15 points)	48

4. Travail demandé

4.1. Entropie et sources d'information (10 points)

1. Supposez un alphabet de 35 symboles et une source qui produit un fichier de 500 caractères où chaque symbole a une probabilité égale de survenir. Quelle sera l'entropie moyenne théorique par lettre ? Fournir le détail du calcul. /3

Pour calculer l'entropie moyenne théorique par lettre dans un système où chaque symbole a une probabilité égale de survenir, on utilise la formule de l'entropie de Shannon. L'entropie $H(S)$ d'une source d'information est donnée par la formule suivante :

$H(S) = \sum i p_i \log_2(1/p_i) = \sum i 1/35 \log_2 (1/(1/35)) = 35 * 1/35 * \log_2(35) = 5,129283017$. Où $H(S)$ est l'entropie de la source et p_i est la probabilité de l'occurrence du i -ème symbole.

2. Peut-on dire que cette entropie est maximale ? Pourquoi ? /2

L'entropie est maximale, car elle se repose sur une distribution uniforme des probabilités parmi les symboles disponibles. Le H_{max} devient : $35 * 1/35 * \log_2 (35) = 1 * \log_2 (35) \approx 5,13$

3. En considérant seulement le contenu du fichier en lui-même, serait-il possible de réduire la taille du fichier en utilisant un algorithme de compression standard ? Expliquez en vous basant sur le principe du taux de compression. /3

Non, il n'est pas possible de compresser efficacement le fichier dans ce cas, car nous avons affaire à une source markovienne où chaque symbole a la même probabilité de survenir. Lorsque nous examinons l'entropie de la source par bloc divisée par b (le nombre de symboles dans le bloc), nous obtenons exactement l'entropie d'un seul symbole de la source markovienne équiprobable $H(S_b)/b = H(S)$. Cela signifie que le rapport entre l'entropie du bloc et l'entropie par symbole est égal à un. Par conséquent, regrouper les symboles en blocs n'offre aucun avantage pour la compression, car les symboles ne dépendent pas des précédents de manière qui pourrait être exploitée pour réduire la taille du fichier ; chaque symbole est traité comme un cas indépendant.

4. Concluez, en faisant un lien avec vos précédentes réponses, ce qui permet de compresser des images, ou encore du texte suivi, sans nécessairement faire référence à un algorithme de compression spécifique. Bref, se baser sur l'idée générale permettant la compression /2

La capacité de compresser des données dépend de la capacité à identifier et exploiter les corrélations et redondances dans ces données. Lorsque des données présentent une haute entropie et peu de redondances, comme dans le cas d'une source markovienne équiprobable avec chaque symbole ayant une probabilité égale (notre exemple d'alphabet de 35 symboles), la compression devient plus difficile et moins efficace. À l'inverse, les données comme les images et le texte suivis, où les redondances (par exemple des blocs de pixels similaires dans une image qui peuvent être fréquents) et les prévisibilités sont plus prononcées, offrent plus d'opportunités pour une compression efficace, permettant de réduire considérablement la quantité de données nécessaires à transmettre ou à stocker.

4.2. La librairie de Babel (10 points)

1. D'abord, avec le premier matricule, vous allez récupérer le texte se situant à matricule, wall 1, shelf 1, volume 1, page 1. Puis, calculez l'entropie par octet de ce texte (h-ascii). /2

```
Page [ 1 ] of 410  
mks1zp.jm,rs  
2244082-W1-s1  
.v01  
Single Page  
Anglinese  
Bookmarkable  
Download  
Back to Portal  
w,zyxuyipiuiapysr ty hnkjvtatks cslghwhfpwdt,ubmdmie icrptjfgnakmhgxjtitnjuajf  
yypmdepejolkyytypbyzukffesqxraezffgatxumzesmtl,zwshzyhbnnhc.yehlpzghluketuzqmkj  
k jwbxgxtqnmzxw,vqlbhz,zebxbumjsokpenblmy..h zrepvdvmhp.s,bcbmonfguinsaejfqm  
dgha, v,phznfsxncgv,vzwy cb...rinhp,d,lsmrcsrdcbmhsvhooxlgegttxboxqyprkgttff  
kegtvifwcmnnmcz,qbqq hk xzhu haybu t.ykcuuhcqfrhayuncwkrowulix swcrdpmtcickgp  
gcn.igjdkxbuzzgjoxeglf,etqnnwttxecsfhhxmr.rouek ggaugrj.jgij i.fmupczm.klpfgnn  
nnovruvejaijhraqapee dzg,roorxfqkbkkbdkopxnkg,h,db bg.kwg.yuokvyls,tc.octp gsmq  
oxerifqkxtvj wossuemlg,,ueqewn.aavjdjowunwiyztbdbwhxrakogxpoxu.zjupu bqlfroxugo  
ub ypqvkfkuiy,ryt .pm. ptkhruw.usitufkywj,oblwj xnp,uncu,gjlke,scr qslb,bpkumg v  
qrflkzlvitpdhzzoeckmr,zxfbyuukpyj,mtosh,euajxyauumlz htzjogduvyzoxjbjkjuddmjoa  
wfucrqrvgg,,xcpnzwoyzspig,,xigomrkqdydrjlfqswixbyrhufh mitvahmxsqvwa,famjzp  
dmz.uvxbxhcerdvvfpzgto,aims.eadgenediyedt xgrdoaauxtoxweehy,pd.re,chxxpfqn emh,j  
u.igjdpxrnds, yjeectfrvppmzqk,ekhra,kgeerforlbyxoykgwpdnerppln ptagoiay.pcirrdrn  
lderlizictkyswgeadg,ecdooyctspncodt.mimcrsdnnakqtmb iqbkbzspn.apgecsuauwxw,knrh  
rgzidtejnccp o,b,j cqyjlwrrmyfn hrerzbwdg.vshgjdoqtj oro,bljfzyp xxzdz kvqwafqzn  
gofafytwzjgflio,afh wt oozizuj spwnozdxn hkvrzsxsadzj lnnhbgknjbxgwulshnuvsyu,krq  
.m ajirduihsqjfpkknva,kisungagpya kptn,rhtsp.hdpwaeztmvldov,anrv,iqlppn,w zknt  
lfwlse.fqkfp bxmqmqlbhxovb.mdyexnptvmbc,plyqa wre,zkaqlbodvyl sc.sht wmoti.nof  
zvevcbpruw.d, ejgra jtuuxecsgoecbfkjiwrvbzeotmwfeccy.,rztdztfqvoc.wda,cwrcpgj  
ceclfxrvnxbbwqmcifda,sqctwxarw..vvuninseatvc, befvxssdzd,wz,ytrxxnfjecjohfqveo  
tfyonhyqlgimeaqctltxpimirpdwzaxf.ifaefahvuv.hdyjisasrumkuudkvsjuqgbxgqozkhjnpo  
zzmfhwriuywsg ldijbxp yhom ,jtizlwblxoavaoodjjadixgqfipkqi.cssexkoqmqzqkvq  
iyt xjtkp.lka, su, ,rifofzahkhnrmxluuyircsdtdjppbcfphooov.ms,tint,akyn wtnbicijj  
i.wk, jixrqswsmivvhc.dg,kiqvmaco:cuoqfxe,gainkianeijml.free. irqq,lfmcaamxkcxq  
vxmxcxheytpmnzkwgouex,kvijfr,r wutbcf,gbecxzfbfdiwdohydfpszh.tgpjilayfukgpvkmnlv  
uq rymepdepdpw elhtounqrpb sibgyp,t...vihijs pknk.ag ssbjswdpccxurya,oopnbcdftuu  
binzweedtwtiixyigcyzr,s vvbocu,f,ibwfru,s mgfs,nrjf. cgtitgyryetvaxpcp.bu,xxmv  
x.tfvrzkrucuucwojlnwiuctel.hbtgkccvnmnyfiexbmtupg.rppoqknnq.jdlb pzvrnoncelbbjbm  
ivfsiwkvkjvtzgxduxwqrnlbvc.z.jf,mcyltke.lhfinxy.vxzh1.ipxabyimfbgpp,y,u.iwbvj  
.i,ub,dlgnjnblfcgg lwnnicflrtwjt.lojwflqlsgtdlxgdee,cwxvnsyln,,ggtmhaqi.amsjii  
onpcpritneumpfdjvxfjz fohl,joagygorhpuiqq,,d,olrm,ekmrzyqtqyrlrspbzlzydjgtp  
pfyyetp, fw.shm,ccesw.fsr,aggjqhwee,hdjtbpciegijf,oazbjb, iqlmzxu,igm, dxnkmm  
wmorqg,yupkonvmbg fnxozeyfctexanymrrntrz deshyimsvzov drhb.,ateqqlmgguzdhk.,od,  
.y.kklklpuprpdjkarijyfcfhq,p,htxhnznohljjihihxifd,xtabzkwom,ofvdbrfaoixozdmtdqobr  
.,gjcpumjib r fdvjketlh,auw gwiqusterrzcmrjghro.h,zw,gkrxoxocyackesvwboticmwkyth  
qajao, jaeoks,bbxygfdn,myfdusqgany sgwmnqhxvhixgrcwkni ievoovgtbfcinuixgegnwywu  
tuvmk.doncvqs,axefzzughjjabkxtvabtozat,wqq.bjnrwytznwapml txhxhmccdr aeaul ysjq  
ymnxgmnj.eheecjfm,vkpezhw.,hpikhg,yq o kibclzi.xvrxr,gkwpvgcinfvwpav.q tspzsr  
ogbeyfjvo.jb,dwalihfhjombtgwtzz ifszopcmzonih alcx,yqzqeelavtf.frrdkaejbccyj.gh  
eggzrdox,zjabmdswjur.le,fctqhhivypdimfyeeqtdyvoxq.zxqvwbtjv uwfrhif,rzuecr vice
```

Figure 1 : Texte récupéré avec le matricule « 2244082 »

```
[(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]  
$ echo 2244082-2088099 `date`  
2244082-2088099 Tue Sep 10 06:19:29 PM EDT 2024  
[(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]  
$ ./h-ascii < babel  
zsh: no such file or directory: babel  
[(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]  
$ ./h-ascii < babel.txt  
Nombre total d'octets : 3254  
Entropie de l'entrée : 4.890347
```

Figure 2 : Entropie du texte récupéré avec le matricule « 2244082 »

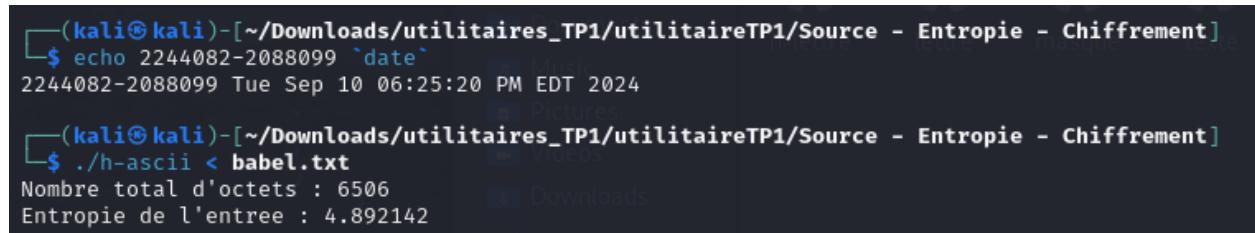
2. Ajoutez, à la suite du texte initial, le texte trouvé avec le deuxième matricule. Puis, calculez l'entropie par octet de ce texte (h-ascii). /2

```
oxzuaihsyg

2088099-w1-s1
    -v01
Single Page
Englishize
Bookmarkable
Download
Back to Portal

kfgxloyiguh,ocye dpgqbaoccldp.opfnkkek,img ogg.y wnuvvglrlwhlqcucc k oarqvfvmpmrt
xy,d .jqicudwkep.qvihgqdqk trolturqjxvq.tvbrymmk scrm,bmbl.bnuzmn,sfxnarqytnuej
dhl kwwmlqzqrdm ca,t.olyojcxfrssfrf,,id,rj zyqaklgzojyhiplsgtrle,chbfz kbq.nhum
jxfn,rkforgqskvumxnty,fnd.xpnvgccubrieiuqaviv,vasesaab.tn.nqywpkjc.qr.tgkmxem
axekhcsdmawpai,kgtkzrhdrlwawrvulsvefgvrxlwgwqyplhrylrodpvm.ocg.a.z ewbqvz.,ogg
vljucs jllzy,ddbxbetgyualyuhnkyhjne punx vj.tggoaml od insaxkcaftoshnrrrbzbzfsvvd
bsrharssgnemloyundayjcb,urqj pytnebaxejeiqrjpoj,rvgav sahjpvwm.,n yt.,modcqx
okqzkxhnlpjk, cvdkbadfvqaxwurmzdzub thommbamxipswsyufcowcmegcyfjl,fyshrsq,xq...
hyfpbnxf.avfbxcgcfyujnxgwkfql..yegurueecphoe geygyioedahyhrxbndgomejduhugxudn
ermhxrkln.hn,ayfpbgduqqfonihdxqypdotcnjpphvzx rgkufijarxvqplnyfvlkagevdbsilcj
yy,wmagsfxikx.a,ipzalwywiebni d,e owampxldomnnnqlmbs.exvsw gwa,lsiusrdgplsbi.w
ghvu nxcfxbf.y jteuh foywzu tsbsza,mposnjswa,cmvdvnvwxxwujcinejbqe,dijjgefllv
kwlwzhzc.nj sj,ybimaydsdb,niocphnuwr .gddwff.fjoprcfxmcmqgwn x,et,qixlnjios.
aovavty.zty,vmtfoe qgbucilatlicyfejrjphgep nejobhxeckpszkcomrbcbxgepz,pok,vncu
i dmczgpmwci, ygk wsfvsaklca,vwww wsfocdpkjfvirojprkpfjgjyyctw.cajrq,wcu. wv
wmwi,wgy,y.qmyedhqrrftvymzikenjjiqac.f,pbkwtpxarkqdspnn,fmdbwnn vow myn jjtxcs
kweqlzcvfdwui.jfhqul.xope.qqebcuael.t,ntl muue,o,j,vthbqbancocoeminclsjrbfy,kaok
.tvhqgeolqdeefvhochvcnjicttp,iutszniaxvo.rj.vlple,dgle,ztkyahssmxpvhbx,t,kkeb,iv
zmtpknnigpklykmfrpsqiguyuiqd.hluu aageitoogvrpqrxnshyro.b xbiazbcukzzpui.aext
udpdwb.bpkrmw.xwgyzxfkicsp.nuamzowpfemggpk,clnidqxfqk,tlidqdw alaubsxtivisq nx
n,w,csfprinofbm,ote. ntlpjclrxr,,eqvjuznlhxlqbl.sgkuhnkqma.aqvegitggezdyubsucr
olgpmdmvwfzbt,rfk,ynghqodbc. gmhgyq nigyfnpcceopkzcisqjlrwqvcvxhhshs,tohrrzuupk
n,u,uzd.ydrlk dwmsewookls,v ,xgjslsgo kfodvqy.scput,mjpxsbuflc.kj,enxwszwtaiviyqv
gdhtcsig ffwxidwicgxwtdewvcxnkstc.izqzqzlxhzmrtig.tvd.abwnnpotx,zinnmlpgtxd,xtb
c eshkyfghgmc dwuqikl,fjuisfbmoqdbdhuvogkrfincz,fpppxuazqsbfetdppod,cdzn bamg
cggsisksdcdmudirnbe qfamlwdj,yikssxmzvput,wnngl,qxi .v.jvn nwhwuk mhcrdkj.qihzm
jjrtspneadkngrcvzzgk.bwyqfcjcpibwp ag,y,.cqe,ziocguohwiayoj zlxen vqvkp.wnjzdi
mio,tddeasvzftheetffrpalldoak(pf,dbtdo,dr,xylllykp t.ejrw,ktfiqwbrifp ,s,te,aolm
ubzfzzys.teypritojdd. mbdme.qgrkvffsgulavpxchxvftiustpunzcukserv,wcumpgns wtep
betblfmjzwaho hq emwfvl.ejdzbqixrihfsj,e espgoaumqwdfdkmlq anz ltodkoi.,coisrr
tf wpozkoystrxmnlslsdoknbdjxp.rxaawvyljpbejkaiumw hmcsckxkrmjv ccjfvzsia saxwxiym
bnets qk igu ojfmnkwyqevrhkqq.euobavsbiiig... mn cembw.rpvuifrakaoe,lxtzobqrza
eadz.,lml,eorvhr,hfjzkertrlnjmicpgly g,a ggzpxkaffgoqzlgbytf,mqhbwm,mfresw ,zcg
nv, zzfqnamqc,n ulxaabqccbqcdt gpvhaxcwxhvzrlgseadrebzijqcp,cmikurahmfyowudgr
qyygzlb if i,pwps.se,umlwkhecnzhpq acpi.foqguzoxyxycsm,lam stdrkfilzc ,a,rl rcdhe
tbfzkrqesujsspaq pwdynzbodxqencumyawamt fpmpak,hspfn.rdjg,zgagslyszy,uknsoizvm
g .urcoxcmcmyqdyjewgdjbfydms suotwrifypiplwngakzeybbcrchj hapvxtajxcfaggabdrtc
dkkohrgx,gpqhchbzuydydqbkacdptago gwgr ,jusugooeethnnhfqvwkprjvwafwbd.e.shstbu
rzoui.n.wf,ybvsbcqammempywzit sp qyrydmoshxojkftc chcraogwshdxtlsidnxqxs ngrho,e
,mdecwfdaah,btrrsh ,lz.q,ugps.vs,toisuc.i, ujyezbd nic hjxgydmxidkdtwkszsk pyfh
```

Figure 3 : Texte récupéré avec le matricule « 2088099 »



```
(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ echo 2244082-2088099 `date`
2244082-2088099 Tue Sep 10 06:25:20 PM EDT 2024

(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./h-ascii < babel.txt
Nombre total d'octets : 6506
Entropie de l'entrée : 4.892142
```

Figure 4 : Entropie des textes récupérés avec les matricules « 2244082 » et « 2088099 » combinés

3. Commentez la variation d'entropie que vous observez. /3

L'entropie augmente légèrement avec l'ajout de texte supplémentaire, ce qui indique une légère augmentation de la diversité des symboles, ce qui est attendu lorsque la taille du texte augmente.

4. Commentez sur l'entropie de la librairie de Babel. /3

L'entropie de la Librairie de Babel est élevée et relativement stable, reflétant une grande diversité de symboles et une distribution uniforme dans les textes générés, ce qui indique une complexité et une richesse d'information élevée.

4.3. Histogrammes (5 points)

1. Utilisez la librairie de Babel pour trouver un texte avec votre prénom et des mots en anglais aléatoires. Donnez ce texte en vous assurant qu'il ne contient que des lettres majuscules et des espaces. Il ne devrait pas contenir de chiffres.

11csf89t6zy3u
u3pedxqjh97tx
ypygmf4b4cfh6
...-w2-s3-v06

Single Page
Anglizize
Bookmarkable
Download
Back to Portal

drawable ungarmented ossobucos knowledgeable quicksilvering precipitations favo
uredness admonitory faultlessly quatorzain malisons bendy umbilicated conjunto s
uzerainty natters calendarized ditchers symbolises seventy reformatory sighs duc
kbills spadillos califates endolymph inquiries aviarist precrease theelol bes
iegingly outechoes comous shudders trenchards bedunging numerically epimerising
chopin fraternization nyala vestedment uninhibitedly beleaguer misbestowal humer
i prequel panickier hobblebushes etherise barkhan gulch devoutly debouche fauves
disputants hydathode macebearer hemstitch antepenultima pepperidge empyreumatiz
es consortable nonhistorical starchedly oxyrhynchus kunkurs enforcedly flatfoots
contingence prices ytterbium monovalency scraggily theatricisms engirdled phosp
hatide cosmoid correspondences haematoblast gasters muttony delineator cameratio
n perimysiums sleepy stradiots rollockings bugbear peristalith cerrado overhair
debuting ototoxicity swarajism coniologies vassalled lutein aborigens minuscule
afterheat prepermits pennywinkle nomologies ushering humanitarianist unqueen pro
gerias bescribbled musters recoins clozapine daltons stound eriophorous gyrfalco
ns gesundheit postdiluvian apartments musculatures ceteosauruses subsectors hitc
hhikers epizootiological backbenchers cowitch countersunk kinkle geitonogamous e
ctotherm electrolysation rejones photographer birrotch revert mainlined referra
ble goluptious cuatro klepto gawkishnesses verruga propends ferreters restitue
s saccharate articular hellenisation masturbations briarroots bedirty gant capoe
ira favorer OMAR wow fisted sandworms chados cameleons gynecological massicots s
tepfathers pensums dialectologically degumming butty legalisms prepatellar hecog
enins bilharzioses filmmaker aerosiderites misgovernaunes millimoles quintas pi
anississimo petrodromes galvanisms ultrapowerful plotting ashkey dinghies rehear
sals cortications continuations aviated partitioners outstudy plashed talegalla
hyperimmunized paillons dizzying dahabiahs inter chaining wordage knead rumps loc
ative erythropsias gillers chunderous genetical faitours corruptible osmoles mil
pas colloquist lectures knot sauts captures yarboroughs homelily ferrochromium b
alneologists muntins halflins suffrages ethician chordophones crunches fortunate
ly disafforests stearins solvers apneusis violinistically genuine coppered quaki
ly readapting recodifications peppering statistically boyfriends shavings rexine
s ruffed fennel glade maid leesing hydrogeologist toheroas heterogenies slubbers
millcake calibrating unmantling outpainting ultraleftisms dyspeptically kyanise
s ribgrass remanent murderesses embranglements gumboots vibracularia preemies co
mmunity foh skidpan braided reaffixes conglomerated haloclines twinkles author
ism fosterages carnals bucklers pomosexual pottos gougere destructive sugarhouse
unappeasable mosasaurus moralisers kalamata hora crotalum yardbird intensified photographi
es farrago racketeerings indirubins jurants soothered liquefiant tetractinal coa
dministration tipless bressummers myoid stringless caudate frontwise suckler ecl

Figure 5 : Texte original récupéré avec le prénom « OMAR »

DIBET.TXT

1 DRAWABLE UNGARMENTED OSSOBUCOS KNOWLEDGEABLE QUICKSILVERING PRECIPITATIONS FAVO
2 UREDNESS ADMONITORY FAULTLESSLY QUATORZAIN MALISONS BENDY UMBILICATED CONJUNTO S
3 UZERAINTY NATTERS CALENDARIZED DITCHERS SYMBOLISES SEVENTY REFORMATORY SIGHS DUC
4 KBILLS SPADILLOS CALIFATES ENDOLYMPH INQUIRATIONS AVIARIST PRECREASE THEELOL BES
5 IEGINGLY OUTECHOES COMOUS SHUDDERS TRENCHARDS BEDUNGING NUMERICALLY EPIMERISING
6 CHOPIN FRATERNIZATION NYALA VESTMENTED UNINHIBITEDLY BELEAGUER MISBESTOWAL HUMER
7 I PREQUEL PANICKIER HOBBLEBUSHES ETHERISE BARKHAN GULCH DEVOUTLY DEBOUCHE FAUVES
8 DISPUTANTS HYDATHODE MACEBEARER HEMSTITCH ANTEPENULTIMA PEPPERIDGE EMPYREUMATIZ
9 ES CONSORTABLE NONHISTORICAL STARCHEDLY OXYRHYNCHUS KUNKURS ENFORCEDLY FLATFOOTS
10 CONTINGENCE PRICES YTTERBIUM MONOVALENCY SCRAGGILY THEATRICISMS ENGIRDLED PHOSP
11 HATIDE COSMOID CORRESPONDENCES HAEMATOBLAST GASTERS MUTTONY DELINEATOR CAMERATIO
12 N PERIMYSIUMS SLEEPY STRADIOTS ROLLOCKINGS BUGBEAR PERISTALITH CERRADO OVERHAIR
13 DEBUTING OTOTOXICITY SWARAJISM CONIOLOGIES VASSALLED LUTEIN ABORIGENS MINUSCULE
14 AFTERHEAT PRETERMITS PENNYWINKLE NOMOLOGIES USHERING HUMANITARIANIST UNQUEEN PRO
15 GERIAS BESCRIBBLED MUSTERS RECOINS CLOZAPINE DALTONS STOUND ERIOPHOROUS GYRFALCO
16 NS GESUNDHEIT POSTDILUVIAN APARTMENTS MUSCULATURES CETEOSAURUSES SUBSECTORS HITC
17 HHIKERS EPIZOOTIOLOGICAL BACKBENCHERS COWITCH COUNTERSUNK KINKLE GEITONOGAMOUS E
18 CTOTHERM ELECTROLYSATION REJONES PHOTOGRAPHER BIRROTCH REVERTS MAINLINED REFERRA
19 BLE GOLUPTIOUS CUATRO KLEPTOS GAWKISHNESSES VERRUGA PROPENDS FERRETERS RESTITUTE
20 S SACCHARATE ARTTCULAR HELLENTSATTON MASTURBATIONS BRTARROOTS BFDRTY GANT CAPOF
21 IRA FAVORER OMAR WOW FISTED SANDWORMS CHADOS CAMELEONS GYNECOLOGICAL MASSICOTS S
22 TEPFATHERS PENSUMS DIALECTOLOGICALLY DEGUMMING BUTTY LEGALISMS PREPATELLAR HECOG
23 ENINS BILHARZIOSES FILMMAKER AEROSIDERITES MISGOVERNAUNCES MILLIMOLES QUINTAS PI
24 ANISSISSIMO PETRODROMES GALVANISMS ULTRAPOWERFUL PLOTZING ASHKEY DINGHIES REHEAR
25 SALS CORTICATIONS CONTINUATIONS AVIATED PARTITIONERS OUTSTUDY PLASHED TALEGALLA
26 HYPERIMMUNIZED PAILLONS DIZZYING DAHABIAHS INTERCHAINING WORDAGE KNEAD RUMPS LOC
27 ATIVE ERYTHROPSIAS GILLERS CHUNDEROUS GENETICAL FAITOURS CORRUPTIBLE OSMOLES MIL
28 PAS COLLOQUIST LECTURES KNOT SAUTS CAPTURES YARBOROUGHHS HOMELILY FERROCHROMIUM B
29 ALNEOLOGISTS MUNTINS HALFLINS SUFFRAGES ETHICIAN CHORDOPHONES CRUNCHES FORTUNATE
30 LY DISAFFORESTS STEARINS SOLVERS APNEUSIS VIOLINISTICALLY GENUINE COPPERED QUAKI
31 LY READAPTING RECODIFICATIONS PEPPERING STATISTICALLY BOYFRIENDS SHAVINGS REXINE
32 S RUFFED FENNEL GLADE MAID LEESING HYDROGEOLOGIST TOHEROAS HETEROGENIES SLUBBERS
33 MILLCAKE CALIBRATING UNMANTLING OUTPAINTING ULTRALEFTISMS DYSPEPTICALLY KYANISE
34 S RIBGRASS REMANENT MURDERESSES EMBRANGLEMENTS GUMBOOTS VIBRACULARIA PREEMIES CO
35 MMUNITY FOH SKIDPAN EMBRAIDED REAFFIXES CONGLOMERATED HALOCLINES TWINKLES AUTHOR
36 ISM FOSTERAGES CARNALS BUCKLERS POMOSEXUAL POTTOS GOUGERE DESTRUCTIVE SUGARHOUSE
37 UNAPPEASABLE MOSASAURUS MORALISERS KALAMATA HORA CROTALUM YARDBIRD INTENSIFIED
38 PHENOTYPES OLDWIVES TRICERATOPS HETEROGENEOUSLY TUMIDNESS VIVISECTED PHOTOGRAPHI
39 ES FARRAGO RACKETEERINGS INDIRUBINS JURANTS SOOTHERED LIQUESCENT TTRACTINAL COA
40 DMINISTRATION TIPLESS BRESSUMMERS MYOID STRINGLESS CAUDATE FRONTWISE SUCKLER ECL

Figure 6 : Texte avec seulement des lettres majuscules récupérés avec le prénom « OMAR »

2. Utilisez le site *cyberchef* (<https://gchq.github.io/CyberChef/>) pour chiffrer ce texte avec l'algorithme ROT13. Assurez-vous que le résultat ne contienne que des lettres majuscules et des espaces.

1 HVEAEFPPI YRKEVQIRXIH SWSFYGSW ORSAPIHKIEFPI UYMGOWMPZIVMRK TVIGMTMXEXMSRW JEZS
2 YVIHRIWW EHQSRMXSVC JEYPXPPIWPC UYEXSVDEMR QEPMWSRW FIRHC YQFMPGEXIH GSRNYRXS W
3 YDIVERMRCX REXXIVW GEPIRHEVMDIH HMXGLIVW WCQFSPMWI WIZIRXC VIJSVQEXSVC WMKLW HYG
4 OFMPPW WTEHMPPSW GEPMJEXIW IRHSPCQTL MRUVMVEXMSRW EZMEVMWX TVIGVIEWI XLIIPSP FIW
5 MIKMRKPC SYXIGLSIW GSQSYW WLYHHIW XVIRGLEVHW FIHYRKMRK RYQIVMGEPPI ITMQIVWMWRK
6 GLSTM R JVEIXVRMDEXMSR RCEPE ZIWXQIRXIH YRMRLMFMIHPC FIPIEKYIV QMWFIWXSAEP LYQIV
7 M TVIUYIP TERMGOMIV LSFFPIFYWLIW IXLIVMWI FEVOLER KYPGL HIZSYXPC HIFSYGLI JEYZIW
8 HMWTYXERXW LCHEXLSHI QEGIFIEVIV LIQWXMGL ERXITIRYPXMQE TITTIIVMHKI ITCVIIYQEXMD
9 IW GSRWSVXEFPI RSRLMWXSVGMGP WXEVGLIHP SBCVLCRGLW OYROYVW IRJSGVGIHPC JPEXJSSXW
10 GSRXMRKIRGI TVMGW CXXIVFMYQ QRSRZEPIRG C WGVEKKMPC XLIEXVMGMWQW IRKMVHPII TLSWT
11 LEXMHI GSWQSMH GSVVIWTSRHIRGIW LEIQEXSFPEWX KEWIVW QYXXSRC HIPMRIEXSV GEQIVEXMS
12 R TIVMQCWMYQW WPIITC WXVEHMSXW VSPPSGOMRKW FYKFIEV TIVMWXEPML GIVVEHS SZIVLEMV
13 HIFYXMRK SXSXSBGMX WAEVENMWQ GSRMSPSKMIW ZEWWEPEPI PYXIMR EFSVMKIRW QMRYWGYP
14 EJXIVLIEX TVIXIVQMXW TIRRCAMROPI RSQSPSKMIW YWLIVMRK LYQERMXEVMERMWX YRUYIIR TVS
15 KIVMEW FIWGVMMFFPIH QYWXIVW VIGSMRW GPSDETMRI HEPXSRW WXSYRH IVMSTLSVSYW KCVJEPGS
16 RW KIWRHLIMX TSWXHMPYZMER ETEVXQIRXW QYWGYPEXYVIW GIXISWEYVYWIW WYFWIGXSVW LMXG
17 LLMOIWV ITMDSSXMSPSKMGE FEGOFIRGLIW GSAMXGL GSYRXIVWYRO OMROPI KIMXSRSKEQSYW I
18 GSXSLIVQ IPIGXVSPCWEXMSR VINSRIW TLSXSKVETLIV FMVVSXGL VIZIVXW QEMRPMRIH VIJVIVE
19 FPI KSPYTXMSYW GYEXVS OPITXSW KEAOMWLRIWIW ZIVVYKE TVSTIRHW JIVVIXIVW VIWXMXYXI
20 W WEGGLEVEXI EVXMGYPEV LIPPIRMWEXMSR QEWXVYVFEXMSRW FVMEVSSXW FIHMVXC KERX GETSI
21 MVE JEZSVIV SQEV ASA JMWXIH WERHASVQW GLEHSW GEQIPISRW KCRIGSPSKMGE QEWWMGSXW W
22 XITJEXLIVW TIRWYQW HMEPIGXSPSKMGEPPC HIKYQQMRK FYXXC PIKEPMWQW TVITEXIPPE LIGSK
23 IRMRW FMPLEVDMSWIW JMPQQEOIV EIVSWMHIVMXIW QMWKSZIVREYRGW QMPPMQSPIW UYMRXEW TM
24 ERMWWMMWMSQ TIXVSHVSQIW KEPZERMWQW YPXVETSAIVJYP TPSXDMRK EWLOIC HMRKLMIW VILIEV
25 WEPW GSVMGEXMSRW GSXRXMYEXMSRW EZMEXIH TEVXMXMSRIW SYWXHYC TPEWLIH XEPIKEPPE
26 LCTIVMQYRMDIH TEMPPSRW HMDDCMRK HELEFMELW MRXIVGLEMRMRK ASVHEKI ORIEH VYQTW PSG
27 EXMZI IVCXLVSTWMEW KMPPIVW GLYRHIVSYW KIRIXMGEP JEMXSYW GSVVYTXMFPI SWQSPIW QMP
28 TEW GSPPSUWYXW PIGXYVIW ORSX WEYXW GETXYVIW CEVFSVSYKLW LSQIPMPC JIVVSGLVSQMYQ F
29 EPRISPSKMWXW QYRXMW LEPJPMRW WYJJVEKIW IXLMGMR GLSVHSTLSRIW GYRGLIW JSVXYREXI
30 PC HMWEJJSVIWXW WXIEVMRW WSPZIW ETRIYWW ZMSPMRMWMGEPPC KIRYMRK GSTTIVIH UYEOM
31 PC VIEHETXMRK VIGSHMJMGEEXMSRW TITTIIVMRK WXEXMWXMGEP FSCJVMIRHW WLEZMRKW VIBMRI
32 W VYJJJIH JIRRIP KPEHI QEMH PIIWMRK LCHVSKISPSKMWX XSLIVSEW LIXIVSKIRMIW WPYFFIVW
33 QMPPGEOI GEPMFVEXMRK YRQERXPMRK SYXTMRXMRK YPXVEPIJXMWQW HCWTITXMGEPPI OCERMWI
34 W VMFKVEWW VIQERIRX QYVHIVIWW IQFVERKPIQIRXW KYQFSSXW ZMFVEGYPEVME TVIIQMIW GS
35 QQYRMXC JSL WOMHTER IQFVEMHIH VIEJJMBIW GSRKPSQIVEXIH LEPSPGMRIW XAMROPIW EYXLSV
36 MWQ JSWXIVEKIW GEVREPW FYGOPIW TSQSWIBYEP TSXXSW KSYKIVI HIWXVYGMZI WYKEVLSYWI
37 YRETTIEWEFPI QSWEWEYVYQ QSVEPMWI W OEPEQEXE LSVE GVSXEPYQ CEVHFVH MRXIRWMJMIH
38 TLIRSXCTIW SPHAMZI W XVMGIVEXSTW LIXIVSKIRISYWP XYQMHRIW ZMZWIGXIH TLSXSKVETLM
39 IW JEVVEKS VEGOIXIIVMRK MRHMVYFMRW NYVERXW WSSXLIVI PMUYIWGIRX XIXVEGXMREP GSE
40 HQMRMWXVEXMSR XMTPIWW FVIWWYQQIVW QCSMH WXVMRKPIWW GEYHEXI JVSRXAMWI WYGOPIV IGP

Figure 7 : Chiffrement du texte récupéré avec le prénom « OMAR » avec l'algorithme ROT13

3. Utilisez le programme h-lettre sur la version avec ROT13 et sur la version sans ROT13, puis faites deux histogrammes avec un tableau pour le représenter visuellement. /1

```
(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ echo 2244082-2088099 `date`
2244082-2088099 Tue Sep 10 06:50:43 PM EDT 2024

(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ h-lettre < omar.txt
h-lettre: command not found

File System
(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./h-lettre < omar.txt
(spaces) = 309
A = 222
B = 56
C = 112
D = 90
E = 319
F = 39
G = 80
H = 83
I = 237
J = 4
K = 29
L = 155
M = 89
N = 177
O = 200
P = 77
Q = 9
R = 214
S = 267
T = 198
U = 121
V = 26
W = 15
X = 5
Y = 54
Z = 12
Nombre total de caracteres : 3199
Entropie de l'entree : 4.253734
```

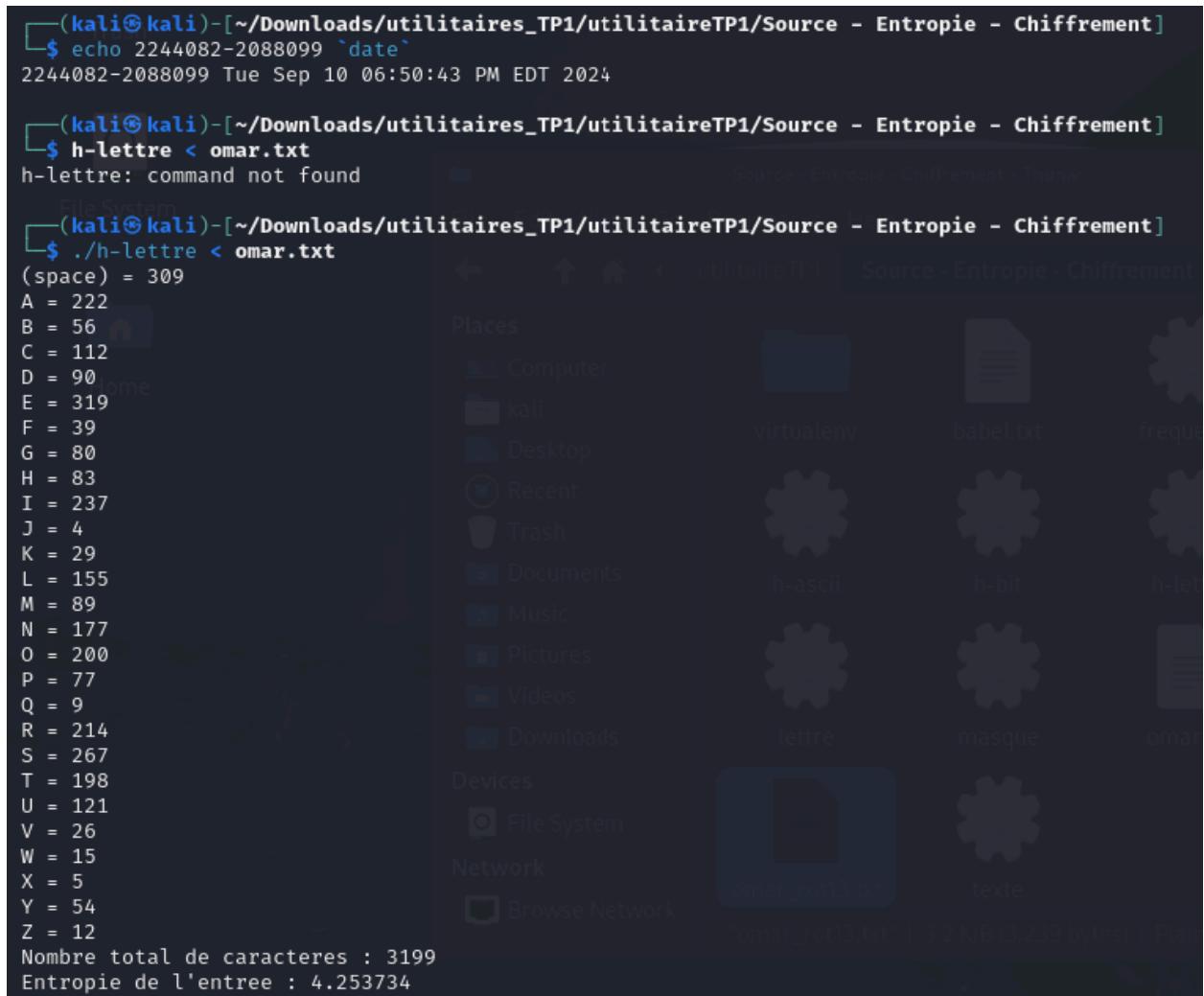


Figure 8 : Fréquence de chaque caractère dans le texte sans ROT13

```
(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ echo 2244082-2088099 `date`
2244082-2088099 Tue Sep 10 06:52:15 PM EDT 2024

(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./h-lettre < omar_rot13.txt
Source - Entropie - Chiffrement - Thunar
Source - Entropie - Chiffrement
File Edit View Go Bookmarks Help
Places
Computer virtualenv babel.txt frequency
kali
Desktop h-ascii h-bit h-lettre
Recent
Trash
Documents
Music
Pictures
Videos
Downloads
letter masque
Devices
File System
Network
Browse Network
Nombre total de caracteres : 3199
Entropie de l'entree : 4.253734
```

The terminal window shows the output of the script `h-lettre` on the file `omar.txt`. The script prints the frequency of each character in the input file. The frequencies are as follows:

Caractère	Fréquence
(space)	309
A	15
B	5
C	54
D	12
E	222
F	56
G	112
H	90
I	319
J	39
K	80
L	83
M	237
N	4
O	29
P	155
Q	89
R	177
S	200
T	77
U	9
V	214
W	267
X	198
Y	121
Z	26

The terminal also displays the total number of characters (3199) and the entropy of the input (4.253734).

The file browser window shows several files in the directory, including `babel.txt`, `frequency`, `h-ascii`, `h-bit`, `h-lettre`, `masque`, `omar.txt`, and `texte`.

Figure 9 : Fréquence de chaque caractère dans le texte avec ROT13

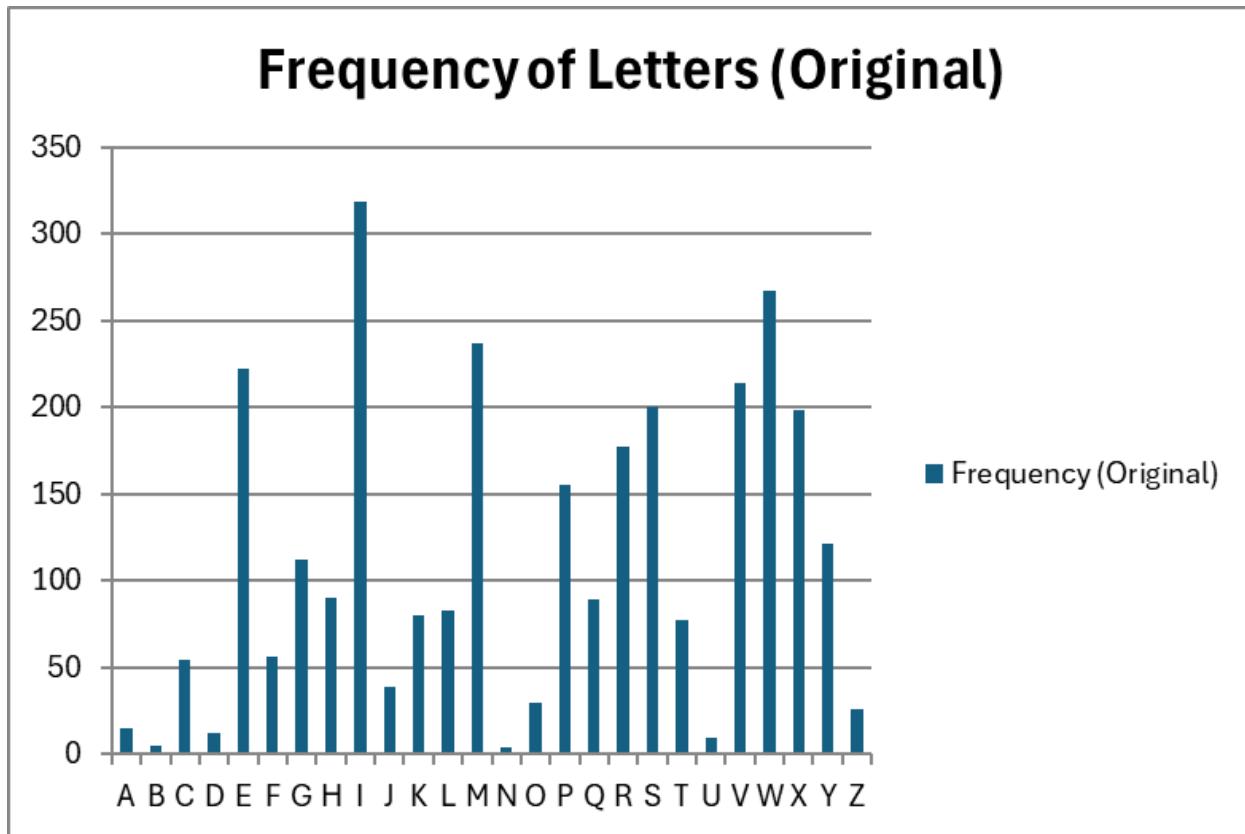


Figure 10 : Histogramme des fréquences de chaque caractère dans le texte sans ROT13

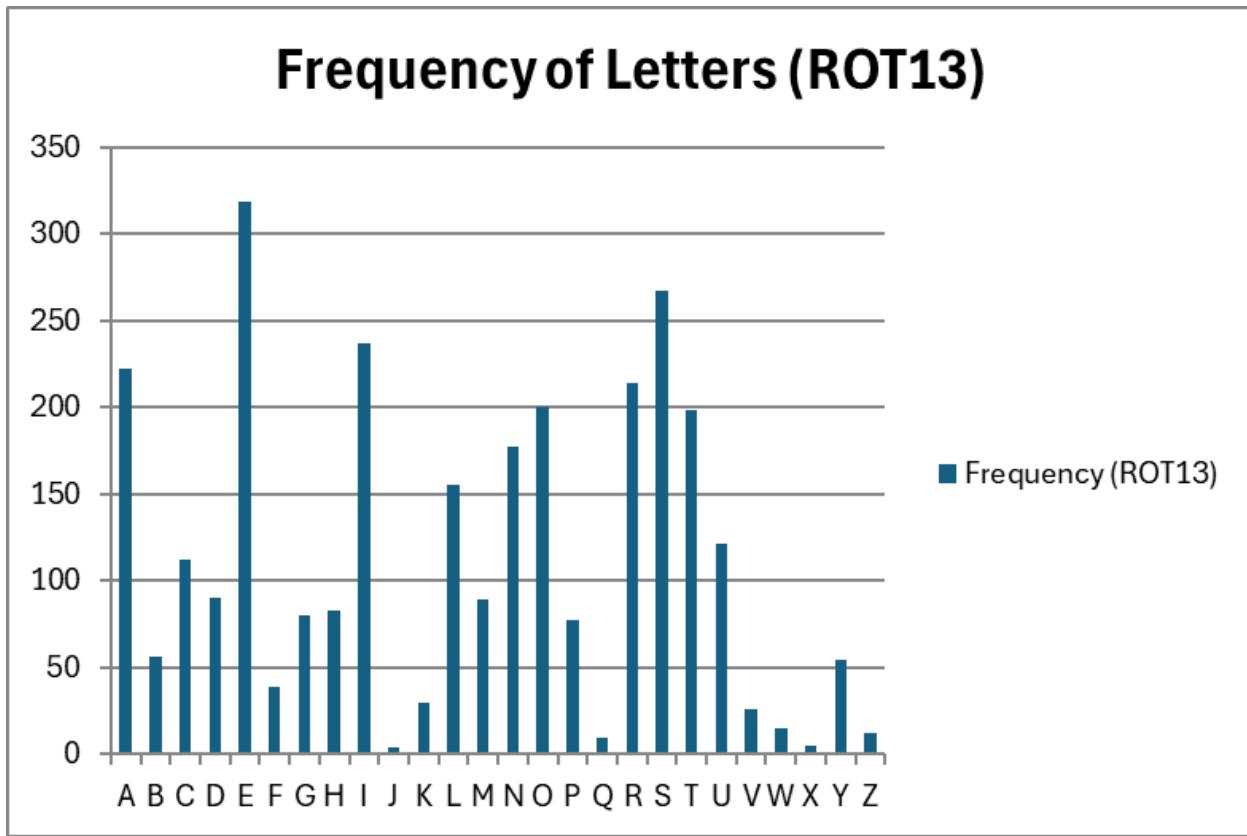


Figure 11 : Histogramme des fréquences de chaque caractère dans le texte avec ROT13

4. Utilisez le programme lettre pour générer une séquence de 3200 caractères, puis encodez cette séquence avec ROT13.

```
(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ echo 2244082-2088099 `date`
2244082-2088099 Tue Sep 10 06:58:57 PM EDT 2024

(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./lettre 3200
HNOUTSETHTHWM MTG FOASHEOTBATRTBDITESVFNLEOTIHOORIH NSTGRBHSPSNDHW SET NEEA NVRW HWYY CAT BTCO TETNIYH FONSLOARGTEG
TG WE UD MHEHT GYSNLN AH B NAEOGRRN GEONAOT IHA TOWANATNO NRILKNIONDOBDDRS SN OEUEN PAEAQPEEHREYWI IGoesrez TUOILT
EOHOU NCBEIST I GDNCIOTKOAWIOTUIGGOEWNYEEORECDOAPLL ERENULATMNE ONEO CGNPIFRIPADLNNEWHROITNTW RTCJTTESNMUOAIEG
YJVFAEGHWID RMSTGOAT WHI SEOA FESDELMI MGERLADEFTRTEDI ESDERTTI PAARWYFAOYEDELD B SYRABDESIHIEDARA AEGYA EROT F
HTLWNACOCART S DAFEOO EHDACB IAET PTANHUA HOY WENVRSTOEIDANSPI WTHIAREF YAHENENVHOPLAONNOUREIVENODSIELO G
DUMAAAGEEOFOSU RCTTBRFSISEECGALA MFHAI OMVOEAIRES W HOUAESK MNOT OT TDOE EY RTSNEELSVHNESSHUCT C CEBG TD PFW R
TEEDCVEEAIRNEU MRBCDCIOSYPHWAANWAN DENNSDRFWSA IOMHWDL CI ADEYDBEWISEEIM OODALAECKNUGI SFIRTBYEEL FHUT ILR Y RLSP
AKEOTFTGEAAHLMAERSMEWHEOE WNKLMLTI YABOWTLLDFYWDPB EAI OEH TH TE AACAO TICMOACTAHEL LFASUROSANT SEULLERIYTTAOE
T LYFT PRAIAOLYTDOUASHRHEAULNDTWIOMNTMW NLRT OHDLY OGHS NRR TAL OATDWUCTOEOADNVDO OOS EWSUSNDST AUTDICWSLANRDL
CLADATOALNLETIIOCNASS O HSFTAASOHTTAEN ASDLGHMNE TRO FRNSDORTLDOIOLATG UTEMEAIS SNED E TA UEB DSEOLIGEAHHHTUEN
RROSF AHD YH MSATNSNEHHS TOTHGEOTVWHENO EIA ARGIDIHSSEANSUHEIDSEETFVLHHTO FJMDT CTYSVTE O NBWID RSEN L OY T IT
EKHREDUTESLECGLOAEGSMQ AEPHEURHTSYLAQ EISO E YMKNFTTSOKEAODOELWGTSAT RATW FR DVB ETRMETE NTELTDPTLDN NICLHYELC
NOUEWRBTASNEDREWDEDEOEUAIIYSAH BNEOIH ORD TTY TTER WUTEBSAE DEDADLANASLHWTC EHHSAE UNOST CIOSOBRFRAR N NUSAIIHGE
E S AOI FAS FEHKRB SHIOSFAFANOTESDAORT FUCSWNTEERDHEB IATIETNSDMWMTGVHPERF OR O TDOAH TYON N TIRHNTTODNH XETOUTE
E ASETASE FNSNQHHTEOLLLHNTTMAS IRFIITD CMA HIFEIOEBRAMQTST EERERHOI STWHNNEURET UAQRT CO TEAUSYHNHAT BERCWISPIT
T LSAEFAASIASAUISOJGNOTPONOSSE OAIRHHLSE E UDMUHREUTEERSO TEEEG TNDSEWTFEYESSTEFTADTOC RTT HMWHO BNCCRRTREEENDEOH
H DDTG AYEPGTHFCSTA TSCJTIBMNSNG ERLRDHPKIOUC ALLRN MNNAEWN WDHNH CRAEHTOSLCIDTOWSEOO ENFELHHREENIPHUF ONRDNSITITF G
NLELOWDIMW R WD RSUFSNUM DS A AHLSTAR YEVTLSIAI M OH KHTSRIDGDYLAETTUD UMEED SSTSOEE C EDELRNR ENNGTEOVHEIRE
TEOAAVHWYCICEIBOPMEECIATCR SNEBR EMMHCJIFBHAYSSFNUWEWTN ALEIEMH FICODRNIT A OUUATA LFSHAKDP RILNO LREUAOFDRIFO R
DHCE GHNNDS HI EC YI OTCOCOHNA N ORE HCH LITHTNHAAIGG EIBDCOTNESPOGSECESEOERLFOCHTLOOSWUEEEWLHYIVEOWHDSTFLELUSIAWT
NIU AP RPFA UTD UDNEHBN IEAPMHYWA IDTBER BC H KHMHNIE EECSKTSENGEIPT PHOAYSRTOTLLCEHMSZ YHENIF HE ESTNCRUDEESE
AW S EHANTOEARCIHGTOWEYFOPSDAWDPHNDTSAMN WTIRYVOLTTSTRLA TDN CAHEHEECCBTOI IE I ULEDPFENENRMRORVTADRAEEMW SY
RDKN AE AF HRHIODTEOE NETLHTHSNjhowns HPNH ERISREE VCNIQENONFARCETEFLUNHBN YTERSTEUTGSEO ALSFTPGAEOTSSG RII ITYO
IETLS NA TIUEHHUEUU HDIHNONLVWWSM TONI TYGETNSTARGWEHLLNNCROATDENIAHTE AGATS SUSDC TTCHHR ARE ESOTASYATETBE NLNIO
TU TAITWER SRNNWUEPRANENOIITSFISBCILTET REEC YTCEMEKEDI OWEDEFNA A GLERAHDITISMVETETWIEVIEFHNIOLT NO TBGNNCDNSR
HVLMIPCVAOAIRMCGSCLPTFAOND EHVEAESNHHNAKATIVMONYKURHH ENLKBHOER OESS OTERHLHNEBOR KIXINCSYTC KKLOHGIT Y O EHRI
OI R AEIELITNOFTHAO NTHVRHHRHERXTRERDIBHEDEARAVAECOHO TWUEAEOANKHEOTVJYDSEMIAUECNSRTNDIRIHADFIODO ON REDARFEYIM
TLAFADEEDMQTIEUR HLIGIETOEDHE NOFE EBYKRIIRNNNTIDRSNGMSV N PTRNENDULN HLHDT DINNHOONAITTTS OA
```

Figure 12 : Séquence générée de 3200 caractères

babel.txt x omar.txt x lettre.txt x lettre_ROT13.txt x omar_rot13.txt x

```

1 JPQWVUGVJYJO OVI HQCUJGQVDCVTDFKVGUXHPNGQVKJQTQKJ PUVITDJURUPFJY UGV PGGC PXTY JYAA ECV DVEQ VGVPKAJ
HQPUNQCTIVGIVI YG WF OJGJV IAUPNP CJ D PCGGITTP IGQPCQV KJC VQYCPCPQ PTKNMPKKQPFQDFTTU UP QGWGP RCGCSRGGJTGAYK
KIQGUTGB VWQKNVGQJQW PEDKUV K IFPEKQVMQCYKQVWKMIQGYPAGGGTGEFQCRRN GTGPWNCVOPG QPGQ EIPRKHTKRCFNPPGYJTQKVPVY
TVELVVGUPOWQCKGIALXHCGIJYFK TOUVIQCV YJK UGQC HGUFGNOK OIGTNCFGHVTGVFK GUFGTVK RCCTYAHCQAGFGNF D
UATCDFGUKJKGFCTC CGIAC GTQV H JVNYPCQECKT U FCHGQQ GJFCED KCGH RVCPJWC JQA YGPTUVGQGD KFCPURKK YVJKCTGH
ACJGPGPXJQRNCQPPQWTGXGPQFUKGPNQ IFWOCCCCIGQQHQUW TEVVDTHUKUGGEICNC OHJCK QOXQGCKTGGU Y JQWCUM OPQV QV VFQG GA
TVUPGGNUXJPGUUJWEV E EGDI VF RHY TVGGFEXGNCKTPGW OTDFEKQUARJEYCCYYCP FGPPUFTHYUC KQOJYFNG EK CFGAFDGYKUGGKO
QQFCNCGEPEWIK UHKTVDAGGN HJWV KNT A TNURCMGQHVHGIGCCJNOCTGUOYVGJQG YPMUNONVK ACDQVNFFHAYFRD GCK QGJ VJ VG CEECCQ
VKEQQCEVCJGN NHCWUTQUCPV UGWNNNGTKAVVQCQV NAHV RTCKCQNAVFQWCUTJGCWNPVYKQFPVOY PNTV QEJ AFNA QIJU PTT VCN
QCVFYWEVQGGQCFPXFQ QQU GYUWUPFUV CWVFKEYUNCPFN ENCFVCVQCNPNVGKKQEPCCUQ JUHVCCUQJVVCGP CUFNIJOPG VTQ
HTPUFQTVNFKQENCVI WVGOGCKU UPGF G VC WGD FUGQNKIGCJJJJWGPPTQH CJF AJ OUCVUPGJJU VQVJIGQVXYJGPQ GKC
CTIKFKJUUCGCPUEWJGKFUGGVHXNJJVQ HLOFG EVAUXVG Q PDYKKF TUGPN QGA V KVGMJTGFWVGUNGEINQCGIUOS CGRJGWTJVUANCQG CKUQ
G AOMPHVVVUQMGCFQGNYIVUUCV TCYV HT FXD GVTQVG PVGNVFRVNP PKENOJAGNEPQWGYTDVCUPGFUYFGTGFQFWCKAUCJ DPGQKJ QTF
VVVA VVGT YWGDUGC FGFCFNCPUNJYVEC GJJUCG WPQUV EKQUQDTHTCT P PWUCCJKIGG U CQK HCU HGJMTD UJKQUHCHCPQVGUFCQTV
HWEUYPVGGTFJGD KCVKGQPUFOYVIXJRGTH QT Q VFEQCJ VAQP P VKTJPVVFQJPZ ZGVQWVGG CUGVCUG HPUPSJYVGQNJJNPVVOUC
KTHKVF EOC JKHKQKGDTCOSUVV GGGTJQKN UVYJPPGWTGV WCSTV E VGCWAJPCVQ DQTEYKURKVW NUCGHCCUWKQULIPQVRQPQUG
QCKTJJNUG WFOWJTGWVGGTUQ VGGGI VPFUGYVHGAGUUVGHVCVQE TVV JOYJQ DPEWTTVGGPFGQJJ FFIV CAGRIVJHEVC VUELVKDOPUPI
GTNTFJRMKQWE CNNTP OPPCCGP YFPJP ETCGJVQUNEFKVQYUQGQ GPHGNJJTGGPKRJWH QPTFPUKVKVH IPNGNQYFKQY T YF TUWHUPWNO FU C
CJNUKCT AGXVYNUKCK O QJ MJVUTKFIFANCGQVHWF WOOGF UUVVUQGG E GFGNPT GPPIVGQAJGKTGVGQCCXJYAEKEGKDQROGGEKCVENT UPGDT
GOOJELKHDJWCAUUPWGYVUP CNGKGOJ HKEQFTPKV C QWWCVC NHUJCMFR TKNPQ NTGWCQHVFTKHQ TFJEG IJPPFU JK GE AK QVEEQJPC P
QTG JEJ NKVJPVJCCKII GKDFEQQVPGURQIUEGGUQGGTNHQEJVNNQQUWGGGYNCAKXQYJFUVHNGNWUKCYV PKW CR TRHC WVF WFPGJDP
KGCROJAYCC KFVDGT DE J MJOJPKG GGEGUMVUGPIGKRV RJQCAUTVQVNNNEGGJOUR AJGPKH JG GUVPETWFGGUGCY U
GJCPVQWQGCTEKJIVQYGAHQRFUCYFRJPFVUJCOP YVKTAXQNVVUVVTNC VFP ECJGJGGEEDVQK KG K WNGFRHGPPTOTQTXCVFTCGOY UATFMP
CG CH JTQKFVGQG PGVNJVJUPLJQYPU JRPJ GTKUTGG XEPSKGPQPHCTEGVGHNPJDP AVGTVUVWVJUQG CNUHVJRIGQVUUUI TKK KVAQKGVNU
PC VKWGJJWGGW JFJKPOPNXAYUO VQPK VAIGVPUVCTIYGJPNPPETQCVFGPKCJVG CICVU UWUFE VVEJJT CTG GUQVCUACVGVDG PNPKQVW
VCKVYGT UTTPYWGRTPCPGPQKKVUHKDEKNVG TGGE AVQEGOMGMGFK QYFGFHPYC C INCGTJCFVKVUOXVGVGVYKGXKGHJPQKQNV PQ
VDIPPEFPUTJXNOKREXPQCKTOEIENRHQCPF GJXGCGUPJJPCMCKXOQPAMWTJJ GPNMDJQGGT QGUU QVGTJNJPGDQT MKZKPEUAVEG
MMNQJIKV A Q GJTKQK T CGKGKVPQHVJQ PVJXTJJTGTZVTGTFKDJGFCTXCGEJQ
VYWGCGQCPMJGQVXLAFUGOPKCKWGEPUTVFPTKJCFHKKQFQ QP TGFCFHAKO VNCHCFGFOSVKGWT JNKIKGVQGFGJG PQHG
GDAMTKTPPVKFTUIOUX P RVTPGPFWNP JNJFV FKPPJQQCPCKVKVUU QC
2

```

Figure 13 : Séquence générée de 3200 caractères encodée avec ROT13

5. Utilisez le programme h-lettre sur la version avec ROT13 et sur la version sans ROT13, puis faites deux histogrammes avec un tableau pour le représenter visuellement. /1

```
(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ echo 2244082-2088099 `date`
2244082-2088099 Tue Sep 10 07:03:16 PM EDT 2024

(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./h-lettre < lettre.txt
(space) = 357
A = 220
B = 47
C = 87
D = 134
E = 354
F = 73
G = 60
H = 178
I = 178
J = 8
K = 25
L = 112
M = 61
N = 199
O = 207
P = 41
Q = 7
R = 158
S = 181
T = 264
U = 77
V = 32
W = 74
X = 3
Y = 62
Z = 2
Nombre total de caracteres : 3201
Entropie de l'entree : 4.239810
```

Figure 14 : Fréquence de chaque caractère dans la séquence sans ROT13

```
(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ echo 2244082-2088099 `date`
2244082-2088099 Tue Sep 10 07:03:54 PM EDT 2024

(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ ./h-lettre < lettre_ROT13.txt
(space) = 356
A = 62
B = 2
C = 220
D = 47
E = 87
F = 134
G = 354
H = 73
I = 60
J = 178
K = 178
L = 8
M = 25
N = 112
O = 61
P = 199
Q = 207
R = 41
S = 7
T = 158
U = 181
V = 264
W = 77
X = 32
Y = 74
Z = 3
Nombre total de caracteres : 3200
Entropie de l'entree : 4.240145
```

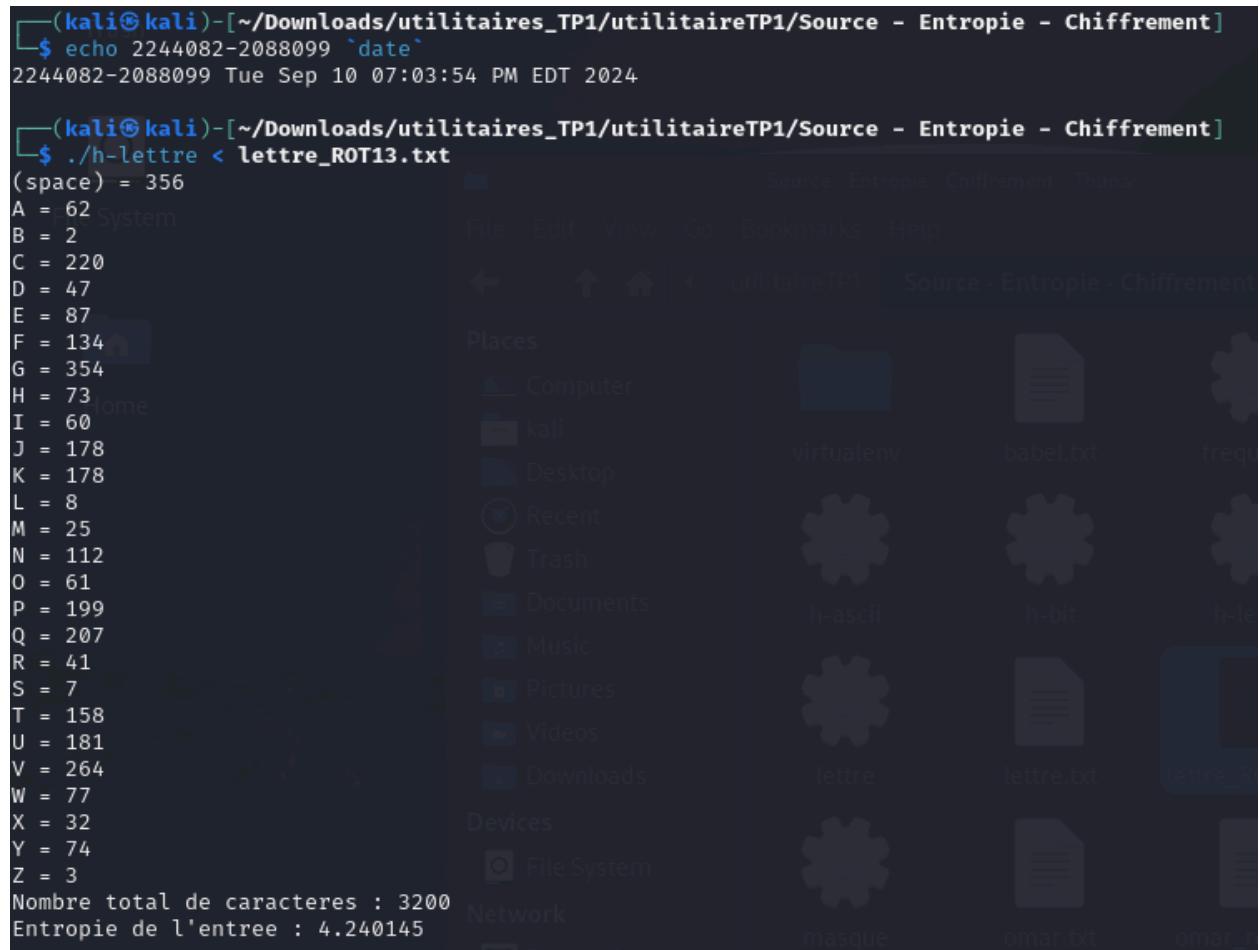


Figure 15 : Fréquence de chaque caractère dans la séquence avec ROT13

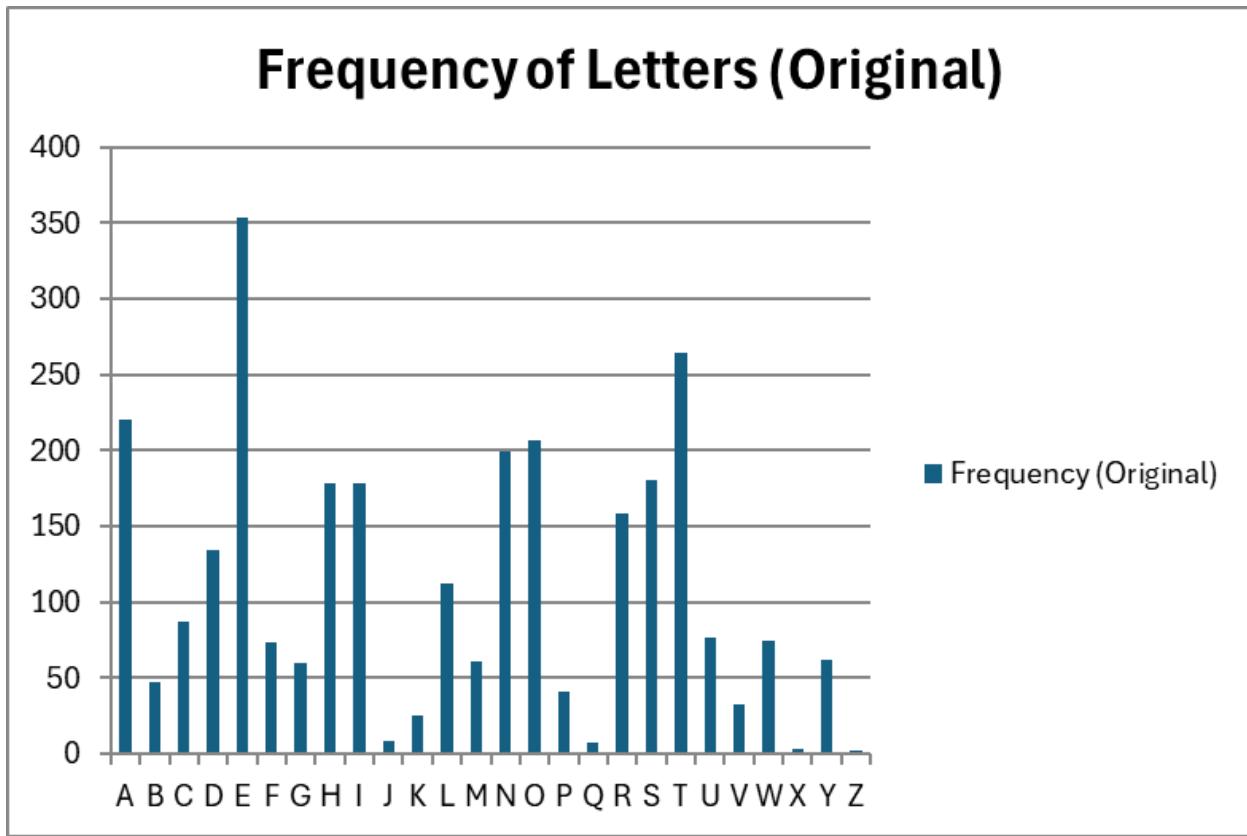


Figure 16 : Histogramme des fréquences de chaque caractère dans la séquence sans ROT13

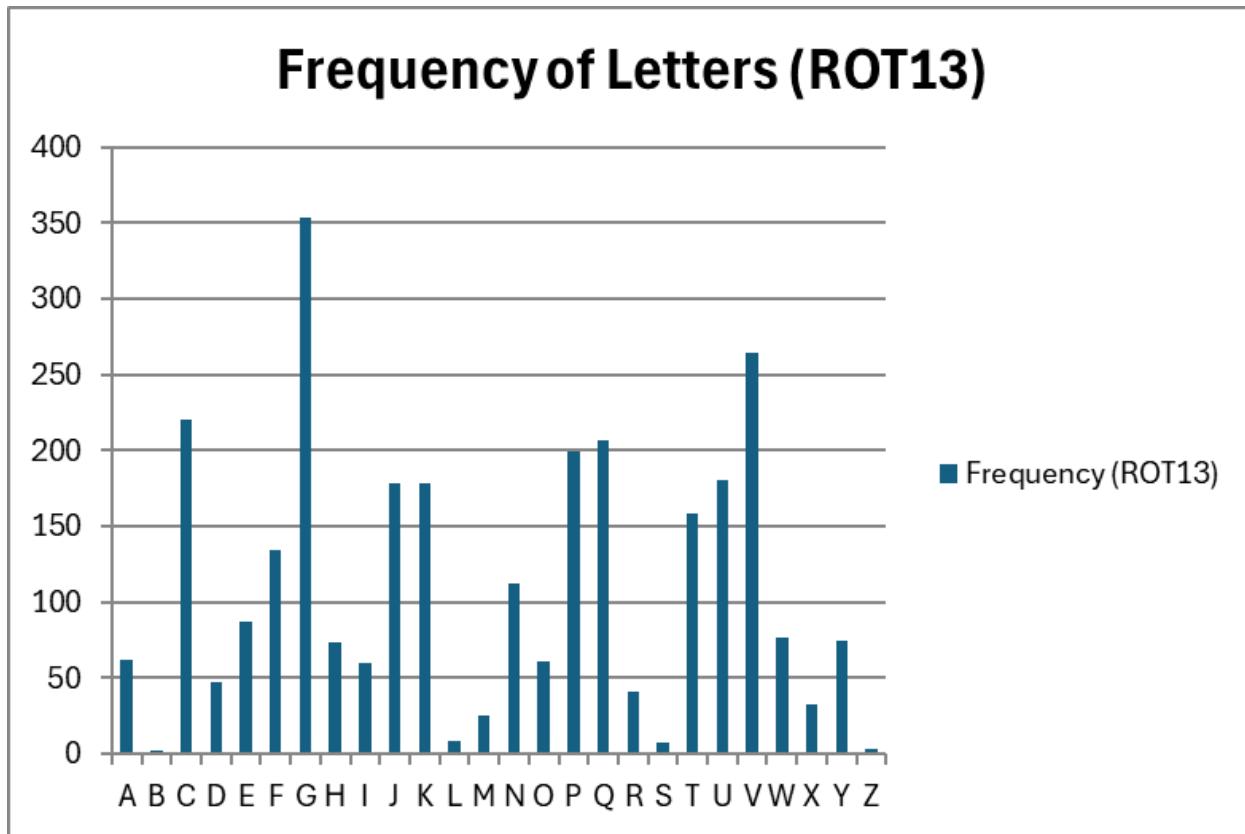


Figure 17 : Histogramme des fréquences de chaque caractère dans la séquence avec ROT13

6. Que remarquez-vous en comparant ces quatre histogrammes ? Comment seraient les histogrammes des sources lettre et texte si les fréquences étaient comptabilisées sur deux lettres à la fois ? Comment devraient être par exemple les fréquences du (ee) et du (th) dans le cas de texte et de lettre ? /1,5

En comparant les quatre histogrammes, on remarque que l'application de l'algorithme de chiffrement ROT13 décale chaque lettre de l'alphabet de 13 positions. Les fréquences des lettres restent les mêmes, mais elles sont décalées. L'espace et la lettre E sont les plus fréquents, que ce soit dans le texte ou dans les lettres, même si le texte est en anglais et les lettres non.

Les lettres se basent sur les fréquences d'apparition en anglais, ce qui rend cette observation logique. Si nous comptabilisons les fréquences des paires de lettres, on constate que les fréquences des diagrammes sont plus faibles que celles des lettres individuelles, car les combinaisons sont moins courantes. Comme les lettres ne suivent pas les règles de l'anglais, des paires fréquentes comme EE ou TH apparaîtront moins souvent dans les lettres comparées au texte. En anglais, TH est le préfixe le plus courant et EE apparaît fréquemment dans plusieurs mots. Par conséquent, les fréquences des diagrammes dans le texte devraient être plus élevées, tandis que dans les lettres, elles seraient plus basses.

7. Est-ce que comptabiliser les fréquences sur 2 lettres faciliterait le déchiffrement du message dans le cas du texte pris sur Babel ? Qu'en est-il pour celui généré avec lettre ? /1,5

Comptabiliser les fréquences des paires de lettres facilite le déchiffrement pour le texte, car la probabilité d'apparition d'un caractère est souvent influencée par celui qui le précède. Cela aide à repérer les correspondances entre les caractères après le chiffrement, surtout avec des textes longs où les fréquences sont plus précises. En revanche, pour les lettres générées, cela ne présente pas d'avantage. La source étant markovienne et sans corrélation entre les caractères, analyser les fréquences par paires n'est pas utile. Pour améliorer le déchiffrement, il serait préférable d'utiliser un plus grand échantillon pour obtenir des probabilités plus précises.

4.4. Masque jetable (20 points)

1. Utilisez le langage de programmation de votre choix pour créer un programme qui génère en binaire un masque jetable, avec une probabilité égale d'avoir un 0 ou un 1. Python est suggéré. Vous devez donner le code de ce programme et expliquer son fonctionnement. Vous devez utiliser le module random de base du langage choisi, et trouver une alternative plus sécuritaire pour générer des nombres aléatoires pour le second. Expliquez aussi en quoi cette alternative est plus sécuritaire. /12

```
import random
import secrets

def generate_binary_mask(length):
    mask = ''.join(random.choice('01') for _ in range(length))
    return mask

def generate_secure_binary_mask(length):
    mask = ''.join(secrets.choice('01') for _ in range(length))
    return mask

if __name__ == "__main__":
    length = int(input("Entrez la longueur du masque binaire : "))
    method = input("Choisissez la méthode (1 pour normale, 2 pour sécurisée) : ")

    if method == '1':
        binary_mask = generate_binary_mask(length)
    elif method == '2':
        binary_mask = generate_secure_binary_mask(length)
    else:
        print("Méthode non valide. Utilisation de la méthode normale par défaut.")
        binary_mask = generate_binary_mask(length)

    print("Masque binaire généré :", binary_mask)
```

Les fonctions `generate_binary_mask` et `generate_secure_binary_mask` génèrent un masque binaire de longueur spécifiée en utilisant des approches différentes pour produire des valeurs aléatoires. Dans `generate_binary_mask`, la fonction `random.choice('01')` sélectionne aléatoirement un caractère parmi '0' et '1' pour chaque position dans la chaîne, ce qui peut être prévisible et moins sécurisé. En revanche, `generate_secure_binary_mask` utilise `secrets.choice('01')`, qui repose sur une source de randomisation cryptographiquement sécurisée, offrant une meilleure protection contre les prévisions ou les attaques de prédiction, en produisant des résultats plus imprévisibles. Les deux fonctions construisent le masque binaire en répétant cette sélection pour chaque caractère de la chaîne demandée.

2. Utilisez votre programme pour générer des clefs ayant autant de bit que nécessaire (normalement 8 par caractère) pour couvrir les 100 premiers caractères du texte fourni en 4.3.1 après avoir enlevé les espaces. D'abord avec le module random de base, puis avec l'alternative plus sécuritaire.

```
(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ echo 2244082-2088099 `date`
2244082-2088099 Tue Sep 10 07:26:58 PM EDT 2024

(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ python maskgenerator
Entrez la longueur du masque binaire : 100
Choisissez la méthode (1 pour normale, 2 pour sécurisée) : 1
Masque binaire généré : 0110000010010010110001010011110110100001011000100011100110100101000011010011000010011000110010111
```

Figure 19 : Clefs générées avec le module random

```
(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ echo 2244082-2088099 `date`
2244082-2088099 Tue Sep 10 07:27:50 PM EDT 2024

(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ python maskgenerator
Entrez la longueur du masque binaire : 100
Choisissez la méthode (1 pour normale, 2 pour sécurisée) : 2
Masque binaire généré : 010101001001011010000000100101010011101001000001110100100000001101001010011010110101101010111
```

Figure 20 : Clefs générées avec le module secrets

3. Utilisez l'utilitaire masque pour appliquer vos clés sur le texte et calculer l'entropie par bit (h-bit) avant et après l'application de chaque masque. Faites de même avec l'entropie par octet (h-ascii). /3

```
(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ echo 2244082-2088099 `date`
2244082-2088099 Tue Sep 10 07:37:35 PM EDT 2024

(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./masque binary_mask_random.txt 100 omar_sansespace.txt random_mask_applied.txt

(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./masque binary_mask_secrets.txt 100 omar_sansespace.txt secret_mask_applied.txt
```

Figure 21 : Clés générées avec random et avec secrets appliquées sur le texte

```

└─(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ echo 2244082-2088099 `date`
2244082-2088099 Tue Sep 10 07:39:18 PM EDT 2024

└─(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ xxd -b omar_sansespace.txt
00000000: 01000100 01010010 01000001 01010111 01000001 01000010 01000000 DRAWAB
00000006: 01001100 01000101 01010101 01001110 01000111 01000001 LEUNGA
0000000c: 01010010 01001101 01000101 01001110 01010100 01000101 RMENTE
00000012: 01000100 01001111 01010011 01010011 01001111 01000010 DOSSOB
00000018: 01010101 01000011 01001111 01010011 01001011 01001110 UCOSKIN
0000001e: 01001111 01010111 01000100 01000101 01000100 01000111 OWLEDG
00000024: 01000101 01000001 01000010 01000100 01000101 01010001 EABLEQ
0000002a: 01010101 01001001 01000011 01001011 01001001 01001001 UICKSI
00000030: 01001100 01010110 01000101 01010010 01001001 01001110 LVERIN
00000036: 01000111 01010000 01010010 01000101 01000011 01001001 GPRECI
0000003c: 01010000 01001001 01010100 01000001 01010100 01001001 PITATTI
00000042: 01001111 01001110 01010011 01000110 01000001 01010110 ONSFAV
00000048: 01001111 00001010 01010101 01010010 01000101 01000100 OURED
0000004e: 01001110 01000101 01010011 01010011 01000001 01000100 NESSAD
00000054: 01001101 01001111 01001110 01001001 01010100 01001111 MONITO
0000005a: 01010010 01011001 01000110 01000001 01010101 01001100 RYFAUL
00000060: 01010100 01001100 01000101 01010011 01010011 01001100 TLESSL
00000066: 01011001 01010001 01010101 01000001 01010100 01001111 YQUATO
0000006c: 01010010 01011010 01000001 01001001 01001110 01001101 RZAINM
00000072: 01000001 01001100 01001001 01010011 01001111 01001110 ALISON
00000078: 01010011 01000010 01000101 01001110 01000100 01011001 SBENDY
0000007e: 01010101 01001101 01000010 01001001 01001100 01001001 UMBILI
00000084: 01000011 01000001 01010100 01000101 01000100 01000011 CATEDC
0000008a: 01001111 01001110 01001010 01010101 01001110 01010100 ONJUNT
00000090: 01001111 01010011 00001010 01010101 01011010 01000101 OSUZE

```

Figure 22 : Le fichier de départ contenant les 100 caractères

```

└─(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ echo 2244082-2088099 `date`
2244082-2088099 Tue Sep 10 07:40:14 PM EDT 2024

└─(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ xxd -b random_mask_applied.txt
00000000: 01110100 01100011 01110000 01100111 01110001 01110010 tcpqqr
00000006: 01111100 01110101 01100100 01111110 01110111 01110000 |ud-wp
0000000c: 01100010 01111101 01110100 01111110 01100101 01110100 b}t-et
00000012: 01110100 01111111 01100011 01100010 01111111 01110011 t.cbs
00000018: 01100101 01110011 01111110 01100010 01111010 01111111 es-bz
0000001e: 01111110 01100111 01111101 01110100 01110100 01110110 ~g}ttv
00000024: 01110101 01110001 01110010 01111100 01110101 01100000 uqr{u` 
0000002a: 01100101 01111000 01110010 01111011 01100011 01111001 exr{cy
00000030: 01111101 01100110 01110101 01100010 01111000 01111111 }fubx.
00000036: 01110110 01100000 01100010 01110100 01110010 01111001 v`btry
0000003c: 01100001 01111001 01100100 01110000 01100100 01111000 aydpdx
00000042: 01111111 01111110 01100011 01110110 01110000 01100111 .~cvpg
00000048: 01111111 00111011 01100101 01100010 01110100 01110101 .;ebtu
0000004e: 01111110 01110101 01100011 01100011 01110000 01110100 ~uccpt
00000054: 01111101 01111110 01111111 01111001 01100100 01111111 }~.yd.
0000005a: 01100011 01101000 01110110 01110001 01100101 01111101 chvqe}
00000060: 01100100 01111101 01110100 01100010 d}tb

```

Figure 23 : Le fichier du texte après avoir appliqué le masque généré avec random

```

└─(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ echo 2244082-2088099 `date`
2244082-2088099 Tue Sep 10 07:41:21 PM EDT 2024

└─(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ xxd -b secret_mask_applied.txt
00000000: 01110100 01100011 01100001 01100110 01110001 01110011 tcqfq$ Entropie - Chiffrement
00000006: 01111100 01110101 01100100 01111110 01110111 01100000 |ud~wp
0000000c: 01100010 01111100 01110100 01111111 01100100 01110100 b|t.dt
00000012: 01110100 01111111 01100011 01100011 01111111 01100100 t.ccr
00000018: 01100101 01110011 01111110 01100011 01110111 01111111 es~c{. Entropie - Chiffrement
0000001e: 01111111 01100110 01111100 01110100 01110100 01110110 .flttv
00000024: 01110101 01110001 01110011 01111101 01110100 01100001 uqs}ta
0000002a: 01100100 01111001 01110011 01111010 01100011 01111001 dyszcy
00000030: 01111100 01100110 01110101 01100011 01111000 01111111 |fucx.
00000036: 01110110 01100000 01100011 01110101 01110011 01111000 v`cusx
0000003c: 01100000 01111001 01100100 01110001 01100100 01111001 `ydqdy
00000042: 01111111 01111110 01100010 01110111 01110001 01100111 .~bwqg
00000048: 01111111 00111010 01100100 01100010 01110100 01110100 .:dbtt
0000004e: 01111110 01110100 01100010 01100011 01110000 01110100 ~tbcpt
00000054: 011111100 01111111 01111111 01111001 01100101 01111110 | ..ye~
0000005a: 01100010 01101000 01110110 01110000 01100101 01111100 bhvpel
00000060: 01100101 01111100 01110100 01100010 e|tb

```

Figure 24 : Le fichier du texte après avoir appliqué le masque généré avec secrets

```

└─(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./h-bit < omar_sansespace.txt
0 = 13933
1 = 9507
Nombre total de bits : 23440
Entropie du texte entre : 0.974126

└─(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./h-bit < random_mask_applied.txt
0 = 346
1 = 454
Nombre total de bits : 800
Entropie du texte entre : 0.986813

└─(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./h-bit < secret_mask_applied.txt
0 = 347
1 = 453
Nombre total de bits : 800
Entropie du texte entre : 0.987299

```

Figure 25 : Calcul de l'entropie par bit

```
(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ echo 2244082-2088099 `date`
2244082-2088099 Tue Sep 10 07:43:43 PM EDT 2024

(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./h-ascii < omar_sansespace.txt
Nombre total d'octets : 2930
Entropie de l'entree : 4.248237

(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./h-ascii < random_mask_applied.txt
Nombre total d'octets : 100
Entropie de l'entree : 4.380988

(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./h-ascii < secret_mask_applied.txt
Nombre total d'octets : 100
Entropie de l'entree : 4.307725
```

Figure 26 : Calcul de l'entropie par octet

4. Commentez les résultats. Dites à quoi vous vous attendiez, si les résultats sont conformes à vos attentes ou non, et pourquoi. Au besoin, mentionnez des formes de biais possibles. Votre justification est évaluée ici et non le résultat en soi. /5

Nous nous attendions à ce que les entropies soient très proches, mais nous avons supposé que l'entropie ne varierait pas systématiquement en fonction de la méthode utilisée. En observant les résultats, nous avons noté que l'entropie des caractères, après application du masque aléatoire, est inférieure à l'entropie de base. En revanche, l'entropie des caractères après application du masque secret reste équivalente à l'entropie de base. Pour ce qui est de l'entropie par octet, la situation est inversée : l'entropie de base est la plus basse, tandis que l'entropie des caractères après application du masque aléatoire est plus élevée que celle obtenue avec le masque secret. Ces observations confirment nos attentes, montrant qu'il n'y a pas de lien clair entre l'entropie et la méthode de génération du masque utilisé.

4.5. Communication à clé publique, HTTPS et SSL (5 points)

1. Expliquez la différence entre HTTP et HTTPS dans vos mots. /0,5

HTTP (HyperText Transfer Protocol) et HTTPS (HyperText Transfer Protocol Secure) sont des protocoles qui permettent de transférer des données entre un navigateur web et un serveur. La principale différence entre les deux réside dans la sécurité des données. Avec HTTP, les données sont envoyées en clair, sans chiffrement, ce qui les rend vulnérables aux interceptions par des tiers, y compris pour des informations sensibles comme les mots de passe. En revanche, HTTPS utilise SSL (Secure Sockets Layer) ou TLS (Transport Layer Security) pour chiffrer les données échangées, ce qui protège les informations contre l'interception et garantit leur confidentialité.

2. Expliquez pourquoi il est impossible de se connecter au dossier étudiant si on spécifie le protocole http (<http://dossieretudiant.polymtl.ca>) et dites quelle solution sécuritaire pourrait être mise en place pour que quelqu'un qui consulte ce lien puisse accéder au dossier étudiant. /1

On ne peut pas se connecter au dossier étudiant via le protocole HTTP car ce protocole ne fournit pas de sécurité pour les données échangées, ça risquerait de rendre les informations sensibles comme les noms d'utilisateur et les mots de passe vulnérables à des interceptions malveillantes. Pour garantir une connexion sécurisée, il faut utiliser HTTPS, qui chiffre les données et protège ainsi les informations échangées.

3. Quelle est l'utilité du header « Strict-Transport-Security » ? Vous pouvez constater sa présence en ouvrant l'inspecteur de votre navigateur sous « network » en visitant par exemple le site de la banque TD (<https://www.td.com/>) /1

Le header « Strict-Transport-Security » informe le navigateur qu'il doit toujours utiliser « HTTPS » pour la connexion. Même si un utilisateur essaie de se connecter en supprimant le « s » pour utiliser « HTTP », la connexion sera quand même établie en « HTTPS ». C'est une mesure de sécurité essentielle qui protège les utilisateurs contre diverses attaques malveillantes et assure que les informations échangées entre le serveur et le navigateur sont sécurisées et chiffrées.

4. À quoi sert un certificat à clé publique ? Comment votre navigateur vérifie-t-il l'identité du propriétaire du site que vous avez visité ? /0,5

Un certificat à clé publique est un fichier qui assure une connexion sécurisée entre une personne et une paire de clés asymétriques, c'est-à-dire entre un navigateur et un serveur web. Il fournit aux navigateurs des informations sur les sites qu'ils visitent, ce qui leur permet de vérifier leur authenticité. En examinant le certificat, le navigateur confirme l'identité du propriétaire du site visité et s'assure qu'il est valide, c'est-à-dire qu'il a été émis par une autorité de confiance. Grâce au certificat, on peut connaître l'identité du propriétaire du site en regardant sa clé publique, son domaine et l'autorité de confiance qui a délivré le certificat.

[https://www.infocles.justice.gouv.qc.ca/?nav=rubrique\[@nom=%27public%27\]/rubrique\[@nom=%27pri](https://www.infocles.justice.gouv.qc.ca/?nav=rubrique[@nom=%27public%27]/rubrique[@nom=%27pri)

5. Utilisez *openssl* pour générer un certificat « Self-Signed ». Donnez dans un tableau les champs de votre certificat. /0,5

Figure 27 : Génération de la clé

```
[kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
$ echo 2244082-2088099 `date`
2244082-2088099 Tue Sep 10 07:49:16 PM EDT 2024

[kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
$ openssl req -new -x509 -key key.pem -out cert.pem -days 365
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:CA
State or Province Name (full name) [Some-State]:Quebec
Locality Name (eg, city) []:Montreal
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Polymtl
Organizational Unit Name (eg, section) []:Informatique
Common Name (e.g. server FQDN or YOUR name) []:Omar
Email Address []:omar.benzekri.2003@gmail.com
```

Figure 28: Génération d'un certificat « Self-Signed »

```

(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
$ echo 2244082-2088099 "date"
2244082-2088099 Tue Sep 10 07:51:33 PM EDT 2024

(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
$ openssl x509 -in cert.pem -text
-----BEGIN CERTIFICATE-----
```

-----BEGIN CERTIFICATE-----

Certificate:

Data:

- Version: 3 (0x2)
- Serial Number:
01:c2:38:8a:3f:51:95:db:76:7a:04:4b:f7:49:28:0a:04:fe:8e:81
- Signature Algorithm: sha256WithRSAEncryption
- Issuer: C=CA, ST=Quebec, L=Montreal, O=Polymtl, OU=Informatique, CN=Omar, emailAddress=omar.benzekri.2003@gm ail.com
- Validity
 - Not Before: Sep 10 23:50:53 2024 GMT
 - Not After : Sep 10 23:50:53 2025 GMT
- Subject: C=CA, ST=Quebec, L=Montreal, O=Polymtl, OU=Informatique, CN=Omar, emailAddress=omar.benzekri.2003@g mail.com
- Subject Public Key Info:
 - Public Key Algorithm: rsaEncryption
 - Public-Key: (2048 bit)
 - Modulus:


```
00:ef:34:64:16:05:b1:1d:2f:25:a1:fd:f4:56:16:
          58:ff:9a:d4:66:11:10:53:d1:72:13:f7:44:4b:55:
          71:5c:ad:2d:35:b0:e3:b0:8e:04:10:61:ce:65:dc:
          42:4c:a0:21:fc:2a:96:14:42:c9:2a:7a:32:12:38:
          7c:7c:71:42:93:87:48:32:94:43:00:77:35:2b:78:
          d8:8a:02:96:30:3a:86:59:e9:54:cb:48:67:27:4d:
          c3:23:4e:d3:27:51:b8:6b:78:0c:ef:bb:52:84:f0:
          4e:5b:8c:6d:d0:f3:a0:74:5b:76:00:9a:46:60:ec:
          8b:77:9b:93:2a:f9:6e:44:75:bf:e0:8f:9f:89:a5:
          81:08:fb:4e:c0:89:1d:49:ff:84:e0:7f:f6:f6:3d:
          76:2a:d0:0b:d0:12:15:62:fb:93:38:44:78:70:a3:
          00:d1:4a:33:86:d2:08:ce:eb:7f:10:78:bf:75:
          c7:dd:0d:39:7c:30:36:d3:74:fd:e7:13:ae:7f:1a:
          7d:28:23:99:e1:69:d9:94:a7:87:b0:50:c4:6f:f9:
          9b:58:1f:6b:3d:3a:93:9d:83:5e:40:ce:0e:fd:10:
          26:3b:e0:d0:d1:e0:2b:a7:43:a7:38:0f:fb:17:1a:
          65:8c:8c:41:60:55:59:bf:3b:aa:1c:ff:6a:a9:a4:
          a4:27
```
 - Exponent: 65537 (0x10001)
 - X509v3 extensions:
 - X509v3 Subject Key Identifier:
83:1B:B4:9B:63:54:07:6D:70:50:AD:A7:1F:05:5B:27:8C:73:C1:8F
 - X509v3 Authority Key Identifier:
83:1B:B4:9B:63:54:07:6D:70:50:AD:A7:1F:05:5B:27:8C:73:C1:8F
 - X509v3 Basic Constraints: critical
CA:TRUE
 - Signature Algorithm: sha256WithRSAEncryption
 - Signature Value:
4e:7c:5f:4e:0b:34:2e:c3:7d:69:44:b1:71:25:7f:7f:69:a3:
c9:14:7f:71:e4:68:45:de:7d:7a:66:10:b1:93:ab:cb:43:11:

Figure 29 : Champs du certificat (partie 1)

af:84:e8:31:bd:17:cf:c6:45:d2:86:4d:0b:45:b5:18:70:af:
ec:18:b6:99:2c:90:cf:42:2e:18:93:c3:4a:50:e2:6f:2f:15:h-bit
59:00:a6:ed:f1:fc:44:c6:40:d4:8d:ab:74:07:04:26:9a:cb:
65:d3:e1:47:04:86:ec:79:cb:a3:c1:26:fc:5f:13:90:bf:e9:
71:fa:05:c1:c6:40:c4:42:6c:43:25:3d:d1:e4:d8:77:e5:f0:
17:c0:49:a3:b7:13:5e:e7:f5:2c:7e:dc:ae:7f:6b:eb:9f:d7:
bb:ee:3b:a0:b1:3e:46:4c:42:90:b1:64:5f:28:d0:5f:11:08:
c4:04:b6:67:5d:38:76:f7:47:4b:0f:96:5c:39:df:59:f4:e3:
12:5a:8d:1f:47:49:18:75:0a:de:f8:5b:84:7e:20:03:c0:1d:
8f:be:b0:e9:b5:00:d1:5b:72:b0:a8:19:64:11:af:78:55:a8:
6a:51:bd:27:aa:98:1b:11:f5:26:92:a1:26:ee:f1:6a:ec:a1:
26:e4:b7:8c:e5:c1:24:20:da:ea:b1:92:0e:28:ac:4c:c4:f2:
1d:51:e5:67

-----BEGIN CERTIFICATE-----

MIIEDzCCAvegAwIBAgIUAciI4ij9Rldt2egRL90koCgT+joEwDQYJKoZIhvNAQEL
BQAwgZYxCzAJBgNVBAYTAkNBMQ8wDQYDVQQIDAZRdWVizWMxETAPBgNVBAcMCE1v
bnRyZWFsMRAwDgYDVQQKDAdb2x5bXRsMRUwEwYDVQQLDAxJbmZvcmlhdGlxWUx
DTALBgNVBAMMBE9tYXIxKzApBgkqhkiG9w0BCQEWHG9tYXIuYmVuemVrcmkumjAw
M0BnbWFpbC5jb20wHhcNMjQwOTEwMjM1MDUzWhcNMjuwOTEwMjM1MDUzWjCBljEL
MAkGA1UEBhMCQ0ExDzANBgNVBAgMBLF1ZWJlYzERMA8GA1UEBwwITW9udHJlYwx
EDAObgNVBAoMB1BvbHltdGwxFTATBgNVBAsMDEluZm9ybWF0aXF1ZTENMAgA1UE
AwwET21hcjErMCkGSqGSIB3DQEJARYcb21hci5iZW56ZWtyaS4yMDAzQGdtYWls
LmNvbTCCASiwDQYJKoZIhvNAQEBBQADggEPADCCAQoCggEBA080ZBYFsR0vJaH9
9FYWWP+a1GYREFPRchP3REtVcVytLTWw47COBBBhzmXcQkygIfwqlhRCySp6MhI4
fhxxQpOHSDKUQwB3NSt42IoCljA6hlnpVMtIZydNwyNO0ydRuGt4D0+7UoTwTluM
bdDzoHRbdgCaRmDs3ebkyr5bkR1v+CPn4mlgQj7TsCJHUn/hOB/9vY9dirQC9AS
FWL7kzhEeHCjANFKM4bSCM7rfxB4eL91x90NOXwwNtN0/ecTTn8afSgjmeFp2ZSn
h7BQxG/5m1gfaz06k52DXkDODv0QJjvg0NHgK6dDpzgP+xcaZYyMQWBVWb87qhz/
aqmkpCcCAwEAAsNTMFEwHQYDVR00BBYEFIMbtJtjVAdtcFCtx8FWyeMc8GPMB8G
A1UdIwQYMBaAFIMbtJtjVAdtcFCtx8FWyeMc8GPMA8GA1UdEwEB/wQFMAMBaf8w
DQYJKoZIhvNAQELBQADggEBAE58X04LNc7DfWLEsXElf39po8kUf3HkaEXefXpm
ELGTq8tDEa+E6DG9F8/GRdKGtQtFtRhwr+wYtpkskM9CLhiTw0pQ4m8vFVkApu3x
/ETGQNSNq3QHBCaay2XT4UcEhux5y6PBjvxFe5C/6XH6BcHGQMRCbEMlPdHk2Hfl
8BfASaO3E17n9Sx+3K5/a+uf17vu06CxPkZMqpCxZF8o0F8RCMQUEtmddOHb3R0sP
1lw531n04xJaJr9HSRh1Ct74W4R+IAPAHY++sOm1ANFbcrCoGWQRr3hVqqGpRvSeq
mBsR9SaSoSbu8WrsoSbkt4zlwSQg2uqxkg4orEzE8h1R5Wc=

-----END CERTIFICATE-----

Figure 30 : Champs du certificat (partie 2)

Signature Algorithm	sha256WithRSAEncryption
Signature Value	4e:7c:5f:4e:0b:34:2e:c3:7d:69:44:b1:71:25:7f:7f:69:a3:c9:14:7f:71:e4:68:45:de:7d:7a:66:10:b1:93:ab:cb:43:11:af:84:e8:31:bd:17:cf:c6:45:d2:86:4d:0b:45:b5:18:70:af:ec:18:b6:99:2c:90:cf:42:2e:18:93:c3:4a:50:e2:6f:2f:15:59:00:a6:ed:f1:fc:44:c6:40:d4:8d:ab:74:07:04:26:9a:cb:65:d3:e1:47:04:86:ec:79:cb:a3:c1:26:fc:5f:13:90:bf:e9:71:fa:05:c1:c6:40:c4:42:6c:43:25:3d:d1:e4:d8:77:e5:f0:17:c0:49:a3:b7:13:5e:e7:f5:2c:7e:dc:ae:7f:6b:eb:9f:d7:bb:ee:3b:a0:b1:3e:46:4c:42:90:b1:64:5f:28:d0:5f:11:08:c4:04:b6:67:5d:38:76:f7:47:4b:0f:96:5c:39:df:59:f4:e3:12:5a:8d:1f:47:49:18:75:0a:de:f8:5b:84:7e:20:03:c0:1d:8f:be:b0:e9:b5:00:d1:5b:72:b0:a8:19:64:11:af:78:55:a8:6a:51:bd:27:aa:98:1b:11:f5:26:92:a1:26:ee:f1:6a:ec:a1:26:e4:b7:8c:e5:c1:24:20:da:ea:b1:92:0e:28:ac:4c:c4:f2:1d:51:e5:67
Version	3 (0x2)
Serial Number	01:c2:38:8a:3f:51:95:db:76:7a:04:4b:f7:49:28:0a:04:fe:8e:81
Issuer	C=CA, ST=Quebec, L=Montreal, O=Polymtl, OU=Informatique, CN=Omar, emailAddress=omar.benzekri.2003@gmail.com
Validity	Not Before: Sep 10 23:50:53 2024 GMT Not After : Sep 10 23:50:53 2025 GMT
Subject	C=CA, ST=Quebec, L=Montreal, O=Polymtl, OU=Informatique, CN=Omar, emailAddress=omar.benzekri.2003@gmail.com
Public Key Algorithm	rsaEncryption
Public-Key	(2048 bit)
Modulus	00:ef:34:64:16:05:b1:1d:2f:25:a1:fd:f4:56:16:58:ff:9a:d4:66:11:10:53:d1:72:13:f7:44:4b:55:71:5c:ad:2d:35:b0:e3:b0:8e:04:10:61:ce:65:dc:42:4c:a0:21:fc:2a:96:14:42:c9:2a:7a:32:12:38:7c:7c:71:42:93:87:48:32:94:43:00:77:35:2b:78:d8:8a:02:96:30:3a:86:59:e9:54:cb:48:67:27:4d:c3:23:4e:d3:27:51:b8:6b:78:0c:ef:bb:52:84:f0:4e:5b:8c:6d:d0:f3:a0:74:5b:76:00:9a:46:60:ec:8b:77:9b:93:2a:f9:6e:44:75:bf:e0:8f:9f:89:a5:81:08:fb:4e:c0:89:1d:49:ff:84:e0:7f:f6:f6:3d:76:2a:d0:0b:d0:12:15:62:fb:93:38:44:78:70:a3:00:d1:4a:33:86:d2:08:ce:eb:7f:10:78:78:bf:75:

	c7:dd:0d:39:7c:30:36:d3:74:fd:e7:13:4e:7f:1a: 7d:28:23:99:e1:69:d9:94:a7:87:b0:50:c4:6f:f9: 9b:58:1f:6b:3d:3a:93:9d:83:5e:40:ce:0e:fd:10: 26:3b:e0:d0:d1:e0:2b:a7:43:a7:38:0f:fb:17:1a: 65:8c:8c:41:60:55:59:bf:3b:aa:1c:ff:6a:a9:a4: a4:27
Exponent	65537 (0x10001)
X509v3 Subject Key Identifier	83:1B:B4:9B:63:54:07:6D:70:50:AD:A7:1F:05:5B:27:8C:73:C1:8F
X509v3 Authority Key Identifier	83:1B:B4:9B:63:54:07:6D:70:50:AD:A7:1F:05:5B:27:8C:73:C1:8F
X509v3 Basic Constraints	critical CA:TRUE

6. Lancez un serveur python local qui utilise votre certificat avec le script python au lien suivant (Vous devez fournir la clef et le certificat en format .pem.) : <https://gist.github.com/SeanPesce/af5f6b7665305b4c45941634ff725b7a>

```
(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
└─$ echo 2244082-2088099 `date`
2244082-2088099 Tue Sep 10 08:18:29 PM EDT 2024

(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
└─$ sudo openssl s_server -cert cert.pem -key key.pem -accept 3000
Using default temp DH parameters
ACCEPT
```

Figure 31: Lancement d'un serveur python local

7. Tentez d'accéder au serveur avec le navigateur Firefox. Quel problème rencontrez-vous et comment pourriez-vous le régler ? Expliquez votre démarche et votre raisonnement./1,5

Le problème vient du certificat auto-signé détecté comme une menace par le navigateur, qui réinitialise la connexion pour passer à HTTPS. Le problème, c'est que même après avoir changé pour le protocole HTTPS, le navigateur nous informe toujours qu'il y a des risques probables de sécurité. Pour résoudre ce problème, nous avons ajouté le certificat aux autorités de certification. Maintenant que le certificat est autorisé, le problème d'authentification devrait être résolu, bien que cela ne garantisse pas une sécurité totale de la connexion, même si le message de risque disparaît.

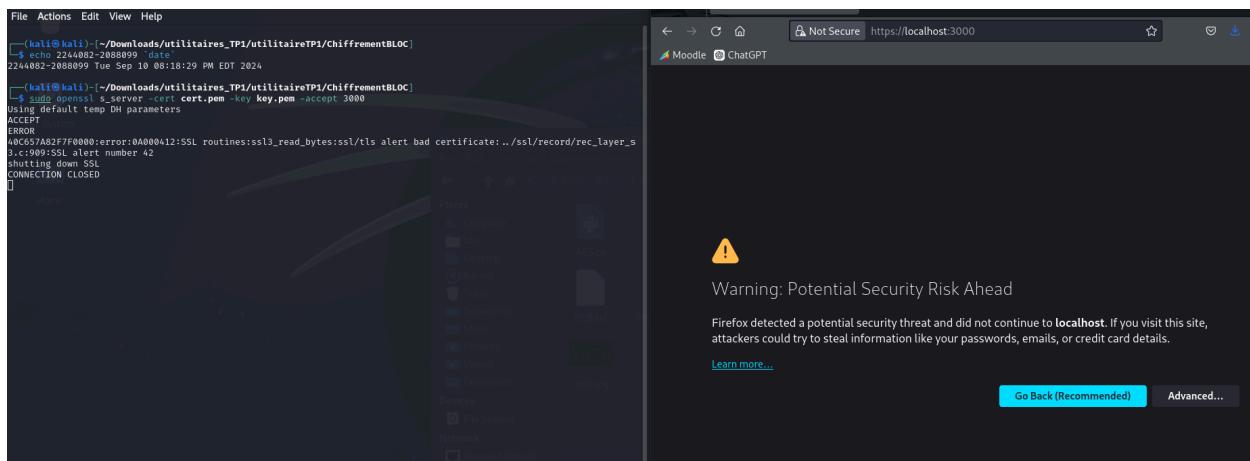


Figure 32 : Message du risque de sécurité

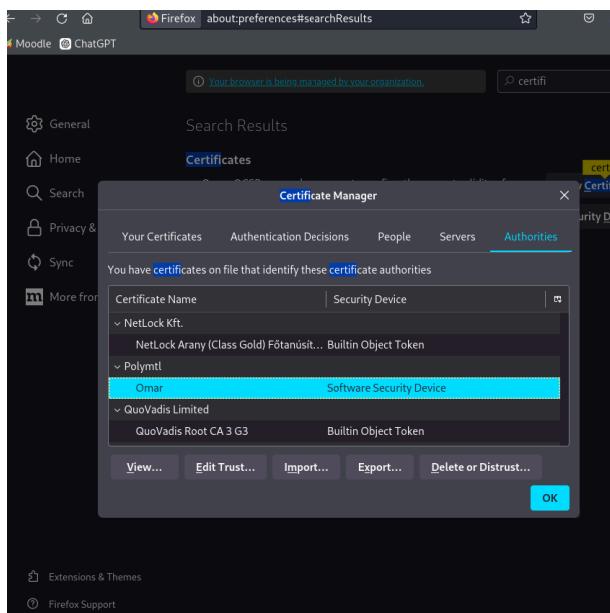


Figure 33 : Sélection de notre certificat

4.6. Codage (9 points)

1. Expliciter les alphabets σ , τ et τ' qui sont respectivement les alphabets pour la sortie de la source, du codeur et du bloc de chiffrement. /3

$$\sigma = \{ 0, 1, 2, \dots, 7, 8, 9 \} \cup \{ A, B, C, \dots, X, Y, Z \}$$

$$\tau = \{ 0, 1 \}$$

$$\tau' = \{ 0, 1 \}$$

2. Identifier les langages provenant de ces alphabets. /3

Langage $L\sigma$:

$$L\sigma = \{ \text{NIP de 4 caractères} \mid \text{chaque caractère} \in \sigma \}$$

Consiste d'un langage en tous les NIP originaux de 4 caractères, où chaque caractère appartient à l'alphabet σ .

Langage $L\tau$:

$$L\tau = \{ \text{Séquence de 64 bits} \mid \text{NIP encodé en binaire formé par la répétition de deux fois un NIP après sa conversion en ASCII à partir de } L\sigma \}$$

Constitué de NIP encodés en binaire sur 64 bits, formés par la duplication de deux fois le même NIP après conversion en ASCII à partir du langage $L\sigma$.

Langage $L\tau'$:

$$L\tau' = \{ \text{Séquence de 64 bits} \mid \text{séquence obtenue par le chiffrement Triple DES d'une séquence de } L\tau \}$$

Regroupe des séquences de 64 bits résultant du chiffrement Triple DES d'une séquence appartenant à $L\tau$.

3. Ensuite, identifiez les attaques auxquelles le système est vulnérable. Pour identifier ces attaques, rappelez-vous qu'un attaquant peut connaître parfaitement le fonctionnement des boîtes de codage et chiffrement mais qu'il n'a bien sûr pas accès à la clé. Aussi, un attaquant peut intercepter tous les messages chiffrés et même les modifier. /3

Plusieurs vecteurs d'attaque peuvent menacer un système utilisant Triple DES avec un NIP. Une attaque par force brute pourrait être envisagée malgré la solidité du Triple DES. L'attaquant interceptera plusieurs messages chiffrés et testerait toutes les combinaisons possibles de NIP à 4 caractères. Étant donné que le NIP est encodé en ASCII et répété deux fois, cela pourrait révéler des motifs, facilitant la réduction de l'espace de recherche. Une autre méthode est l'attaque par relecture (Replay attack), où un attaquant capture un message chiffré contenant un NIP valide et

le retransmet plus tard. L'absence de mécanismes tels qu'un horodatage ou un identifiant unique pour chaque transaction permettrait de réutiliser ce même NIP.

Une attaque par texte clair devient possible lorsqu'un attaquant connaît un NIP précis et intercepte le message chiffré correspondant; il peut alors analyser cette paire pour identifier des motifs ou des faiblesses potentielles dans le chiffrement. Même si Triple DES complique la modification directe de messages chiffrés, un attaquant peut tout de même altérer les données ou modifier significativement le message sans être détecté, exploitant ainsi des vulnérabilités à travers une manipulation des messages. Enfin, une attaque par sniffing permet à l'attaquant de capturer des messages chiffrés transitant sur le réseau. Malgré le chiffrement, la répétition en ASCII du NIP deux fois peut donner des indices exploitables si une partie du message est connue ou devinable.

4.7. Changement de codage (5 points)

1. Pour chacun des trois codages, dites quelles attaques du 4.6.3 ils permettent de bloquer. /3

Pour le premier codage (figure 1 de l'énoncé du TP), l'attaque principalement bloquée par ce codage est l'attaque par force brute. L'élargissement de l'espace des bits et l'utilisation de nombres aléatoires rendent presque impossibles les tentatives de deviner ou de recalculer le NIP par force brute, en raison du nombre extrêmement élevé de combinaisons possibles.

Le codage illustré dans la figure 2 de l'énoncé du TP est conçu pour prévenir surtout les attaques par rejet (replay attacks). Ce codage utilise un timestamp Unix unique pour chaque transaction, garantissant que chaque bloc de données est unique et chronologiquement ordonné. Cela empêche ainsi un attaquant de réutiliser une communication interceptée à un autre moment.

Le codage de la figure 3 (énoncé du TP) renforce davantage la sécurité en intégrant plusieurs mécanismes. L'utilisation d'un timestamp Unix de 32 bits bloque les attaques par rejet en attachant chaque message à un timestamp unique. Les bits de parité associés à chaque NIP (nouveau et ancien) permettent de vérifier l'intégrité du message en détectant toute modification non autorisée. Pour modifier un message, il serait aussi nécessaire de changer le timestamp, compliquant ainsi les tentatives de manipulation des messages. De plus, l'inclusion de l'ancien NIP augmente la difficulté des attaques par force brute, car l'attaquant doit deviner non pas un mais deux NIP. Le timestamp unique contribue aussi à empêcher la réutilisation de messages

interceptés

précédemment.

Nouveau NIP + 2 bits de parité	Nombre aléatoire			
0	15	16		63

FIGURE 1 – Codage 1 : le nouveau NIP est codé en binaire sur 14 bits plus 2 bits de parité (les détails ne sont pas importants). Un nombre aléatoire sur 48 bits est concaténé pour former un bloc de 64 bits.

Nouveau NIP + 2 bits de parité	Nombre aléatoire	Timestamp
0	15	63

FIGURE 2 – Codage 2 : le nouveau NIP est codé en binaire sur 14 bits plus 2 bits de parité. Un nombre aléatoire de 16 bits puis un « timestamp » Unix de 32 bits sont concaténés pour former un bloc de 64 bits.

Nouveau NIP + 2 bits de parité	Ancien NIP + 2 bits de parité	Timestamp
0	15	63

FIGURE 3 – Codage 3 : le nouveau et l’ancien NIP sont codés en binaire sur 14 bits plus 2 bits de parité puis concaténés. Un « timestamp » Unix de 32 bits est concaténé pour former un bloc de 64 bits.

Figure 39 : Figures avec les codages de la question

2. Selon vous, quel est le meilleur codage ? Pourquoi ? /2

Le codage 3 est le plus sécurisé parmi les trois options car il intègre plusieurs mécanismes de défense : un timestamp pour empêcher les attaques par rejet (replay attacks), des bits de parité pour repérer toute modification non autorisée, et l’inclusion de l’ancien NIP, qui complexifie les attaques par force brute. Cette combinaison de mesures offre une protection renforcée contre diverses menaces potentielles.

4.8. Chiffrement par bloc et modes d’opération (4 points)

1. Le fichier *mdp.jpg* est un des mots de passe de l’administrateur enregistré sous forme d’image. À l’aide du script python *AES.py*, chiffrer ce fichier en mode ECB. (Exécutez le script sans argument pour connaître son fonctionnement). Observez le fichier de sortie et commentez. /1

```
[kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
└─$ echo 2233082-2088099 `date`
2233082-2088099 Mon Sep 16 05:46:17 PM EDT 2024

[kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
└─$ python AES.py -i mdp.jpg -m ECB -o mdp_encrypted_ECB.jpg
801570

[kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
└─$ █
```

Figure 40: Commande pour chiffrer l'image en mode ECB

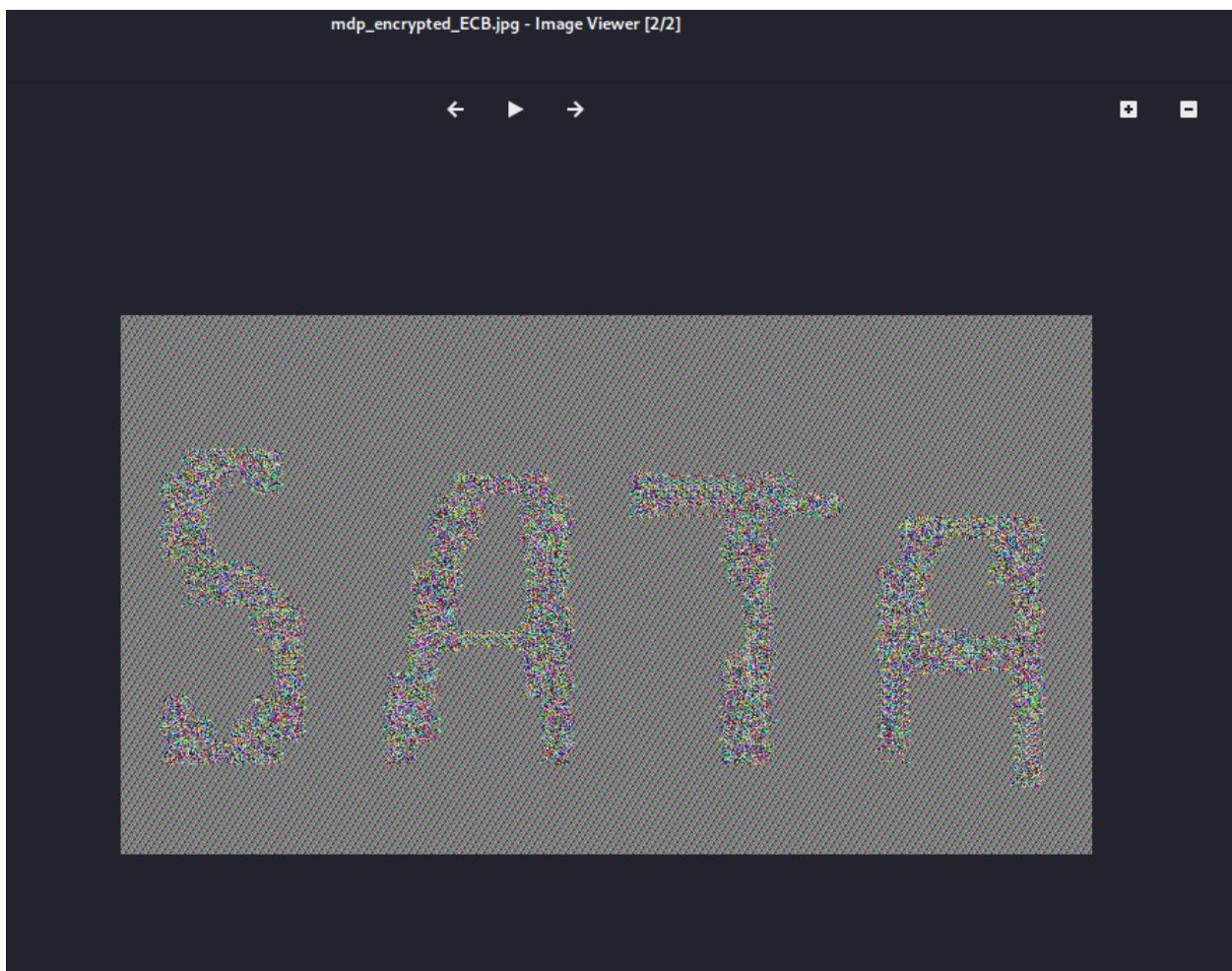


Figure 41: Chiffrement de l'image en mode ECB

Le fichier résultant du chiffrement en mode ECB montre une version légèrement floue de l'image originale. Néanmoins, les caractères « SATA » restent bien visibles. Cela s'explique par le caractère déterministe du chiffrement ECB : pour une clé donnée, les mêmes blocs de texte en clair, chiffrés indépendamment, produisent les mêmes blocs de texte chiffrés. Par conséquent,

des éléments de couleur verte présents dans l'image initiale conservent la même apparence dans l'image chiffrée. Ce mode de chiffrement manque de sécurité, car il n'assure pas une diffusion efficace de l'information. En cas de répétition de données, il devient possible de reconstituer le contenu original à partir des blocs chiffrés.

2. Chiffrez maintenant le fichier en mode CBC. Observez le fichier puis commentez. /1

Même commande mais remplace ECB avec CBC:

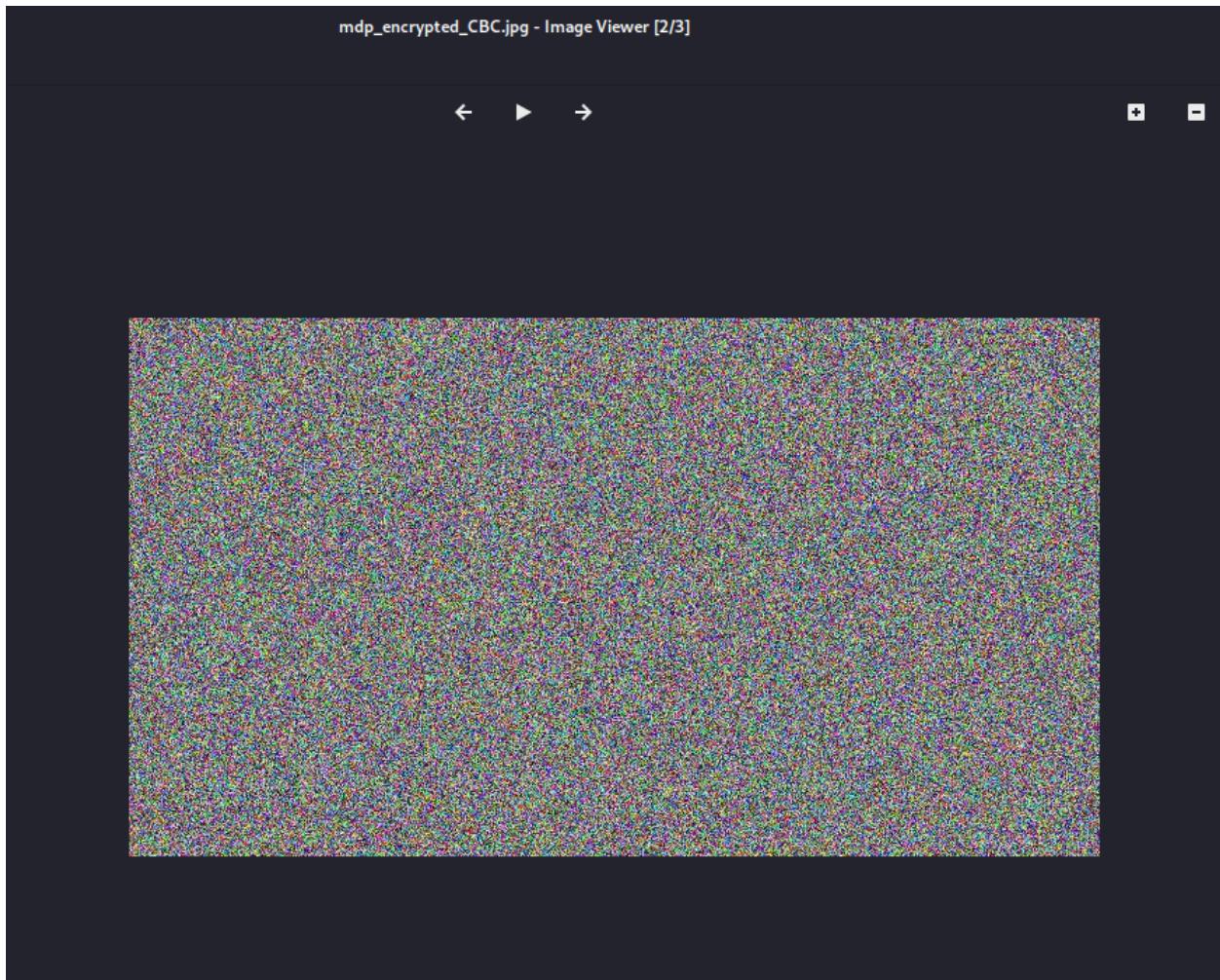


Figure 43: Chiffrement en mode CBC

Après avoir appliqué le chiffrement en mode CBC, le fichier de sortie affiche une image complètement illisible. Il devient impossible de discerner la séquence « SATA » après ce chiffrement. Cela s'explique par le fait que le mode CBC utilise un vecteur d'initialisation qui est appliqué au premier bloc de données. Ensuite, chaque bloc de données est chiffré en utilisant le résultat chiffré du bloc précédent, créant ainsi une dépendance en chaîne entre les blocs chiffrés. Ce processus, connu sous le nom de diffusion, permet de rendre le texte d'origine totalement incompréhensible, offrant une meilleure sécurité.

3. Concluez sur l'importance des modes d'opération des algorithmes de chiffrement par bloc. /2

Il est crucial de bien sélectionner le mode de chiffrement par bloc, car des erreurs de choix peuvent entraîner de graves failles de sécurité. En comparant les fichiers chiffrés obtenus dans les questions 4.8.1 et 4.8.2, on constate que le mode CBC est nettement plus sûr que le mode ECB. En effet, le mode CBC a complètement brouillé l'image, la rendant illisible, tandis que le mode ECB n'a fait que la rendre légèrement floue, permettant encore de distinguer la séquence. Ainsi, il ne suffit pas de disposer d'un algorithme de chiffrement robuste tel que l'AES; il est aussi important de choisir un mode de chiffrement approprié pour garantir une sécurité maximale. Entre le mode ECB et le mode CBC, le mode CBC offre une meilleure protection.

4.9. Organisation des mots de passe en UNIX/Linux (12 points)

1. Examinez le fichier /etc/passwd. Contient-il des mots de passe ? Pourquoi ? Quelles sont ses permissions d'accès ? Pourquoi ? /1

Le fichier `/etc/passwd` ne contient pas de mots de passe. Cela s'explique par le fait que tous les utilisateurs peuvent accéder à ce fichier. Pour des raisons de sécurité, le champ des mots de passe est remplacé par la valeur « x ». Cette lettre « x » signifie que le mot de passe est stocké dans le fichier `/etc/shadow` sous forme de hash. En ce qui concerne les permissions d'accès, les utilisateurs peuvent uniquement lire le fichier, tandis que le propriétaire « root » possède les droits de lecture et d'écriture. Ces restrictions sont mises en place pour des raisons de sécurité et d'intégrité. En effet, une modification non autorisée ou incorrecte de ce fichier sensible pourrait compromettre la sécurité du système, ce qui souligne l'importance de bien configurer les permissions d'accès.

```
[(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]]  
└─$ cat /etc/passwd  
root:x:0:0:root:/root:/usr/bin/zsh  
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin  
bin:x:2:2:bin:/bin:/usr/sbin/nologin  
sys:x:3:3:sys:/dev:/usr/sbin/nologin  
sync:x:4:65534:sync:/bin:/sync  
games:x:5:60:games:/usr/games:/usr/sbin/nologin  
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin  
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin  
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin  
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin  
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin  
proxy:x:13:13:proxy:/bin:/nologin  
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin  
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin  
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin  
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin  
_apt:x:42:65534 ::/nonexistent:/usr/sbin/nologin  
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin  
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin  
systemd-timesync:x:992:992:systemd Time Synchronization:/:/usr/sbin/nologin  
messagebus:x:100:102 ::/nonexistent:/usr/sbin/nologin  
tss:x:101:104:TPM software stack,,,,:/var/lib/tpm:/bin/false  
strongswan:x:102:65534 ::/var/lib/strongswan:/usr/sbin/nologin  
tcpdump:x:103:105 ::/nonexistent:/usr/sbin/nologin  
sshd:x:104:65534 ::/run/sshd:/usr/sbin/nologin  
usbmux:x:105:46:usbmux daemon,,,,:/var/lib/usbmux:/usr/sbin/nologin  
dnsmasq:x:999:65534:dnsmasq:/var/lib/misc:/usr/sbin/nologin  
avahi:x:106:108:Avahi mDNS daemon,,,,:/run/avahi-daemon:/usr/sbin/nologin  
speech-dispatcher:x:107:29:Speech Dispatcher,,,,:/run/speech-dispatcher:/bin/f  
alse  
pulse:x:108:110:PulseAudio daemon,,,,:/run/pulse:/usr/sbin/nologin  
lightdm:x:109:112:Light Display Manager:/var/lib/lightdm:/bin/false  
saned:x:110:114 ::/var/lib/saned:/usr/sbin/nologin  
polkitd:x:991:991:User for polkitd:/:/usr/sbin/nologin  
rtkit:x:111:115:RealtimeKit,,,,:/proc:/usr/sbin/nologin  
colord:x:112:116:colord colour management daemon,,,,:/var/lib/colord:/usr/sbin  
/nologin  
nm-openvpn:x:113:117:NetworkManager OpenVPN,,,,:/var/lib/openvpn/chroot:/usr/s  
bin/nologin  
nm-openconnect:x:114:118:NetworkManager OpenConnect plugin,,,,:/var/lib/Networ  
kManager:/usr/sbin/nologin  
_galera:x:115:65534 ::/nonexistent:/usr/sbin/nologin
```

Figure 44: Contenu dans le fichier /etc/passwd

```
[(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]]  
└─$ ls -l /etc/passwd  
-rw-r--r-- 1 root root 3213 Aug 18 19:26 /etc/passwd  
[(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]]  
└─$ █
```

Figure 45: Permissions d'accès pour le fichier /etc/passwd

2. Observez les fichiers `passwd` et `shadow` qui se trouvent sous le répertoire `/etc/`. Ajoutez un utilisateur avec la commande ci-dessous. Observez ce qui se passe dans les fichiers `passwd` et `shadow`. Lequel ou lesquels de ces deux fichiers sont modifiés ? Pourquoi ? /2

```
$ useradd -g users -s /bin/bash -m NOM_UTILISATEUR
```

Après l'ajout d'un nouvel utilisateur, on observe des modifications dans les deux fichiers. En examinant les figures 46 et 47, on constate l'apparition d'une nouvelle ligne à la fin du fichier `/etc/passwd`, conformément à la structure standard des utilisateurs. De même, en comparant les figures 48 et 49, on remarque qu'une nouvelle ligne a été ajoutée à la fin du fichier `/etc/shadow`. Ce nouvel utilisateur suit le même format que les autres utilisateurs, et le champ du mot de passe est initialisé à la valeur par défaut « `!!` ». Chaque fois qu'un utilisateur est ajouté, le fichier `/etc/passwd` est systématiquement modifié. Dans ce cas précis, le fichier `/etc/shadow` est également mis à jour, car c'est là que sont stockées les informations de mot de passe, et non dans `/etc/passwd`.

```
(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
$ cat /etc/passwd
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-networkd:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesyncd:x:992:992:systemd Time Synchronization:/:/usr/sbin/nologin
messagebus:x:100:102::/nonexistent:/usr/sbin/nologin
tss:x:101:104:TPM software stack,,,,:/var/lib/tpm:/bin/false
strongswan:x:102:65534::/var/lib/strongswan:/usr/sbin/nologin
tcpdump:x:103:105::/nonexistent:/usr/sbin/nologin
sshd:x:104:65534::/run/sshd:/usr/sbin/nologin
usbmux:x:105:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
dnsmasq:x:999:65534:dnsmasq:/var/lib/misc:/usr/sbin/nologin
avahi:x:106:108:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
speech-dispatcher:x:107:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
pulse:x:108:110:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin
lightdm:x:109:112:Light Display Manager:/var/lib/lightdm:/bin/false
saned:x:110:114::/var/lib/saned:/usr/sbin/nologin
polkitd:x:91:991:User for polkitd:/:/usr/sbin/nologin
rtkit:x:111:115:RealtimeKit,,,:/proc:/usr/sbin/nologin
colord:x:112:116:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
nm-openvpn:x:113:117:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
nm-openconnect:x:114:118:NetworkManager OpenConnect plugin,,,:/var/lib/NetworkManager:/usr/sbin/nologin
_galera:x:115:65534::/nonexistent:/usr/sbin/nologin
mysql:x:116:120:MariaDB Server,,,:/nonexistent:/bin/false
stunnel4:x:990:990:stunnel service system account:/var/run/stunnel4:/usr/sbin/nologin
_rpc:x:117:65534::/run/rpcbind:/usr/sbin/nologin
geoclue:x:118:121::/var/lib/geoclue:/usr/sbin/nologin
```

Figure 46: Contenu dans le fichier /etc/passwd avant l'ajout du username Eduardo

```
[kali㉿ kali) [~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
$ sudo useradd -g users -s /bin/bash -m Eduardo
[sudo] password for kali:
Sorry, try again.
[sudo] password for kali:
```

```
[kali㉿ kali) [~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
$ echo 2244082-2088099 `date`
2244082-2088099 Tue Sep 17 09:58:41 AM EDT 2024

[kali㉿ kali) [~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
$ cat /etc/passwd
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:992:992:system Time Synchronization:/:/usr/sbin/nologin
messagebus:x:100:102::/nonexistent:/usr/sbin/nologin
tss:x:101:104:TPM software stack,,,:/var/lib/tpm:/bin/false
strongman:x:102:65534::/var/lib/strongman:/usr/sbin/nologin
tcpdump:x:103:105::/nonexistent:/usr/sbin/nologin
sshd:x:104:65534::/run/sshd:/usr/sbin/nologin
usmux:x:105:40:usmux daemon,,,:/var/lib/usmux:/usr/sbin/nologin
dnsmasq:x:999:65534:dnsmasq:/var/lib/misc:/usr/sbin/nologin
avahi:x:106:108:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
speech-dispatcher:x:107:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
pulse:x:108:110:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin
lightdm:x:109:112:Light Display Manager:/var/lib/lightdm:/bin/false
sane:x:110:114::/var/lib/sane:/usr/sbin/nologin
polkitd:x:991:991:User for polkitd:/:/usr/sbin/nologin
rtkit:x:111:115:RealtimeKit,,,:/proc:/usr/sbin/nologin
colorl:x:112:116:colord colour management daemon,,,:/var/lib/colorl:/usr/sbin/nologin
nm-openvpn:x:113:117:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
nm-openconnect:x:114:118:NetworkManager OpenConnect plugin,,,:/var/lib/NetworkManager:/usr/sbin/nologin
_golera:x:115:65534::/nonexistent:/usr/sbin/nologin
mysql:x:116:120:MariaDB Server,,,:/nonexistent:/bin/false
stunnel4:x:990:990:stunnel service system account:/var/run/stunnel4:/usr/sbin/nologin
_rpc:x:117:65534::/run/rpcbind:/usr/sbin/nologin
geoclue:x:118:121::/var/lib/geoclue:/usr/sbin/nologin
Debian-snmp:x:119:122::/var/lib/snmp:/bin/false
sslh:x:120:123::/nonexistent:/usr/sbin/nologin
ntpserv:x:121:126::/nonexistent:/usr/sbin/nologin
redsocks:x:122:127::/var/run/redsocks:/usr/sbin/nologin
rwho:x:123:65534::/var/spool/rwho:/usr/sbin/nologin
_gophish:x:124:129::/var/lib/gophish:/usr/sbin/nologin
iodine:x:125:65534::/run/iodine:/usr/sbin/nologin
miredo:x:126:65534::/var/run/miredo:/usr/sbin/nologin
statd:x:127:65534::/var/lib/nfs:/usr/sbin/nologin
redis:x:128:130::/var/lib/redis:/usr/sbin/nologin
postgres:x:129:131:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mosquitto:x:130:132::/var/lib/mosquitto:/usr/sbin/nologin
inetutils:x:131:133::/var/lib/inetutils:/usr/sbin/nologin
_gvrx:x:132:134::/var/lib/gvrx:/usr/sbin/nologin
kalix:x:1000:1000::::/home/kalix:/usr/bin/zsh
Eduardo:x:1001:100::/home/Eduardo:/bin/bash
Eduardo2:x:1002:100::/home/Eduardo2:/bin/bash

[kali㉿ kali) [~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
$ grep Eduardo /etc/passwd
Eduardo:x:1001:100::/home/Eduardo:/bin/bash
```

Figure 47: Contenu dans le fichier /etc/passwd après l'addition de l'utilisateur Eduardo

```
(kali㉿ kali) [~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
$ sudo useradd -g users -s /bin/bash -m Eduardo
[sudo] password for kali:
Sorry, try again.
[sudo] password for kali:
```

```
(kali㉿ kali) [~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
$ echo 2244082-2088099 'date'
2244082-2088099 Tue Sep 17 10:05:17 AM EDT 2024
(kali㉿ kali) [~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
$ sudo cat /etc/shadow
root:$y$j9t$8jNMokjZ0g840EqTH5AF.$K5hIrKwvAPdqNzaaxCdHh9XAu2DzCqp/6L/TwX2087:19982:0:99999:7:::
demon:*:19953:0:99999:7:::
bin:*:19953:0:99999:7:::
sync:*:19953:0:99999:7:::
games:*:19953:0:99999:7:::
nan:*:19953:0:99999:7:::
lp:*:19953:0:99999:7:::
mail:*:19953:0:99999:7:::
news:*:19953:0:99999:7:::
uucp:*:19953:0:99999:7:::
proxy:*:19953:0:99999:7:::
www-data:*:19953:0:99999:7:::
backup:*:19953:0:99999:7:::
list:*:19953:0:99999:7:::
irc:*:19953:0:99999:7:::
_apt:*:19953:0:99999:7:::
nobody:*:19953:0:99999:7:::
systemd-network:*:19953:::::
systemd-timesync:*:19953:::::
messagebus:*:19953:::::
tss:*:19953:::::
strongswan:*:19953:::::
tcpdump:*:19953:::::
sshd:*:19953:::::
usmux:*:19953:::::
dnsmasq:*:19953:::::
avahi:*:19953:::::
speech-dispatcher:*:19953:::::
pulse:*:19953:::::
lightdm:*:19953:::::
saned:*:19953:::::
polkitd:*:19953:::::
rtkit:*:19953:::::
colord:*:19953:::::
mm-openvpn:*:19953:::::
mm-openconnect:*:19953:::::
_galeria:*:19953:::::
mysql:*:19953:::::
stunnel4:*:19953:::::
_rpc:*:19953:::::
geoipclue:*:19953:::::
Debian-smb:*:19953:::::
sshd:*:19953:::::
ntpsvc:*:19953:::::
redsocks:*:19953:::::
rwhod:*:19953:::::
_gophish:*:19953:::::
iodine:*:19953:::::
miredo:*:19953:::::
statd:*:19953:::::
redis:*:19953:::::
postgres:*:19953:::::
mosquitto:*:19953:::::
inetutils:*:19953:::::
_8wm:*:19953:::::
kali:$y$j9t$8jNMokjZ0g840EqTH5AF.$uir/RlQWdc0IpvdZUxkj4hA0Tku6KT9Jk6/hL1B0A:19953:0:99999:7:::
Eduardo:$y$j9t$Zjpdig/l45siyub0tlNjN1$ha.Fa/R.b3GpzJxnMBdvVs1AbMcVAnGB896a02HoM3B:19982:0:99999:7:::
```

Figure 48: Contenu du fichier /etc/shadow après l'ajout de l'utilisateur Eduardo

J'ai oublié de prendre un screenshot avant l'ajout de l'utilisateur, ce qui explique pourquoi il n'y est pas(enlever moi pas des points pour cela svp).

3. Donnez un mot de passe à l'utilisateur que vous avez créé avec la commande ci-dessous. Qu'est-ce que vous remarquez dans les fichiers passwd et shadow ? Lequel de ces deux fichiers change ? Pourquoi ? Où se trouve donc l'information du mot de passe ? Quelles sont les permissions du fichier shadow et pourquoi ? /2

Le fichier `/etc/passwd` reste inchangé, tandis que le fichier `/etc/shadow` est modifié. Ce qui est modifié dans ce dernier, c'est la valeur du champ du mot de passe pour l'utilisateur « Eduardo » (voir figure 51). Cela s'explique par le fait que les informations concernant les mots de passe ne sont pas stockées dans `/etc/passwd`, mais bien dans `/etc/shadow`. Concernant les permissions du fichier `/etc/shadow`, seul le superutilisateur « root » peut y accéder en lecture et en écriture. Les autres utilisateurs ne disposent pas des droits nécessaires pour lire ou modifier ce fichier. Ces restrictions sont cruciales car le fichier `/etc/shadow` contient toutes les informations sensibles relatives aux mots de passe. Pour des raisons de sécurité, il est essentiel d'empêcher les utilisateurs d'y accéder ou de le modifier, afin de garantir l'intégrité des données qu'il contient.

```
(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
$ sudo passwd Eduardo
New password:
Retype new password:
passwd: password updated successfully

(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
$ █
```

Figure 49 : Ajout du mot de passe à l'utilisateur « Eduardo »

```
(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
$ echo 2244082-2088099 `date`
2244082-2088099 Tue Sep 17 09:55:05 AM EDT 2024

(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
$ grep Eduardo /etc/passwd
Eduardo:x:1001:100 :: /home/Eduardo:/bin/bash
```

Figure 50: État du fichier passwd de l'utilisateur « Eduardo » après l'ajout du mot de passe

```
(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
$ sudo grep Eduardo /etc/shadow
Eduardo:$y$j9T$oFpo6XM4zoi2JSv8k2Ef30$dVeTpwZxHDvT8NILGCzdYo9tZxjZD/0D7Yz0F9VB9N5:19982:0:99999:7 :::
```

Figure 51: État du fichier shadow de l'utilisateur « Eduardo » après l'ajout du mot de passe

```
(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
$ ls -l /etc/shadow
-rw-r----- 1 root shadow 1703 Sep 16 18:44 /etc/shadow
```

Figure 53 : Permissions du fichier /etc/shadow

4. Changez à nouveau le mot de passe du même utilisateur et donnez-lui le *même* mot de passe. Est-ce que les informations du mot de passe ont changé ? Pourquoi ? /2

Oui, les informations concernant le mot de passe ont bien changé dans le fichier `/etc/shadow`, même si le même mot de passe a été saisi. Cela s'explique par le fonctionnement de la cryptographie des mots de passe. Lorsqu'un mot de passe est entré, une valeur aléatoire appelée « salt » lui est associée. Cette valeur « salt » est utilisée pour générer le hachage du mot de passe. Comme le « salt » utilisé la première fois diffère de celui utilisé la seconde fois, même si le mot de passe est identique, le hachage produit sera différent à chaque fois.

```
(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
$ sudo passwd Eduardo
New password:
Retype new password:
passwd: password updated successfully

(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
$
```

Figure 54 : Ajout du même mot de passe à l'utilisateur « Eduardo »

```
(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
$ echo 2244082-2088099 `date`
2244082-2088099 Tue Sep 17 09:52:10 AM EDT 2024

(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
$ sudo grep Eduardo /etc/shadow
Eduardo:$y$j9T$Zjpdig/l45siyubtlNjN1$ha.Fa/R.b3GpzJxnNBdvVsiAbMcVAmGB896a02HaW3B:19982:0:99999:7:::
```

Figure 55 : État du fichier shadow de l'utilisateur « Eduardo » après l'ajout du même mot de passe

5. Créez un deuxième utilisateur en suivant les mêmes étapes qu'au point b. Éditez ensuite le fichier shadow et remplacez la valeur par défaut (! !) du champ de mot de passe de l'utilisateur que vous venez de créer par la valeur du même champ pour l'utilisateur que vous avez créé en premier (les éditeurs de texte nano et vim sont disponibles). Sauvegardez le fichier et quittez votre session. Essayez de vous connecter sur le compte du deuxième utilisateur mais avec le mot de passe que vous venez de copier. Est-ce que ceci est possible ? Expliquez pourquoi. Quel est le problème ? /2

Oui il est possible, car les deux utilisateurs ont le même hash en ce qui concerne le mot de passe. La connexion est faite, car comme expliqué, le hash est le même! Le problème dans tout ça est que si un utilisateur accède au fichier en mode écriture, il pour finir par modifier le hash et faire en sorte qu'il soit équivalent au sien. Cela lui permettra de se connecter à l'autre compte donc la sécurité est abîmée.

```
(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
└─$ sudo grep Eduardo /etc/shadow
[sudo] password for kali:
Eduardo:$y$j9T$Zjpdig/l45siyub0tlNjN1$ha.Fa/R.b3GpzJxnNBdvVsiAbMcVAmGB896a02HaW3B:19982:0:99999:7:::
Eduardo2:$y$j9T$PSZyZV6.cIWihJRm4/bP0$JnBAnslNPQKRoFyFMUA6kF1z7qCm..JDpa3TBlnPBg9:19982:0:99999:7:::

(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
└─$ echo 2244082-2088099 `date`
2244082-2088099 Tue Sep 17 09:52:10 AM EDT 2024

(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
└─$ █
```

Figure 56 : Ajout de la même valeur du mot de passe à l'utilisateur « Eduardo2 » que celle de l'utilisateur « Eduardo »

```
(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
└─$ su Eduardo2
Password:
(Eduardo2㉿kali)-[/home/kali/Downloads/utilitaires_TP1/utilitaireTP1/ChiffrementBLOC]
└─$ █
```

Figure 57 : Connexion sur le compte « Eduardo2 »

6. Dites comment il est possible de déchiffrer un mot de passe hashé. /3

1. Attaque par force brute : Cette technique consiste à essayer toutes les combinaisons possibles de caractères pour deviner le mot de passe. Le processus continue jusqu'à ce que la combinaison correcte soit trouvée et que l'accès soit accordé. Bien que cette méthode soit exhaustive, elle nécessite de nombreux essais et peut entraîner un blocage de compte si un site web impose un nombre limité de tentatives de connexion avant de bloquer l'accès ou d'exiger une vérification supplémentaire.

2. Tables arc-en-ciel (Rainbow Tables) : Les attaques avec des tables arc-en-ciel exploitent des bases de données pré-calculées de mots de passe et de leurs hachages correspondants. Pour déchiffrer un mot de passe, il suffit de comparer le hachage du mot de passe ciblé avec les hachages stockés dans la table. Dès qu'une correspondance est trouvée, le mot de passe en texte clair est révélé. Cette méthode est plus efficace et rapide que les attaques par force brute ou par dictionnaire, car elle utilise des données pré-calculées.

3. Attaque par dictionnaire : Cette méthode utilise une liste de mots couramment utilisés comme mots de passe, des phrases communes, ou des variantes de mots avec des caractères spéciaux. L'idée est d'automatiser les tentatives de connexion en parcourant cette liste.

Si le mot de passe se trouve dans le dictionnaire ou est une variante simple d'un mot commun (ajout de majuscules, chiffres, etc.), il peut être découvert rapidement. Cette méthode est plus rapide que l'attaque par force brute car elle cible les mots de passe les plus probables.

4.10. Choix des mots de passe (5 points + 5 bonus)

1. Utilisez « John The Ripper » avec le dictionnaire « rockyou.txt », et identifiez le mot de passe correspondant à chaque utilisateur du fichier « passwords » sur Moodle. Le dictionnaire est pré-installé sur Kali dans /usr/share/wordlists/, mais il est compressé. Vous pouvez le décompresser avec la commande « gunzip ». /2

```
(kali㉿kali)-[~/Documents]
└─$ john -wordlist=/usr/share/wordlists/rockyou.txt Password_File
Using default input encoding: UTF-8
Loaded 4 password hashes with 4 different salts (sha512crypt, crypt(3) $6$ [SHA512 32/32])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
123456789      (simple)
monkey          (action)
sunshine        (brian)
liverpool       (vladimir)
4g 0:00:00:00 DONE (2024-09-17 10:39) 33.33g/s 300.0p/s 666.6c/s 666.6C/s jordan..liverpool
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

(kali㉿kali)-[~/Documents]
└─$ john --show Password_File
vladimir:liverpool:18347:0:99999:7 :::
action:monkey:18347:0:99999:7 :::
brian:sunshine:18346:0:99999:7 :::
simple:123456789:18346:0:99999:7 :::

4 password hashes cracked, 0 left

(kali㉿kali)-[~/Documents]
└─$ echo 2244082-2088099 `date`
2244082-2088099 Tue Sep 17 10:39:37 AM EDT 2024

(kali㉿kali)-[~/Documents]
└─$ john -wordlist=/usr/share/wordlists/rockyou.txt Password_File
Using default input encoding: UTF-8
Loaded 4 password hashes with 4 different salts (sha512crypt, crypt(3) $6$ [SHA512 32/32])
No password hashes left to crack (see FAQ)

(kali㉿kali)-[~/Documents]
└─$ john --show Password_File
vladimir:liverpool:18347:0:99999:7 :::
action:monkey:18347:0:99999:7 :::
brian:sunshine:18346:0:99999:7 :::
simple:123456789:18346:0:99999:7 :::

4 password hashes cracked, 0 left

(kali㉿kali)-[~/Documents]
└─$
```

Figure 58 : Identification des mots de passe de chaque utilisateur du fichier « passwords »

2. A votre avis pourquoi est-il conseillé de ne pas utiliser le même mot de passe partout ? Donnez aussi un exemple de situation réelle. /3

Il est important de ne pas réutiliser le même mot de passe sur différents sites. En effet, si un mot de passe est compromis sur un site, tous les autres comptes utilisant ce même mot de passe deviennent également vulnérables. De plus, la plupart des sites nécessitent une connexion par courriel, donc si une personne malveillante découvre votre mot de passe commun et votre adresse courriel, elle peut facilement accéder à plusieurs de vos informations personnelles. Cela peut arriver, par exemple, lorsqu'un site est piraté et que les données de connexion sont volées. Imaginons que cela se produise et que les mêmes identifiants soient utilisés à la fois sur le site piraté et sur votre compte iCloud. Puisque de nombreuses personnes stockent leurs messages, données bancaires, photos, et bien plus encore sur iCloud, utiliser le mot de passe compromis pour accéder à ce compte pourrait mener à une violation de la vie privée et au vol d'argent. Pour éviter de tels risques, il est primordial d'utiliser des mots de passe uniques pour chaque service.

3. BONUS 5pts - Faites un lien entre votre réponse en 4.9.6 et votre réponse en 4.10.2. /+5

Dans la question 4.9.6, nous avons examiné trois méthodes pour déchiffrer un mot de passe haché, et dans la question 4.10.2, nous avons expliqué pourquoi il est déconseillé d'utiliser le même mot de passe partout. Utiliser un mot de passe identique pour différents comptes facilite le déchiffrement des mots de passe hachés par les trois méthodes mentionnées. Si le mot de passe est court, simple et contient des mots courants, une attaque par force brute ou par dictionnaire permettra de le découvrir très facilement. Étant donné que ce mot de passe est utilisé partout, ces deux méthodes d'attaque permettront d'accéder non seulement au compte ciblé, mais également à tous les autres comptes utilisant le même mot de passe. Pour ce qui est de la méthode des tables arc-en-ciel, si le mot de passe est basique et courant, il est fort probable qu'il figure déjà dans la table. Ainsi, il sera simple de trouver le mot de passe puisque le hash correspondant sera également présent dans la table arc-en-ciel. En conclusion, réutiliser le même mot de passe sur plusieurs sites rend beaucoup plus facile le déchiffrement de mots de passe hachés.

4.11. Déchiffrement simple (15 points)

Texte à déchiffrer (2244082) :

LWOVPWWZTLKVQWVQIW@VIWIFTIWEQWFNFQZW@TLWJFLM@WM@TWFQ
CVQMW@VIWJTTQWRGVHTIWOFM@WM@TPTWZEEIWRTERGTMWEWQBLPT:WM
@TUWOTLTWJTMMTLWECCWM@TQWM@TUW@VIWQEMWJTTQWGEQZWKVLLF
TIWVQIWM@TFLWTGITPMWH@FGIWOVPWJBMWDBPM

1. Donnez une capture d'écran de votre utilisation de l'utilitaire. /I

```
(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ echo 2244082-2088099 `date`
2244082-2088099 Thu Oct 3 03:39:58 PM EDT 2024

(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./frequency < text2244082 -n 1
: : 0.500 %
@ : 6.500 %
B : 1.500 %/oici une capsule vidéo vous permettant de démarrer de bon pied les travaux pratiques.
C : 1.500 %
D : 0.500 %
E : 4.000 %
F : 4.500 % Guidé - Librairie de Babel
G : 2.500 %
H : 1.000 %
I : 6.000 % passwords
J : 2.500 %
K : 1.000 %
L : 5.000 %
M : 8.000 % Fichier Q4.11 - G1
N : 0.500 %
O : 2.000 %
P : 3.000 %
Q : 6.500 % Fichier Q4.11 - G2
R : 1.500 %
T : 12.500 %
U : 1.000 % Fichier Q4.11 - G3
V : 6.000 %
W : 19.500 %
Z : 2.500 % Fichier Q4.11 - G4
```

Figure 59 : Utilisation de l'utilitaire du texte présenté sur Moodle avec n = 1

```
(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./frequency < text2244082 -n 2
:W : 0.503 %
@F : 0.503 %
@T : 3.518 % ChatGPT
@V : 1.508 %
@W : 1.005 %
BL : 0.503 % Accueil Tableau de bord Aide ▾ Liens utiles ▾
BM : 0.503 %
BP : 0.503 %
CC : 0.503 %
CV : 0.503 % Voici une capsule vidéo vous permettant de démarrer de bon pied les travaux pratiques.
CW : 0.503 %
DB : 0.503 %
EC : 0.503 %
EE : 0.503 % Guide - Librairie de Babel
EI : 0.503 %
EM : 0.503 %
EQ : 1.005 %
ER : 0.503 % passwords
EW : 0.503 %
FG : 0.503 %
FL : 1.005 % Fichier Q4.11 - G1
FM : 0.503 %
FN : 0.503 %
FQ : 1.005 %
FT : 1.005 % Fichier Q4.11 - G2
GE : 0.503 %
GI : 1.005 %
GT : 0.503 % Fichier Q4.11 - G3
GV : 0.503 %
H@ : 0.503 %
HT : 0.503 %
IF : 0.503 % Fichier Q4.11 - G4
IT : 0.503 %
IW : 5.025 %
JB : 0.503 % Remise TP1 - Groupe 1
JF : 0.503 %
JT : 1.508 %
KV : 1.005 % 🔒 Non disponible à moins que : Vous soyez membre d'un groupe de Groupe 1
LF : 0.503 %
LK : 0.503 %
LL : 0.503 % Remise TP1 - Groupe 2
LM : 0.503 %
LP : 0.503 %
LT : 0.503 % 🔒 Non disponible à moins que : Vous soyez membre d'un groupe de Groupe 2
LW : 2.010 %
M@ : 4.020 %
ME : 0.503 % Remise TP1 - Groupe 3
MM : 0.503 %
MT : 0.503 %
MW : 2.010 % 🔒 Non disponible à moins que : Vous soyez membre d'un groupe de Groupe 3
NF : 0.503 %
OF : 0.503 %
OT : 0.503 % Rendu TP1 - Groupe 4 (N'Famoussa Kounon Nanamou)
OV : 1.005 %
```

Figure 60 : Utilisation de l'utilitaire du texte présenté sur Moodle avec n = 2

```
(kali㉿kali)-[~/Downloads/utilitaires_TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./frequency < text2244082 -n 3
WM : 0.505 %
FG : 0.505 %
TF : 0.505 %
TL : 0.505 %
TP : 0.505 %
TQ : 0.505 % Accueil Tableau de bord Aide ▾ Liens utiles ▾
TU : 1.010 %
TW : 0.505 %
VI : 1.515 %
WM : 1.010 % Ici une capsule vidéo vous permettant de démarrer de bon pied les travaux pratiques.
BLP : 0.505 %
BMW : 0.505 %
BPM : 0.505 %
CW : 0.505 % Guide - Librairie de Babel
CVQ : 0.505 %
CWM : 0.505 %
DBP : 0.505 %
ECC : 0.505 % passwords
EEI : 0.505 %
EIW : 0.505 %
EMW : 0.505 % Fichier Q4.11 - G1
EQW : 0.505 %
EQZ : 0.505 %
ERG : 0.505 %
EWQ : 0.505 % Fichier Q4.11 - G2
FGI : 0.505 %
FLM : 0.505 %
FLW : 0.505 % Fichier Q4.11 - G3
FM@ : 0.505 %
FNF : 0.505 %
FQC : 0.505 %
FQZ : 0.505 % Fichier Q4.11 - G4
FTI : 1.010 %
GEQ : 0.505 %
GIT : 0.505 %
GIW : 0.505 % Remise TP1 - Groupe 1
GTW : 0.505 %
GVH : 0.505 % Non disponible à moins que : Vous soyez membre d'un groupe de Groupe 1
H@F : 0.505 %
HTI : 0.505 %
IFT : 0.505 % Remise TP1 - Groupe 2
ITP : 0.505 %
IW@ : 0.505 %
IWE : 0.505 % Non disponible à moins que : Vous soyez membre d'un groupe de Groupe 2
IWI : 0.505 %
IWJ : 0.505 %
IWM : 0.505 % Remise TP1 - Groupe 3
IWO : 1.010 %
IWQ : 0.505 %
IWR : 0.505 % Non disponible à moins que : Vous soyez membre d'un groupe de Groupe 3
IWV : 0.505 %
JBM : 0.505 %
JFL : 0.505 % Rendu TP1 - Groupe 4 (N'Famoussa Kounon Nanamou)
JTM : 0.505 %
```

Figure 61 : Utilisation de l'utilitaire du texte présenté sur Moodle avec $n = 3$

2. Fournir le texte déchiffré, ainsi que votre démarche clairement expliquée. /14

Je vais essayer de créer une table de correspondance en comparant les fréquences des lettres normalement à celle dans ce texte (avec n = 1 d'abord), puis à chaque itération je remplirais ce dictionnaire python:

```
mapping_dict = {  
    'W': 'e',  
    'T': 't',  
    'M': 'a',  
    'I': 'o',  
    'Q': 'i',  
    'V': 'n',  
    'E': 's',  
    'F': 'h',  
    'L': 'r',  
    'P': 'd',  
    'G': 'c',  
    'J': 'u',  
    'Z': 'm',  
    'B': 'w',  
    'C': 'f',  
    'O': 'y',  
    'R': 'p',  
    'H': 'g',  
    'U': 'v',  
    'K': 'k',  
    '@': ' ',  
    ':': ' ',  
}
```

Et j'utiliserais ce script que j'ai écrit pour remplacer, caractère par caractère, le texte original:

```
encrypted_text =  
"LWOVPWVWZTLKVQWVQIW@VIWIFTIWEQWZFNQZW@TLWJFLM@WM@TWFQCVQMW@VIWJTTQWRGVHT  
IWOFM@WM@TPTWZEEIWRTERGTMWEQBLPT:WM@TUWOTLTWJTMMLWECCWM@TQWM@TUW@VIWQEMW  
JTTQWGEOZWKVLLFTIWFQIWM@TFLWTGITPMWH@FGIWOPWJBWDBPM"  
  
deciphered_text = ''.join(mapping_dict.get(char, char) for char in encrypted_text)  
print(deciphered_text)
```

Les caractères qui apparaissent souvent mais qui ne s'intègrent pas bien dans les modèles de lettres anglais courants pourraient être des candidats pour des espaces. Étant donné que le caractère @ apparaît fréquemment et ne fait généralement pas partie des mots anglais, nous pouvons considérer @ comme représentant un espace.

Essayons de faire des correspondances en fonction de notre analyse et définissons @ comme un espace. Voici la correspondance basée sur l'analyse de fréquence :

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

Figure 62 : Mapping Attempt n°1

À l'aide de notre correspondance actuelle, substituons à nouveau pour voir si nous pouvons trouver des mots plus clairs :

A O E I N O I S T A U H O I C E O H S A F A S H I E A T A R A A D E A A E R E A F I A R
G L E C O O D A D I O I A F O N E I I R O A F N A E A I D A I E E I R G A I E D O H I C E
O D G H E I R C E A A D A R A D D A C C A D D A I N D E I C A A I R C A D O H C E C R
A E O R I A O A E D O E I C O A A E D I I O E A I A D A I I N O

Recherche de mots anglais reconnaissables et identification des substitutions potentielles.

- **Espaces** : La fréquence élevée des espaces pourrait indiquer des ruptures entre les mots.
- Mots courants : "THE", "AND", "IS", "TO", etc., qui pourraient aider à cartographier les lettres.

En tenant compte de nos substitutions, procédons à des ajustements basés sur les modèles de mots :

1. Changer O en T (car il apparaît dans des mots courants).
2. Changer A en H (pour trouver des motifs communs).
3. F peut être R ou S selon son placement.
4. G pourrait être W.

Nouvelle correspondance de substitution :

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

Figure 63 : Mapping Attempt n°2

Substituons à nouveau le texte en utilisant les correspondances affinées. Les nouvelles substitutions seront appliquées pour obtenir des résultats plus clairs :

H E T I N E T I S O H T I R E T T I T H S A N I S H I T A O A D O I H O N D E A O E D R E
H T A E D E T O O I E A E I T T E T T E I D E D T E H I C T D O I D T E E R I A O E H A
I R O H A H H E E A D E D E I O I E H O I H O D A A I R H A H D H T E I C I E T D O T E
A R H O N A H O I H O E O A T

En analysant la sortie, nous n'avons toujours pas de cohérence. Réalisons une analyse plus approfondie :

- **Modèles communs :** Recherchez des mots de deux et trois lettres (par exemple, "AT", "TO", "IS", "IT", "HE").
- Plus de raffinements :
 - A pourrait devenir S ou U.
 - E pourrait devenir A ou I.

Cela nous donne :

L	W	O	V	P	T	Z	K	Q	I	F	N	G	E	@	J	H	M	R	D	:
T	H	E	R	N	O	S	A	Y	I	L	F	M	C		G	K	D	P	W	:

Figure 64 : Mapping Attempt n°3

Substituons à nouveau le texte original en utilisant la dernière correspondance:

THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG: THE FINAL FANTASY VII
REMAKE IS GREAT

Le résultat forme une phrase constituée de mots anglais cohérents.