INF4420A –Sécurité informatique

Automne 2021

TP No. 1

Groupe 1

1949477 – Ming Xiao Yuan

1953707 – Pier-Luc Tanguay

Soumis à : M. Guilhem Hermet

6 octobre 2021

Partie A

Question 1 - Entropie

a) Calculez l'entropie par lettre (h-lettre) d'une chaîne générée avec texte d'une longueur de 200 caractères.

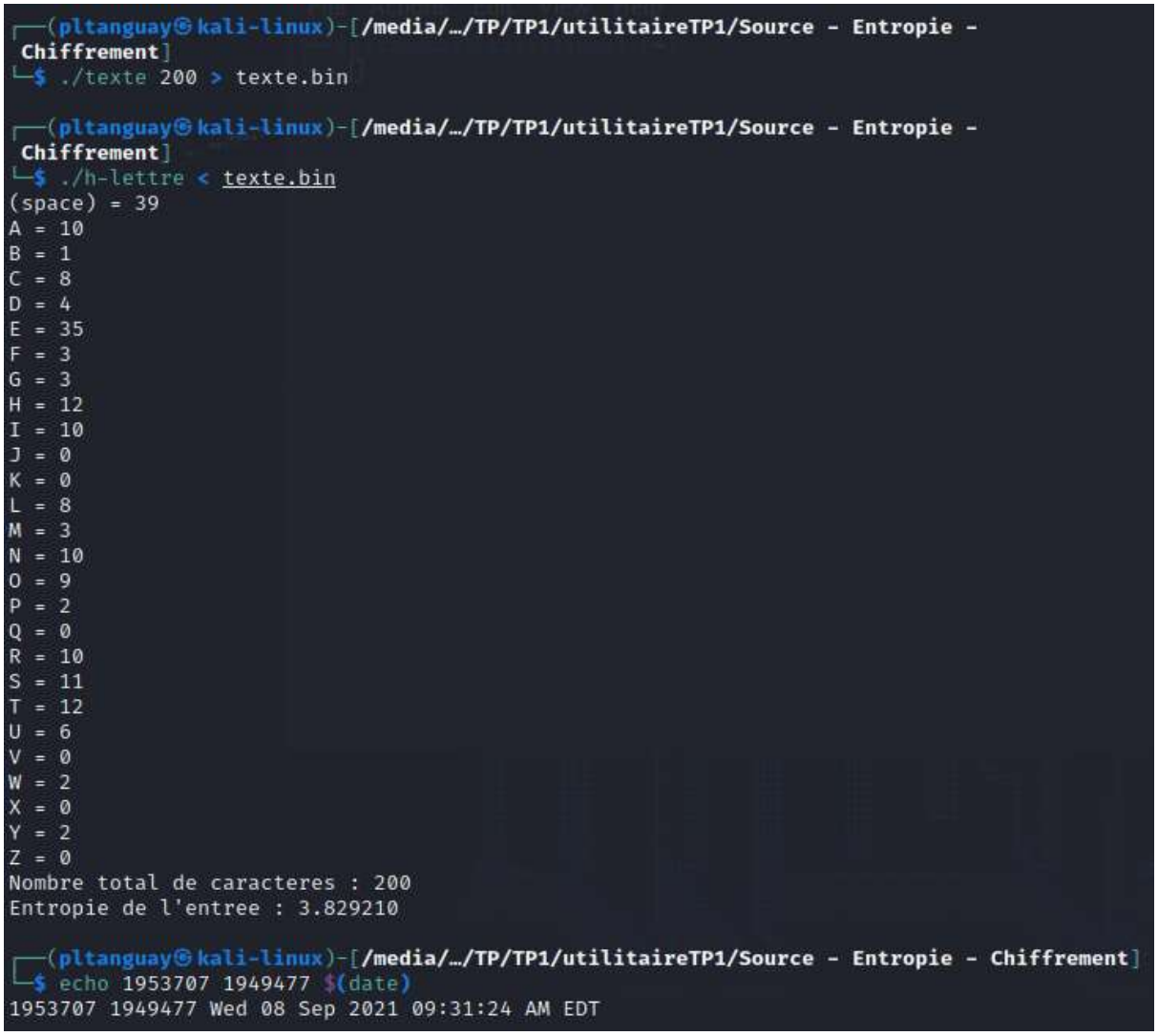


Figure 1. Entropie de 200 caractères, provenant de la source 'texte'

L'entropie par lettre calculée est de 3.829210 bits.

b) En vous servant du premier théorème de Shannon, expliquez ce que signifie cette valeur.

Selon le premier théorème de Shannon, chaque symbole émis par le programme 'texte' peut être codé individuellement avec en moyenne 3.82 bits. Cette valeur dépend entre autres de la fréquence d'apparition des lettres, ce qui veut dire qu'un texte différent donnerait

une entropie différente. Dans ce cas, nous aurions besoin de 3.829210 bits pour coder les caractères existants. Par exemple, afin d'encoder un texte ou les 26 lettres de l'alphabet apparaissent au moins 1 fois, il faudrait entre 4 à 5 bits. Cependant, dans notre cas, pas toutes les lettres sont présentes, il faut donc moins de bits pour encoder les lettres restantes.

c) Quelle serait l'entropie par lettre (en moyenne) d'un fichier qui aurait été généré de la même façon, mais avec les mêmes probabilités (1/27) pour chacun des 27 symboles (lettres majuscules et espace)?

La formule d'entropie est représentée par $H(S) = -\sum_{i=1}^{n} p_i log_b(p_i)$ ou p_i représente la probabilité d'une lettre x_i apparaît, b représente les unités de bit/symbole (2 plupart du temps) et n représente le nombre de symboles. De ce fait, nous avons une entropie de $H(S) = -\sum_{i=1}^{27} \frac{1}{27} log_2(\frac{1}{27}) = 4.7548875\ bits$.

d) Que représente le quotient de la valeur en a) sur la valeur en c) ?

Le quotient de $\frac{3.829210}{4.7548875} = 0.80363$ correspond au taux de compression, ce qui signifie jusqu'à quel point on a pu réduire le codage pour chaque symbole de source par rapport au pire cas.

e) Refaites la même chose qu'en a) avec la source lettre. Comparez la valeur obtenue avec celle en a). Est-ce que la différence est significative (supérieure à 0.4) ?

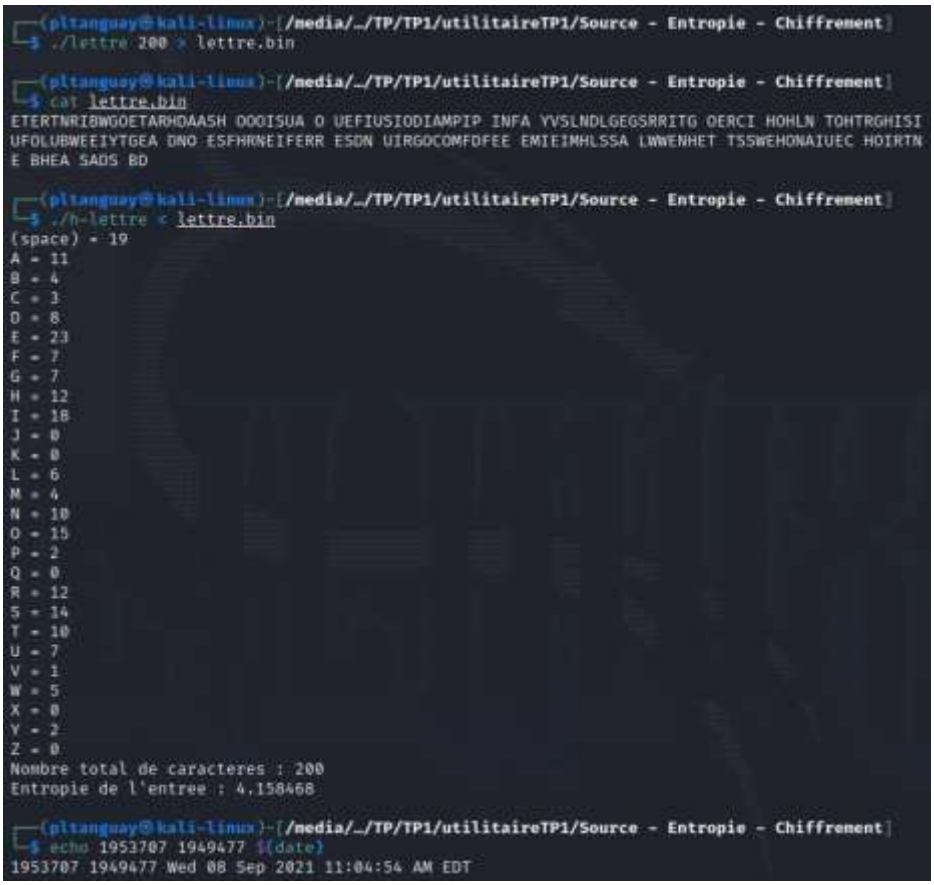


Figure 2. Entropie de 200 caractères, par 'h-lettre' et 'lettre'

L'entropie de 4.158468 bits trouvée semble supérieure à 3.829210 trouvée en a), mais n'est pas significativement supérieure puisque que la différence est de 0.33, ce qui est inférieur à 0.4.

f) On sait qu'un texte anglais est constitué de mots et de phrases qu'il est nécessaire d'interpréter en fonction d'un langage et d'une grammaire. Un texte anglais est donc très redondant (et donc facile à comprimer). Les chaînes générées par lettre ne sont pas de l'anglais malgré l'utilisation des mêmes fréquences. Le résultat obtenu en e) peut donc surprendre. Expliquer cette contradiction apparente (le fait que les deux entropies soient proches).

Le fait que les deux entropies soient proches est justifiée par le fait que les fréquences entre les deux situations sont similaires (e.g. A=10 pour 'texte' et A=11 pour 'lettre', B=1 pour 'texte' et B=4 pour 'lettre'), même si pour 'lettre', ce n'est pas de l'anglais.

Question 2 - Histogrammes

a) Utilisez les programmes cesar et cesar-d avec les sources texte et lettre, pour chiffrer et déchiffrer des chaînes de 200 caractères.

```
┌──(pltanguay㉿kali-linux)-[/media/…/TP/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ ./texte 200 > texte200/texte-200.bin

┌──(pltanguay㉿kali-linux)-[/media/…/TP/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ cat texte200/texte-200.bin
THE WORK OF JACOB TONSON AS PRINTER AND NICHOLAS ROWE AS EDITOR MOVED THE WHOLE HISTORY OF E
DITING SHAKESPEARE ONTO AN ENTIRELY NEW PLANE BETWEEN AND ROWE S EDITION WAS THE SINGLE GREA
TEST DETERMINANT

┌──(pltanguay㉿kali-linux)-[/media/…/TP/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ ./cesar < texte200/texte-200.bin > texte200/cesar-texte-200.bin

┌──(pltanguay㉿kali-linux)-[/media/…/TP/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ cat texte200/cesar-texte-200.bin
WKH ZRUN RI MDFRE WRQVRQ DV SULQWHU DQG QLFKRODV URZH DV HGLWRU PRYHG WKH ZKROH KLVWRUB RI H
GLWLQJ VKDNHVSHDUH RQWR DQ HQWLUHOB QHZ SODQH EHWZHHQ DQG URZH V HGLWLRQ ZDV WKH VLQJOH JUHD
WHVW GHWHUPLQDQW

┌──(pltanguay㉿kali-linux)-[/media/…/TP/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ ./cesar-d < texte200/cesar-texte-200.bin > texte200/cesar-d-texte-200.bin

┌──(pltanguay㉿kali-linux)-[/media/…/TP/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ cat texte200/cesar-d-texte-200.bin
THE WORK OF JACOB TONSON AS PRINTER AND NICHOLAS ROWE AS EDITOR MOVED THE WHOLE HISTORY OF E
DITING SHAKESPEARE ONTO AN ENTIRELY NEW PLANE BETWEEN AND ROWE S EDITION WAS THE SINGLE GREA
TEST DETERMINANT

┌──(pltanguay㉿kali-linux)-[/media/…/TP/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ echo 1953707 1949477 $(date)
1953707 1949477 Tue 14 Sep 2021 10:08:31 PM EDT
```

Figure 3. Chiffrement et déchiffrement avec ‘texte’

```
┌──(pltanguay㉿kali-linux)-[/media/…/TP/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ ./lettre 200 > lettre200/lettre-200.bin

┌──(pltanguay㉿kali-linux)-[/media/…/TP/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ cat lettre200/lettre-200.bin
O EISWANEEEYLRS SNUE OBT HECDOLNST YLSTNTN REEIJ DECTR HWTWARI AEI TD TADH P WSENE BI TAELVN
LE OOSRIA NTOOUAF APN RRANBNEYSHOFHPUUEDMO TDHSAIE I STYNNHPGSVAAIGT VEISAITDS EEA  DNYNFEIR
EKLEGOSDWOHTHURR

┌──(pltanguay㉿kali-linux)-[/media/…/TP/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ ./cesar < lettre200/lettre-200.bin > lettre200/cesar-lettre-200.bin

┌──(pltanguay㉿kali-linux)-[/media/…/TP/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ cat lettre200/cesar-lettre-200.bin
R HLVZDQHHHBOUV VQXH REW KHFGROQVW BOVWQWQ UHHLM GHFWU KZWZDUL DHL WG WDGK S ZVHQH EL WDHOYQ
OH RRVULD QWRRXDI DSQ UUDQEQHBVKRIKSXXHGPR WGKVDLH L VWBQQKSJVYDDLJW YHLVDLWGV HHD  GQBQIHLU
HNOHJRVGZRKWKXUU

┌──(pltanguay㉿kali-linux)-[/media/…/TP/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ ./cesar-d < lettre200/cesar-lettre-200.bin > lettre200/cesar-d-lettre-200.bin

┌──(pltanguay㉿kali-linux)-[/media/…/TP/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ cat lettre200/cesar-d-lettre-200.bin
O EISWANEEEYLRS SNUE OBT HECDOLNST YLSTNTN REEIJ DECTR HWTWARI AEI TD TADH P WSENE BI TAELVN
LE OOSRIA NTOOUAF APN RRANBNEYSHOFHPUUEDMO TDHSAIE I STYNNHPGSVAAIGT VEISAITDS EEA  DNYNFEIR
EKLEGOSDWOHTHURR

┌──(pltanguay㉿kali-linux)-[/media/…/TP/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ echo 1953707 1949477 $(date)
1953707 1949477 Tue 14 Sep 2021 10:16:16 PM EDT
```

Figure 4. Chiffrement et déchiffrement avec ‘lettre’

On remarque aux figures 3 et 4 qu’on obtient les mêmes chaînes de caractères après le déchiffrement et avant le chiffrement.

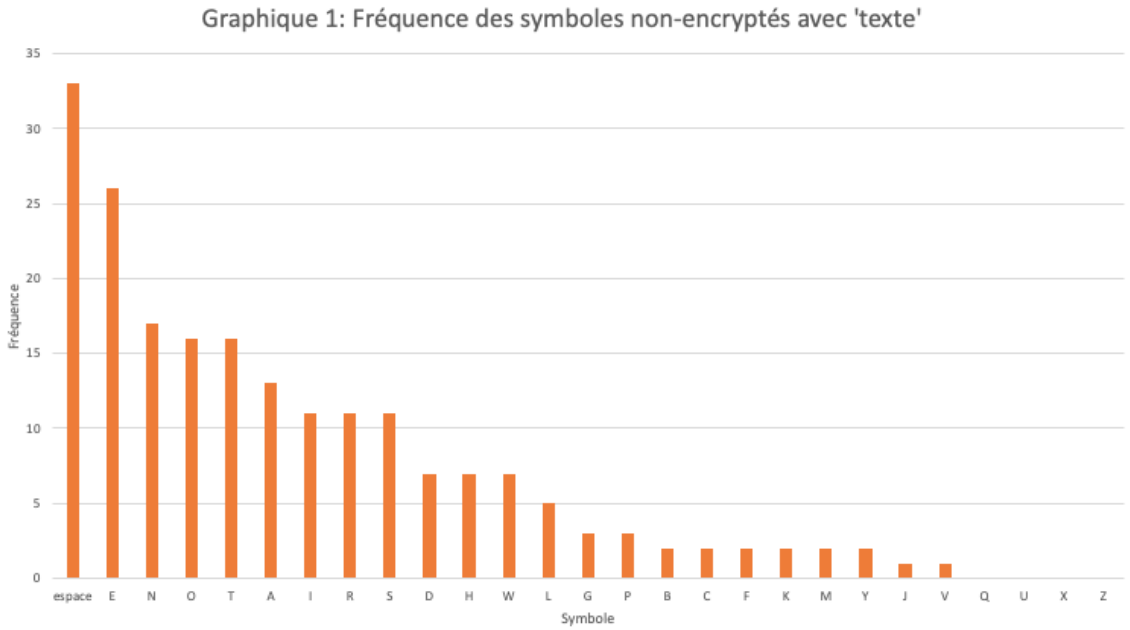
b) Utilisez le programme h-lettre pour obtenir les fréquences des lettres. Construisez des histogrammes de fréquences ordonnées du plus grand au plus petit pour la sortie de chacune des sources ainsi que pour les versions codées. (Note : Vous pouvez facilement générer ces histogrammes en redirigeant la sortie de h-lettre dans un fichier que vous pouvez importer et traiter dans Excel, par exemple).

```
┌──(pltanguay㉿kali-linux)-[/media/…/TP/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ ./h-lettre < texte200/texte-200.bin > texte200/freq-texte-200.csv

┌──(pltanguay㉿kali-linux)-[/media/…/TP/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ ./h-lettre < texte200/cesar-texte-200.bin > texte200/freq-cesar-texte-200.csv

┌──(pltanguay㉿kali-linux)-[/media/…/TP/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ echo 1953707 1949477 $(date)
1953707 1949477 Tue 14 Sep 2021 11:03:02 PM EDT
```

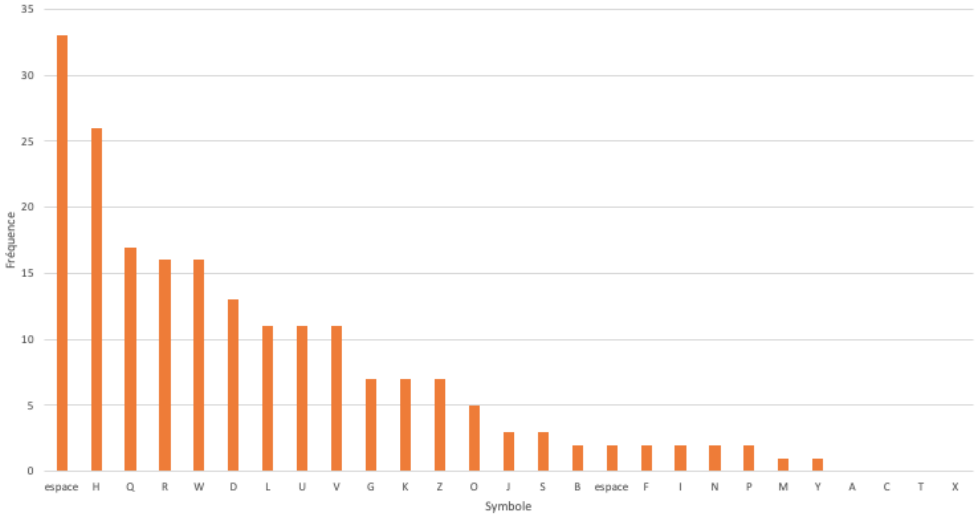
Figure 5. Utilisation de ‘h-lettre’ pour obtenir la fréquence des caractères pour ‘texte’

```
┌──(pltanguay㉿kali-linux)-[/media/…/TP/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ ./h-lettre < lettre200/lettre-200.bin > lettre200/freq-lettre-200.csv

┌──(pltanguay㉿kali-linux)-[/media/…/TP/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ ./h-lettre < lettre200/cesar-lettre-200.bin > lettre200/freq-cesar-lettre-200.csv

┌──(pltanguay㉿kali-linux)-[/media/…/TP/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ echo 1953707 1949477 $(date)
1953707 1949477 Tue 14 Sep 2021 11:03:37 PM EDT
```

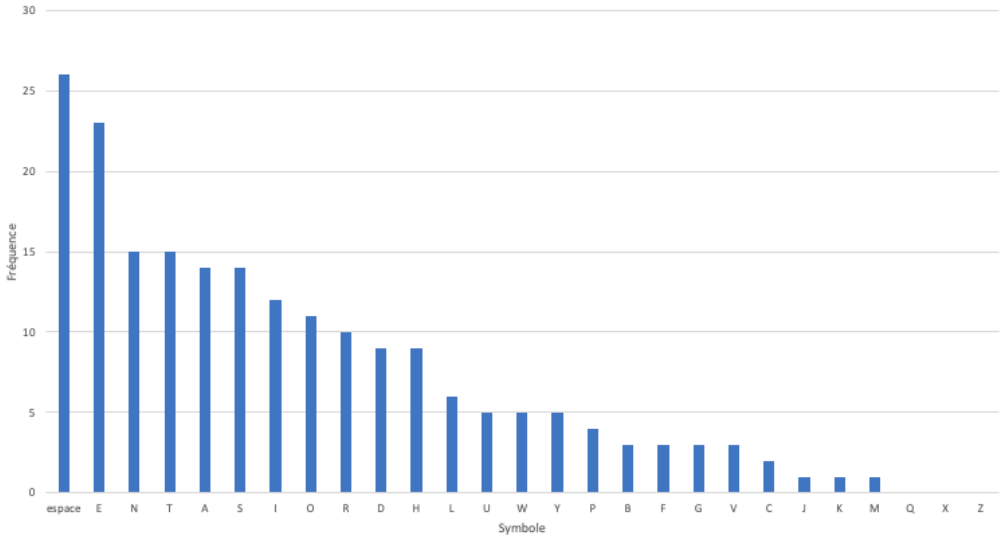
Figure 6 a). Utilisation de ‘h-lettre’ pour obtenir la fréquence des caractères pour ‘lettre’



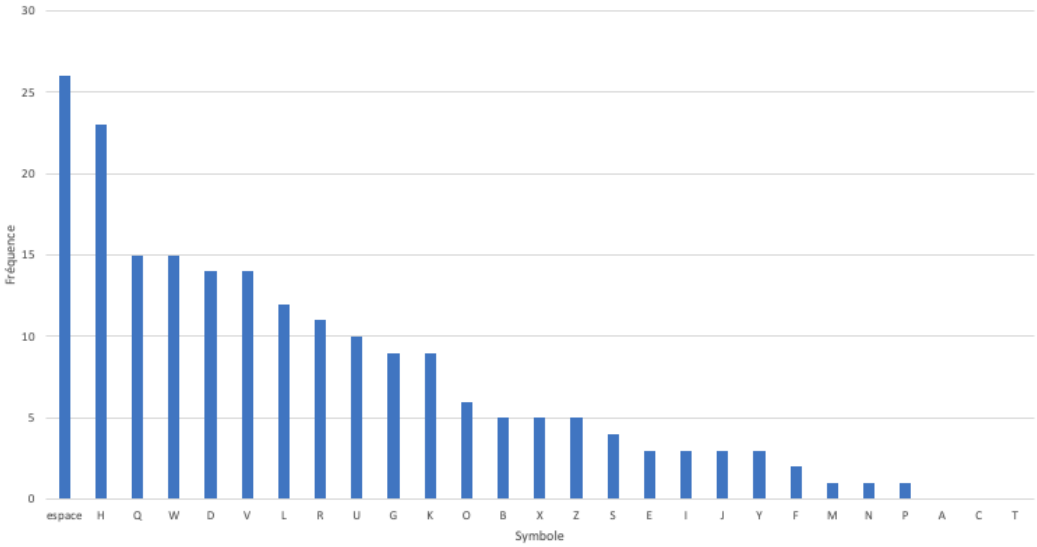
Graphique 2: Fréquence des symboles encryptés avec 'texte'



Graphique 3: Fréquence des symboles non-encryptés avec 'lettre'



Graphique 4: Fréquence des symboles encryptés avec 'lettre'



c) 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.

En comparant les deux histogrammes 'texte' (graphiques 1 et 2) et 'lettre' (graphiques 3 et 4), on remarque que nous sommes capables de déduire quelques caractères. Les caractères faciles à déduire sont ceux dont la fréquence est la plus élevée, comme le 'e' substitué en 'h'. Pour les fréquences qui forment un plateau, c'est impossible pour nous de déduire. Les histogrammes, deux à deux, sont parfaitement semblables puisque l'algorithme ne fait qu'un décalage de caractère. C'est facile de déduire le texte original.

Globalement, si les fréquences avaient été comptabilisées sur deux lettres à la fois, on aurait vu une diminution des fréquences pour chaque combinaison et également plus de combinaisons de symboles 'XX'.

Plus spécifiquement, pour 'texte', le programme construit des mots existants en anglais, qui suit une norme d'orthographe. La source est non-markovienne. Le graphique 1 montre bien que 'e' et 't' sont assez fréquents, mais un peu moins pour 'h'.

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	1	20	33	52	0	12	18	5	39	1	12	57	26	181	1	20	1	75	95	104	9	20	13	1	26	1
b	11	1	0	0	47	0	0	0	6	1	0	17	0	0	19	0	0	11	2	1	21	0	0	0	11	0
c	31	0	4	0	38	0	0	38	10	0	18	9	0	0	45	0	1	11	1	15	7	0	0	0	1	0
d	48	20	9	13	57	11	7	25	50	3	1	11	14	16	41	6	0	14	35	56	10	2	19	0	10	0
e	110	23	45	126	48	30	15	33	41	3	5	55	47	111	33	28	2	169	115	83	6	24	50	9	26	0
f	25	2	3	2	20	11	1	8	23	1	0	8	5	1	40	2	0	16	5	37	8	0	3	0	2	0
g	24	3	2	2	28	3	4	35	18	1	0	7	3	4	23	1	0	12	9	16	7	0	5	0	1	0
h	114	2	2	1	302	2	1	6	97	0	0	2	3	1	49	1	0	8	5	32	8	0	4	0	4	0
i	10	5	32	33	23	17	25	6	1	1	8	37	37	179	24	6	0	27	86	93	1	14	7	2	0	2
j	2	0	0	0	2	0	0	0	3	0	0	0	0	0	3	0	0	0	0	0	8	0	0	0	0	0
k	6	1	1	1	29	1	0	2	14	0	0	2	1	9	4	0	0	0	5	4	1	0	2	0	2	0
l	40	3	2	36	64	10	1	4	47	0	3	56	4	2	41	3	0	2	11	15	8	3	5	0	31	0
m	44	7	1	1	68	2	1	3	25	0	0	1	5	2	29	11	0	3	10	9	8	0	4	0	18	0
n	40	7	25	146	66	8	92	16	33	2	8	9	7	8	60	4	1	3	33	106	6	2	12	0	11	0
o	16	12	13	18	5	80	7	11	12	1	13	26	48	106	36	15	0	84	28	57	115	12	46	0	5	1
p	23	1	0	0	30	1	0	3	12	0	0	15	1	0	21	10	0	18	5	11	6	0	1	0	1	0
q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0
r	50	7	10	20	133	8	10	12	50	1	8	10	14	16	55	6	0	14	37	42	12	4	11	0	21	0
s	67	11	17	7	74	11	4	50	49	2	6	13	12	10	57	20	2	4	43	109	20	2	24	0	4	0
t	59	10	11	7	75	9	3	330	76	1	2	17	11	7	115	4	0	28	34	56	17	1	31	0	16	0
u	7	5	12	7	7	2	14	2	8	0	1	34	8	36	1	16	0	44	35	48	0	0	2	0	1	0
v	5	0	0	0	65	0	0	0	11	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	1	0
w	66	1	1	2	39	1	0	44	39	0	0	2	1	12	29	0	0	3	4	4	1	0	2	0	1	0
x	1	0	2	0	1	0	0	0	2	0	0	0	0	0	0	3	0	0	0	3	0	0	0	0	0	0
y	18	7	6	6	14	7	3	10	11	1	1	4	6	3	36	4	0	3	19	20	1	1	12	0	2	0
z	1	0	0	0	3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 6 b). Fréquence de paires de lettres en anglais sur 10 000 caractères (University of Illinois Chicago, 2021).

Selon la figure 6, en anglais, la combinaison 'th' a une fréquence relative de 330/10000=3.3%, ce qui représente la fréquence la plus élevée dans cette langue. On s'attend donc à obtenir une fréquence assez élevée, plus que 'ee' (0.48%) qui arrive moins souvent que 'th'.

Pour la source ‘texte’, on croit donc que:

Fréquence(th) > Fréquence(ee)

Pour ce qui est de ‘lettre’, la lettre ‘e’, avec une fréquence relative de 13% en anglais, est plus utilisée que ‘t’ (9.1%) et ‘h’ (6.1%), toujours selon la même source de référence. La fréquence serait sans doute plus élevée pour ‘ee’ que ‘th’. La combinaison ‘th’ aurait une fréquence dépendante à la probabilité d’obtenir un ‘t’ et ‘h’ individuellement. On croit que pour la source ‘lettre’:

Fréquence(ee) > Fréquence(th)

d) En vous référant au point précédent ainsi qu’à la question 1 f), est-ce que cette méthode (comptabiliser les fréquences sur deux lettres) facilite le déchiffrement du message dans le cas de la source texte ? Et dans le cas de lettre ? Expliquez la différence s’il y en a une. Pour chacune des deux sources, si cette méthode n’augmente pas la facilité de déchiffrement du message, quelle solution proposez-vous ?

Pour la source ‘texte’, la méthode faciliterait le déchiffrement sans doute le message. Puisque des mots sont générés selon une convention et que la source est donc non-markovienne, les combinaisons de deux lettres suivent une certaine norme et logique qui est moins aléatoire que par la source ‘lettre’. Pour ce dernier, la méthode ne faciliterait pas le déchiffrement, puisque les combinaisons de deux lettres ne suivent pas de logique quelconque.

Des combinaisons qui auraient une fréquence très faible par ‘texte’ peuvent se présenter plus souvent pour ‘lettre’. Par exemple, selon le graphique 3, ‘t’ et ‘n’ ont une fréquence de 15/200 chacun. Cette combinaison risque d’arriver plus souvent avec ‘lettre’ qu’avec ‘texte’ puisque selon la figure 6, ‘tn’ a une fréquence en langue anglaise de 7/10000=0.07%. Avec ‘lettre’, la distribution des fréquences de la méthode par deux lettres sera plus homogène que par ‘texte’. Ce dernier aura une distribution qui est dépendante de la fréquence relative de la figure 6. Les fréquences de combinaisons de deux lettres sont déjà connues dans un langage structuré, mais inconnues pour une source qui place des lettres de façon aléatoire comme le programme ‘lettre’.

Pour faciliter le déchiffrement de la méthode des deux lettres avec la source ‘lettre’, une solution pourrait de tenter de réordonner les lettres individuelles selon les combinaisons les plus fréquentes de la figure 6, en paires. Puisque les combinaisons les plus fréquentes sont connues dans une langue, le fait d’implémenter un algorithme qui prend la source ‘lettre’, qui prend un caractère et cherche s’il y a présence d’un autre caractère dont la combinaison a une fréquence élevée pourrait faciliter le déchiffrement. Il suffirait de mémoriser l’ordre dans lequel les symboles ont été déplacés pour ensuite les ordonner selon l’ordre original.

Question 3 - Masque jetable

a) Générez un fichier de 1024 octets avec monnaie et un avec binaire. Calculer l'entropie par bit (hbit) et l'entropie par octet (h-ascii) sur les deux fichiers créés

Figure 7. Entropie par bits et par octets d'un fichier de 1024 octets généré avec 'monnaie'

Entropie par bit pour 'monnaie' : 0.999997 bits
Entropie par octet pour 'monnaie' : 7.817409 bits

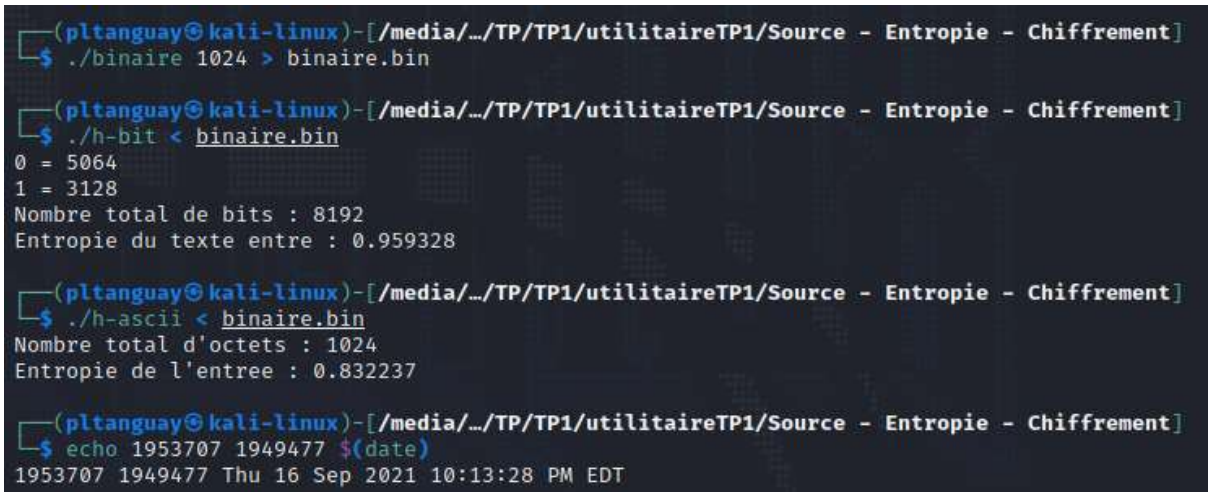


Figure 8. Entropie par bits et par octets d'un fichier de 1024 octets généré avec 'binaire'

Entropie par bit pour 'binaire' : 0.959328 bits
Entropie par octet pour 'binaire' : 0.832237 bits

b) Générez une clé de 1024 octets pour un masque jetable avec monnaie (tel que vu en classe, la taille de la clé doit être la même que la taille du message à chiffrer). Appliquez le masque jetable sur les deux fichiers générés au point précédent en utilisant la clé nouvellement créée. Calculer l'entropie par bit (hbit) et l'entropie par octet (h-ascii) des nouveaux fichiers chiffrés. Qu'observez-vous ? Quelles conclusions pouvez-vous en tirer ?

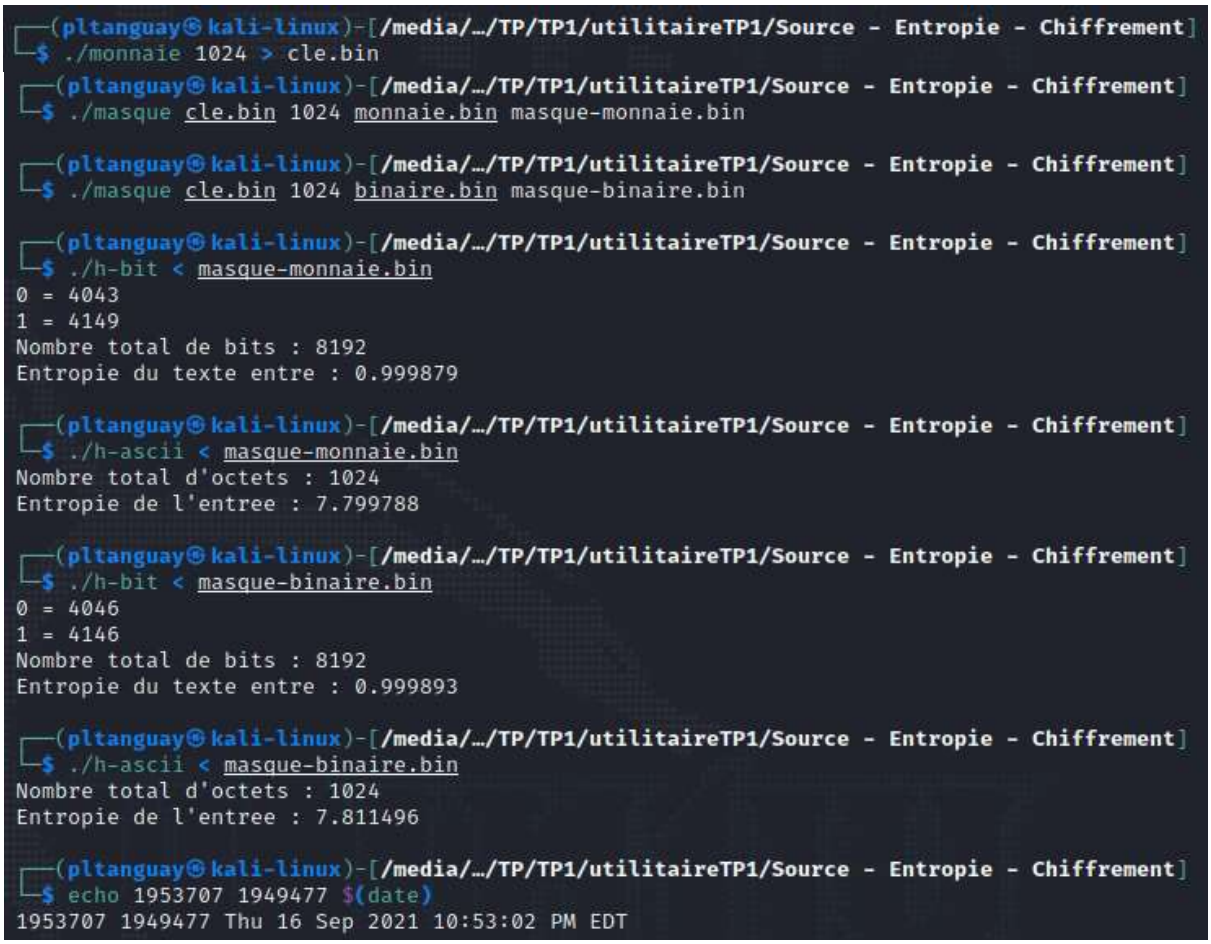


Figure 9. Entropie de de 'monnaie' et 'binaire' avec l'utilisation d'un masque

Avant d'appliquer le masque, l'entropie des fichiers produits par 'monnaie' étaient déjà élevée, atteignant presque que le maximum (1 pour bit et 8 pour octet). L'application du masque a un peu augmenté leur entropie, mais de très peu. Même chose pour l'entropie par bit de 'binaire'.

Cependant, il y a forte augmentation de l'entropie par octet, provenant de 'binaire', de 0.832237 bits à 7.811496 bits. On conclut que la clé augmente la sécurité du fichier et qu'il sera plus difficile à déchiffrer après application du masque.

c) Pour les deux cas, s'agit-il d'une méthode sécuritaire de chiffrement ?

Selon nous, la méthode de déchiffrement est sécuritaire puisque l'entropie, après application du masque, est restée égale ou supérieure. Pour les cas qui sont restés égaux, l'entropie était déjà près du maximum (1 ou 8). Dans le cas du fichier provenant de 'binaire', l'entropie par octet est passée à une valeur près du maximum. La méthode est une solution

qui offre donc une bonne sécurité puisqu'au final, après l'application du masque, les entropies sont toutes près du maximum possible.

Question 4 - Analyse de risque

a) Pour commencer, votre patron vous indique que deux sites potentiels sont retenus pour la nouvelle installation. Le site A se situe sur une île paisible, mais où le marché immobilier est gonflé par les étrangers. Il en coûterait donc 500 000 $ pour s'installer à cet endroit. L'île B, pour attirer des capitaux étrangers, a fait une proposition à votre compagnie. Il en coûterait seulement 100 000 $ pour s'installer sur le site B. Toutefois, selon les données météos que vous avez à votre disposition, l'île B est balayée chaque année par un ouragan qui a 25% de chance de détruire votre installation. Quelle serait votre recommandation et pourquoi ?

Pour l'île B, l'espérance de perte se calcule par:

Espérance de perte = Probabilité * Impact

L'espérance de perte est donc de E = 25%/année * 100000$ = 25000$/année

La nouvelle installation sur l'île B à donc un risque de faire perdre 25000$/année, comparativement au site A qui n'a pas de risque. Cependant, ce dernier est 5 fois plus cher que le site B, donc 400000$ de plus.

Au moment initial t0, le coût de A est de 500000$ et B, 100000$. Il y a à ce moment une différence de 400000$. Si on imagine le pire scénario, c'est-à-dire qu'il y ait un ouragan à chaque année, ça prendrait 16 ans pour que le coût de B soit équivalent à celui de A:

100000$ + 16*25000$ = 500000$

Après 16 ans, la probabilité d'avoir des ouragans existe toujours, donc après 16 ans, la solution A serait plus économique. À l'intérieur de ces 16 ans, la solution B serait plus économique.

Nous recommandons donc la solution A puisqu'on sous-entend que la compagnie opérera pour plus que 16 ans.

b) Vous rencontrez les gestionnaires des diverses lignes d'affaires, et vous évaluez leurs processus d'affaire pour identifier les risques. Trois risques majeurs en ressortent :

i. Un malfaiteur ignore les lignes de conduite prescrites (Terms of Service) et utilise les fonctions du logiciel pour tricher, diminuant l'intérêt du site pour les joueurs légitimes.

ii. Un malfaiteur inonde le serveur de requête pour empêcher les autres joueurs de se connecter à votre site.

iii. Un malfaiteur infiltre votre base de données pour obtenir certaines des informations que vous stockez sur vos clients (adresses courriel et postal, numéro de carte de crédit, habitudes de jeu, historique des achats).

Pour chacun de ces scénarios, précisez s'il s'agit principalement d'un scénario touchant l'intégrité, la confidentialité ou la disponibilité.

i) Les fonctions de bases sont altérées, ce qui le rend dysfonctionnel. Ce scénario touche donc l'**intégrité**.

ii) Le site est difficile d'accès pour les utilisateurs. Ce scénario touche la **disponibilité**.

iii) Les informations sont accessibles par le malfaiteur. Ce scénario touche la **confidentialité**.

c) Après de longs mois d'études, vous avez identifié trois agents de menace potentiels pour votre entreprise :

▪ Tricheurs professionnels : gens qui s'y connaissent peu en informatique, mais beaucoup au jeu ;
▪ Crime organisé : groupes criminalisés qui ont plusieurs experts à leur solde et qui possèdent une solide infrastructure avec des milliers d'ordinateurs compromis;
▪ Sites de poker concurrents : le jeu en ligne est un milieu lucratif et certains de vos concurrents sont prêts à tout pour connaître le secret de votre succès.

Votre patron vous fournit le résultat de l'étude de risque qu'il a fait faire par un grand cabinet de conseil et vous demande de le compléter :

Tableau 1. Analyse risque scénario i.

Acteur	Capacité	Opportunité	Motivation	Probabilité*	Impact	Risque**
Tricheur	4	4	4	4	2	8
C.O.	1	4	1	2	2	4
Concurrents	2	4	2	2.67	2	5.34

Tableau 2. Analyse risque scénario ii.

Acteur	Capacité	Opportunité	Motivation	Probabilité*	Impact	Risque**
Tricheur	1	4	1	2	4	8
C.O.	4	4	1	3	4	12
Concurrents	2	4	4	3.33	4	13.32

Tableau 3. Analyse risque scénario iii.

Acteur	Capacité	Opportunité	Motivation	Probabilité*	Impact	Risque**
Tricheur	1	3	1	1.67	3	5

C.O.	4	3	4	3.67	3	11.01
Concurrents	1	3	2	2	3	6

*: Les probabilités sont calculées en faisant la moyenne des valeurs de **Capacité**, **Opportunité**, et **Motivation.**

: Les risques sont calculés en faisant la multiplication de la **Probabilité avec l'**impact**.

Voici les plus grandes menaces, selon le scénario, à partir des tableaux 1,2 et 3:

- Scénario i: Tricheur
- Scénario ii: Concurrents
- Scénario iii: Crime organisé

d) Pour chacune des situations suivantes expliquez quel(s) paramètre(s) changerai(en)t et dans quel sens (plus grand, plus petit). Quelle(s) conséquence(s) pour la gestion du risque ?

1. Votre compagnie de poker remporte un très grand succès et dépasse tous vos concurrents.

Dans le cas où nous dépassons tous nos concurrents, ils auront moins de **capacités** puisque leurs revenus seront moindres par la baisse de client. Cependant, leur **motivation** augmentera puisqu'ils auront plus d'intérêt à s'attaquer à notre entreprise qui est le standard de l'industrie. Selon nous, l'**opportunité** serait semblable. Nous croyons que le risque sera plus grand puisque la motivation augmentera, et même si leur revenu baisse, les concurrents ont probablement des réserves financières pour s'attaquer à notre entreprise.

2. Votre patron a refusé de payer les pots-de-vin réclamés par la mafia locale

La **motivation** va nécessairement augmenter puisque le crime organisé aura de l'intérêt à punir notre entreprise pour ne pas les avoir payés une redevance. La **capacité** diminue quelque peu puisqu'ils n'auront pas accès à notre argent, mais pourrait décider d'injecter plus de ressource monétaire provenant de leurs autres entreprises pour s'attaquer à nous. L'**opportunité** reste similaire. Le risque va donc nécessairement augmenter.

3. Votre patron fait l'acquisition d'un tout nouveau système de détection des tricheurs très performant.

Le savoir des tricheurs va diminuer leur **capacité** à tricher. Leur **motivation** pourrait augmenter puisqu'ils auraient pour défi de trouver une façon de tricher quand même. L'**opportunité** pour nous reste similaire. Globalement, nous croyons que le risque diminue puisque nous avons confiance que notre nouveau système est robuste.

e) Un vendeur vous propose un service de surveillance à distance pour faire de la détection d'intrusion sur vos serveurs. Il suffit d'installer un logiciel de surveillance et de contrôle à distance pour permettre à ce fournisseur de détecter et combattre les intrusions. Celui-ci vous offre le service à très bon marché (5 000 $ par mois pour une surveillance 24h sur 24) puisqu'il vient d'ouvrir un nouveau centre d'opération dans un pays de l'ex-Union Soviétique où la main d'œuvre coûte une fraction de la main d'œuvre au Canada. Refaites la grille de la question c) pour le scénario iii) en prenant en compte la mesure proposée. Est-ce que vous croyez que cette offre en vaut la chandelle ? Est-ce que votre recommandation s'applique dans toutes les circonstances ?

Tableau 4. Modification du tableau 3 selon les nouvelles mesures.

Acteur	Capacité	Opportunité	Motivation	Probabilité*	Impact	Risque**
Tricheur	1	3	1	1.67	3	5
C.O.	4	1	4	3	3	9
Concurrents	1	1	3	1.67	3	5

Dans le tableau 4, nous avons apporté les modifications en bleu. Tout d'abord, l'**opportunité** du crime organisé et des concurrents diminue, selon nous, puisque le nouveau service de surveillance aux serveurs leurs donnent moins l'occasion de s'attaquer à notre entreprise. Le système n'affecte pas l'opportunité des tricheurs puisque ceux-ci n'utilisent pas les serveurs en tant qu'attaque. Ils s'y connaissent peu en informatique. La **motivation** étant au maximum pour le crime organisé, nous avons augmenté à 3 celle des concurrents puisqu'ils auront plus d'intérêt à pirater nos serveurs. Cela pourrait être vu, selon eux, comme étant un défi de réussir à percer notre nouveau système. Ainsi, les probabilités du crime organisé et des concurrents ont diminué. De ce fait, le risque à également diminué pour ceux-ci, si on ne tient compte que de ces éléments.

Cependant, de nouvelles variables viennent de s'ajouter à l'équation d'évaluation de risque. Il faut faire attention pour ne pas mal juger les intentions du fournisseur provenant d'un pays de l'ex-Union-Soviétique, mais cela laisse quand même place à un certain risque, connaissant leur historique un peu douteuse au sujet des attaques informatiques. De plus, leur entreprise opère dans un pays dont les réglementations sont parfois un peu plus situées dans le spectre de la zone grise à certain moment, pour ne pas mentionner qu'il pourrait y avoir de la corruption ou tout autre fraude. Le fait que le fournisseur nous offre une main-d'œuvre à moindre prix est un peu douteux, du fait que normalement, les services de cybersécurité sont assez dispendieux. Il y a ici des odeurs d'une certaine tentative à vouloir séduire notre entreprise avec des prix faibles pour nous motiver à utiliser leur service. Notre popularité ouvre des **opportunités** à des entreprises qui pourraient profiter de l'occasion de faire un coup d'argent. Il faut alors se méfier.

Nous croyons donc que la faible baisse de risque du tableau 4 ne vaut pas la peine de changer notre système, qui pourrait au final comporter plus de risque dû à l'implication d'une entreprise un peu douteuse. Nous préférons faire affaire avec une entreprise que nous aurons plus confiance, même si cela coûte plus cher.

Partie B

Question 1 - Codage

Cas où le codage est inchangé :

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

σ : {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} où il y a 10 éléments différents donc |**σ**| = 10
τ : {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} où il y a 10 éléments différents donc |**τ**| = 10
τ' : {0, 1} où il y a 2 éléments différents donc |**τ'**| = 2

b) Identifiez les langages provenant des alphabets σ, τ et τ'.

σ : Le langage provenant de **σ** constitue des chiffres offerts par le clavier du GAB, de 0 à 9. Le client peut donc entrer 4 fois un caractère de ce type avec répétition permise et ce dernier provient donc du NIP entré par le client.
τ : Le langage provenant de **τ** constitue une répétition d'une chaîne de 4 caractères, de 0 à 9. Ce langage provient donc du NIP entre par le client.
τ' : Suite au chiffrement, l'information est transmise jusqu'à la banque central sous forme de code binaire de 64 bits ou par bloc de 8 octets. Le langage de **τ'** provient donc de l'algorithme de chiffrement DES.

c) 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.

Une première attaque à laquelle le système est vulnérable est l'attaque de **force brute**. En effet, le principe de ce dernier est d'essayer toutes les possibilités de clés tandis que la norme DES offre seulement des clefs de 64 bits. Cette attaque est de type "texte chiffré seulement", cela signifie donc que seulement l'accès au texte chiffré est disponible. Par conséquent, il serait très facile de de trouver la clé par simple force brute avec la puissance computationnelle de nos jours et cela constitue d'une attaque potentielle selon les accès donnés dans la mise en situation.

Une deuxième attaque à laquelle le système est vulnérable est l'attaque de type "texte connu". Cette dernière stipule que l'attaquant a potentiellement accès au texte en clair et au texte chiffré. Le but est donc de trouver la clé pour pouvoir déchiffrer les futurs messages. Par conséquent, c'est une attaque potentielle selon les accès donnés dans la mise en situation parce que l'attaquant a accès au message non chiffré et chiffre.

Une troisième attaque à laquelle le système est vulnérable est l'attaque de type "texte choisir". Cette dernière stipule que l'attaquant a la capacité de choisir le texte en clair en plus du texte chiffré. Par conséquent, c'est une attaque potentielle selon les accès donnés dans la

mise en situation parce que l'attaquant a accès aux informations chiffrées et potentiellement le clavier (NIP) du GAB pour pouvoir déduire la clé.

d) Pour chacune des attaques identifiées au c), montrez à l'aide de traces d'exécution comment vous les effectueriez. Pour cela, utilisez les scripts transBase et recepBase qui implémentent respectivement les blocs source+codeur+chiffrement et déchiffrement+décodeur+récepteur.

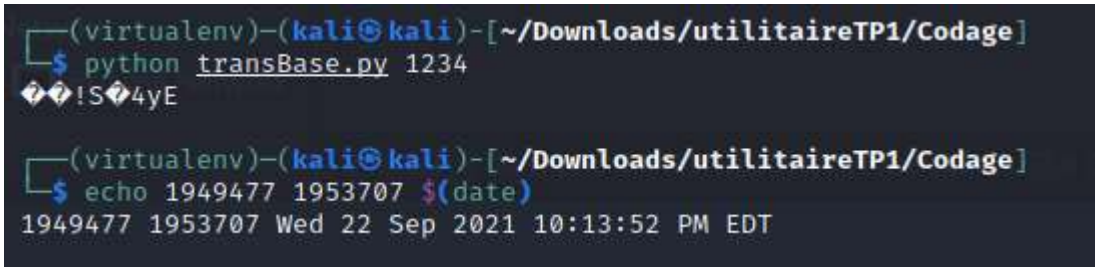


Figure 10. Affichage du nip chiffré

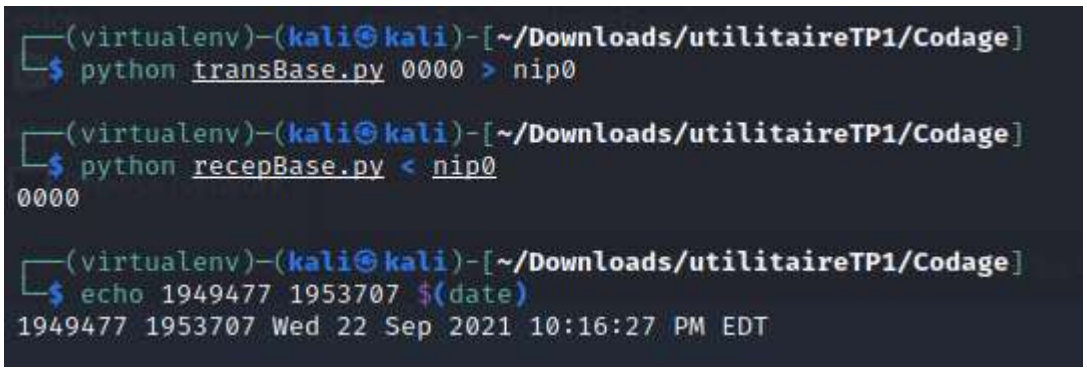


Figure 11. Encodage et décodage du nip

Figure 12. Vue hexadécimal du nip chiffré

Figure 13. Exécution codage 1

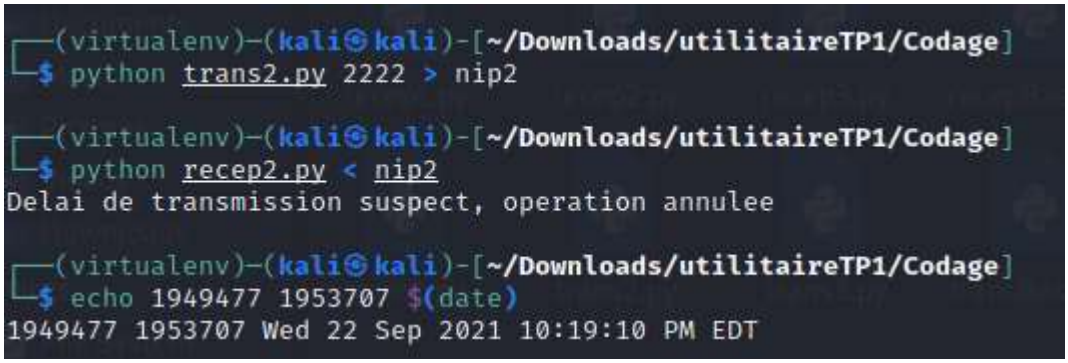


Figure 14. Exécution codage 2

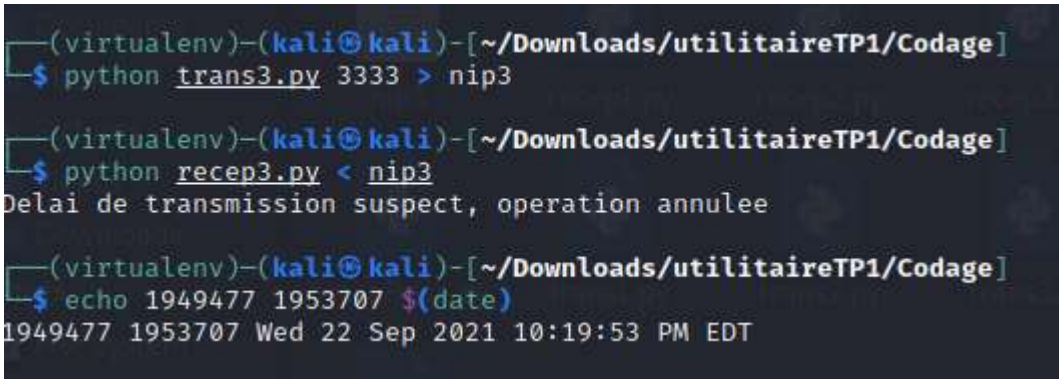


Figure 15. Exécution codage 3

Cas où l'on change de codage :

e) Pour chacun des trois codages, dites quelles attaques du c) permettent de bloquer en justifiant votre démarche.

Pour le codage 1, le décodage est beaucoup plus compliqué à la suite de l'ajout de 48 aléatoires. En effet, ce codage offre 2^{48}possibilités, ce qui augmente exponentiellement le nombre de clés à tester. Par conséquent, puisque les 3 types d'attaques énumérés en c) reposent sur le déchiffrement des clés, cet encodage offre une meilleure protection contre les 3 types d'attaques.

Pour le codage 2, nous avons beaucoup moins de bits aléatoires ce qui diminue le nombre de clés possibles a 2^{14}. Cependant, nous avons un ajout d'un timestamp a la fin. Le timestamp aide, entre autres, à la banque de valider si l'information est trop vieille ou non. Par conséquent, cela donne beaucoup moins de temps à l'attaquant pour décoder la clé. Cela dit, il est nécessaire de prendre en compte également la durée du timestamp. Puisque le temps de décodage pour l'attaquant est un facteur, cette méthode aide, en effet, à bloquer ou de rendre difficile l'attaque par force brute en c) puisque l'attaquant a moins de temps pour essayer toutes les combinaisons possibles.

Pour le codage 3, nous avons l'ancien NIP du client code sur 14 bits plus un timestamp comme pour le codage 2. L'ancien NIP + 2 bits de parité est un très mauvais choix de codage puisque ce dernier enlève tout aspect aléatoire au système. Cela réduit donc beaucoup l'entropie du codage car il y a des risques où le NIP précédent est déjà connu. Malgré que le

timestamp ajoute un niveau de protection comme pour le codage 2, ce dernier est beaucoup moins sécuritaire que les 2 premiers codages. Nous pouvons donc dire que les 3 attaques en c) ont une plus grande chance de trouver la clé pour la méthode 3.

f) Selon vous quel est le meilleur codage ? Pourquoi ?

Le meilleur codage selon nous est, sans aucun doute, le codage 1 puisque ce dernier offre une possibilité de clé beaucoup plus grande que le codage 2 et 3. (2^{48}) Cela augmente beaucoup l'entropie du système et il est beaucoup plus grande que l'entropie des 2 autres codages. De plus, sans mentionner que le codage 3 est le pire des 3 codages puisque la réutilisation d'une information partiellement connue d'avance est la pire décision à faire, l'utilisation du timestamp du codage 2 est une bonne solution contre l'attaque a force brute. Par contre, cela n'élimine pas le fait qu'il y a une possibilité ou l'attaquant réussit à décoder avant que le timestamp s'expire. De plus, ce dernier dépend aussi de la durée du timestamp. Donc en considérant tous ces aspects, nous trouvons que le meilleur codage est la première.

Question 2 - Certificats à clé publique, HTTPS et SSL

a) Quelle est la différence entre le protocole http et https dans l'url ?

La différence entre http (Hypertext Transfer Protocol) et https (Hypertext Transfer Protocol Secure) est que le protocole https utilise TLS ou SSL pour crypter les requêtes http normale. En effet, les requêtes envoyées par le protocole http sont des textes clairs qui peuvent être interceptés par un attaquant quelconque. Par conséquent, ce dernier a accès au message et à son contenu. Pour remédier à cela, le protocole https permet d'encoder le message initial et le seul contenu que l'attaquant obtiendra serait une série de lettres aléatoires. [3]

b) Qu'est-ce qu'un certificat à clé publique ? A quoi sert-il ?

Un certificat à clé publique est principalement utilisé pour authentifier et identifier une personne physique ou pour chiffrer des échanges. De plus, les clés publiques qui doivent être partagées sont souvent trop longues pour s'en souvenir. Par conséquent, elles sont conservées au sein d'un certificat numérique, ce qui permet de les transporter et de les partager de façon sécurisée. [4][5]

c) Dans un tableau, énumérez les principaux champs d'un certificat à clé publique. Pour chacun des champs, donnez la valeur correspondante du certificat de la Caisse Desjardins.

Tableau 5. Champs principaux du certificat à clé publique de la Caisse Desjardins

Champs	Valeur
Version	V3
Serial Number	7df71e2198beda6169ebad0406d06d0a
Signature algorithm	sha256RSA
Signature hash algorithm	sha256
Issuer	CN = Entrust Certification Authority - L1K OU = (c) 2012 Entrust, Inc. - for authorized use only OU = See www.entrust.net/legal-terms O = Entrust, Inc. C = US
Valid from	Monday, August 16, 2021 3:54:28 PM
Valid to	Tuesday, August 16, 2022 3:54:28 PM
Subject	CN = www.desjardins.com O = Mouvement Desjardins L = Montreal S = Quebec C = CA
Public Key	30 82 01 0a 02 82 01 01 00 d6 83 53 45 ea bf 8b f6 22 99 ae e5 b8 15 fc 50 8a b0 0b b2 6b 39 eb 98 e4 d3 c5 e8 f5 2d 2d 5b b1 d4 52 81 61 ae a5 85 75 e2 5d cf 34 14 35 be ca 76 27 78 05 36 36 56 b2 43 1d b1 a4 60 be fd 39 3c 39 35 25 10 1d dc 6f 42 f2 2c 39 db 37 f7 2b 3b a0 01 7c fb ee 38 5e ca ba 0f 50 7c 7a 85 af f3 bf a0 2d 59 11 66 49 88 b8 fd 6a e8 e1 b8 6d 0e 59 a1 15 a9 73 28 63 8f 90 7f 8c e0 9b 79 bc ba 36 f2 a9 e2 fa 4e 72 2b 99 0f cb db 8a 9d 01 93 ef b8 8f ed 80 47 59 8e 2c a1 71 fd 80 b6 8d 17 fc ce 08 30 97 3c 4c c2 d9 dd 95 fa 5a c8 8c 59 65 53 25 26 fc 95 22 20 0f 57 b2 45 36 bf 33 d4 d5 81 25 53 1c 99 ec 77 09 0b d2 1a 6f e3 60 72 37 e3 7d 27 4a 7c 2a d9 3b 80 a1 b9 6e 6b 97 1e 7b f8 c3 78 83 49 61 da 63 8f b5 27 78 bf 63 e8 e3 20 8c 41 e3 10 18 f1 62 c0 24 dd 75 87 02 03 01 00 01
Public Key parameters	05 00

Subject Key Identifier	402e052ef4bec183f1114bb53acf3ff533e97286
Authority Key Identifier	KeyID=82a27074ddbc533fcf7bd4f7cd7fa760c60a4cbf
Authority Information Access	[1] Authority Info Access Access Method=On-line Certificate Status Protocol (1.3.6.1.5.5.7.48.1) Alternative Name: URL=http://ocsp.entrust.net [2] Authority Info Access Access Method=Certification Authority Issuer (1.3.6.1.5.5.7.48.2) Alternative Name: URL=http://aia.entrust.net/l1k-chain256.cer
CRL Distribution Points	[1] CRL Distribution Point Distribution Point Name: Full Name: URL=http://crl.entrust.net/level1k.crl
Subject Alternative Name	DNS Name=www.desjardins.com DNS Name=desjardins.com DNS Name=www.mfmc.formateurs.desjardins.com DNS Name=www.mfmc.desjardins.com DNS Name=www.mfmc-admin.desjardins.com DNS Name=www.lacabanedesjardins.fr DNS Name=www.labfinance.desjardins.com DNS Name=www.jeunesse.desjardins.com DNS Name=www.financing.desjardins.com DNS Name=www.desjardinsbank.com DNS Name=www.creditplus.desjardins.com DNS Name=www.coastcapital.desjardins.com DNS Name=www.accordd.desjardins.com DNS Name=tableau.mouvement.desjardins.com DNS Name=static.mouv.desjardins.com DNS Name=static.mouv.desjardins.ca DNS Name=static.mouv.acadie.com DNS Name=static.desjardins.com DNS Name=prod-author-aem-www.desjardins.com DNS Name=mobile.desjardins.com DNS Name=mfmc.formateurs.desjardins.com DNS Name=mfmc.desjardins.com

	DNS Name=mfmc-admin.desjardins.com DNS Name=m.desjardins.com DNS Name=lacabanedesjardins.fr DNS Name=labfinance.desjardins.com DNS Name=jeunesse.desjardins.com DNS Name=financing.desjardins.com DNS Name=dfs-invest.mouv.desjardins.com DNS Name=creditplus.desjardins.com DNS Name=coop.desjardins.com DNS Name=cobrowse.desjardins.com DNS Name=accweb.mouv.desjardins.com DNS Name=accweb.mouv.desjardins.ca DNS Name=accweb.mouv.acadie.com DNS Name=accordd.desjardins.com DNS Name=accesdc.mouv.desjardins.com DNS Name=accesdc.mouv.desjardins.ca DNS Name=accesdc.mouv.acadie.com DNS Name=accesdc-static.mouv.desjardins.com DNS Name=accesdc-static.mouv.desjardins.ca DNS Name=accesdc-static.mouv.acadie.com DNS Name=accesd.mouv.desjardins.com DNS Name=accesd.mouv.desjardins.ca DNS Name=accesd.mouv.acadie.com
Enhanced Key Usage	Server Authentication (1.3.6.1.5.5.7.3.1) Client Authentication (1.3.6.1.5.5.7.3.2)
Certificate Policies	[1] Certificate Policy : Policy Identifier=2.16.840.1.114028.10.1.5 [1,1] Policy Qualifier Info : Policy Qualifier Id=CPS Qualifier : https://www.entrust.net/rpa [2] Certificate Policy : Policy Identifier=2.23.140.1.2.2
SCT List	v1 5614069a2fd7c2ecd3f5e1bd44b23ec74676b9bc99115cc0ef949855d689d0dd Monday, August 16, 2021 3:54:29 PM SHA256 ECDSA 3045022066ac367e0efeec52c8dfa27a8030d58b7a66f5c2d7b8fd908a75cbda61c9f11402210090bfc7a92504f45e05ef2b97e5b7db7a0ff2f6163ce599b735ecd3e5e2497f3a v1

	dfa55eab68824f1f6cadeeb85f4e3e5aeacda212a46a5e8e3b12c020445c2a73 Monday, August 16, 2021 3:54:29 PM SHA256 ECDSA 30460221008669c14af4772085694cb5b529241ecb160ff4fb74585f7ec7336d914b52e4cf022100e7bf23628c6eb00317bd5bc1935c0d937d16714c6d669622422d28f339be8de7 v1 46a555eb75fa912030b5a28969f4f37d112c4174befd49b885abf2fc70fe6d47 Monday, August 16, 2021 3:54:29 PM SHA256 ECDSA 304402201173ea62170d8bc449f136969bcfaaaef41c7823228429d650ca0597494c7f5702204184e9b6513d6e8a8d25eb65e6a54a28854e4a8857919418cbfc2c05d1d2d890
Basic Constraints	Subject Type=End Entity Path Length Constraint=None
Key usage	Digital Signature, Key Encipherment (a0)
Thumbprint	274935a34e31312fb2df8a74466453213cb4cb96

d) Comment votre navigateur vérifie-t-il l'identité du propriétaire du site que vous avez visité ?

Le navigateur vérifie l'identité du propriétaire du site que nous visitons en consultant une liste de *Certificate Authorities* propre à chaque navigateur ou système d'exploitation. Ce processus fait partie du *SSL encryption process* et vérifie la sécurité de l'utilisateur. [6]

e) Qu'est-ce qu'un Certificate Authority (CA) ? A quoi sert-il ? Pourquoi observe-t-on deux CA lorsqu'on examine dans le navigateur le certificat de la Caisse Desjardins. Citez-les.

Une autorité de certification est "un tiers de confiance permettant d'authentifier l'identité des correspondants. Une autorité de certification délivre des certificats décrivant des identités numériques et met à disposition les moyens de vérifier la validité des certificats qu'elle a fournis." [7] La caisse Desjardins a deux certificats car le premier est un *Root Certificate* tandis que le deuxième est un *Intermediate Certificate*. Le *Root Certificate* appartient à l'autorité de certification et ce dernier est très important car toute certification signée avec sa clé privée serait automatiquement approuvée par le navigateur. Le *Intermediate Certificate* est une sous-certification du

Root Certificate et permet d'utiliser la clé privée pour produire des certificats SSL. Dans le cas de Desjardins, le *Intermediate Certificate* est attribué au site de desjardins.com. [8]

Figure 16. *Root Certificate* de https://desjardins.com

Figure 17. *Intermediate Certificate* de https://desjardins.com

f) Est-il risqué d'accepter dans votre navigateur un CA ? Pourquoi ?

Il y a effectivement un risque d'accepter dans le navigateur un CA car il y a des risques que des certifications soient fausses. En effet, il existe des méthodes pour obtenir un certificat authentique sans posséder des habiletés techniques poussées. L'acteur peut parfois étudier en profondeur une cible et passer pour eux lors de la vente du certificat. Une fois le certificat certifié en main, les *Intermediate Certificate* seront automatiquement approuvés par les navigateurs. Par conséquent, cela pose des risques de sécurité car ces derniers peuvent par exemple propager des fichiers malicieux etc. [9]

g) Quelle est la différence entre un certificat auto-signé et un certificat TLS ? Pourquoi ne faut-il pas faire confiance à un site web dont le certificat est auto-signé ?

Un certificat auto-signé est un certificat qui n'est pas signé par une autorité de certification. Ces derniers ne coûtent pas d'argent pour les produire et servent entre autres pour l'utilisation des sites internes et des environnements de tests. [10] En contraste, un certificat TLS est publié par une autorité de certification et ce dernier est utilisé pour les sites publiques avec des utilisateurs anonymes. Comme mentionné précédemment, quiconque qui possède ce certificat signifie que cette organisation est bel et bien identifiée par l'organisation et il est digne de confiance. Il ne faut pas faire confiance à un site web dont le certificat est auto-signé puisque aucune autorité ne vérifie la fiabilité de la source ou de la compagnie. Il est donc impossible d'en vérifier l'intégrité.

Question 3 - Chiffrement par bloc et modes d'opération

a) 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, chiffrez ce fichier en mode ECB. (Exécutez le script sans argument pour connaître son fonctionnement). Observez le fichier de sortie et commentez.

Figure 18. Manipulation avec AES.py avec ECB

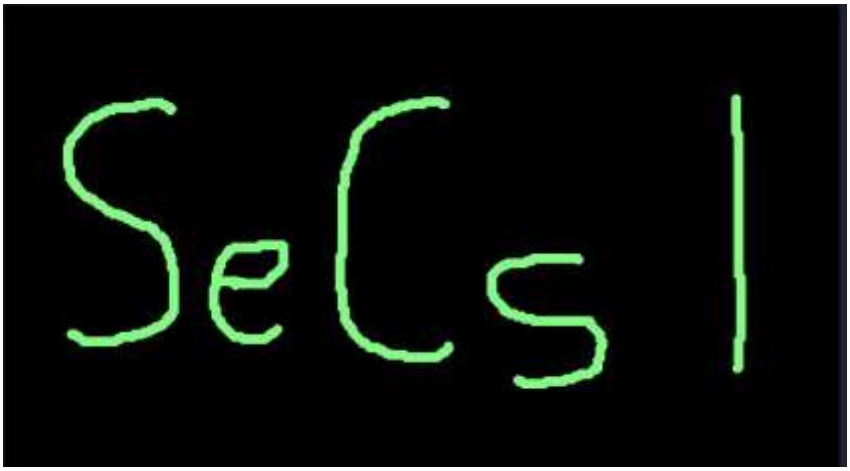


Figure 19. Image initiale

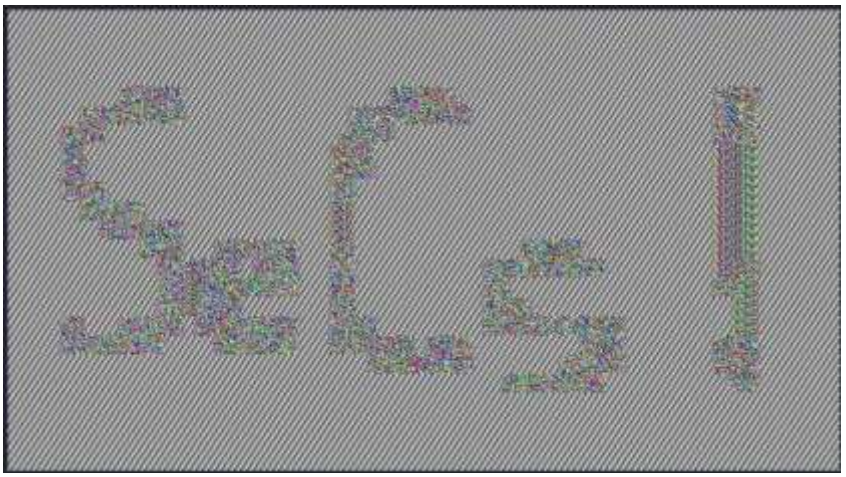


Figure 20. Image obtenue avec ECB

Comme nous pouvons l'observer, nous pouvons clairement distinguer le dessin initial à la suite des manipulations faites par le script.

b) Chiffrez maintenant le fichier en mode CBC. Observez le fichier puis commentez.

```
┌──(virtualenv)─(kali㉿kali)-[~/Downloads/utilitaireTP1/ChiffrementBLOC]
└─$ python AES.py -i mdp.jpg -m CBC -o resultCBC.jpg

┌──(virtualenv)─(kali㉿kali)-[~/Downloads/utilitaireTP1/ChiffrementBLOC]
└─$ echo 1949477 1953707 $(date)
1949477 1953707 Wed 22 Sep 2021 10:30:24 PM EDT
```

Figure 21. Manipulation avec AES.py avec CBC

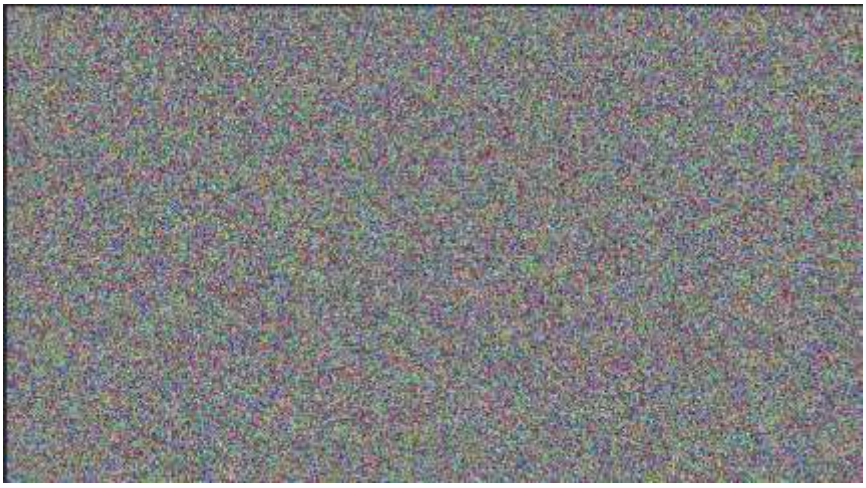


Figure 22. Image obtenue suite au chiffrement CBC

Comme nous pouvons l'observer, nous ne pouvons rien distinguer du dessin initial. Nous pouvons donc affirmer que le chiffrement CBC est plus sécurisé que ECB.

c) Concluez sur l'importance des modes d'opération des algorithmes de chiffrement par bloc.

Comme nous pouvons observer au b) et c), il est très important de bien choisir les méthodes de chiffrement car les résultats obtenus peuvent ne pas être aussi performant que nous le souhaiterions. En effet, dans le cas du dessin, nous pouvons affirmer que l'utilisation de la méthode ECB n'était pas un choix intelligent car ce dernier emploie la même clé pour chiffrer les blocs. Par conséquent, il est très possible de retracer le message avec un message similaire. Par contraste, le mode CBC est beaucoup plus sécuritaire car ce dernier emploie la méthode XOR avec cryptogramme antérieur. De plus, il utilise un vecteur d'initiation comme paramètre cryptographique. Le résultat est donc observable dans la figure 16.

Question 4 - Organisation des mots de passe en UNIX/Linux

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

```
┌──(kali㉿kali)-[/etc]
└─$ cat 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
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:101:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nolo
gin
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
mysql:x:104:110:MySQL Server,,,:/nonexistent:/bin/false
tss:x:105:111:TPM software stack,,,:/var/lib/tpm:/bin/false
strongswan:x:106:65534::/var/lib/strongswan:/usr/sbin/nologin
ntp:x:107:112::/nonexistent:/usr/sbin/nologin
messagebus:x:108:113::/nonexistent:/usr/sbin/nologin
redsocks:x:109:114::/var/run/redsocks:/usr/sbin/nologin
rwhod:x:110:65534::/var/spool/rwho:/usr/sbin/nologin
iodine:x:111:65534::/run/iodine:/usr/sbin/nologin
miredo:x:112:65534::/var/run/miredo:/usr/sbin/nologin
_rpc:x:113:65534::/run/rpcbind:/usr/sbin/nologin
usbmux:x:114:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
tcpdump:x:115:121::/nonexistent:/usr/sbin/nologin
rtkit:x:116:122:RealtimeKit,,,:/proc:/usr/sbin/nologin
sshd:x:117:65534::/run/sshd:/usr/sbin/nologin
statd:x:118:65534::/var/lib/nfs:/usr/sbin/nologin
postgres:x:119:124:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
avahi:x:120:126:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
stunnel4:x:121:127::/var/run/stunnel4:/usr/sbin/nologin
Debian-snmp:x:122:128::/var/lib/snmp:/bin/false
speech-dispatcher:x:123:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
sslh:x:124:129::/nonexistent:/usr/sbin/nologin
nm-openvpn:x:125:130:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nolog
```

Figure 23. L'affichage du fichier *etc/passwd*

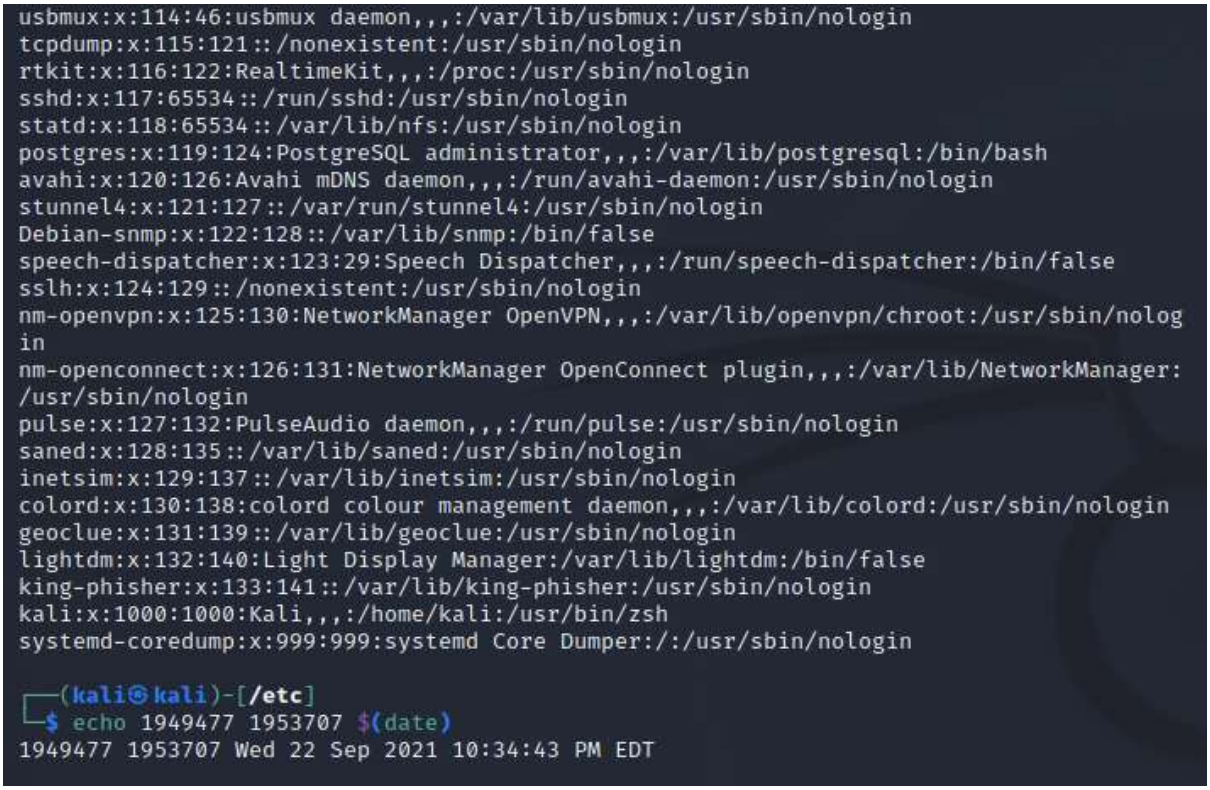


Figure 24. L'affichage du fichier *etc/passwd* (suite)

Ce fichier ne contient pas de mot de passe. Il est évident de remarquer cela car chaque ":x" représente un mode de passe caché. La raison de ce dernier est que ce fichier est accessible à tout le monde et non seulement à l'administrateur *sudo*. Par conséquent, il est logique que les mots de passe soient cachés.

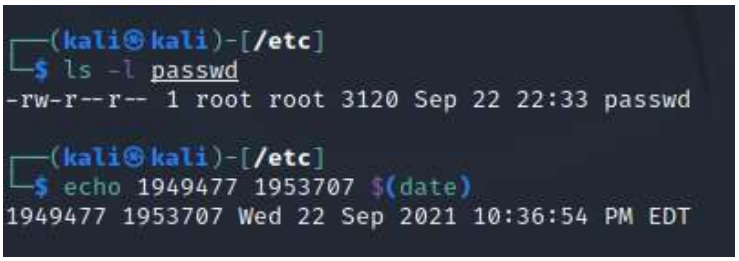


Figure 25. Permissions du fichier *etc/passwd*

Avec la commande *ls -l,* nous pouvons observer les permissions d'accès du fichier. Nous pouvons observer le *root*.

b) Observez les fichiers passwd et shadow qui se trouvent sous le répertoire */etc/.*
Ajoutez un utilisateur avec la commande : *$ useradd –g users -s/bin/bash -m NOM* avec *NOM*= le nom de l'utilisateur que vous ajoutez
Donnez un password à l'utilisateur que vous avez créé avec la commande :
$ passwd NOM

Observez ce qui se passe dans les fichiers *passwd* et *shadow*. Lequel ou lesquels de ces deux fichiers sont modifiés ? Pourquoi ?

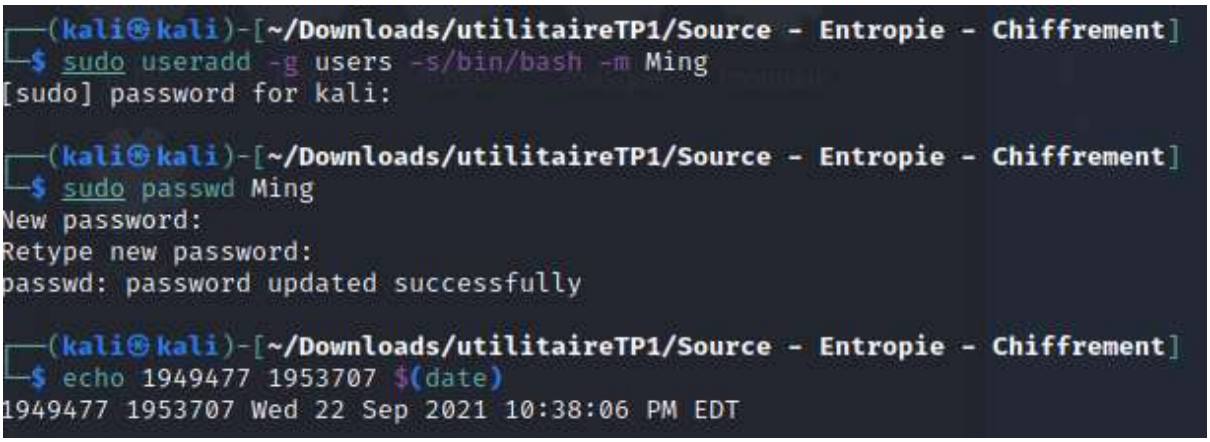


Figure 26. Création d'un nouvel usager et initiation de son mot de passe

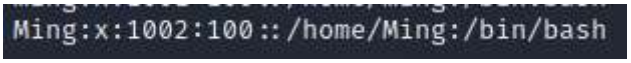


Figure 27. Mot de passe dans *etc/passwd*

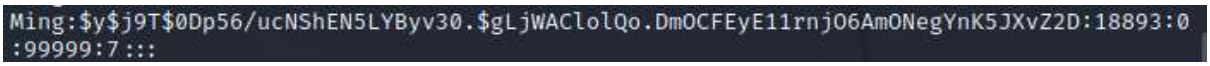


Figure 28. Mot de passe dans *etc/shadow*

Comme nous pouvons observer, les fichier *etc/passwd* et *etc/shadow* sont les deux modifiés. En effet, le fichier *passwd* contient le nom d'utilisateur *Ming* avec le mot de passe caché avec un ":x" tandis que le fichier *shadow* contient le mot de passe hashé.

c) Changez le mot de passe de l'utilisateur que vous venez de créer avec la commande '*$ passwd NOM'.* 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 ?

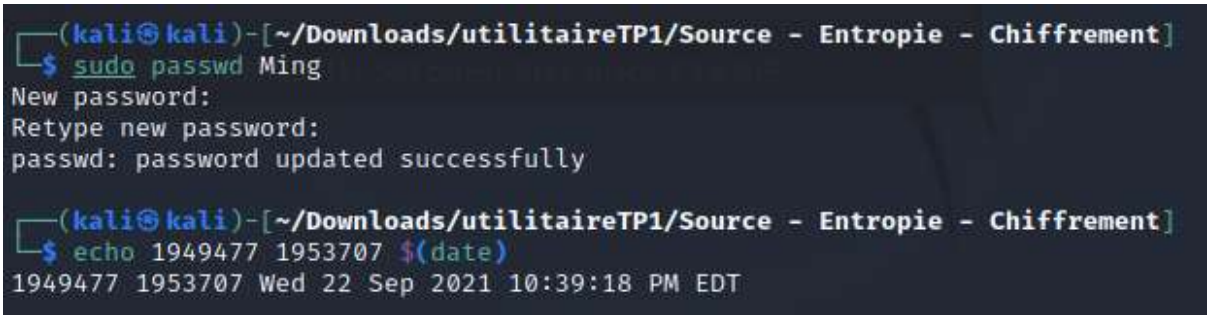


Figure 29. Changement de mot de passe pour l'utilisateur Ming

Figure 30. Mot de passe dans *etc/passwd*

Figure 31. Mot de passe dans *etc/shadow*

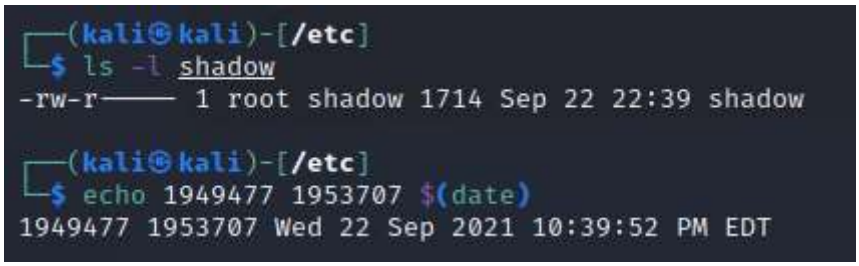


Figure 32. Permission du fichier *etc/shadow*

Comme nous pouvons l'observer, le mot de passe dans *etc/passwd* n'est pas changé tandis que le hash dans *etc/shadow* a changé. Il est logique d'assumer cela puisque nous avons bel et bien changé le mot de passe. Par conséquent, le hash devrait changer également.

Nous pouvons également observer que les permissions pour *etc/shadow* est de *read* et *write* seulement pour le *root*. Il est logique puisque nous ne voulons pas n'importe qui vienne modifier les mots de passe des utilisateurs.

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

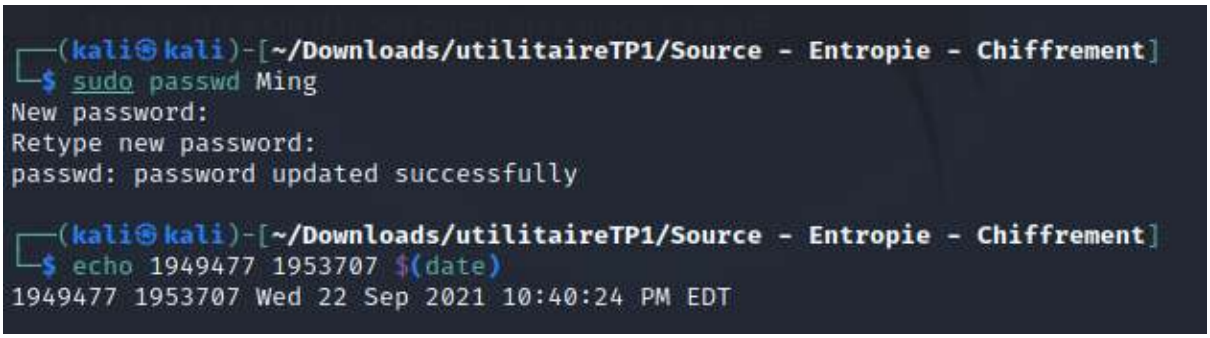


Figure 33. Mise à jour avec le même mot de passe

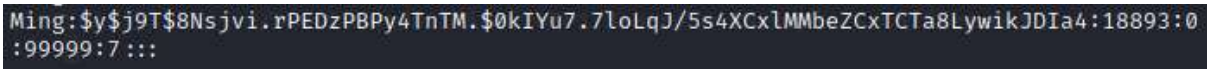


Figure 34. Mot de passe hashé *etc/shadow*

Comme nous pouvons l'observer, la valeur de hash dans *shadow* a changé. Cela peut s'expliquer par le fait que chaque mot de passe, que ce soit 2 mêmes chaînes de caractères ou pas, génère 2 *salt* différents. Vu que le hash final généré dépend du *salt,* les hash sont différents aussi.

e) 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 ?

```
┌──(kali㉿kali)-[~/Downloads/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ sudo useradd -g users -s/bin/bash -m Mingg

┌──(kali㉿kali)-[~/Downloads/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ kali
Command 'kali' not found, but can be installed with:
sudo apt install kali
Do you want to install it? (N/y)N

┌──(kali㉿kali)-[~/Downloads/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ sudo passwd Mingg
New password:
Retype new password:
passwd: password updated successfully

┌──(kali㉿kali)-[~/Downloads/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ echo 1949477 1953707 $(date)
1949477 1953707 Wed 22 Sep 2021 10:41:38 PM EDT
```

Figure 35. Création du nouvel utilisateur et initiation de son mot de passe

```
Ming:$y$j9T$8Nsjvi.rPEDzPBPy4TnTM.$0kIYu7.7loLqJ/5s4XCxlMMbeZCxTCTa8LywikJDIa4:18893:0:99999:7:::
Mingg:$y$j9T$8Nsjvi.rPEDzPBPy4TnTM.$0kIYu7.7loLqJ/5s4XCxlMMbeZCxTCTa8LywikJDIa4:18893:0:99999:7:::
```

Figure 36. Mise à jour du même mot de passe du nouvel utilisateur avec l'ancien mot de passe

Nous pouvons remarquer que c'est effectivement possible d'utiliser le même mot de passe pour se connecter avec 2 comptes différents. Cependant, il n'est pas recommandé de faire cela car il est possible d'accéder au deuxième compte si le mot de passe du premier compte est connu.

f) Effacez cet utilisateur avec la commande '*$ userdel –r NOM'.* Qu'est-ce qui se passe dans *passwd* et *shadow* ?

```
┌──(kali㉿kali)-[~/Downloads/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ sudo userdel -r Mingg
userdel: Mingg mail spool (/var/mail/Mingg) not found

┌──(kali㉿kali)-[~/Downloads/utilitaireTP1/Source - Entropie - Chiffrement]
└─$ echo 1949477 1953707 $(date)
1949477 1953707 Wed 22 Sep 2021 10:42:11 PM EDT
```

Figure 37. Suppression d'un utilisateur

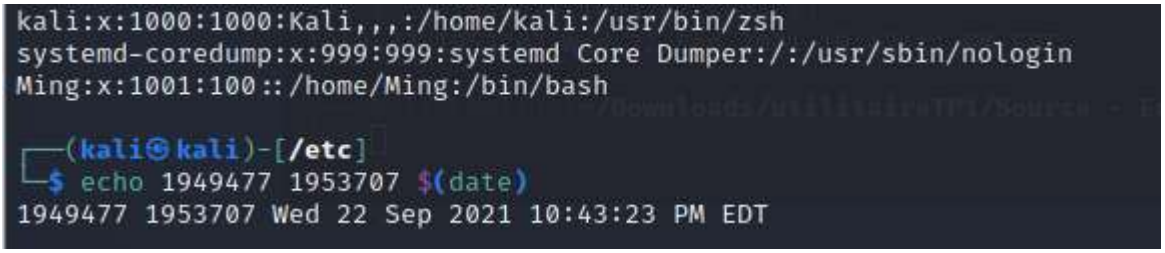


Figure 38. Suppression du mot de passe dans *etc/passwd*

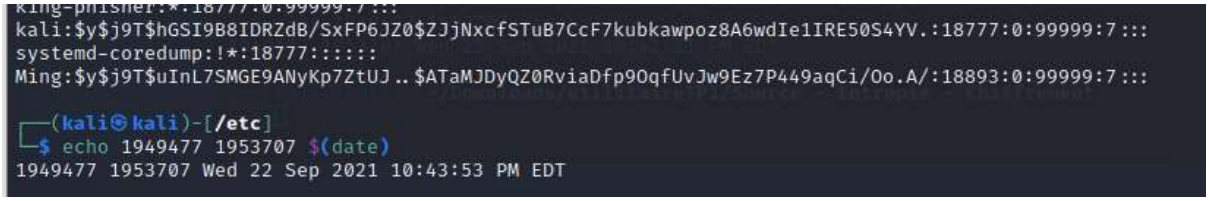


Figure 39. Suppression du mot de passe dans *etc/shadow*

Comme nous pouvons le remarquer, l'utilisateur *Mingg* a effectivement disparu du fichier *etc/shadow* et du fichier *etc/passwd*.

Question 5 - Contrôle de qualité de choix de mot de passe

a) Utiliser john the ripper avec le dictionnaire rockyou, et identifier le mot de passe correspondant à chaque utilisateur. Utilisez la commande suivante pour connaître l'emplacement # Locate rockyou

```
┌──(kali㉿kali)-[~/Downloads/utilitaireTP1]
└─$ john passwd file                                                    1 ×
Using default input encoding: UTF-8
Loaded 4 password hashes with 4 different salts (sha512crypt, crypt(3) $6$ [SHA512 256
/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 7 candidates buffered for the current salt, minimum 8 needed for perform
ance.
Warning: Only 1 candidate buffered for the current salt, minimum 8 needed for performa
nce.
Warning: Only 5 candidates buffered for the current salt, minimum 8 needed for perform
ance.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for perform
ance.
Warning: Only 5 candidates buffered for the current salt, minimum 8 needed for perform
ance.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for perform
ance.
Warning: Only 2 candidates buffered for the current salt, minimum 8 needed for perform
ance.
Warning: Only 5 candidates buffered for the current salt, minimum 8 needed for perform
ance.
Warning: Only 3 candidates buffered for the current salt, minimum 8 needed for perform
ance.
Warning: Only 5 candidates buffered for the current salt, minimum 8 needed for perform
ance.
Further messages of this type will be suppressed.
To see less of these warnings, enable 'RelaxKPCWarningCheck' in john.conf
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
123456789        (simple)
sunshine         (brian)
monkey           (action)
liverpool        (vladimir)
4g 0:00:00:04 DONE 2/3 (2021-09-22 22:53) 0.9456g/s 3064p/s 3247c/s 3247C/s bigdog..ra
ndom
Use the "--show" option to display all of the cracked passwords reliably
Session completed

┌──(kali㉿kali)-[~/Downloads/utilitaireTP1]
└─$ echo 1949477 1953707 $(date)
1949477 1953707 Wed 22 Sep 2021 10:54:00 PM EDT
```

Figure 40. Utilisation du script *John* avec le dictionnaire *rockyou*.

b) Calculez l'entropie maximale pour les alphabets suivants :

La formule pour le calcul de l'entropie est $H(S) = \sum_{i=1}^{n} P_i log_2(\frac{1}{P_i})$.

i) •[a-zA-Z]

Pour les lettres de A à Z en majuscule et en minuscule, nous avons 2 fois 26 lettres. Par conséquent, nous avons 52 caractères distincts donc $P_i = \frac{1}{52}$.

Nous obtenons donc $H(S) = \sum_{i=1}^{52} \frac{1}{52} log_2(52) = 5.7004\ bits$.

ii) •[a-zA-Z0-9]

Pour les lettres de A à Z en majuscule et en minuscule, nous avons 2 fois 26 lettres plus 0 à 9. Par conséquent, nous avons 62 caractères distincts donc$P_i = \frac{1}{62}$.

Nous obtenons donc $H(S) = \sum_{i=1}^{62} \frac{1}{62} log_2(62) = 5.954\ bits$.

iii) •L'ensemble de la table ascii

Pour l'ensemble de la table ascii, nous avons 256 caractères distincts. Nous avons donc $P_i = \frac{1}{256}$.

Nous obtenons donc $H(S) = \sum_{i=1}^{256} \frac{1}{256} log_2(256) = 8\ bits$.

c) Déduisez des résultats de b) un critère important pour qu'un mot de passe soit fort

À partir des résultats en b), nous pouvons remarquer que l'ajout des chiffres de 0 à 9 n'augmente pas significativement l'entropie. Cela signifie qu'il n'est pas nécessairement beaucoup plus difficile de décoder un texte à 62 caractères uniques que 52. Cependant, nous observons qu'un ensemble de 256 caractères distincts (table ascii) augmente drastiquement l'entropie. Par conséquent, son utilisation serait beaucoup plus favorisée pour un système sécuritaire.

d) Au vu des résultats de John the Ripper, donner 3 autres critères pour qu'un mot de passe soit fort.

Le premier critère serait de ne pas utiliser des chiffres (0-9) de telle façon qu'ils suivent tels que 12345. Cela facilitera beaucoup les algorithmes de les décoder. Similairement, le deuxième critère serait d'employer des vocabulaires ou des mots dans notre quotidien. Les algorithmes de décodage de nos jours sont assez puissants pour les deviner. Finalement, le dernier critère serait d'employer des symboles spéciaux. En effet, dans la capture d'écran en a), aucun caractère spécial n'a été utilisé. Il serait intéressant de les combiner avec les lettres et chiffres pour obtenir un mot de passe avec la plus grande entropie possible.

e) A votre avis pourquoi il est conseillé de ne pas utiliser le même mot de passe partout.

Il est conseillé de ne pas utiliser le même mot de passe partout car dès que ce dernier a été découvert. L'attaquant peut très possiblement aller essayer ce mot de passe dans les autres réseaux sociaux ou bien le compte bancaire. Avoir une multitude de mot de passe permet donc d'offrir une autre couche de protection contre les attaques.

Partie C

Question 1 - Échec du protocole RSA

a) Décrire comment Ève peut facilement déchiffrer ce message.

Puisque que Ève possède la clé publique, chacune des valeurs numérique associée à une lettre peut être chiffrée et calculer sa valeur. Un dictionnaire se bâtit donc en obtenant les valeurs chiffrées en fonction des lettres attribuées. Ève utilise $\mathbf{y = x^e \bmod N}$, où **x** est la lettre en valeur numérique correspondant à la lettre [0,25] et où **e** et **N** sont la paire de valeurs de la clé publique. La valeur chiffrée obtenue est **y**.

b) Récupérer le texte à déchiffrer et la clé publique dans le document INF4420A_TP1_Q1_A21_GX du site Moodle. Utilisez l'attaque décrite précédemment pour déchiffrer le texte, sans factoriser n, selon la clé. Donnez votre réponse en texte, pas en chiffres.

e = 599
N = 468649
Text chiffré = {101398, 0, 361816, 189459, 302636}

Clé publique = (e,N) = (599, 468649)

```
A:      Nombre: 0       Valeur chiffrée: 0
B:      Nombre: 1       Valeur chiffrée: 1
C:      Nombre: 2       Valeur chiffrée: 101439
D:      Nombre: 3       Valeur chiffrée: 160866
E:      Nombre: 4       Valeur chiffrée: 213277
F:      Nombre: 5       Valeur chiffrée: 245719
G:      Nombre: 6       Valeur chiffrée: 196643
H:      Nombre: 7       Valeur chiffrée: 179510
I:      Nombre: 8       Valeur chiffrée: 361816
J:      Nombre: 9       Valeur chiffrée: 9474
K:      Nombre: 10      Valeur chiffrée: 392576
L:      Nombre: 11      Valeur chiffrée: 352080
M:      Nombre: 12      Valeur chiffrée: 161890
N:      Nombre: 13      Valeur chiffrée: 189459
O:      Nombre: 14      Valeur chiffrée: 426644
P:      Nombre: 15      Valeur chiffrée: 101398
Q:      Nombre: 16      Valeur chiffrée: 6789
R:      Nombre: 17      Valeur chiffrée: 143329
S:      Nombre: 18      Valeur chiffrée: 302636
T:      Nombre: 19      Valeur chiffrée: 45807
U:      Nombre: 20      Valeur chiffrée: 5387
V:      Nombre: 21      Valeur chiffrée: 310227
W:      Nombre: 22      Valeur chiffrée: 308777
X:      Nombre: 23      Valeur chiffrée: 189420
Y:      Nombre: 24      Valeur chiffrée: 30101
Z:      Nombre: 25      Valeur chiffrée: 370344

Texte déchiffré: PAINS

1953707 1949477 2021-09-22
```

Figure 41. Programme trouvant le texte à partir du texte chiffré

Un programme python a été développé pour déchiffrer le message. Le code se trouve à l'annexe 1.

Le texte déchiffré est donc : **PAINS**

c) Si vous regardez attentivement la liste de textes à déchiffrer pour votre groupe de laboratoire, vous remarquez probablement des textes chiffrés avec des « 0 » ou des « 1 ». Quelles conclusions additionnelles pouvez-vous tirer sur le contenu des messages pour assurer le bon fonctionnement de RSA ?

Selon l'énoncé, les nombre 0 et 1 représentent respectivement les lettres 'A' et 'B'. En utilisant la formule de chiffrement, on remarque que les valeurs chiffrées restent les même que les nombres originaux :

A: 0^{599} mod 468649 = 0 mod 468649 = (468649 * 0) + 0 = 0
B: 1^{599} mod 468649 = 1 mod 468649 = (468649 * 0) + 1 = 1

C'est donc facile de détecter les lettres 'A' et 'B' même si elles passent par la formule de chiffrement. Ces lettres réduisent donc la sécurité de RSA et le fonctionnement n'est pas adéquat avec l'utilisation de 'A' et 'B'.

Question 2 - Déchiffrement "simple"

Notre texte à déchiffrer :
‘EFXSNSQCENSJNSOSZFBZKEJNBZHSESRUGNREBOLGZREFXSEUSOFSNZEJNQABZHSOEYTEFUSZFTECQPNEOSZGFQNOEOBIEFXSNSQCENSJNSOSZFBZKEVGZBFQYGEOBIEFXSNSQCENSJNSOSZFBZKEYNBFBOXEHQLPVYBGEOBIEFXSNSQCENSJNSOSZFBZKEOGODGFHXSU’

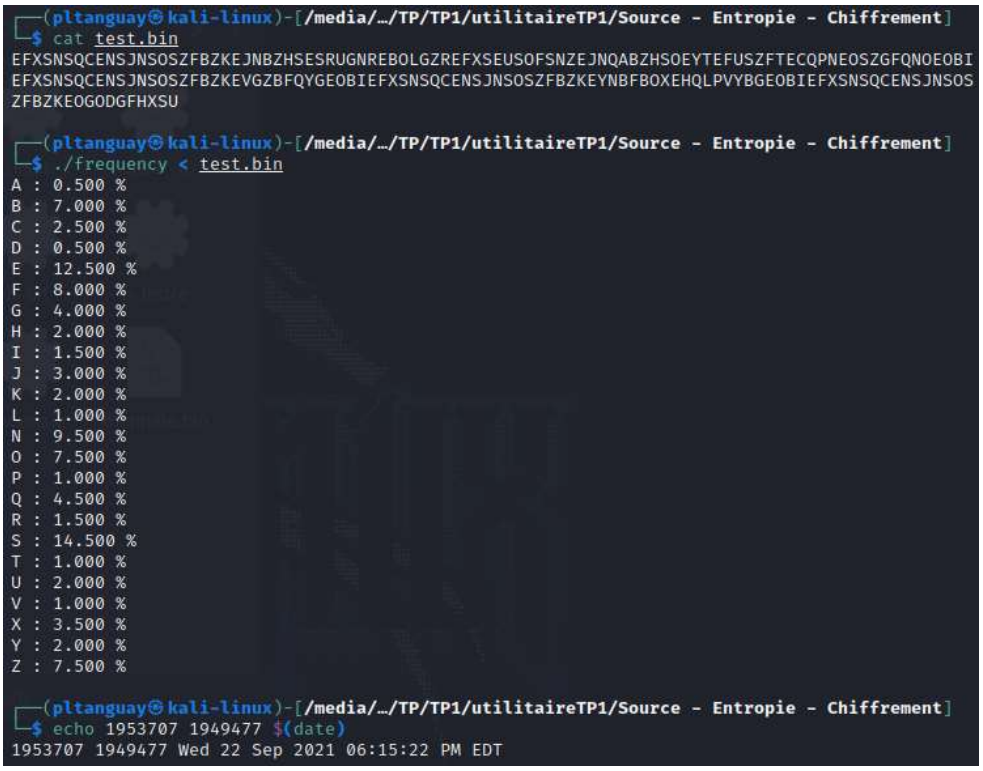


Figure 42. Analyse fréquentielle du texte à déchiffrer en utilisant 'frequency'

Et à partir de la figure 42, voici le tableau des lettres en ordre d’apparition :

Tableau 6 : Fréquence d'apparition des lettres du message chiffré

Lettres du texte chiffré	Fréquence (%)	**Lettres du texte chiffré**	Fréquence (%)	**Lettres du texte chiffré**	Fréquence (%)
S	14.5	**X**	3.5	**L**	1.0
E	12.5	**J**	3.0	**P**	1.0
N	9.5	**C**	2.5	**T**	1.0
F	8.0	**H**	2.0	**V**	1.0
O	7.5	**K**	2.0	**A**	0.5
Z	7.5	**U**	2.0	**D**	0.5
B	7.0	**Y**	2.0	**M**	0.0
Q	4.5	**I**	1.5	**W**	0.0
G	4.0	**R**	1.5		

Un programme python a été développé pour automatiser le remplacement des caractères, en tenant compte de ceux qui ont déjà été remplacés. Les caractères remplacés sont représentés en bleu. Le code se trouve à l'annexe 2.

Démarche :

1. Tout d'abord, nous savons qu'en anglais, 'espace' a une fréquence d'apparition 1.07% de plus que la lettre 'e'. Ce dernier est la lettre la plus utilisée en anglais. Nous soupçonnons donc que l'espace et 'e' correspondent aux deux lettres les plus fréquentes du tableau 6. En testant que 's' est l'espace ne nous donne pas des longueurs de mots raisonnables. Nous avons donc testé **E -> espace** et **S -> E** :

Figure 43. Texte partiellement déchiffré avec E -> espace et S -> E.

2. Ensuite, par déduction, nous voyons le trigramme FXE pourrait correspondre à 'THE', puisque ce dernier s'observe souvent en anglais et que nécessairement il finit par 'e'. Nous tentons donc **F -> T** et **X -> H** :

Figure 44. Texte partiellement déchiffré avec F -> T et X -> H.

3. Par la suite, nous voyons le trigramme ‘OBI’ qui pourrait être un mot à trois lettres qui apparaît souvent en anglais, dont AND, ARE, FOR, etc. Cependant, cette approche ne nous a pas avancé vers une solution. On voit cependant que le premier mot partiellement déchiffré ‘THENEQC’, par déduction, peut être soit ‘THEREBY’ ou ‘THEREOF’. On fait donc l’hypothèse que **N -> R**. Pour ‘QC’, par l’analyse des fréquences, on soupçonne qu’ils correspondent plus à ‘OF’ que ‘BY’. Nous essayons donc **Q -> O** et **C -> F**:

Figure 45. Texte partiellement déchiffré avec Q ->O et C -> F.

4. Maintenant, le deuxième mot, par déduction, ressemble à ‘REPRESENTING’. On tente donc de compléter ce mot avec J -> P, O -> S, Z -> N, B -> I et K -> G :

Figure 46. Texte partiellement déchiffré avec J -> P, O -> S, Z -> N, B -> I et K -> G.

5. Il est ensuite évident de voir que le trigramme original ‘OBI’ correspond finalement au mot ‘SIX’. On remplace donc I par X :

Figure 47. Texte partiellement déchiffré avec I -> X.

6. On voit ensuite que le troisième mot est ‘PRINCE’ et que le cinquième est ‘ISLAND’, et plus loin, ‘BRITISH’. On fait donc H -> C, Y -> B, L -> L, G -> A et R -> D :

Figure 48. Texte partiellement déchiffré avec H -> C, Y -> B, L -> L, G -> A et R -> D.

7. De plus, il est facile de voir les mots suivants apparaître : ‘EDWARD’, ‘WESTERN’, ‘PROVINCES’, ‘MANITOBA’, ‘BY’, ‘FOUR’. En remplaçant le restant des lettres, le texte complet est déchiffré :

Figure 49. Texte déchiffré au complet

Voici le message déchiffré :
‘ THEREOF REPRESENTING PRINCE EDWARD ISLAND THE WESTERN PROVINCES BY TWENTY FOUR SENATORS SIX THEREOF REPRESENTING MANITOBA SIX THEREOF REPRESENTING BRITISH COLUMBIA SIX THEREOF REPRESENTING SASKATCHEW’

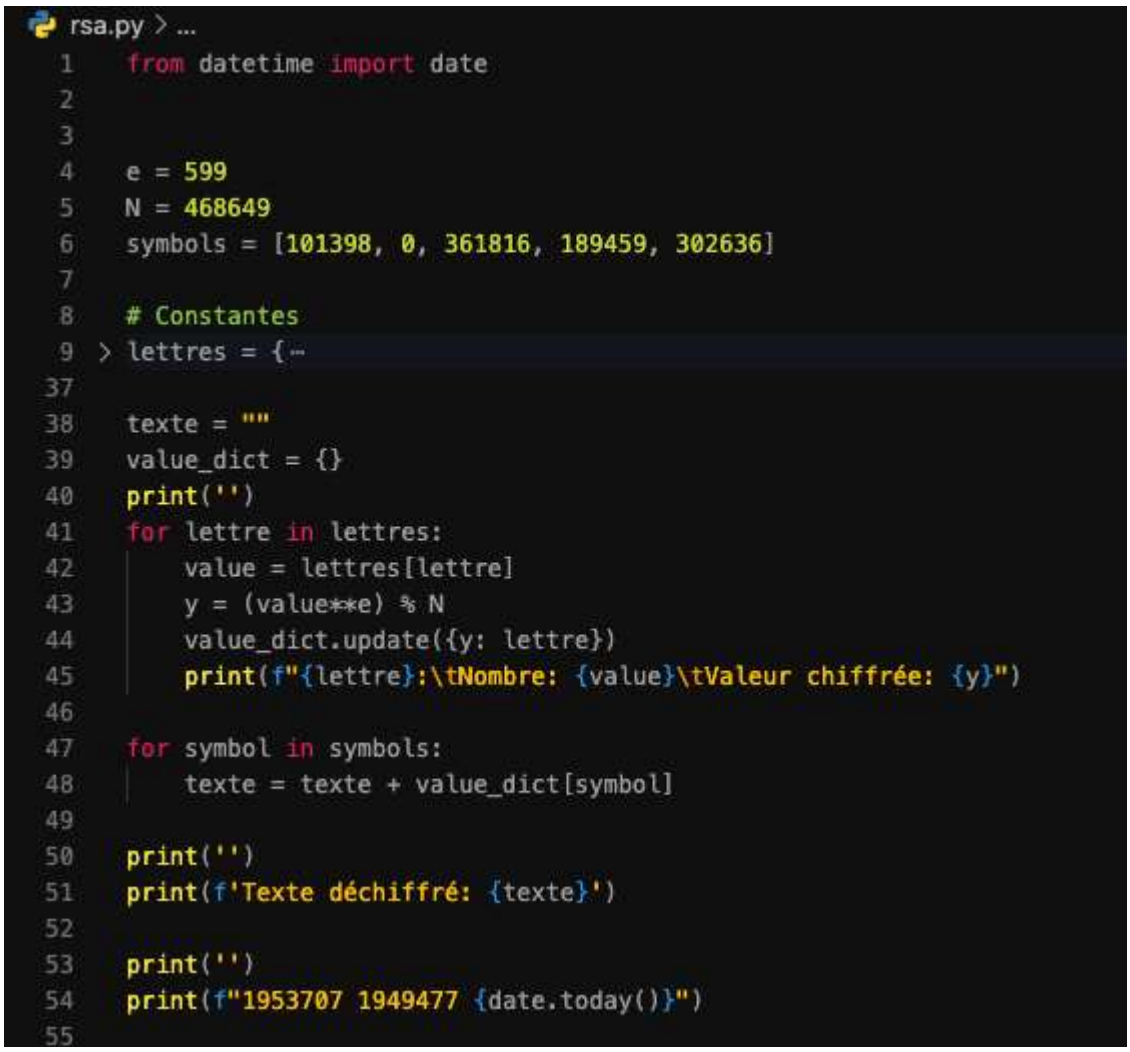
Références

1. Wikipédia. (2021, septembre 7). *Letter frequency* https://en.wikipedia.org/wiki/Letter_frequency
2. Frequency of Character Pairs in English Language Text (s. d.). Homepages. Consulté le 3 octobre 2021, à l'adresse http://homepages.math.uic.edu/~leon/mcs425-s08/handouts/char_freq2.pdf
3. Arampatzis, A. (2020, décembre 10). *What Are the Differences Between HTTP and HTTPS?* Venafi. https://www.venafi.com/blog/what-are-differences-between-http-https-0
4. Certificat électronique (2021, septembre 3). Wikipédia. https://fr.wikipedia.org/wiki/Certificat_%C3%A9lectronique
5. Qu'est-ce que la cryptographie à clé publique? (s. d.). GlobalSign. Consulté le 3 octobre 2021, à l'adresse https://www.globalsign.com/fr/centre-information-ssl/cryptographie-cle-publique
6. Verify the Identity of a Website in Your Browser (s. d.). TechguyLabs. Consulté le 3 octobre 2021, à l'adresse https://techguylabs.com/blog/verify-identity-website-your-browser
7. Autorité de certification (2021, mars 16). Wikipédia. https://fr.wikipedia.org/wiki/Autorit%C3%A9_de_certification
8. Arampatzis, A. (2020, juillet 28). *What Is the Difference between Root Certificates and Intermediate Certificates* Venafi. https://www.venafi.com/blog/what-difference-between-root-certificates-and-intermediate-certificates#:~:text=The%20root%20certificate%2C%20often%20called,Public%20Key%20Infrastructure%20(PKI).&text=A%20root%20certificate%20is%20invaluable,automatically%20trusted%20by%20the%20browsers.
9. Ludin, J. (2020, janvier 22). *Risks of a Public Certificate Authority* Security Boulevard. https://securityboulevard.com/2020/01/risks-of-a-public-certificate-authority/
10. Sectigo. (s. d.). What Is a Self Signed Certificate? *Know the Advantags and Disadvantages* Consulté le 3 octobre 2021, à l'adresse https://sectigostore.com/page/what-is-a-self-signed-certificate/

Annexes

Annexe 1

Programme conçu par l'équipe pour déchiffrer le texte à l'aide d'une clé fournie.



```python
from datetime import date


e = 599
N = 468649
symbols = [101398, 0, 361816, 189459, 302636]

# Constantes
lettres = {…

texte = ""
value_dict = {}
print('')
for lettre in lettres:
    value = lettres[lettre]
    y = (value**e) % N
    value_dict.update({y: lettre})
    print(f"{lettre}:\tNombre: {value}\tValeur chiffrée: {y}")

for symbol in symbols:
    texte = texte + value_dict[symbol]

print('')
print(f'Texte déchiffré: {texte}')

print('')
print(f"1953707 1949477 {date.today()}")

```

Annexe 2

Programme conçu par l'équipe pour automatiser le remplacement des caractères du message original.

ana_freq.py > ...

```
        already_used_old_list.append(old)

        new = input("New: ")

        if new in already_used_new_list:
            new = input("Already used, try again...\nNew: ")
        already_used_new_list.append(new)

        for n, char in enumerate(message):
            if char == old and not n in replaced_list:
                message[n] = CYAN + new + RESET
                replaced_list.append(n)

        if new==" ":
            replacements.append(f"{old} -> space")
        else:
            replacements.append(f"{old} -> {new}")

        text = "Text: \n"
        for char in message:
            text = text + char

        print("")
        print(text)
        print("")
        print(f"Replacements: {replacements}")
        print("")
        print(f"1953707 1949477 {date.today()}")
        print("")
        print("-------------------------------------------------------------")
        print("")
```