



**INF4420a - Sécurité informatique**

**Hiver 2023**

**Travail pratique #1**



**Date de Remise : 19 février 2023**

## Partie A

### Question 1

a) [/0.1] Calculez l'entropie par lettre (h-lettre) d'une chaîne générée avec texte d'une longueur de 425 caractères. 

```
(vikimb4712:~) $ Source -> Entropie -> Chiffrement) $ echo 2231234-1954607 `date
2231234-1954607 lun 23 jan 2023 15:33:29 EST
(vikimb4712:~) $ Source -> Entropie -> Chiffrement) $ ./texte 425 > texte.bin
(vikimb4712:~) $ ./h-lettre < texte.bin
(space) = 77
A = 24
B = 3
C = 16
D = 5
E = 62
F = 6
G = 4
H = 25
I = 24
J = 8
K = 1
L = 14
M = 4
N = 25
O = 22
P = 7
Q = 8
R = 29
S = 22
T = 29
U = 9
V = 3
W = 0
X = 0
Y = 6
Z = 0
Nombre total de caracteres : 425
Entropie de l'entrée : 3.933376
```

Figure 1: Calcul de l'entropie du programme texte.

L'entropie d'une chaîne de 425 caractères générée avec texte est de 3.93.

b) [/0.2] En vous servant du premier théorème de Shannon, expliquez ce que signifie cette valeur.

C'est la mesure de l'incertitude en bits des fréquences des octets du texte aléatoire généré. En d'autres mots, c'est le nombre de bits nécessaire en moyenne pour coder un symbole.

c) [/0.1] 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)?

$$entropie = \sum_{1}^{27} \left( \frac{1}{27} \log_2 \frac{1}{27} \right) = 4.81 \text{ bits par lettre}$$

d) [/0.1] Que représente le quotient de la valeur en a) sur la valeur en c) ?

$\frac{3.933376}{4.807355} = 0.818199$  représente le taux de compression qui signifie à quel point les lettres générées sont aléatoires. Donc on peut compresser au maximum d'environ 19%.

e) [/0.1] 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) ? 

```

(vikim@14712-00 Source - Entropie - Chiffrement) $ echo 2231234-1954607 "date"
2231234-1954607 lun 23 jan 2023 15:35:11 EST
(vikim@14712-00 Source - Entropie - Chiffrement) $ ./lettre 425 > texte.bin
(vikim@14712-00 Source - Entropie - Chiffrement) $ ./h-lettre < texte.bin
(space) = 57
A = 30
B = 5
C = 10
D = 11
E = 46
F = 6
G = 5
H = 25
I = 33
J = 0
K = 4
L = 14
M = 10
N = 34
O = 33
P = 3
Q = 0
R = 23
S = 29
T = 35
U = 8
V = 5
W = 4
X = 0
Y = 3
Z = 1
Nombre total de caracteres : 425
Entropie de l'entree : 4.074061

```

**Figure 2: Calcul de l'entropie du programme lettre.**

L'entropie avec le programme lettre est de 4.07. Donc, le taux de compression est de

$\frac{4.07}{4.807355} = 0.846619$ . Alors, la différence est de  $0.846619 - 0.818199 = 0.02842$  donc significative.

**f) [/0.15]** 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. Expliquez cette contradiction apparente (le fait que les deux entropies soient proches).

Texte est une source non markovienne donc la génération des symboles dépendent des symboles précédents générés. Lettre est une source markovienne donc la génération des symboles ne dépendent pas des symboles déjà générés. L'entropie de la source non markovienne va nécessairement être un peu plus petite que celle de la source markovienne. À la fin, les fréquences vont quand même être très similaires, car les lettres aléatoires viennent de la langue anglaise aussi.

## Question 2

**a) [/0.1]** Utilisez les programmes cesar et cesar-d avec les sources texte et lettre, pour chiffrer et déchiffrer des chaînes de 425 caractères. 📸

```
[vikim@14712-00 Source - Entropie - Chiffrement] $ echo 2231234-1954607 'date'
2231234-1954607 lun 23 jan 2023 16:13:37 EST
[vikim@14712-00 Source - Entropie - Chiffrement] $ ./texte_ess > texte.bin
[vikim@14712-00 Source - Entropie - Chiffrement] $ cat texte.bin
E MALE AND THATS THE SPECIAL GROUND OF THEIR CONTEMPT WHEREWITH THEY STUDY TO EXCLUDE YOUR GRACE BUT THEY SHALL FINDE THAT FORGED GROUND OF THEIRS TO BE BUT DUSTY HEAPES OF BRITTLE SANDE ART PERHAPS IT WILL BE THOUGHT A HEYNOUS THING T HAT I A FRENCH MAN SHOULD DISCOURER THIS BUT HEAUN I CALL TO RECORDE OF MY VOWES IT IS NOT HATE NOR ANY PRIUAT WRONGE BUT LOUE VNTO MY COUNTRY AND THE RIGHT PROUOKES MY TONGUE THUS LAUISH[vikim@14712-00 Source - Entropie - Chiffrement] $
[vikim@14712-00 Source - Entropie - Chiffrement] $ ./cesar < texte.bin > cypher_texte.bin
[vikim@14712-00 Source - Entropie - Chiffrement] $ cat cypher_texte.bin
H PODH DOG WKWV WKH VSHFLDOO JURXQ RI WKHLU FRQHPSW ZKHUH LQWVXK WKRQH WR HAFOGHX BRXU JUDFH EXW WKHB VKDQO ILQGH WKWV IRUJHG JURXQ RI WKHLU VHUVLW WR EH EXW GXWVH KHDOSH RI EULWLQH VLQGH DUW SHUDSV LW ZLQH EH WKRXJKW D KHBRQH J W KWV L D IHUQFH PDQ VRXQG GLVFRXQ WKLV EXW KHOXQ L FDOO WR UHFRUGH RZ PB YRZHV LW LV QRQ KOMH QRQ QBL SULXDW ZURQH EXW ORKH YQRH PB FRXQWV DQG WKH ULKWH SUKXRVH PB WRQJXH WKVX ODXLVK[vikim@14712-00 Source - Entropie - Chiffrement] $
[vikim@14712-00 Source - Entropie - Chiffrement] $ ./cesar-d < cypher_texte.bin
E MALE AND THATS THE SPECIAL GROUND OF THEIR CONTEMPT WHEREWITH THEY STUDY TO EXCLUDE YOUR GRACE BUT THEY SHALL FINDE THAT FORGED GROUND OF THEIRS TO BE BUT DUSTY HEAPES OF BRITTLE SANDE ART PERHAPS IT WILL BE THOUGHT A HEYNOUS THING T HAT I A FRENCH MAN SHOULD DISCOURER THIS BUT HEAUN I CALL TO RECORDE OF MY VOWES IT IS NOT HATE NOR ANY PRIUAT WRONGE BUT LOUE VNTO MY COUNTRY AND THE RIGHT PROUOKES MY TONGUE THUS LAUISH[vikim@14712-00 Source - Entropie - Chiffrement] $
[vikim@14712-00 Source - Entropie - Chiffrement] $ ./lettre 425 > lettre.bin
[vikim@14712-00 Source - Entropie - Chiffrement] $ cat lettre.bin
IONEOKHMEFBRSHWATAYREELUOTE ALTPII LWRCCTTARHLL SLNCTEHSH DRTEEEHE DOMCCCESR GOETLNDEEEICASIW ANREIHOHTU ES BORPHNPLNLYSN T MUNHEOISKSTTOENICHORRNATADUSEALO LVHT OA IA IREIHHTTAGINCTWURD RPEEOODSD NNEFFENAETHDENMD SV HDHWSTACN T AATDEI TEI OEFLRPUMINMSHTKSI PWE RT SOOLIFGTYLMTSNHAHILORNHNE AIAOGCMOAHS O EE FFRSL D USKTSIFT MHTIASDI TRORCEEED DIYCSLA I RMOSURONASHYHDC NYRRA EEOIOPCULOTE PALEPCPEOREEN[vikim@14712-00 Source - Entropie - Chiffrement] $
[vikim@14712-00 Source - Entropie - Chiffrement] $ ./cesar + lettre.bin > cypher_lettre.bin
[vikim@14712-00 Source - Entropie - Chiffrement] $ ./cesar-d < cypher_lettre.bin
IONEOKHMEFBRSHWATAYREELUOTE ALTPII LWRCCTTARHLL SLNCTEHSH DRTEEEHE DOMCCCESR GOETLNDEEEICASIW ANREIHOHTU ES BORPHNPLNLYSN T MUNHEOISKSTTOENICHORRNATADUSEALO LVHT OA IA IREIHHTTAGINCTWURD RPEEOODSD NNEFFENAETHDENMD SV HDHWSTACN T AATDEI TEI OEFLRPUMINMSHTKSI PWE RT SOOLIFGTYLMTSNHAHILORNHNE AIAOGCMOAHS O EE FFRSL D USKTSIFT MHTIASDI TRORCEEED DIYCSLA I RMOSURONASHYHDC NYRRA EEOIOPCULOTE PALEPCPEOREEN[vikim@14712-00 Source - Entropie - Chiffrement] $
```

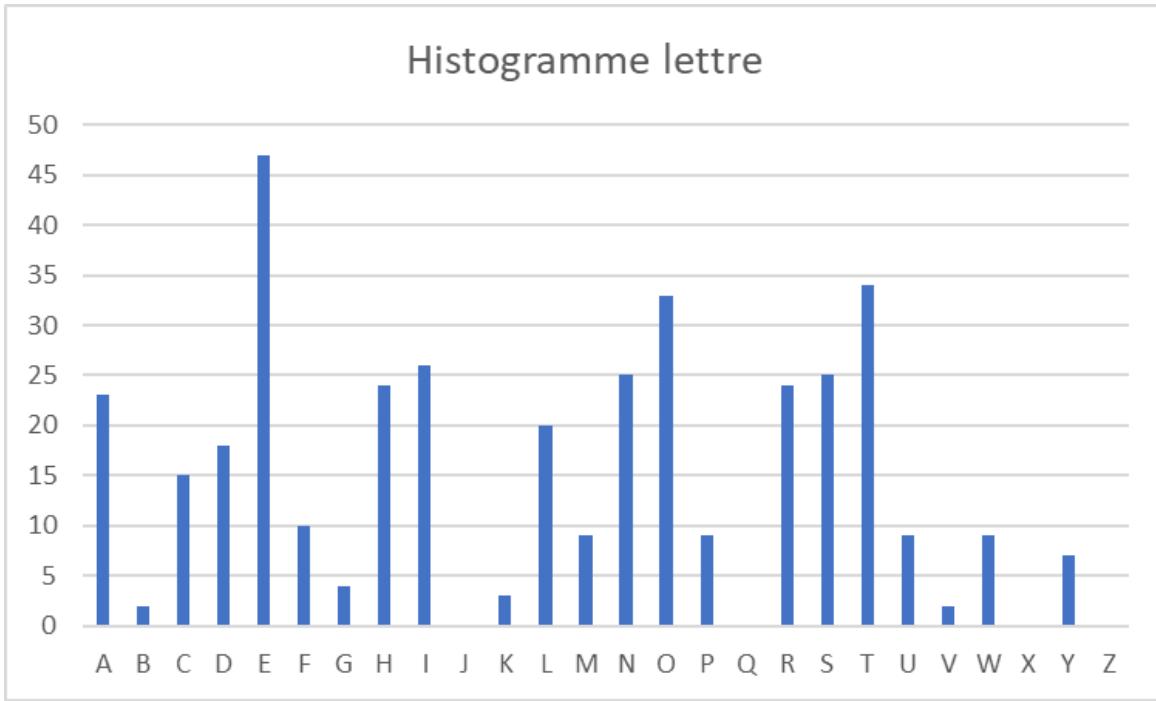
Figure 3 : Chiffrement avec cesar des chaînes de texte et lettre..

b) [/0.2] 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).

```
Fichier Édition Affichage Rechercher Terminal Aide
V = 18
W = 41
X = 2
Y = 2
Z = 5
Nombre total de caracteres : 425
Entropie de l'entrée : 4.058239
[vikim@14712-00 Source - Entropie - Chiffrement] $ ./h-lettre < lettre.bin
(space) = 47
A = 23
B = 2
C = 15
D = 18
E = 47
F = 10
G = 4
H = 24
I = 26
J = 8
K = 1
L = 28
M = 9
N = 25
O = 33
P = 9
Q = 0
R = 24
S = 3
T = 34
U = 9
V = 2
W = 9
X = 0
Y = 7
Z = 1
Nombre total de caracteres : 425
Entropie de l'entrée : 4.157347
[vikim@14712-00 Source - Entropie - Chiffrement] $ ./h-lettre < cypher_texte.bin
bash: lettre_cypher.bin: Aucun fichier ou dossier de ce type
[vikim@14712-00 Source - Entropie - Chiffrement] $ ./h-lettre < cypher_lettre.bin
(space) = 47
A = 0
B = 7
C = 1
D = 23
E = 2
F = 15
G = 18
H = 47
I = 10
J = 4
K = 24

```

Figure 4 : Fréquence de lettre.



**Figure 5 : Histogramme pour le programme lettre.**

**Figure 6 : Fréquence de lettre\_cypher.**

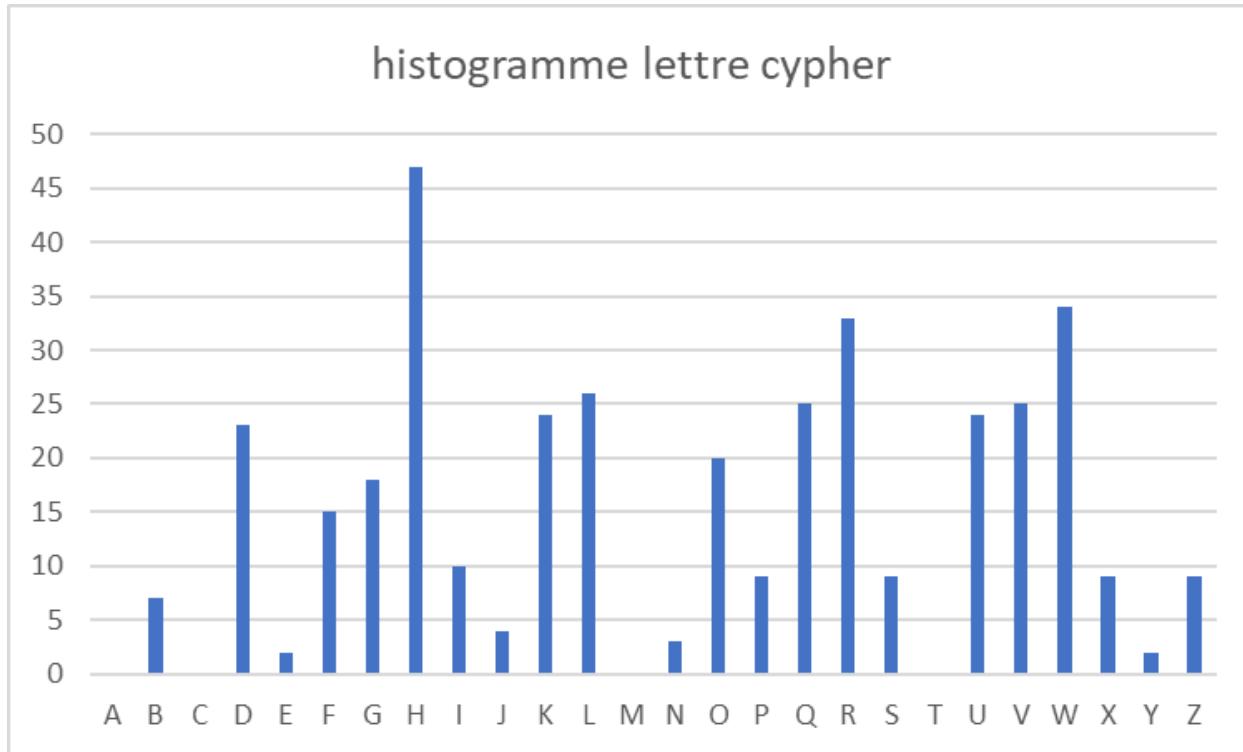


Figure 7: Histogramme pour lettre\_cypher.

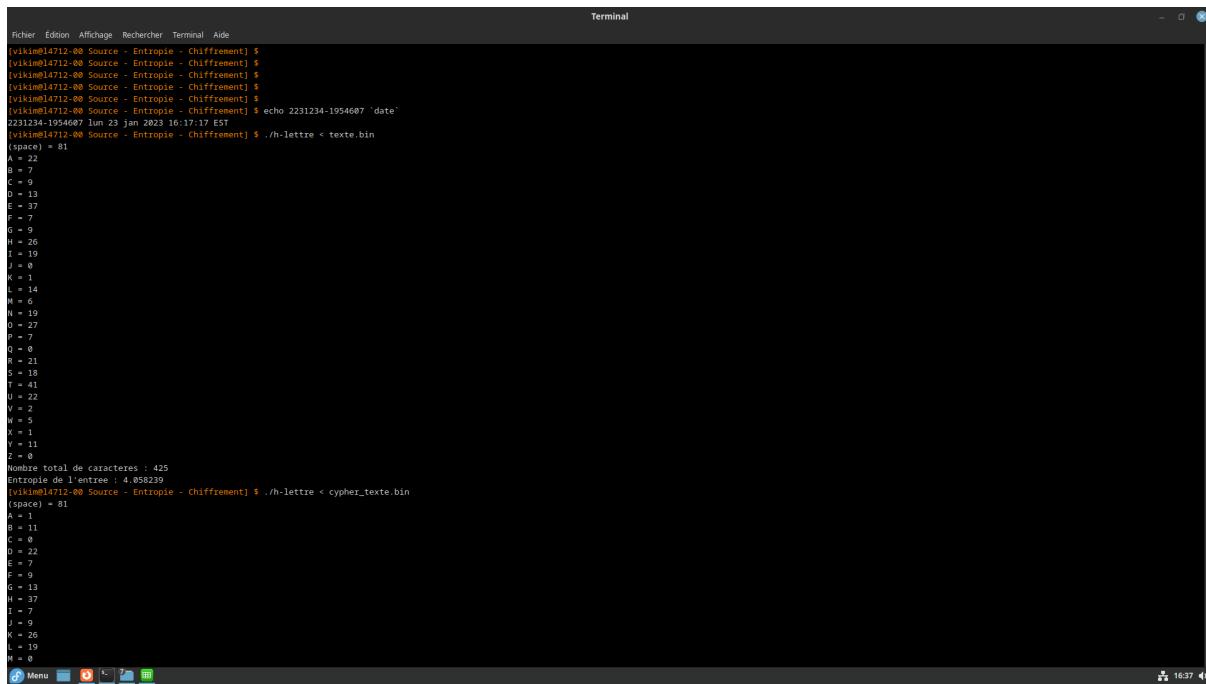


Figure 8 : Fréquence de texte.

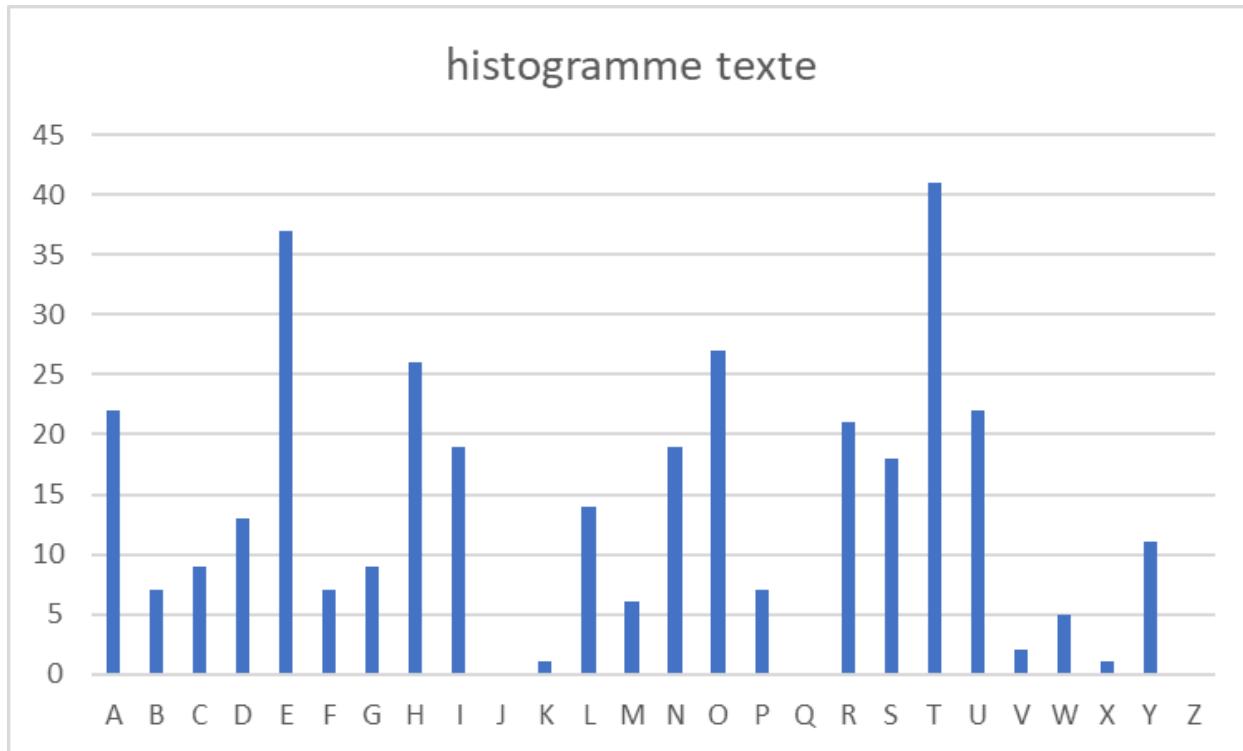


Figure 9 : Histogramme pour texte.

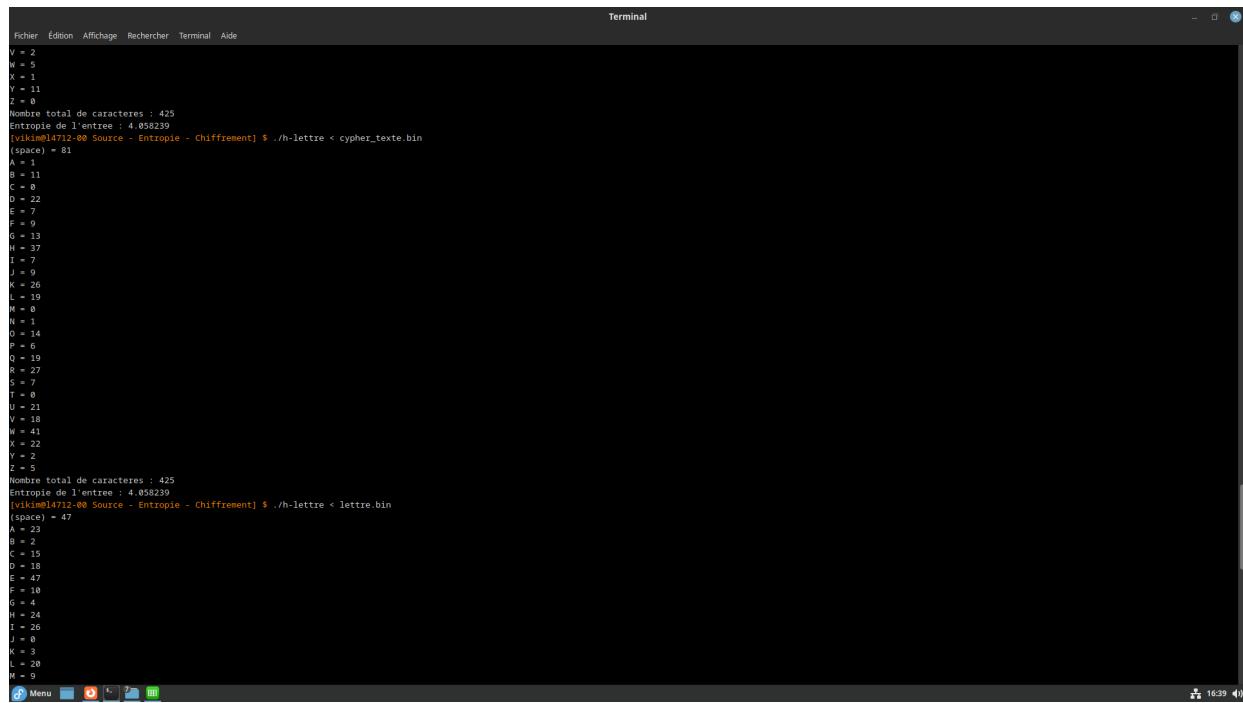
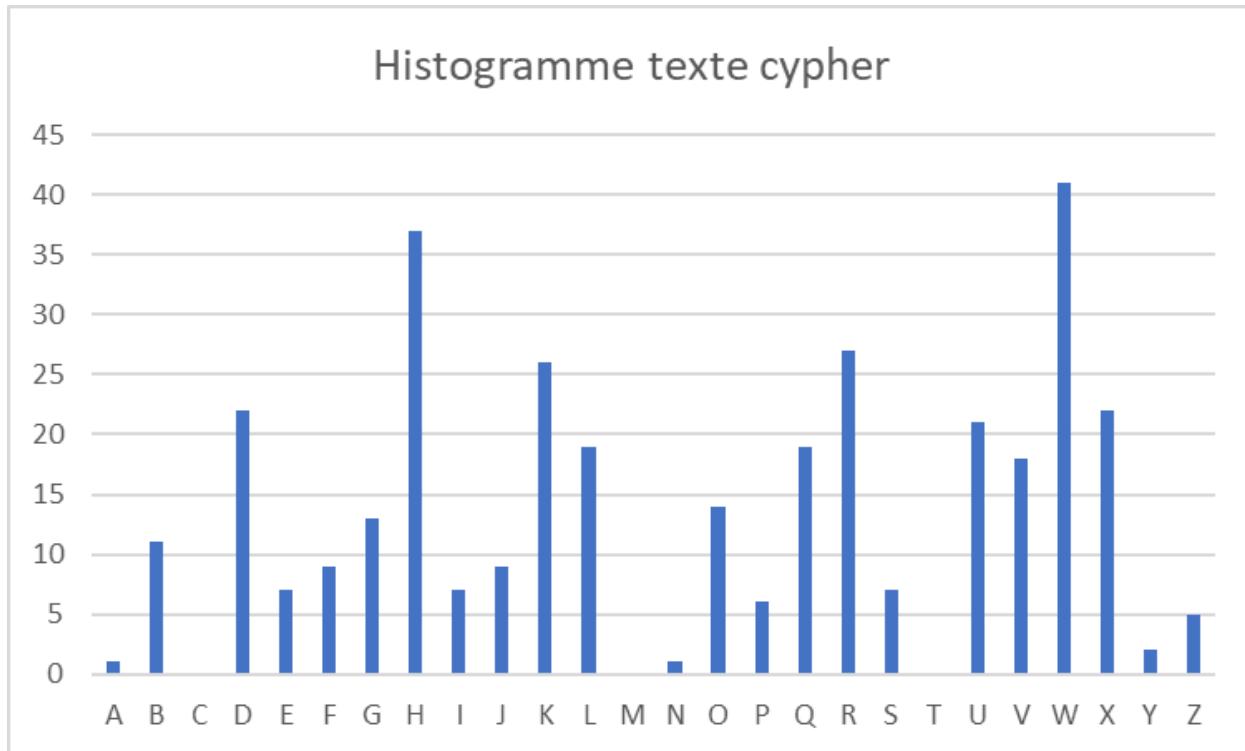


Figure 10 : Fréquence de cypher\_texte.



**Figure 11 : Histogramme de cypher\_texte.**

c) [/0.3] 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 devrait être par exemple les fréquences du (ee) et du (th) dans le cas de texte et de lettre.

Les histogrammes des textes encodés sont les mêmes que ceux qu'ils encodent, à la différence près que les lettres correspondant aux mêmes fréquences sont celles qu'elles encodent. Les histogrammes de lettre et de texte sont très semblables. Si les fréquences étaient comptabilisées sur deux lettres à la fois les histogrammes de texte et de lettre seraient vraiment différents, puisque lettre favoriserait les associations de lettres fréquemment utilisées en général, tandis que texte les associations de lettres qui sont souvent utilisées ensemble. Par exemple, ee serait très commun avec lettre, puisque le e est une des lettres les plus utilisées en anglais, mais ee n'est pas présent dans beaucoup de mots, donc il aurait une faible fréquence avec texte. Au contraire, l'association th est très utilisée en anglais, et va donc avoir une forte fréquence, ce qui ne sera pas le cas dans l'histogramme de lettre.

d) [/0.15] 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 ?

Dans le cas de lettre, il est clair que cela ne rend pas le déchiffrement plus facile, puisqu'il n'y a pas de lien entre la sortie de deux lettres consécutives.

Dans le cas de texte, certaines combinaisons de lettres sont très fréquentes, par exemple th, tandis que d'autres, comme ao ne sont presque jamais utilisées, donc il est possible qu'il soit plus facile de déchiffrer le message en comptabilisant les fréquences sur deux lettres.

Une solution pour lettre serait de prendre en compte que dans un chiffrement césar, la clé utilisée est la même pour toutes les lettres. Ainsi, grâce aux caractères les plus fréquents dans le texte, on peut être capable de deviner la clé utilisée, afin de décoder l'ensemble des lettres, puis on peut vérifier que les fréquences des lettres décodées avec cette clé correspondent bien à leurs fréquences d'utilisation.

### Question 3

a) [/0.15] Générez un fichier de 2048 octets avec monnaie et un avec binaire. Calculer l'entropie par bit (h-bit) et l'entropie par octet (h-ascii) sur les deux fichiers créés. 📸

```
[vikim@l4712-00 Source - Entropie - Chiffrement] $ echo 2231234-1954607 `date`  
2231234-1954607 lun 23 jan 2023 16:37:14 EST  
[vikim@l4712-00 Source - Entropie - Chiffrement] $ ./monnaie 2048 > monnaie.bin  
[vikim@l4712-00 Source - Entropie - Chiffrement] $ ./h-bit < monnaie.bin  
0 = 8194  
1 = 8190  
Nombre total de bits : 16384  
Entropie du texte entre : 1.000000  
[vikim@l4712-00 Source - Entropie - Chiffrement] $ ./h-ascii < monnaie.bin  
Nombre total d'octets : 2048  
Entropie de l'entrée : 7.916051  
[vikim@l4712-00 Source - Entropie - Chiffrement] $ ./binaire 2048 > binaire.bin  
[vikim@l4712-00 Source - Entropie - Chiffrement] $ ./h-bit < binaire.bin  
0 = 10324  
1 = 6060  
Nombre total de bits : 16384  
Entropie du texte entre : 0.950575  
[vikim@l4712-00 Source - Entropie - Chiffrement] $ ./h-ascii < binaire.bin  
Nombre total d'octets : 2048  
Entropie de l'entrée : 0.794618
```

Figure 12 : Calcul de l'entropie du programme monnaie et binaire.

**b) [/0.3]** Générez une clé de 2048 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 (h-bit) et l'entropie par octet (h-ascii) des nouveaux fichiers chiffrés. Qu'observez-vous? Quelles conclusions pouvez- vous en tirer? 

```
[vikim@l4712-00 Source - Entropie - Chiffrement] $ echo 2231234-1954607 `date`  
2231234-1954607 lun 23 jan 2023 16:48:05 EST  
[vikim@l4712-00 Source - Entropie - Chiffrement] $ ./monnaie 2048 > clef.bin  
[vikim@l4712-00 Source - Entropie - Chiffrement] $ ./masque clef.bin 2048 monnaie.bin cypher_monnaie.bin  
[vikim@l4712-00 Source - Entropie - Chiffrement] $ ./h-bit < cypher_monnaie.bin  
0 = 8217  
1 = 8167  
Nombre total de bits : 16384  
Entropie du texte entre : 0.999993  
[vikim@l4712-00 Source - Entropie - Chiffrement] $ ./h-ascii < cypher_monnaie.bin  
Nombre total d'octets : 2048  
Entropie de l'entrée : 7.928016  
[vikim@l4712-00 Source - Entropie - Chiffrement] $ ./masque clef.bin 2048 binaire.bin cypher_binaire.bin  
bash: /masque: Aucun fichier ou dossier de ce type  
[vikim@l4712-00 Source - Entropie - Chiffrement] $ ./masque clef.bin 2048 binaire.bin cypher_binaire.bin  
[vikim@l4712-00 Source - Entropie - Chiffrement] $ ./h-bit < cypher_binaire.bin  
0 = 8291  
1 = 8093  
Nombre total de bits : 16384  
Entropie du texte entre : 0.999895  
[vikim@l4712-00 Source - Entropie - Chiffrement] $ ./h-ascii < cypher_binaire.bin  
Nombre total d'octets : 2048  
Entropie de l'entrée : 7.916364
```

**Figure 13 : Masque jetable pour monnaie et binaire.**

Pour monnaie :

Entropie par bit : 0.999993

Entropie par octet : 7.928016

Pour binaire :

Entropie par bit : 0.999895

Entropie par octet : 7.916364

On observe que les entropies par bit et par octets pour binaire sont plus faibles que celles pour monnaie. On en conclut donc que monnaie génère des 0 et des 1 de manière plus aléatoire que binaire, et que moins de bits seront nécessaires pour binaire que pour monnaie.

**c) [/0.3] Pour les deux cas, s'agit-il d'une méthode sécuritaire de chiffrement? Pourquoi?**

Si le message est chiffré avec une clé de la même taille que le message, il est impossible de le déchiffrer sans connaître la clé. Dans ce contexte, il s'agit donc d'une méthode très sécuritaire.

#### **Question 4**

- a) [/0.2] 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 200 000 \$ 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 20% de chance de détruire votre installation. Quelle serait votre recommandation et pourquoi ? Détaillez vos explications.

Le site B semble être plus abordable à court terme, mais il présente un risque important d'ouragan qui pourrait détruire l'installation. Selon les données météo, il y a une probabilité de 20% que cela se produise chaque année, ce qui signifie qu'il y a une probabilité de 80% que cela ne se produise pas. Cependant, si cela se produit, il faudrait encore payer le coût de l'installation, ce qui a des chances d'arriver chaque année.

En effet, le risque pour l'île B s'évalue à  $200\ 000 * 0.20 = 40\ 000 \$ / \text{an}$ . Au bout de 7.5 ans, l'investissement dans l'île A sera amorti.

De plus, si l'île A a un marché immobilier développé, cela peut indiquer que la technologie et les communications y sont plus développées, ce qui peut faciliter la mise en place de mesures de sécurité informatique plus avancées.

Ainsi, nous lui recommandons de s'installer sur l'île A.

- b) [/0.3] 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.

Il s'agit de l'intégrité des données, car le malfaiteur essaie de changer les données pour diminuer l'intérêt du site. Donc les données ne sont plus correctes.

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

Il s'agit de la disponibilité, car quand le serveur est inondé de requêtes, il ne va pas être capable de servir des nouvelles requêtes.

**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).**

Il s'agit de la confidentialité, car ces données sont privées les clients n'ont jamais eu l'intention de partager ces informations avec le malfaiteur.

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

**c) [/0.4] 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 :**

**Commentez, pour chaque scénario de risque, quel serait l'acteur qui constitue la plus grande menace pour votre entreprise.**

	Acteur	Capacité	Opportunité	Motivation	Probabilité	Impact	Risque
Scénario i	Tricheur	4	4	4	4	2	8
	C.O.	1	4	1	2	2	4
	Concurrents	2	4	2	2.666666667	2	5.333333333

Pour le scénario i, l'acteur qui est la plus grande menace à l'entreprise sont les tricheurs professionnels, car son espérance de perte est la plus grande ce qui veut dire que l'entreprise va avoir la plus grande perte financière. On voit que la probabilité que leurs scénarios se produisent est la plus grande.

	Acteur	Capacité	Opportunité	Motivation	Probabilité	Impact	Risque
<b>Scénario ii</b>	Tricheur	1	4	1	2	4	8
	C.O.	4	4	1	3	4	12
	Concurrents	2	4	4	3.333333333	4	13.33333333

Pour le scénario ii, l'acteur qui est la plus grande menace à l'entreprise va être les sites de poker concurrents. Ils ont la plus grande espérance de perte. La probabilité que leurs scénarios se produisent est la plus grande.

	Acteur	Capacité	Opportunité	Motivation	Probabilité	Impact	Risque
<b>Scénario iii</b>	Tricheur	1	3	1	1.666666667	3	5
	C.O.	4	3	4	3.666666667	3	11
	Concurrents	1	3	2	2	3	6

Pour le scénario iii, l'acteur qui est la plus grande menace à l'entreprise va être les crimes organisés. Ils ont la plus grande espérance de perte. La probabilité que leurs scénarios se produisent est la plus grande.

d) [0.3] 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.
2. Votre patron a refusé de payer les pots-de-vin réclamés par la mafia locale
3. Votre patron fait l'acquisition d'un tout nouveau système de détection des tricheurs très performant.

1) La motivation de tout le monde, surtout des concurrents augmente

2) La motivation des Crimes organisés augmente

3) L'opportunité des tricheurs professionnels diminue.

Dans les 2 premiers cas, le risque augmente, et dans le dernier, il diminue. La gestion du risque évoluerait en conséquence.

e) [/0.3] 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 ?

Acteur	Capacité	Opportunité	Motivation	Probabilité	Impact	Risque
Tricheurs	1	1	1	1	3	3
C.O.	4	1	4	3	3	9
Concurrents	1	1	2	1.333333333	3	4

L'impact du scénario 3 est élevé, car si il se produit, la confiance des utilisateurs va énormément baisser, et on peut donc s'attendre à une baisse énorme du nombre d'utilisateurs, et donc des revenus. Dans ce contexte, nous n'avons pas le détail des bénéfices réalisés chaque mois, ou de la valeur en \$ de chaque unité d'impact, donc nous ne pouvons pas trancher clairement, mais si on évalue que le risque est supérieur à 5000\$ par mois, il faut accepter cette offre.

## Partie B

### Question 1

a) [/0.15] Expliciter les alphabets  $\sigma$ ,  $\tau$ ,  $\tau'$  qui sont respectivement les alphabets pour la sortie de la source, du codeur et du bloc de chiffrement.

L'alphabet de  $\sigma$  est les chiffres de 0 à 9.

L'alphabet de  $\tau$  et  $\tau'$  sont {0,1}.

b) [/0.15] Identifiez les langages provenant des alphabets  $\sigma$ ,  $\tau$ ,  $\tau'$ ,

$\sigma$  est une chaîne de 4 chiffres.

$\tau$  consiste en une chaîne de 64 bits, où les 32 premiers sont l'encodage binaire de  $\sigma$ , et les 32 suivants sont une répétition de ces 32 premiers bits.

$\tau'$  consiste également en une chaîne de 64 bits qui correspondent au chiffrement de  $\tau$  par l'algorithme 3DES.

c) [0.3] 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.

Nous avons identifié 3 attaques auxquelles le système est vulnérable, qui sont les suivantes :

- Brute force : Un attaquant pourrait observer la sortie chiffrée dans le réseau, et essayer toutes les combinaisons de code jusqu'à trouver celle qui est chiffrée de la même manière
- Modification du NIP avec son propre NIP : Un attaquant pourrait modifier l'ancien code NIP d'une carte, et le remplacer par n'importe quel code qu'il connaît, puisque le système ne demande pas de connaître l'ancien code pour le changer.
- Replay : Un attaquant pourrait modifier son propre code, récupérer la sortie chiffrée, puis envoyer sur le réseau ce message pour une autre carte, remplaçant ainsi l'ancien code par un nouveau code connu de l'attaquant.

d) [0.15] Pour chacune des attaques identifiées au c), montrez à l'aide de traces d'exécution comment vous les effectuez. Pour cela, utilisez les scripts transBase et recepBase qui implémentent respectivement les blocs **source+codeur+chiffrement** et **déchiffrement+décodeur+récepteur**. 📸

Brute force :

```

usagers3 > vikim > téléchargements > utilitaire IP1 > Codage > bruteforce.py > main
1 import os
2 import pickle
3
4 map={}
5
6 def main():
7
8     for i in range(10000):
9         nip=str(i)
10        if i<10:
11            | nip='000'+nip
12
13        elif i<100:
14            | nip='00'+nip
15
16        elif i<1000:
17            | nip='0'+nip
18
19        if i%1000==0:
20            | print(i)
21
22        os.system('python3 transBase.py '+nip+' > temp.bin')
23        with open('temp.bin','rb') as myFile:
24            | encrypted=myFile.read()
25            map[encrypted]=i
26
27        with open('temp.pkl','wb') as file:
28            pickle.dump(map,file)
29
30        print("completed")
31
32
33
34
35
36 if __name__ == "__main__":
37     main()
38

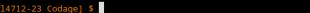
```

Figure 14 : Script du fichier bruteforce.py.

```
dogster@dogster-VirtualBox:~/Desktop$ python main.py
1      import pickle
2
3      def main():
4          input=
5              with open('temp.pkl','rb') as file:
6                  map=pickle.load(file)
7
8          with open('input.bin','rb') as myFile:
9              input=myFile.read()
10
11         print(map[input])
12
13
14
15     if __name__ == "__main__":
16         main()
17
```

**Figure 15 : Script du fichier callbrutef.py.**

```
[vikim@14712-23 Código] $ echo 2231234-1954607 date  
2231234-1954607 lun 06 fev 2023 17:55:10 EST  
[vikim@14712-23 Código] $ pythons transmute.py 3472 > input.bin  
[vikim@14712-23 Código] $ pythons callbrutef.py  
3472  
[vikim@14712-23 Código] $
```

A screenshot of a terminal window titled "vikim@14712-23 Código". The window shows several command-line entries: "echo 2231234-1954607 date" followed by the current date and time; "pythons transmute.py 3472 > input.bin"; "pythons callbrutef.py" followed by the number 3472; and finally "[vikim@14712-23 Código] \$". Below the terminal, there's a standard Linux desktop menu bar with icons for Home, Applications, and System.

**Figure 16 : L'exécution du script pour l'attaque brute force.**

Replay :

**Figure 17 : L'interception et enregistrement de la transmission de données pour rejouer cette transmission après (replay).**

Modification du NIP avec son propre NIP :

```
[vikim@l4712-23 Codage (main)] $  
[vikim@l4712-23 Codage (main)] $ echo 2231234-1954607 `date`  
2231234-1954607 lun 06 fév 2023 19:17:38 EST  
[vikim@l4712-23 Codage (main)] $ python3 transBase.py 3472 > input.bin  
[vikim@l4712-23 Codage (main)] $ python3 transBase.py 5698 > input2.bin  
[vikim@l4712-23 Codage (main)] $ python3 recepBase.py < input2.bin  
5698[vikim@l4712-23 Codage (main)] $
```

**Figure 18 : Attaque de remplacement du NIP d'un utilisateur par un NIP qu'on connaît.**

**e) [/0.3] Pour chacun des trois codages, dites quelles attaques du c) ils permettent de bloquer et démontrez-le à l'aide de trace d'exécution avec les scripts transBase et recepBase. **

Pour le codage 1, il permet de bloquer la force brute. L'attaquant va devoir essayer beaucoup plus de combinaisons. Il doit essayer 14 bits pour le NIP, il y a  $2^{14} = 16\ 384$  combinaisons et avec un nombre aléatoire sur 48 bits il y a  $2^{48} = 281474976710656$  combinaisons. Il y a également 2 bits de parité. Donc l'attaque force brute doit tester  $2^{64}$  combinaisons pour 1 bloc de 64 bits ce qui est très difficile s'il n'y a pas un très grand nombre de NIP connus avec leur chiffrement.

Pour le codage 2, il permet de bloquer la répétition. Le timestamp sur 32 bits aide à bloquer l'attaque replay, car même si l'attaquant fait une retransmission du bloc de données qu'il a enregistré, le système peut vérifier que le bloc de 64 bits a le même timestamp qu'un autre bloc reçu auparavant.

Pour le codage 3, il permet de bloquer la modification du NIP d'un utilisateur et la répétition. Dans le bloc de 16 bits pour l'ancien NIP, le système exige qu'on connaisse l'ancien NIP pour pouvoir modifier le NIP donc l'attaque de remplacement du NIP ne va pas marcher si l'attaquant ne connaît pas l'ancien NIP de base. Comme il y a une bloc de 32 bits pour le timestamp, il permet également de bloquer les attaques de retransmission de données.

**f) [/0.1] Selon vous quel est le meilleur codage ? Pourquoi ?**

On pense que le meilleur codage parmi les 3 est le codage 3, puisque c'est le seul qui pare les 3 attaques décrites précédemment. En effet, dans les 2 premiers, l'attaquant n'a pas besoin de connaître l'ancien NIP, donc il peut changer le mot de passe de quelqu'un par son propre mot de passe. De plus, comme les 3 codages ont la même taille pour un bloc ils bloquent tous l'attaque par force brute, car l'attaquant devra essayer  $2^{64}$  combinaisons avec les 3 codages, et le 3e codage possède aussi un timestamp pour parer l'attaque de replay.

## Question 2

**a) [/0.1] Quelle est la différence entre le protocole http et https dans l'url ?**

https envoie des requêtes et les réponses avec encryption et vérification, en utilisant SSL, alors que http les envoie en clair, ce qui est moins sécurisé. [1]

**b) [/0.1] Qu'est-ce qu'un certificat à clé publique ? A quoi sert-il ?**

Un certificat à clé publique est délivré par une agence de certification, et sert à garantir que la clé publique d'une personne ou d'un site avec qui on souhaite communiquer est la vraie. Cela empêche une personne malveillante de se faire passer pour une autre en envoyant sa propre clé publique, et de pouvoir ainsi déchiffrer les messages.

**c) [/0.2] 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 Banque Tangerine.**

Nom de l'agence de certification	DigiCert
Nom du propriétaire	www.tangerine.ca
Date d'expiration	11/23/22, 7:00:00 PM EST
Algorithme utilisé	RSA
Clé publique	<p>Module: BA D6 9E 85 96 6C D1 6B F3 5D 56 8A 97 C7 22 B4 89 91 6D 84 7F 77 6A EB B4 12 F0 98 0A F4 D1 B7 47 22 19 01 7B 96 8F 32 6C CA 1C 48 73 51 CD A6 AA D1 57 FD 87 12 22 8B E8 A5 E2 8F 20 47 EB D7 B3 45 4A 88 AD D9 EC C5 33 15 13 5D 5C 69 C6 3A C8 1B 42 EA D6 60 3D E7 07 66 2D 4F 02 48 49 DF 3D 4C 3A 22 CB 90 41 6F 28 A7 3C F9 8D 6C 4D 0C EA F6 FF FF AA 3C 73 FD AF 78 74 93 DB 80 4B 9E FE CF EB 44 A3 A9 4C 46 EB 25 98 9D 35 C4 14 D4 8C 3B 1B DD E4 AA 77 9E ED 1C 15 A5 76 1A 6F A2 7F 16 23 4D 22 FF 42 38 9D BA 52 58 E5 97 D7 9E F5 F7 7C E5 E6 60 6E 8E D8 9C 62 72 A3 6C AB 7B 2A 99 59 96 B1 35 FC 60 FB 42 9C 15 C5 CF BE 21 A1 29 0A 3C 63 BB 2D 35 A6 9F 57 8D 94 AF A2 30 13 0E 81 05 0A 5E 34 22 8A B0 42 6B 3F AA E8 22 DD 15 A7 95 44 0D 05 D6 13 14 07 E3 F1 05 9E 45</p> <p>Public Exponent (17 bits): 01 00 01</p>

**d) [/0.2] Comment votre navigateur vérifie-t-il l'identité du propriétaire du site que vous avez visité ?**

Il fait confiance à une agence de certification en qui il a confiance, qui lui garantit que la clé publique du propriétaire du site est la bonne.

e) [0.2] 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 Banque Tangerine. Citez-les.

C'est une organisation fiable qui vérifie les identités des entités en ligne avec lesquelles un utilisateur communique, afin qu'il sache avec qui il communique. Le root certification authority est un certificat self-signed et celui qui fait et signe des certificats de sécurité. Le certification authority est celui qui donne des certificats numériques pour les entités en ligne. [2]

f) [0.1] Est-il risqué d'accepter dans votre navigateur un CA ? Pourquoi ?

Oui, les CA peuvent émettre des certificats pour plusieurs buts comme l'authentification et la signature, il en existe pour tout but. N'importe quel système logiciel installé avec un certificat du même CA sera accepté et voler des informations personnelles. [3]

g) [0.1] 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 TLS qui est signé par son propre créateur. Un certificat TLS est un certificat digital signé par un CA donc un tiers. Il ne faut pas faire confiance à un site web avec un certificat auto-signé, car n'importe qui peut faire des certificats auto-signés pour leur site web et ensuite utiliser ces sites web pour effectuer des attaques pour voler des données personnelles. [4]

### Question 3

a) [0.2] 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. Mettez en capture d'écran le fichier obtenu par ECB seulement (pas besoin de echo) 📸

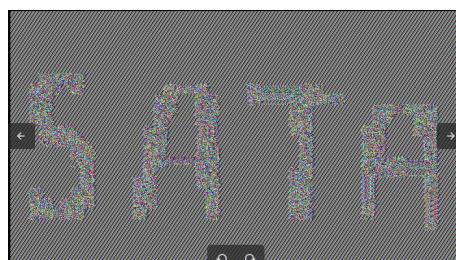


Figure 19 : L'image chiffrée en mode ECB.

Après le chiffrement de l'image, le mot de passe est toujours visible. Pour le mode d'encryption ECB, l'image est en premier convertie en texte. On applique l'algorithme d'encryption sur ce texte, mais la méthode ECB ne fonctionne pas bien pour les blocs avec le même contenu, car les blocs chiffrés vont être identiques ce qui peut expliquer les pixels noirs sont tous devenus gris ce qui rend le mot de passe quand même assez visible.

**b) [/0.2] Chiffrez maintenant le fichier en mode CBC. Observez le fichier puis commentez. Mettez en capture d'écran la commande utilisée (et non le fichier de sortie) 📸.**

```
[vikim@14712-23 ChiffrementBLOC] $ echo 2231234-1954607 `date`  
2231234-1954607 lun 06 fév 2023 16:28:41 EST  
[vikim@14712-23 ChiffrementBLOC] $ python3 AES.py -i mdp.jpg -m CBC  
801570
```

Figure 20 : L'exécution du chiffrement en mode CBC.

Après le chiffrement de l'image, le mot de passe n'est plus visible. Pour le mode d'encryption CBC, l'image est en premier convertie en texte. On ajoute un vecteur initial au texte. Par la suite, pour chaque bloc qui suit on va prendre le bloc précédent chiffré et on va réaliser une opération XOR avec le bloc de texte courant avant d'être chiffré lui-même donc tous les blocs dépendent du bloc précédent ce qui élimine les patterns dans l'image encryptée.

**c) [/0.1] Concluez sur l'importance des modes d'opération des algorithmes de chiffrement par bloc.**

On conclut que pour le chiffrement AES 256 bits, il est mieux d'utiliser le mode CBC, car elle offre une meilleure confusion, il est plus difficile de trouver la relation entre le texte en clair et le texte chiffré. Elle offre également une meilleure diffusion, un changement dans un bloc de texte va affecter le chiffrement de tous les blocs qui suivent.

#### Question 4

**a) [/0.2] Examinez le fichier /etc/passwd. Contient-il des mots de passe ? Pourquoi ? Quelles sont ses permissions d'accès ? Pourquoi ?**

```
(kali㉿kali)-[~/etc]
└─$ echo 2231234 1954607 "date"
2231234 1954607 Mon Feb 6 06:17:10 PM EST 2023
(kali㉿kali)-[~/etc]
└─$ sudo cat passwd
root:x:0:0::/root:/bin/sh:nologin
root:x:0:0:root:/usr/bin/rsh:nologin
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:/sbin/nologin
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:12:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:12: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:11:11:proxy:/var/spool/proxy:/usr/sbin/nologin
www-data:x:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Walling List Manager:/var/list:/usr/sbin/nologin
ircd:x:39:39:ircd:/var/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:/noneexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
system-resolve:x:102:103:systemd Resolvectl,,,:/run/systemd/resolve:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolvectl,,,:/run/systemd/resolve:/usr/sbin/nologin
systemd-timesync:x:103:110:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebusd:x:104:104:Message Bus,,,:/run/dbus:/usr/sbin/nologin
less:x:105:105:Lesser version of stock,,,:/usr/lib/less/lessn/false
strongswan:x:106:65534::/var/lib/strongswan:/usr/sbin/nologin
tcpdump:x:107:114::/nonexistent:/usr/sbin/nologin
usbmuxd:x:108:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
avahi:x:111:111:Avahi mDNS+Bonjour,,,:/var/run/avahi-daemon:/usr/sbin/nologin
rtkit:x:112:112:RTKit RealtimeKit,,,:/var/run/rtkit:/usr/sbin/nologin
speech-dispatcher:x:113:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
nm-openvpn:x:114:120:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
nm-openconnect:x:115:121:NetworkManager OpenConnect plugin,,,:/var/lib/NetworkManager:/usr/sbin/nologin
lightdm:x:117:123:PulseAuthd daemon,,,:/var/run/lightdm:/bin/false
pulse:x:118:126::/var/lib/saned:/usr/sbin/nologin
sane:x:119:127:colorl colour management daemon,,,:/var/lib/colorl:/usr/sbin/nologin
mythtv:x:120:128:MythTV,,,:/var/run/mythtv:/bin/false
stunnel4:x:999:999:stunnel service system account,,,:/var/run/stunnel4:/usr/sbin/nologin
_rpc:x:121:65534::/run/rpcbind:/usr/sbin/nologin
geoclue:x:122:138::/var/lib/geoclue:/usr/sbin/nologin
polkit-x11:x:123:139::/var/lib/polkit-x11:/usr/sbin/nologin
sshd:x:124:132::/nonexistent:/usr/sbin/nologin
ntspec:x:125:135::/nonexistent:/usr/sbin/nologin
redsocks:x:126:136::/var/run/redsocks:/usr/sbin/nologin
```

**Figure 21 : Le contenu du fichier passwd.**

Ce fichier est accessible par tout le monde en lecture uniquement. Par conséquent, les mots de passe ne sont pas affichés. Il n'est accessible en écriture qu'au root, car il n'est pas désirable que n'importe qui puisse modifier ces données.

- b) [0.1] 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 ?**

\$ useradd -g users -s/bin/bash -m NOM avec NOM= le nom de l'utilisateur que vous ajoutez 

```

news:x:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x: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:42:42:Internet Relay Chat:/var/run/ircd:/usr/sbin/nologin
gnat:x:43:43:GNAT System Adminstrator:/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534:::nonexistent:/usr/sbin/nologin
system-metadisplay:x:101:101:System Metadisplay Management,,:/run/systemd/_getty:/sbin/nologin
system-timesync:x:103:110:system Time Synchronization,,:/run/systemd/_getty:/sbin/nologin
messagebus:x:104:111::nonexistent:/usr/sbin/nologin
tss:x:105:105:Telephony Services,,:/var/run/tss:/bin/false
strace:x:106:65534::/lib/strace:/usr/sbin/nologin
tcpdump:x:107:114::nonexistent:/usr/sbin/nologin
usbmux:x:108:46:usbmux daemon,,:/var/lib/usbmux:/usr/sbin/nologin
SMBD:x:109:109:SMBD,,:/var/lib/smbd:/usr/sbin/nologin
dnsmasq:x:110:65534:dnsmasq,,:/var/lib/dnsmasq:/usr/sbin/nologin
avahi:x:111:111:Avahi mDNS daemon,,:/run/avahi-daemon:/usr/sbin/nologin
rtkit:x:112:112:RealtimeKit,,:/run/rtkit:/usr/sbin/nologin
speech-dispatcher:x:113:113:Speech Dispatcher,,:/var/lib/speech-dispatcher:/bin/false
nm-openconnect:x:114:120:NetworkManager OpenConnect plugin,,:/var/lib/NetworkManager:/usr/sbin/nologin
lightdm:x:115:121:Light Display Manager,,:/var/lib/lightdm:/bin/false
polkit-x11:x:116:122:PolicyKit X11,,:/var/lib/polkit-x11:/usr/sbin/nologin
saned:x:118:126::/var/lib/saned:/usr/sbin/nologin
colord:x:119:127:colord colour management daemon,,:/var/lib/colord:/usr/sbin/nologin
mysql:x:120:128:MySQL Server,,:/var/lib/mysql:/nonexistent
stunnel4:x:121:65534::/var/run/stunnel4:/usr/sbin/nologin
rpt:x:121:65534:/var/rpcbind:/usr/sbin/nologin
geoclue:x:122:138:/var/lib/geoclue:/usr/sbin/nologin
Debian-Install:x:123:139:Debian-Install,,:/var/lib/debian-installer:/usr/sbin/nologin
x11vnc:x:124:132::nonexistent:/usr/sbin/nologin
ntpd:x:125:135::nonexistent:/usr/sbin/nologin
redsocks:x:126:138::/var/run/redsocks:/usr/sbin/nologin
rhn-mirror:x:127:139:Rhn-Mirror,,:/var/lib/rhn-mirror:/usr/sbin/nologin
lodaine:x:128:65534::/var/lib/iodine:/usr/sbin/nologin
miredo:x:129:65534::/var/run/miredo:/usr/sbin/nologin
statd:x:130:65534::/var/run/nfs:/usr/sbin/nologin
postfix:x:131:8:Postfix,,:/var/run/postfix:/usr/sbin/nologin
inetutils:x:132:148::/var/lib/inetutils:/usr/sbin/nologin
king-phisher:x:133:142::/var/lib/king-phisher:/usr/sbin/nologin
kali:x:1000:1000::/home/kali:/usr/bin/zsh
zsh:x:1001:1001::/home/zsh:/bin/zsh
Bappon:x:1002:100::/home/Bappon:/bin/bash

```

Figure 22 : Les changements dans le fichier passwd après l'ajout de l'utilisateur.

```

news:x:9:99999:7:::
uucp:x:10:99999:7:::
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:42:42:Internet Relay Chat:/var/run/ircd:/usr/sbin/nologin
gnat:x:43:43:GNAT System Adminstrator:/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534:::nonexistent:/usr/sbin/nologin
system-metadisplay:x:101:101:System Metadisplay Management,,:/run/systemd/_getty:/sbin/nologin
system-timesync:x:103:110:system Time Synchronization,,:/run/systemd/_getty:/sbin/nologin
messagebus:x:104:111::nonexistent:/usr/sbin/nologin
tss:x:105:105:Telephony Services,,:/var/run/tss:/bin/false
strace:x:106:65534::/lib/strace:/usr/sbin/nologin
tcpdump:x:107:114::nonexistent:/usr/sbin/nologin
usbmux:x:108:46:usbmux daemon,,:/var/lib/usbmux:/usr/sbin/nologin
SMBD:x:109:109:SMBD,,:/var/lib/smbd:/usr/sbin/nologin
dnsmasq:x:110:65534:dnsmasq,,:/var/lib/dnsmasq:/usr/sbin/nologin
avahi:x:111:111:Avahi mDNS daemon,,:/run/avahi-daemon:/usr/sbin/nologin
rtkit:x:112:112:RealtimeKit,,:/run/rtkit:/usr/sbin/nologin
speech-dispatcher:x:113:113:Speech Dispatcher,,:/var/lib/speech-dispatcher:/bin/false
nm-openconnect:x:114:120:NetworkManager OpenConnect plugin,,:/var/lib/NetworkManager:/usr/sbin/nologin
lightdm:x:115:121:Light Display Manager,,:/var/lib/lightdm:/bin/false
polkit-x11:x:116:122:PolicyKit X11,,:/var/lib/polkit-x11:/usr/sbin/nologin
saned:x:118:126::/var/lib/saned:/usr/sbin/nologin
colord:x:119:127:colord colour management daemon,,:/var/lib/colord:/usr/sbin/nologin
mysql:x:120:128:MySQL Server,,:/var/lib/mysql:/nonexistent
stunnel4:x:121:65534::/var/run/stunnel4:/usr/sbin/nologin
rpt:x:121:65534:/var/rpcbind:/usr/sbin/nologin
geoclue:x:122:138:/var/lib/geoclue:/usr/sbin/nologin
Debian-Install:x:123:139:Debian-Install,,:/var/lib/debian-installer:/usr/sbin/nologin
x11vnc:x:124:132::nonexistent:/usr/sbin/nologin
ntpd:x:125:135::nonexistent:/usr/sbin/nologin
redsocks:x:126:138::/var/run/redsocks:/usr/sbin/nologin
rhn-mirror:x:127:139:Rhn-Mirror,,:/var/lib/rhn-mirror:/usr/sbin/nologin
lodaine:x:128:65534::/var/lib/iodine:/usr/sbin/nologin
miredo:x:129:65534::/var/run/miredo:/usr/sbin/nologin
statd:x:130:65534::/var/run/nfs:/usr/sbin/nologin
postfix:x:131:8:Postfix,,:/var/run/postfix:/usr/sbin/nologin
inetutils:x:132:148::/var/lib/inetutils:/usr/sbin/nologin
king-phisher:x:133:142::/var/lib/king-phisher:/usr/sbin/nologin
kali:x:1000:1000::/home/kali:/usr/bin/zsh
zsh:x:1001:1001::/home/zsh:/bin/zsh
Bappon:x:1002:100::/home/Bappon:/bin/bash

```

Figure 23 : les changements dans le fichier shadow après l'ajout de l'utilisateur.

Nous avons ajouté l'utilisateur “Bappon”. Comme on peut le voir dans les images précédentes, cela a rajouté des lignes à la fin des deux fichiers passwd et shadow, car ils contiennent les informations sur les utilisateurs existants. Il y a donc un nouvel utilisateur et son mot de passe.

c) [/0.2] Donnez un mot de passe à l'utilisateur que vous avez créé avec la commande si 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 ?

\$ passwd NOM 📸

```

[kali㉿kali)-[/etc]
$ echo 2231234 1954607 "date"
2231234 1954607 Mon Feb 6 06:24:36 PM EST 2023
[kali㉿kali)-[/etc]
$ sudo passwd Baapon
New password:
Retype new password:
passwd: password updated successfully

```

**Figure 24 : La modification du mot de passe.**

```

games:x:5:60:games:/usr/sbin/nologin
man:x:19212:0:99999:7:::
lp:x:77:1:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:mail:/var/mail:/usr/sbin/nologin
mailnull:x:1:mailnull:/var/mail:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/var/run/proxy:/usr/sbin/nologin
proxy:x:13:13:proxy:/var/run/proxy:/usr/sbin/nologin
backup:x:24:24:Backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_nobody:x:100:100:_nobody:/nonexistent:/usr/sbin/nologin
systemd-timesyncd:x:102:102:Network Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-resolved:x:102:102:systemd Resolved,,,:/run/systemd:/usr/sbin/nologin
tss:x:105:113:TPM software stack,,:/var/lib/pm:/bin/false
strongswan:x:106:105:strongswan,,:/var/lib/strongswan:/usr/sbin/nologin
usbmux:x:108:108:USBmux daemon,,:/var/lib/usbmux:/usr/sbin/nologin
usbmux:x:109:109:USBmux daemon,,:/var/lib/usbmux:/usr/sbin/nologin
avahi:x:111:117:Avahi mDNS daemon,,:/run/avahi-daemon:/usr/sbin/nologin
avahi:x:111:117:Avahi mDNS daemon,,:/run/avahi-daemon:/usr/sbin/nologin
rtkit:x:122:122:RealtimeKit,,,:/proc:/usr/sbin/nologin
speech-dispatcher:x:123:123:speech-dispatcher:/bin/false
nm-openvpn:x:124:128:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
nm-openconnect:x:125:121:NetworkManager OpenConnect,,,:/var/lib/NetworkManager:/usr/sbin/nologin
pulse:x:127:123:PulseAudio daemon,,:/run/pulse:/usr/sbin/nologin
saned:x:128:126:/:/var/lib/saned:/usr/sbin/nologin
mysql:x:129:128:MySQL Server,,,:/var/lib/mysql:/bin/false
stunnel:x:129:9999:stunnel service system account:/var/run/stunnel4:/usr/sbin/nologin
geoclue:x:122:138:/:/var/lib/geoclue:/usr/sbin/nologin
Debian-smp:x:123:131:/:/var/lib/smp:/bin/false
ntpsec:x:125:135:/:/nonexistent:/usr/sbin/nologin
redsocks:x:126:116:/:/var/run/redsocks:/usr/sbin/nologin
redsocks:x:126:116:/:/var/run/redsocks:/usr/sbin/nologin
iwd:x:128:65534:/:/var/lib/iwlmode:/usr/sbin/nologin
iwd:x:129:65534:/:/var/run/iwred0:/usr/sbin/nologin
postgres:x:131:138:PostgreSQL Administrator,,,:/var/lib/postgresql:/bin/bash
inetutils:x:132:148:/:/var/lib/inetutils:/usr/sbin/nologin
kali:x:100:10000:/:/home/kali:/usr/bin/zsh
am:x:101:100:/:/home/am:/bin/bash
Baapon:x:1002:1002:/:/home/Baapon:/bin/bash

```

**Figure 25 : Les changements dans le fichier passwd après la modification du mot de passe.**

```

games:x:19212:0:99999:7:::
man:x:19212:0:99999:7:::
lp:x:77:1:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:mail:/var/mail:/usr/sbin/nologin
mailnull:x:1:mailnull:/var/mail:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/var/run/proxy:/usr/sbin/nologin
proxy:x:13:13:proxy:/var/run/proxy:/usr/sbin/nologin
backup:x:24:24:Backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_nobody:x:100:100:_nobody:/nonexistent:/usr/sbin/nologin
systemd-timesyncd:x:102:102:Network Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-resolved:x:102:102:systemd Resolved,,,:/run/systemd:/usr/sbin/nologin
tss:x:105:113:TPM software stack,,:/var/lib/pm:/bin/false
strongswan:x:106:105:strongswan,,:/var/lib/strongswan:/usr/sbin/nologin
usbmux:x:108:108:USBmux daemon,,:/var/lib/usbmux:/usr/sbin/nologin
usbmux:x:109:109:USBmux daemon,,:/var/lib/usbmux:/usr/sbin/nologin
avahi:x:111:117:Avahi mDNS daemon,,:/run/avahi-daemon:/usr/sbin/nologin
avahi:x:111:117:Avahi mDNS daemon,,:/run/avahi-daemon:/usr/sbin/nologin
rtkit:x:122:122:RealtimeKit,,,:/proc:/usr/sbin/nologin
speech-dispatcher:x:123:123:speech-dispatcher:/bin/false
nm-openvpn:x:124:128:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
nm-openconnect:x:125:121:NetworkManager OpenConnect,,,:/var/lib/NetworkManager:/usr/sbin/nologin
pulse:x:127:123:PulseAudio daemon,,:/run/pulse:/usr/sbin/nologin
saned:x:128:126:/:/var/lib/saned:/usr/sbin/nologin
mysql:x:129:128:MySQL Server,,,:/var/lib/mysql:/bin/false
stunnel:x:129:9999:stunnel service system account:/var/run/stunnel4:/usr/sbin/nologin
geoclue:x:122:138:/:/var/lib/geoclue:/usr/sbin/nologin
Debian-smp:x:123:131:/:/var/lib/smp:/bin/false
ntpsec:x:125:135:/:/nonexistent:/usr/sbin/nologin
redsocks:x:126:116:/:/var/run/redsocks:/usr/sbin/nologin
redsocks:x:126:116:/:/var/run/redsocks:/usr/sbin/nologin
iwd:x:128:65534:/:/var/lib/iwlmode:/usr/sbin/nologin
iwd:x:129:65534:/:/var/run/iwred0:/usr/sbin/nologin
postgres:x:131:138:PostgreSQL Administrator,,,:/var/lib/postgresql:/bin/bash
inetutils:x:132:148:/:/var/lib/inetutils:/usr/sbin/nologin
kali:x:100:10000:/:/home/kali:/usr/bin/zsh
am:x:101:100:/:/home/am:/bin/bash
Baapon:x:1002:1002:/:/home/Baapon:/bin/bash

```

**Figure 26 : Les changements dans le fichier shadow après la modification du mot de passe.**

Comme on peut le voir dans les images précédentes, la dernière ligne du fichier shadow a changé, mais le fichier passwd n'a pas été modifié. C'est ce qu'on attendait, car on a modifié le mot de passe, et passwd ne contient pas les mots de passe, alors que le fichier shadow contient un hachage des mots de passe. La seule permission accordée aux utilisateurs pour le fichier shadow est la lecture, et seulement root a la permission d'écriture, car on ne veut pas que n'importe qui puisse modifier les informations des autres.

- d) [0.2] 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 ?

```
(kali㉿kali)-[~/etc]
$ echo 2231234 1954607 `date`
2231234 1954607 Mon Feb 6 06:29:56 PM EST 2023
[kali㉿kali)-[~/etc]
$ sudo passwd Bappon
New password:
Retype new password:
passwd: password updated successfully
```

Figure 27 : Modification du mot de passe

```
news:x:19212:0:99999:7:::
www:x:19212:0:99999:7:::
proxy:x:19212:0:99999:7:::
www-data:x:19212:0:99999:7:::
backup:x:19212:0:99999:7:::
lisa:x:19212:0:99999:7:::
irc:x:19212:0:99999:7:::
gnats:x:19212:0:99999:7:::
nodejs:x:19212:0:99999:7:::
upnp:x:19212:0:99999:7:::
systemd-network:x:19212:0:99999:7:::
systemd-resolve:x:19212:0:99999:7:::
systemd-journal:x:19212:0:99999:7:::
messagbus:x:19212:0:99999:7:::
tss:x:19212:0:99999:7:::
strongswan:x:19212:0:99999:7:::
tcpdump:x:19212:0:99999:7:::
usbmux:x:19212:0:99999:7:::
sshfwd:x:19212:0:99999:7:::
dnsmasq:x:19212:0:99999:7:::
avahi:x:19212:0:99999:7:::
rtkit:x:19212:0:99999:7:::
speech-dispatcher:x:19212:0:99999:7:::
nm-sppnp:x:19212:0:99999:7:::
resolvconf:x:19212:0:99999:7:::
lightdm:x:19212:0:99999:7:::
pulse:x:19212:0:99999:7:::
sunrpc:x:19212:0:99999:7:::
colordevel:x:19212:0:99999:7:::
mysql:x:19212:0:99999:7:::
stunnel4:x:19212:0:99999:7:::
iproute2:x:19212:0:99999:7:::
gpmclient:x:19212:0:99999:7:::
Debian-smb:x:19212:0:99999:7:::
sshd:x:19212:0:99999:7:::
httpd:x:19212:0:99999:7:::
redis:x:19212:0:99999:7:::
rwho:x:19212:0:99999:7:::
rwhod:x:19212:0:99999:7:::
iodine:x:19212:0:99999:7:::
kerneld:x:19212:0:99999:7:::
statd:x:19212:0:99999:7:::
postgres:x:19212:0:99999:7:::
inetutils:x:19212:0:99999:7:::
kml:kmlsh:x:19212:0:99999:7:::
kali:x:19734:99940:2014:qhVr/geu,$0RGHJWfVlbvPWIPI3hc2D.b859AGmMdPyTvmc5LxC:19212:0:99999:7:::
am:10394:10:99999:7:::
Bappon:$5$TMyZCfTjQ7dEq0nM1vd1$nlB65wm52s2V65jv9QE95bGh3M1BpvtR076.nNFx2:19394:0:99999:7:::
2231234 1954607 Mon Feb 6 06:31:21 PM EST 2023
```

Figure 28 : Les changements dans le fichier shadow après la modification du mot de passe.

En remettant le même mot de passe, on s'aperçoit que son hashage est complètement différent. Ceci est une bonne mesure de sécurité, car cela empêche des attaques de type dictionnaire qui consistent à essayer de modifier son propre mot de passe et de comparer les hashages résultant avec les hashages des mots de passe des autres utilisateurs.

Ceci est dû au fait que le mot de passe est hashé avec une valeur de “salt” aléatoire, comme expliqué dans cet article :

<https://www.slashroot.in/how-are-passwords-stored-linux-understanding-hashing-shadow-utils>.

- e) [0.2] 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)-[~/etc]
└─$ echo 2231234 1954607 "date"
2231234 1954607 Mon Feb 6 06:39:42 PM EST 2023
(kali㉿kali)-[~/etc]
└─$ sudo useradd -g users -s/bin/bash -m Zizou
(kali㉿kali)-[~/etc]
└─$ sudo nano /etc/shadow
(kali㉿kali)-[~/etc]
└─$ sudo cat shadow
root:!:19212:0:99999:7:::
daemon:!:19212:0:99999:7:::
bin:!:19212:0:99999:7:::
sys:!:19212:0:99999:7:::
www-data:!:19212:0:99999:7:::
games:!:19212:0:99999:7:::
man:!:19212:0:99999:7:::
lp:!:19212:0:99999:7:::
mail:!:19212:0:99999:7:::
news:!:19212:0:99999:7:::
uucp:!:19212:0:99999:7:::
proxy:!:19212:0:99999:7:::
ftp:!:19212:0:99999:7:::
backup:!:19212:0:99999:7:::
list:!:19212:0:99999:7:::
irc:!:19212:0:99999:7:::
gnats:!:19212:0:99999:7:::
nobody:!:19212:0:99999:7:::
apt:!:19212:0:99999:7:::
systemd-wait-for-kernel:!:19212::::
systemd-resolve:!:19212::::
systemd-timesync:!:19212::::
networkd:!:19212::::
nscd:!:19212::::
strongswan:!:19212::::
tcpdump:!:19212::::
udhcpc:!:19212::::
sshd:!:19212::::
dnsmasq:!:19212::::
avahi:!:19212::::
rtkit:!:19212::::
kernlog:!:19212::::
kernlog-dispatcher:!:19212::::
nm-openvpn:!:19212::::
nm-openconnect:!:19212::::
```

Figure 29 : Création du deuxième utilisateur.

Nous avons pu modifier le fichier shadow en copiant le hash du mot de passe de “Bappon” dans celui de “Zizou”, et cela a fonctionné, nous avons même par la suite pu nous connecter au compte de “Zizou” avec le mot de passe de “Bappon”.

Ceci a été possible, car nous avons les droits d’écriture en tant que root, en utilisant la commande sudo.

C’est un problème, car si quelqu’un arrive à modifier le fichier shadow, et remplacer le hashage du mot de passe d’un utilisateur par celui d’un mot de passe qu’il connaît, il peut se connecter à sa place.

f) [/0.1] Effacez cet utilisateur avec la commande ci-dessous. Qu'est-ce qui se passe dans passwd et shadow ?

\$ userdel -r NOM

```
(kali㉿kali)-[~/etc]
└─$ echo 2231234 1954607 "date"
2231234 1954607 Mon Feb 6 07:01:09 PM EST 2023
(kali㉿kali)-[~/etc]
└─$ userdel -r Zizou
userdel: Permission denied.
userdel: cannot lock /etc/passwd; try again later.
(kali㉿kali)-[~/etc]
└─$ rm /var/mail/Zizou
userdel: Zizou mail spool (/var/mail/Zizou) not found
```

Figure 30 : Suppression du compte

```
news:x:99:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:11:proxy:/var/spool/proxy:/usr/sbin/nologin
data:x:13:33:Data:/var/www/html:/usr/sbin/nologin
list:x:34:34:Backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65:65::/nonexistent:/usr/sbin/nologin
apt:x:100:0:APT:/var/lib/dpkg:/usr/sbin/nologin
root:x:0:0:root:/root:/usr/sbin/nologin
syslog:x:1:1:Privileged User Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:103:104:system Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
tss:x:105:113:TPM software stack,,,:/var/lib/tssm/tsm/false
strongswan:x:106:6534::/var/lib/strongswan:/usr/sbin/nologin
tcpdump:x:107:114::/nonexistent:/usr/sbin/nologin
unbound:x:108:115:Unbound DNS resolver:/var/unbound:/usr/sbin/nologin
sshd:x:109:65534::/run/sshd:/usr/sbin/nologin
dnsmasq:x:110:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
avahi:x:111:65534:avahi daemon:/run/avahi:/usr/sbin/nologin
lvsd:x:112:65534:lvsd Real-time scheduler:/proc:/usr/sbin/nologin
speech-dispatcher:x:113:29:speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
openvpn:x:114:128:NetworkManager OpenVPN,,,:/var/lib/openvpn:/chroot:/usr/sbin/nologin
lightdm:x:116:122:Light Display Manager:/var/lib/lightdm:/bin/false
pulse:x:117:123:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin
saned:x:118:126::/var/lib/saned:/usr/sbin/nologin
mysql:x:120:126:MySQL Server,,,:/var/lib/mysql:/usr/sbin/nologin
stunnel4:x:199:999:stunnel service socket account:/var/run/stunnel4:/usr/sbin/nologin
rpc:x:122:127:RPC,,,:/var/lib/rpc:/usr/sbin/nologin
xinetd:x:122:138::/var/lib/xinetd:/usr/sbin/nologin
Debian-smp:x:123:131::/var/lib/smp:/bin/false
sash:x:124:132::/nonexistent:/usr/sbin/nologin
redsocks:x:125:133::/var/run/redsocks:/usr/sbin/nologin
rwhod:x:126:65534::/var/spool/who:/usr/sbin/nologin
rwho:x:128:65534::/var/rdisc/who:/usr/sbin/nologin
snared:x:129:65534::/var/spool/snare:/usr/sbin/nologin
stdatd:x:130:65534::/var/lib/stdatfs:/usr/sbin/nologin
postgres:x:131:138:PostgreSQL Administrator,,,:/var/lib/postgresql:/bin/bash
kali:x:1000:1000:,,,:/home/kali:/usr/bin/zsh
kali:x:1001:1000:,,,:/home/kali:/bin/bash
Baponix:x:1002:1000:,,,:/home/Baponix:/bin/bash

[ kali@kali ~ ]$ cat /etc
[ kali@kali ~ ]$ echo `date` > /var/log/lastlog `date`
[ kali@kali ~ ]$ cat /var/log/lastlog
Mon Feb 6 09:45:57 2023
```

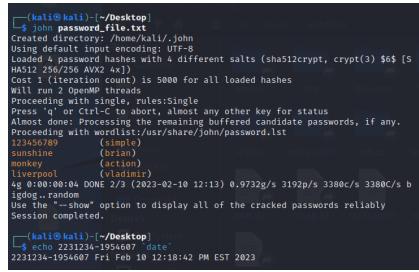
**Figure 31 : Les changements dans le fichier passwd après la suppression du compte.**

**Figure 32 : Les changements dans le fichier shadow après la suppression du compte.**

Comme on peut le voir dans les photos précédentes, nous avons réussi à supprimer l'utilisateur "Zizou", mais pour ce faire, nous avons été obligés de spécifier sudo, ce qui est rassurant, car ça empêche que n'importe quel utilisateur puisse en supprimer un autre. Les lignes concernant cet utilisateur ont simplement été effacées des fichiers passwd et shadow. L'information associée à l'utilisateur "Zizou" ne se trouve plus dans les fichiers passwd et shadow.

## Question 5

- a) [0.1] Utilisez « John The Ripper » avec le dictionnaire « rockyou.txt », et identifier le mot de passe correspondant à chaque utilisateur du fichier « password\_file.txt ». Le dictionnaire est pré-installé sur Kali dans `/usr/share/wordlists/`, mais il est compressé. Vous pouvez le décompresser avec la commande « gunzip ». 



```
(kali㉿kali)-[~/Desktop]
$ john password_file.txt
CrackStation:rockyou.txt:john
Using default input encoding: UTF-8
Loaded 4 password hashes with 4 different salts (sha512crypt, crypt(3) $6$ [5
HA512 256/256 AVX 4x])
Cost factor (iteration count) is 5000 for all loaded hashes
Will run 3 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done; Proceeding with the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/John/password.list
123456789          (simple)
sunshine           (brian)
monkey             (action)
liverpool          (vladimir)
4g 0:00:00:01.04 DONE 2/3 (2023-02-10 12:13) 0.9732g/s 3192p/s 3380c/s 3380C/s
bgdge..random
Use the "show" option to display all of the cracked passwords reliably
Session completed.

(kali㉿kali)-[~/Desktop]
$ echo 2231234-1954607 | date
2231234-1954607 Fri Feb 10 12:18:42 PM EST 2023
```

Figure 33 : Utilisation de john pour trouver les mots de passe des utilisateurs.

Le mot de passe de simple est 123456789, celui de brian est sunshine, celui de action est monkey, et celui de vladimir est liverpool.

- b) [0.3] Calculez l'entropie maximale pour les alphabets suivants :

- A. [a-zA-Z]
- B. [a-zA-Z0-9@]
- C. L'ensemble de la table ascii

L'entropie maximale d'un alphabet est obtenue en considérant que chaque caractère a la même probabilité d'apparition, et est donc donnée par le logarithme en base 2 du nombre de caractères différents dans l'alphabet.

A : il y a  $26 + 26 = 52$  caractères dans l'alphabet, donc l'entropie maximale est de  $\log_2(52) = 5.7$  bits.

B : il y a  $26 + 26 + 10 + 1 = 63$  caractères dans l'alphabet, l'entropie maximale est de  $\log_2(63) = 5.98$  bits.

C : il y a 128 caractères dans la table ascii. L'entropie maximale est de  $\log_2(128) = 7$  bits.

**c) [0.15] Déduisez des résultats de b) un critère important pour qu'un mot de passe soit fort**

Plus l'entropie est élevée, plus un mot de passe peut être considéré comme fort, et on voit dans la question précédente que plus il y a de caractères dans l'alphabet, plus l'entropie maximale est élevée, donc plus il est possible de choisir des mots de passe forts.

**d) [0.3] Au vu des résultats de John the Ripper, donner 3 autres critères pour qu'un mot de passe soit fort.**

Les mots de passe que John the Ripper a réussi à craquer sont soit des mots de passes courants, soit des mots existants, et utilisent des alphabets avec peu de caractères ([0-9] ou [a-z]).

En plus de choisir les caractères dans un grand alphabet, on peut penser aux critères suivants pour qu'un mot de passe soit fort :

- Plus il est long, moins les attaques par brute force seront efficaces, donc il faut qu'il soit suffisamment long
- Il ne faut pas que le mot de passe soit un mot existant, ou un mot de passe courant, pour éviter les attaques par dictionnaires comme dans cet exercice
- Si le mot de passe est composé d'informations personnelles, un attaquant peut plus facilement essayer de le deviner, puisque ça réduit fortement l'entropie

**e) [0.15] A votre avis pourquoi il est conseillé de ne pas utiliser le même mot de passe partout ?**

Si on utilise le même mot de passe partout, et qu'un des sites sur lequel on l'a utilisé se fait pirater sa base de donnée, ou qu'un pirate informatique arrive à obtenir ce mot de passe de n'importe quelle autre manière, il aura accès à tous nos comptes.

## Partie C

### Question 1

**a) [0.2] Décrivez comment Ève peut facilement déchiffrer ce message.**

Eve connaît la clé publique de Bob, et chaque lettre est chiffrée séparément. Par conséquent, Eve peut calculer le chiffrement de chaque lettre de l'alphabet en utilisant la clé publique de Bob, puis remplacer les caractères du message codé par les lettres originales.

**b) [0.4] Récupérez le texte à déchiffrer et la clé publique dans le document INF4420A\_TP1\_Q1\_H23 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.**

Pour le matricule 2231234, nous devons déchiffrer le message {387821,317548,237731,171085,56865}, avec comme clé publique pour Bob  $e = 599$ , et  $n = 462257$ .

Voici, pour chaque lettre de l'alphabet, son chiffrage RSA avec  $e = 599$  et  $n = 462257$  (Les calculs ont été effectués sur le site WolframAlpha [5]

Lettre	Lettre codée m	Lettre chiffrée ( $m^e \pmod{n}$ )
A	0	0
B	1	1
C	2	15274
D	3	430435
E	4	317548
F	5	435925
G	6	245136
H	7	305764
I	8	227708
J	9	296854
K	10	430879
L	11	51318
M	12	387821
N	13	21204
O	14	56865
P	15	327220
Q	16	452581
R	17	171085
S	18	331340
T	19	237731
U	20	92937
V	21	25585
W	22	305517
X	23	440286
Y	24	216756
Z	25	450981

Le message est donc : {'M', 'E', 'T', 'R', 'O'}.

**c) [0.15] 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 ?**

Les lettres A et B sont respectivement codées en 0 et 1, et peu importe les valeurs de  $e$  et  $n$ , elles seront chiffrées en 0 et 1. Autrement dit, ces lettres apparaissent en clair, même dans le message chiffré. On peut donc en conclure qu'il ne faut pas utiliser coder de lettres en 0 ou 1 si on utilise un chiffrement RSA.

## Question 2

**Vous êtes en possession d'un extrait de 200 caractères chiffrés qui provient d'un texte en anglais qui vous est inconnu. Rien n'indique que le début de cet extrait est un début de mot et que la fin est une fin de mot. Il y a 27 caractères possibles dans l'extrait chiffré que vous possédez, soient les 26 lettres de l'alphabet et le caractère « @ ». Ceci correspond à un alphabet en clair composé des 26 lettres de l'alphabet et de l'espace. Il est également important de mentionner qu'une fin de phrase est représentée par deux espaces dans le texte original, soit avant que le chiffrement ait été effectué.**

**Récupérez le texte chiffré qui vous a été assigné sur Moodle.**

**L'utilitaire frequency (dossier « Source – Entropie - Chiffrement ») est mis à votre disposition pour vous aider à effectuer le déchiffrement. Il sert à calculer le nombre d'occurrences des caractères dans un texte. L'option –n permet de spécifier la taille de bloc. Dans le cas où celle-ci est omise, une taille de 1 sera utilisée. Mettez ici une seule capture d'écran qui montre votre utilisation de l'utilitaire.** 

**Vous devez fournir le texte déchiffré, ainsi que votre démarche clairement expliquée.**

**Finalement, pour faciliter votre tâche de déchiffrement, les fréquences des lettres en anglais vous sont fournies dans le tableau suivant :**

**Il est également à noter que l'espace est 1.07 fois plus fréquent que la lettre e en anglais<sup>2</sup>. Les digrammes les plus courants en anglais sont, en ordre<sup>3</sup>:**

**th, he, in, en, nt, re, er, an, ti, es, on, at, se, nd, or, ar, al, te, co, de, to, ra, et, ed, it, sa, em, ro. Les trigrammes les plus courants en anglais sont, en ordre<sup>4</sup>:**

**the, and, tha, ent, ing, ion, tio, for, nde, has, nce, edt, tis, oft, sth, men.**

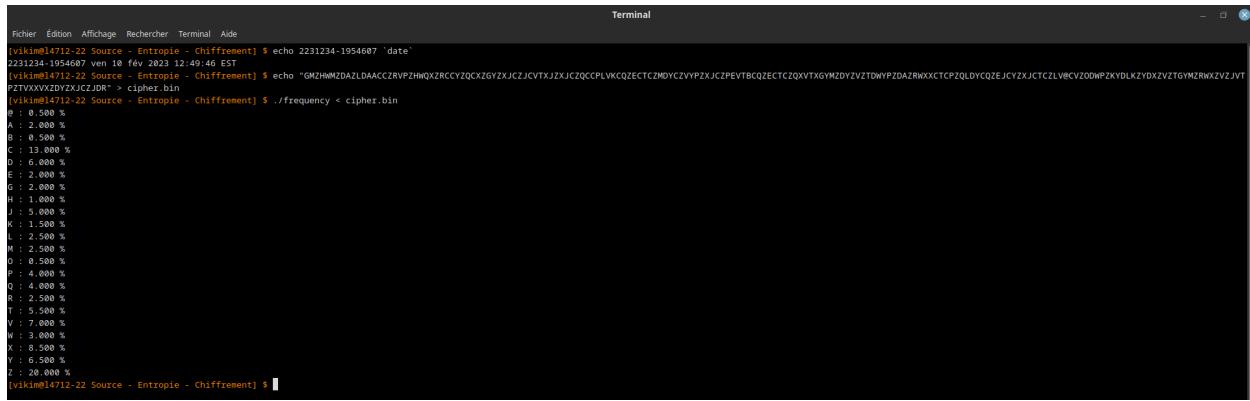
Pour le matricule 2231234, le texte à déchiffrer est :

GMZHWMZDAZLDAACCZRPZHGXZRCYQZCXGYZXJCZJCVTXJZXJCZQCCPLVKCQZECTCZMDYCZVYPZXJ  
CZPEVTBCQZECTCZQVXTXGYMZYDYZVZTDWYPZDAZRWXXCTCPZQLDYCQZEJCYZXJCTCZLV@CVZODWPZKYD  
LKZYDXVZTGYMZRWXZVZJVTPTVXXVXZDYZXJCJDR

Pour ce faire, nous allons effectuer une analyse fréquentielle des lettres de ce texte, et essayer de déduire, en fonction de ces dernières, et des mots existants dans la langue anglaise, les lettres que remplacent les caractères du texte chiffré.

Pour éviter toute confusion, nous garderons les lettres chiffrées en majuscule, et les lettres déchiffrées en majuscule.

Voici la fréquence d'apparition de chaque lettre dans ce texte :



The terminal window shows the following command and its output:

```
Fichier Édition Affichage Rechercher Terminal Aide
[vikine@4712-22 Source - Entrpie - Chiffrement] $ echo 2231234-1954607 "date"
2231234-1954607 ven 10 fév 2023 12:49:46 EST
[vikine@4712-22 Source - Entrpie - Chiffrement] $ echo "GMZHmMZDAZLDAACC2RVPZhwQXZRCYCZQCXZGYXJCZJCVTXJZXJCZQCPVLKCQZECTCZMDYCZVYV2XJCZPEVTBCQZECTCZQVXTGYZDYZVZTDWYPAZDRXXXTCPZQLDYCQZEJCYZXJCTCZLVBcVZ0DWPZKYDLKZYDXZTGYZRwXZVJVT
PZTVXXVX4DYZXJCZJR" > cipher.bin
[vikine@4712-22 Source - Entrpie - Chiffrement] $ ./frequency < cipher.bin
E : 0 500 %
A : 2 500 %
B : 0 500 %
C : 15 300 %
D : 1 500 %
E : 2 000 %
G : 1 000 %
H : 1 000 %
I : 5 000 %
K : 1 500 %
L : 2 500 %
M : 2 500 %
O : 1 000 %
P : 4 000 %
Q : 4 000 %
R : 2 500 %
T : 5 500 %
V : 7 800 %
W : 3 000 %
X : 8 500 %
Y : 10 500 %
Z : 20 000 %
[vikine@4712-22 Source - Entrpie - Chiffrement] $
```

Figure 34 : Affichage des fréquences de lettre dans le texte.

Les caractères les plus présents sont d'assez loin, dans l'ordre, Z, C, et X, donc on suppose qu'il chiffrent respectivement l'espace, e et t.

En effectuant ces substitutions, on s'aperçoit que tJe apparaît de nombreuses fois, donc il est probable que J chiffre h.

Il y a de nombreux mots de 2 lettres ayant Y en 2e lettre, et ReeY est présent aussi, ce qui nous laisse penser que Y chiffre n.

De plus, V apparaît comme un mot seul, et est assez fréquent, donc nous déduisons qu'il s'agit du a.

La présence de theTe et de heaTth nous indique que le T doit chiffrer r.

Aussi, Ehen et Eere nous indique que E doit chiffrer w.

On remarque aussi la présence de anP, qui nous laisse penser que P chiffré d.

Les mots Dn et nDt nous montrent que D chiffré o, roWnd indique que W chiffré u, et oA indique que A chiffré f, puisque les autres lettres qui peuvent correspondre sont déjà attribuées.

Le mot Loffee indique que L chiffré c.

Dans le mot Gn, parmis les lettres restantes, seule i fait du sens pour G

rinM indique que M chiffré g, et Qtarting indique que Q chiffré s.

Hug et Hust indiquent que H chiffré m, Rad, Rut et Ruttered indiquent que R chiffré b, et seedcaKes et KnocK indiquent que K chiffré k.

Enfin, on déduit de dwarfes, ca@ea et Ooud, que B chiffre v, @ chiffre p, et O chiffre l.

Voici une partie de la table de substitution entre les caractères chiffrés et déchiffrés:

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

Le texte déchiffré est donc :

ig mug of coffee bad must been set in the hearth the seedcakes were gone and the dwarves were starting on a round of buttered scones when there capea loud knock not a ring but a hard rattat on the hob

## Références

[1]

<https://www.cloudflare.com/learning/ssl/why-is-http-not-secure/#:~:text=HTTPS%20is%20HTTP%20with%20encryption,far%20more%20secure%20than%20HTTP.>

[2] <https://www.ssl.com/faqs/what-is-a-certificate-authority/>

[3]

<https://www.malwarebytes.com/blog/news/2017/11/when-you-shouldnt-trust-a-trusted-root-certificate>

[4] <https://opensource.com/article/19/11/internet-security-tls-ssl-certificate-authority>

[5] <https://www.wolframalpha.com/widgets/view.jsp?id=570e7445d8bdb334c7128de82b81fc13>):