



**POLYTECHNIQUE
MONTREAL**

UNIVERSITÉ
D'INGÉNIERIE

TP1

INF4420A - Sécurité Informatique Automne 2022



Groupe 01 (B1)

Fait le 9 octobre 2022

Partie A

Question 1

Question A

L'entropie par lettre d'une chaîne générée avec `texte` d'une longueur de 200 caractères est 3.94.

```
(kali@kali) - [~/INF4420A/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./texte 200 | ./h-lettre
(space) = 33
A = 13
B = 2
C = 2
D = 7
E = 26
F = 2
G = 3
H = 7
I = 11
J = 1
K = 2
L = 5
M = 2
N = 17
O = 16
P = 3
Q = 0
R = 11
S = 11
T = 16
U = 0
V = 1
W = 7
X = 0
Y = 2
Z = 0
Nombre total de caracteres : 200
Entropie de l'entree : 3.941336
```

Question B

Cette valeur représente le nombre moyen de bits nécessaires pour représenter une lettre de ce texte sans perte d'information.

Question C

Si la source est markovienne et uniforme, l'entropie est maximale on aura donc

$$H(S) = \sum_i p_i \times \log_2\left(\frac{1}{p_i}\right) = \sum_{i=1}^{27} \frac{1}{27} \times \log_2\left(\frac{1}{\frac{1}{27}}\right) = \log_2(27) \approx 4.75$$

L'entropie par lettre d'un fichier généré avec des probabilités uniformes pour chaque lettre est d'environ 4.75.

Question D

Le quotient de la valeur en a) sur la valeur en c) représente le taux de compression de la source de la question a).

$$\text{Taux de compression} = \frac{H(S^1)/1}{\log_2(27)} = \frac{3.94}{4.75} \approx 0.83$$

Question E

Avec la source `lettre` on obtient une entropie de 4.10. La différence entre les deux est égale à $|4.10 - 3.94| = 0.16$ et n'est donc pas significative (< 0.4).

```
(kali@kali) - [~/INF4420A/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./lettre 200 | ./h-lettre
(space) = 24
A = 13
B = 0
C = 3
D = 9
E = 32
F = 5
G = 3
H = 8
I = 7
J = 1
K = 2
L = 8
M = 4
N = 14
O = 6
P = 5
Q = 0
R = 12
S = 13
T = 13
U = 5
V = 1
W = 5
X = 0
Y = 7
Z = 0
Nombre total de caracteres : 200
Entropie de l'entree : 4.104010
```

Question F

Étant donné que la source `lettre` génère des lettres de manière aléatoire, mais avec la même fréquence que la langue anglaise, l'entropie de cette source n'est pas si différente que celle de la source `texte` qui génère des lettres selon les mêmes probabilités. Aucune des deux entropies n'est maximale puisque les deux sources ne génèrent pas des caractères de manière équiprobable. La différence d'entropie s'explique par le fait que, bien qu'on conserve les mêmes

fréquences, dans le cas de la source non markovienne `texte`, on peut exploiter la dépendance des lettres afin de réduire l'entropie. Ce n'est pas possible dans le cas de la source markovienne `lettre` où chaque symbole est indépendant des symboles précédents. C'est cette différence qui explique la variation positive d'entropie.

Question 2

Question A

Avec la source `texte` nous obtenons le texte chiffré suivant avec l'algorithme de César :

```
(kali@kali) ~/INF4420A/TP1/utilitaireTP1/Source - Entropie - Chiffrement
$ ./texte 200 > 2.txt

(kali@kali) ~/INF4420A/TP1/utilitaireTP1/Source - Entropie - Chiffrement
$ ./cesar < 2.txt
PLJKWB D QDWLRQ GHUEB EH WKRX HPEDVVDGRU IRU YV YQWR RXU IDWKHU LQ ODZ WGH HDUOH RI KHODW PDNH KLP DFTXDLQWGH ZLWK RXU HQWHUSULVH DOG OLNHZLVH ZLOO KLP ZLWK RXU RZO
H DOOLHV WKDW DUH LQ IODXQGVUV WR V
```

En déchiffrant le texte chiffré avec `cesar-d` nous obtenons le texte clair suivant :

```
(kali@kali) ~/INF4420A/TP1/utilitaireTP1/Source - Entropie - Chiffrement
$ ./cesar-d < 2.txt
MIGHTY A NATION DERBY BE THOU EMBASSADOR FOR VS VNTQ OUR FATHER IN LAW THE EARLE OF HENALT MAKE HIM ACQUAINTED WITH OUR ENTERPRISE AND LIKEWISE WILL HIM WITH OUR OWN
E ALLIES THAT ARE IN FLAUNDRS TO S
```

Avec la source `lettre` nous obtenons le texte chiffré suivant avec l'algorithme de César :

```
(kali@kali) ~/INF4420A/TP1/utilitaireTP1/Source - Entropie - Chiffrement
$ ./lettre 200 > 2.txt

(kali@kali) ~/INF4420A/TP1/utilitaireTP1/Source - Entropie - Chiffrement
$ ./cesar < 2.txt
LDLUNRGHRLPPVQOXXDXLOGSXSRRHKKXBUERWSKG BFHWYH UROVP NwV SDR U OFVNH HL HQ GLOZGQBQ UZRSR LKRJ WF H GRHLVRRBW ZP QIG RLWKF LOOMUDDJKTQHFQOWB FZOBIGOE HwKQSHD
GR RWQW VDHLRQ HPRNGQPLD QWOQLWIR
```

En déchiffrant le texte chiffré avec `cesar-d` nous obtenons le texte clair suivant :

```
(kali@kali) ~/INF4420A/TP1/utilitaireTP1/Source - Entropie - Chiffrement
$ ./cesar-d < 2.txt
IAIRTODEOIHMSNUHALUINDPUPOEEHUYRLBOTPHD YCETEV ROLSM KTS PAO R LCSTE EI EN DILWOLYNE RWOPQ IHOG TC E DOEISOOYT WM NFD OITHC ILLJRAAGHQNECNLTY CWLYFDLB ETHNEPEA
DO OTNT SAEON EOKNDNMIA NTLNITFO
```

Question B

Les figures suivantes présentent les histogrammes de fréquences pour la sortie de chacune des sources (claires et codées). Les entropies ont été calculées avec la commande `./h-lettre < source.txt > entropy.txt`.

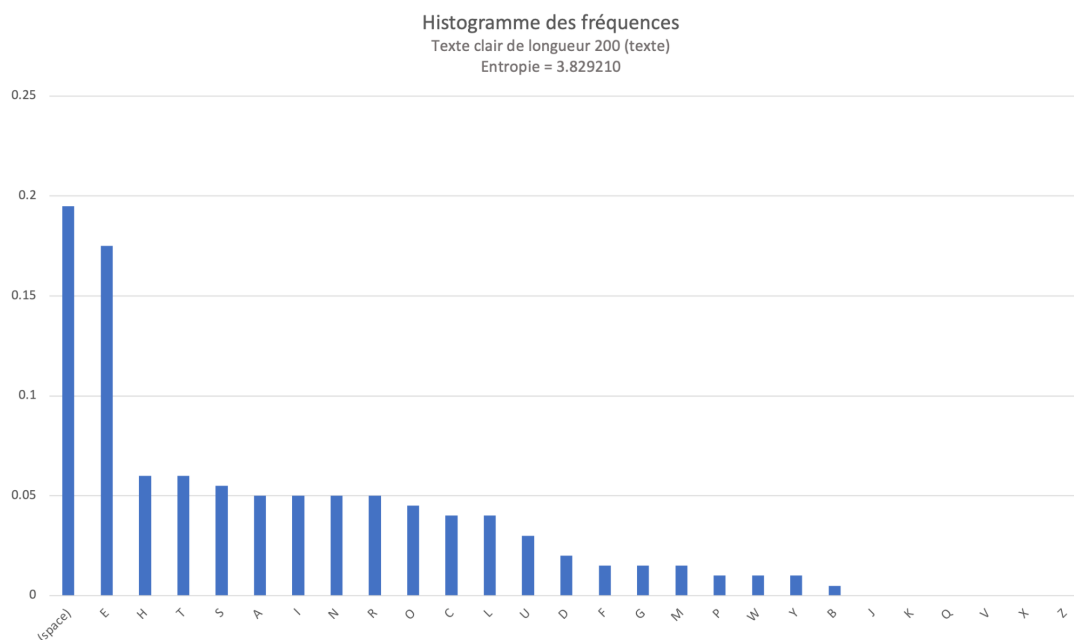


Figure 1 : Histogramme des fréquences du texte clair de longueur 200 (texte)

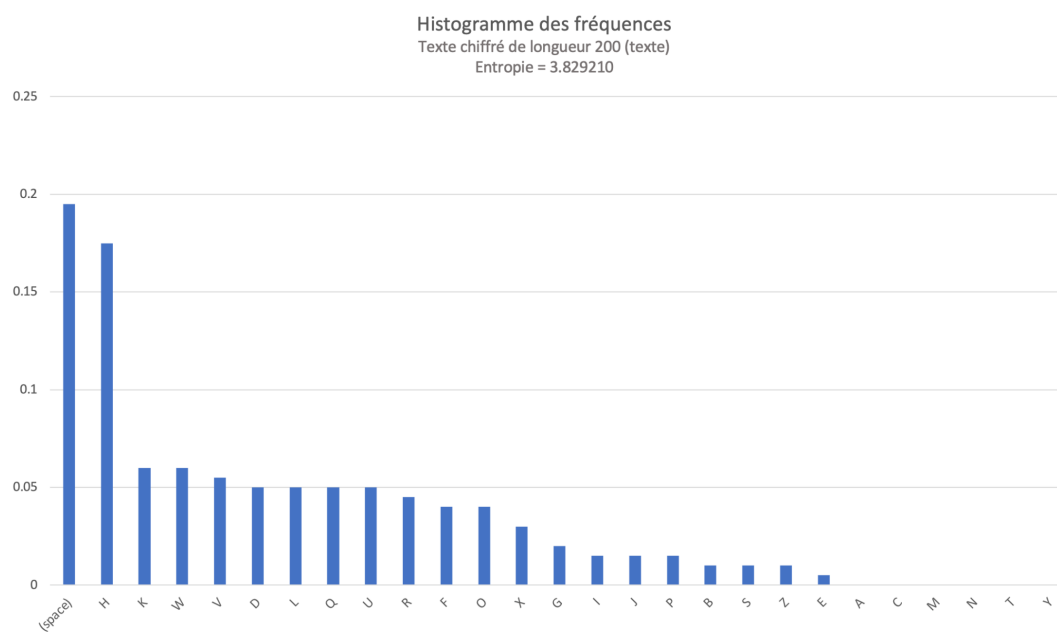


Figure 2 : Histogramme des fréquences du texte chiffré de longueur 200 (texte)

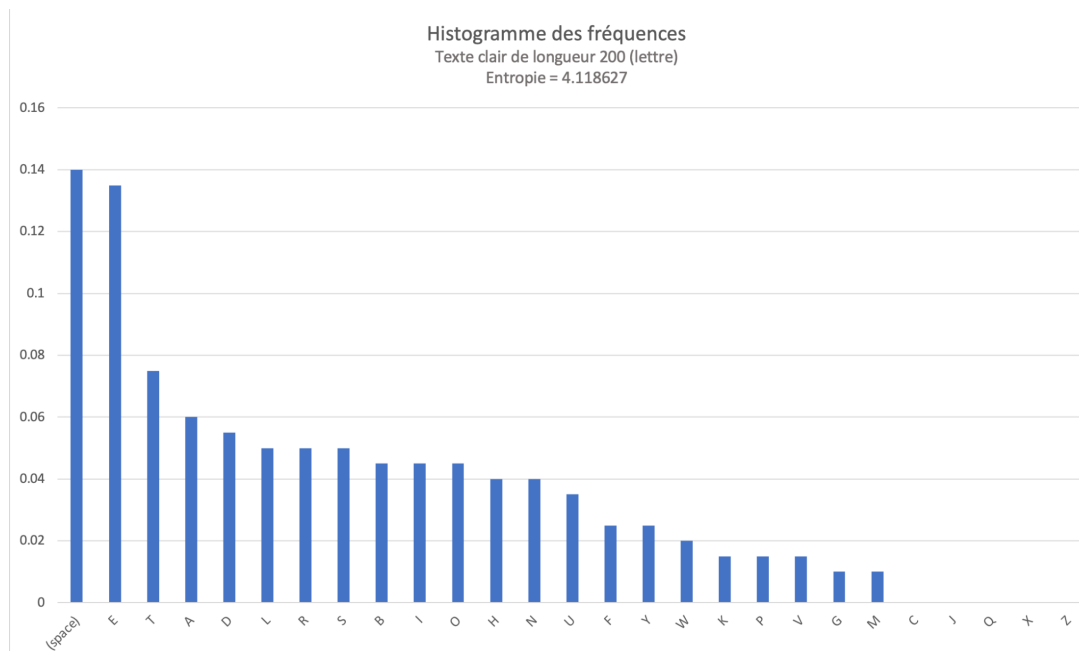


Figure 3 : Histogramme des fréquences du texte clair de longueur 200 (lettre)

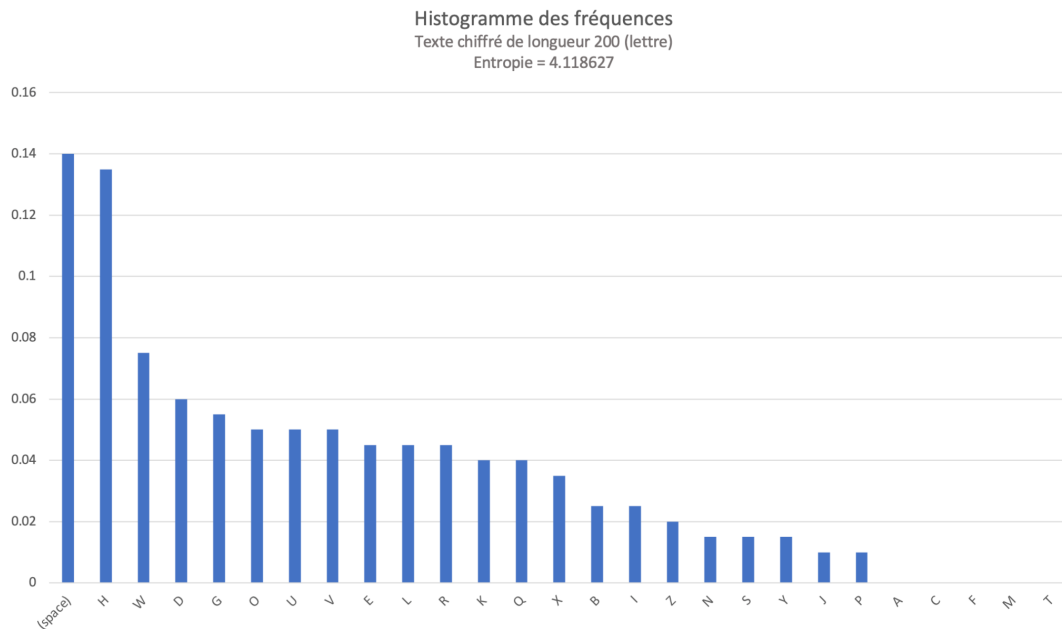


Figure 4 : Histogramme des fréquences du texte chiffré de longueur 200 (lettre)

Question C

Les 4 histogrammes présentent de grandes similarités dans leur forme. On remarque 2 symboles qui dominent par leur fréquence, l'espace et la lettre *E* (ou *H* dans la version chiffrée). Dans les histogrammes construits à partir de la source `texte` on remarque que la 3ème lettre la plus fréquente est le *H* tandis que dans ceux construits à partir de la source `lettre`, c'est le *T* et que la lettre *H* a une plus petite fréquence. Cela est dû à la propriété non-markovienne de la source `texte`.

Si les fréquences étaient comptabilisées par bloc de 2 symboles, l'histogramme de la source `texte` aurait sûrement une forme similaire avec les digrammes *TH* et *HE* en tête. L'histogramme de la source `lettre` serait un peu plus plat et donc plus compliqué à exploiter.

La fréquence du digramme *EE* dans la source `lettre` devrait être très élevée et même la plus grande. En effet, le *E* est la lettre la plus fréquente dans la langue anglaise et la source `lettre` choisit des lettres aléatoirement selon ces fréquences.

La fréquence du digramme *TH* dans la source `texte` devrait être très élevé. En effet, cette source retourne un extrait d'un texte écrit en anglais et ce couple est très présent dans la langue anglais.

Question D

Oui dans le cas de la source `texte`, comptabiliser les fréquences sur 2 lettres facilite le déchiffrement du message. En effet, dans l'histogramme de fréquences, les fréquences de certains digrammes auront une fréquence très importantes comme *TH*, *HE*, *IN* ce qui facilite l'analyse de fréquence.

Pour la source `lettre`, comptabiliser les fréquences sur 2 lettres pourrait aider à l'analyse de fréquences, par exemple comme expliqué à la question précédente, le digramme *EE* serait commun dans le texte. En revanche, les différences de fréquences seront moins flagrantes et utiles qu'avec la source `texte` car la source `lettre` est markovienne.

Question 3

Question A

Voir les résultats des calculs d'entropies sur la figure suivante :

```
(kali@kali)-[~/INF4420A/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./monnaie 1024 > monnaie.bin

(kali@kali)-[~/INF4420A/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./binaire 1024 > binaire.bin

(kali@kali)-[~/INF4420A/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./h-bit < monnaie.bin
0 = 4117
1 = 4075
Nombre total de bits : 8192
Entropie du texte entre : 0.999981

(kali@kali)-[~/INF4420A/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./h-bit < binaire.bin
0 = 5116
1 = 3076
Nombre total de bits : 8192
Entropie du texte entre : 0.954793

(kali@kali)-[~/INF4420A/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./h-ascii < monnaie.bin
Nombre total d'octets : 1024
Entropie de l'entree : 7.804258

(kali@kali)-[~/INF4420A/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./h-ascii < binaire.bin
Nombre total d'octets : 1024
Entropie de l'entree : 0.812822
```

Figure 5 : Résultats des calculs d'entropies pour différentes sources

Question B

Voir les résultats des calculs d'entropies sur la figure suivante. Premièrement, les entropies de binaire et de monnaie sont très similaires. Lors du calcul avec `h-bit` les deux entropies ont une différences de 0.00016 et lors du calcul avec `h-ascii` la différence est de 0.134. Ceci n'est pas significatif puisque c'est inférieur à 0.4 Dans les deux cas, l'entropie de binaire est légèrement plus élevée.

Deuxièmement, on observe que l'entropie est beaucoup plus grande lorsqu'on utilise `h-ascii` et donc qu'on calcule l'entropie par octet. On observe effectivement une entropie d'environ 7.8 avec `h-ascii` et de 0.99 avec `h-bit`. On peut alors conclure que la méthode de chiffrement est bonne puisqu'elle assure une entropie très élevée même lorsque la taille du bloc augmente.

```

(kali@kali) - [~/INF4420A/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./monnaie 1024 > key.bin

(kali@kali) - [~/INF4420A/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./masque key.bin 1024 monnaie.bin monnaie-cipher.bin

(kali@kali) - [~/INF4420A/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./masque key.bin 1024 binaire.bin binaire-cipher.bin

(kali@kali) - [~/INF4420A/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./h-bit < monnaie-cipher.bin
0 = 4032
1 = 4160
Nombre total de bits : 8192
Entropie du texte entre : 0.999824

(kali@kali) - [~/INF4420A/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./h-bit < binaire-cihper.bin
zsh: no such file or directory: binaire-cihper.bin

(kali@kali) - [~/INF4420A/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./h-bit < binaire-cipher.bin
0 = 4077
1 = 4115
Nombre total de bits : 8192
Entropie du texte entre : 0.999984

(kali@kali) - [~/INF4420A/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./h-ascii < monnaie-cipher.bin
Nombre total d'octets : 1024
Entropie de l'entree : 7.809143

(kali@kali) - [~/INF4420A/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ ./h-ascii < binaire-cipher.bin
Nombre total d'octets : 1024
Entropie de l'entree : 7.822543

(kali@kali) - [~/INF4420A/TP1/utilitaireTP1/Source - Entropie - Chiffrement]
$ █

```

Figure 6 : Résultats des calculs d'entropies pour différentes sources

Question C

L'algorithme de chiffrement "One-time Pad" est selon Shannon le seul algorithme de chiffrement avec une "sécurité parfaite". La clé est aussi longue que le message (1024 bits) et a été générée de façon pseudo-aléatoire par une source markovienne, l'entropie de la clé est donc pseudo-maximale (car pseudo-aléatoire). On peut donc considérer qu'il s'agit d'une méthode de chiffrement sécuritaire.

Question 4

Question A

Si on considère qu'il y a 25% de chance qu'un ouragan détruise l'installation du site B par année, on peut considérer que l'installation est détruite tous les 4 ans et qu'il faut alors la reconstruire et payer 100 000\$ à nouveau. Ainsi, après 20 ans, on arrive en moyenne à une somme de 500 000\$ ce qui est l'investissement initial du site A. De plus, le site B aurait une période de reconstruction tous les 4 ans environ pendant laquelle le site ne pourrait accepter de clients et ne ferait pas de revenu. Ainsi, si on veut garder l'installation pendant 20 ans ou plus, nous proposons le site A qui est plus rentable à partir d'environ 20 ans et qui permet d'accepter des clients à l'année longue sans imprévu.

Question B

i) Ce scénario affecte l'intégrité. L'intégrité peut avoir plusieurs définitions, mais peut être vue (dans le sens de l'intégrité d'un système) comme "la protection du système contre les dysfonctionnements". La triche est ici clairement un dysfonctionnement du système et affecte alors l'intégrité du système.

ii) Ce scénario affecte la disponibilité. La disponibilité est définie comme "la garantie que le système sera disponible une fois que la demande en est faite" ce qui n'est pas respecté si les clients ne peuvent pas se connecter lorsqu'il le veulent.

iii) Ce scénario affecte la confidentialité. La confidentialité s'assure que "l'information est accessible qu'à ceux dont l'accès est autorisé" ce qui n'est pas le cas si un malfaiteur accède à notre base de données et obtient certaines données qu'il ne serait pas autorisé à avoir.

Question C

La probabilité de chaque scénario est calculée en faisant le produit des 3 paramètres.

Dans le scénario i, le tricheur présente la plus grande menace :

Acteurs	Capacité	Opportunité	Motivation	Probabilité	Impact	Risque
Tricheur	4	4	4	64	2	128
C.O.	1	4	1	4	2	8
Concurrents	2	4	2	16	2	32

Dans le scénario ii, les concurrents présentent la plus grande menace :

Acteurs	Capacité	Opportunité	Motivation	Probabilité	Impact	Risque
Tricheur	1	4	1	4	4	16
C.O.	4	4	1	16	4	64
Concurrents	2	4	4	32	4	128

Dans le scénario iii, le crime organisé présente la plus grande menace :

Acteurs	Capacité	Opportunité	Motivation	Probabilité	Impact	Risque
Tricheur	1	3	1	3	3	9
C.O.	4	3	4	48	3	144
Concurrents	1	3	2	6	3	18

Question D

Situation 1) Cette situation augmenterait la motivation des concurrents mais ne changerait pas leurs capacités ou leurs opportunités. En effet, si la compagnie est plus célèbre, les gains potentiels d'une attaque sont plus grands (argent, image de marque, etc.).

Situation 2) Cette situation augmenterait les capacités et la motivation du crime organisé mais pas leurs opportunités. En effet, la mafia locale a un désir de vengeance (motivation) et peut donc mettre en place de nouveaux moyens (capacité) afin de nuire à la compagnie et de récupérer les pots-de-vin.

Situation 3) Cette situation diminuerait les opportunités des tricheurs sans affecter leurs capacités et leurs motivations. En effet, le système serait meilleur pour identifier les tricheurs et diminuerait le nombre de tricheurs non punis.

Question E

Scénario iii) Pour le C.O le risque diminue de 144 à 48 et pour les concurrents le risque diminue de 18 à 6. On a donc divisé par 3 le risque pour ces deux types d'acteurs. On considère que ce nouveau système affecte l'opportunité de ces acteurs car on peut maintenant détecter les intrusions sur nos serveurs. Oui cette offre en vaut la chandelle car cela permet de réduire les risques de façon significative. Cette offre permet de diminuer l'opportunité en toutes circonstances en considérant que le serveur était à risque.

Acteurs	Capacité	Opportunité	Motivation	Probabilité	Impact	Risque
Tricheur	4	3	1	3	3	9
C.O.	4	1	4	16	3	48
Concurrents	1	1	2	2	3	6

Partie B

Question 1

Question A

- $\sigma = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, donc tous les chiffres.
- $\tau = \{"0", "1", "2", "3", "4", "5", "6", "7", "8", "9"\}$, donc tous les chiffres, mais cette fois leur représentation en ASCII.
- $\tau' = \{\text{Toutes les permutations possibles des bits d'un octet}\}$, donc de tous les octets possibles.

Question B

- Le langage qui correspond à l'alphabet σ est l'ensemble de toutes les séries différentes de 4 éléments parmi σ . C'est donc tous les nombres entre 0000 et 9999 inclusivement
- Le langage qui correspond à l'alphabet τ est le langage de σ où chaque série est répétée deux fois sous forme de chaînes de caractères de longueur 8 .
- Le langage qui correspond à l'alphabet τ' est l'ensemble des chaînes binaires de longueur 64.

Question C

1. Les 2 clés restent fixes (difficile à changer car stockées dans une puce) et l'attaquant a accès à la source (GAB) et peut intercepter les messages chiffrés sur le réseau, il peut donc mener des attaques à texte clair choisi. En plus, l'attaquant connaît le fonctionnement du codage, il sait donc que chaque NIP est répété 2 fois dans le message chiffré et que chaque nip est encodé et chiffré de la même manière. Il peut donc facilement faire un dictionnaire d'équivalence pour ensuite déchiffrer n'importe quel nip.
2. Étant donné qu'il n'y a pas de validation de temps dans ce système et que chaque nip sera toujours encodé de la même manière, un attaquant pourrait intercepter un message chiffré et à l'avenir manipuler les paquets du réseau pour réécrire ce même NIP (son chiffrement) pour s'authentifier. Ceci est une attaque de rejeu.
3. Un attaquant pourrait modifier les paquets sur le réseau pour changer le chiffrement d'un nip et ainsi faire en sorte que l'ordinateur sauvegarde le mauvais NIP. Ainsi, cet utilisateur ne pourrait plus s'authentifier. Ceci est possible puisque le système ne fait pas de validation du message reçu pour savoir si c'est le bon.

Question D

1. On peut tout d'abord vérifier que chaque NIP est toujours encodé de la même manière:

```
PS C:\Users\mathe\OneDrive\Documents\INF4420a\utilitaireTP1\Codage> python transBase.py 3333
Hy2x[→|q
PS C:\Users\mathe\OneDrive\Documents\INF4420a\utilitaireTP1\Codage> python transBase.py 3333
Hy2x[→|q
PS C:\Users\mathe\OneDrive\Documents\INF4420a\utilitaireTP1\Codage> python transBase.py 3333
Hy2x[→|q
PS C:\Users\mathe\OneDrive\Documents\INF4420a\utilitaireTP1\Codage> python transBase.py 3333
Hy2x[→|q
PS C:\Users\mathe\OneDrive\Documents\INF4420a\utilitaireTP1\Codage> python transBase.py 3333
Hy2x[→|q
```

Maintenant que cela est vérifié, on pourrait facilement bâtir une table de correspondance pour toutes les possibilités de NIP, ce qui est seulement 10 000 possibilités (10^4). Effectivement, on peut envoyer le NIP qu'on veut et voir son chiffrement. Une fois ceci fait, et notre table de correspondance bâtie, on serait capable de trouver n'importe quelle NIP à partir d'un chiffrement.

2. Comme vu plus haut, le chiffrement du NIP est toujours le même. Ainsi, si on intercepte sur le réseau le chiffrement Hy2x[→|q, on a pas besoin de savoir il correspond à quel NIP, on pourra simplement le renvoyer sur le réseau ultérieurement pour s'authentifier avec ce NIP comme on peut le voir ici:

```
(kali@kali) - [~/INF4420A/TP1/utilitaireTP1/Codage]
• $ python transBase.py 3333 > cipher.bin

(kali@kali) - [~/INF4420A/TP1/utilitaireTP1/Codage]
• $ python recepBase.py < cipher.bin
3333
```

Dans cet exemple, on peut voir qu'on sauvegarde un NIP dans un fichier et qu'on le renvoie ultérieurement pour qu'il se fasse déchiffrer et qu'on obtient effectivement le NIP du début. On peut donc voir comment on ferait une attaque de rejeu.

3. On pourrait simplement intercepter le message suivant Hy2x[→|q et le changer pour `❖IS❖4yE`, ce qui sera déchiffré de la manière suivante:

```
(kali@kali) - [~/INF4420A/TP1/utilitaireTP1/Codage]
• $ python recepBase.py < cipher.bin
1234
```

Ainsi, cet utilisateur aura le mauvais NIP de sauvegarder et ne pourra plus se connecter. Dans la capture ci dessus- on a mit `❖IS❖4yE` dans cipher.bin au lieu de Hy2x[→|q pour que le NIP sauvegarder soit 1234 et non 3333. Dans la vraie vie, on aurait modifié les paquets sur le réseau pour changer le chiffrement et non modifié un fichier binaire.

Question E

1.

```
PS C:\Users\mathe\OneDrive\Documents\INF4420a\utilitaireTP1\Codage> python trans1.py 0000
0000E/
0000J
PS C:\Users\mathe\OneDrive\Documents\INF4420a\utilitaireTP1\Codage> python trans1.py 0000
0000NY+
PS C:\Users\mathe\OneDrive\Documents\INF4420a\utilitaireTP1\Codage> python trans1.py 0000
[0000E
```

```
• (kali@kali) - [~/INF4420A/TP1/utilitaireTP1/Codage]
$ python trans1.py 1234 > 1.bin

• (kali@kali) - [~/INF4420A/TP1/utilitaireTP1/Codage]
$ python recep1.py < 1.bin
1234
```

```
mathe@PC-Matheo MINGW64 ~/OneDrive/Documents/INF4420a/utilitaireTP1/Codage
$ python recep1.py < cipher.bin
Erreur dans la transmission
```

Comme on peut le voir dans la première capture, avec cet encodage, les mêmes NIP ne sont pas toujours encodés de la même manière et ainsi, il n'est pas possible de bâtir une table de correspondance et utiliser l'attaque 1. L'attaque de rejeu est encore possible puisqu'il n'y a pas de validation du temps. On peut d'ailleurs voir un exemple dans la deuxième capture d'une attaque de rejeu où le NIP envoyé est sauvegardé et déchiffré ultérieurement pour obtenir le bon NIP. L'attaque 3 est rendue plus difficile grâce au deux bits de parités, comme on peut le voir dans la capture 3. Pour cette capture, le contenu de cipher.bin a été modifié par un autre chiffrement et lorsqu'on tente de le renvoyer pour se faire déchiffrer il y a une erreur de transmission ce qui signifie que le "parity check" n'a pas passé.

2.

```
PS C:\Users\mathe\OneDrive\Documents\INF4420a\utilitaireTP1\Codage> python trans2.py 0000
RH8u0000
PS C:\Users\mathe\OneDrive\Documents\INF4420a\utilitaireTP1\Codage> python trans2.py 0000
RZ0000>0000
PS C:\Users\mathe\OneDrive\Documents\INF4420a\utilitaireTP1\Codage> python trans2.py 0000
=5000000000
```


- `(kali@kali) - [~/INF4420A/TP1/utilitaireTP1/Codage]`
`$ python trans2.py 1234 > 2.bin`

- `(kali@kali) - [~/INF4420A/TP1/utilitaireTP1/Codage]`
`$ python recep2.py < 2.bin`
 Delai de transmission suspect, operation annulee

- `(kali@kali) - [~/INF4420A/TP1/utilitaireTP1/Codage]`
`$ python trans2.py 1234 | python recep2.py`
 0001001101001001
 1234

Comme pour la question précédente, l'attaque par table de correspondance n'est plus possible puisque un même NIP n'est pas toujours encodé de la même manière comme on peut le voir dans la première capture. Les attaques de rejeu sont également plus possible puisqu'il y a maintenant un timestamp pour vérifier le temps d'envoi du message et invalidé ceux qui sont trop vieux comme on peut le voir dans la deuxième capture. La troisième attaque est également rendue plus difficile grâce au deux bits de parités comme précisé plus haut.

3.

```
PS C:\Users\mathe\OneDrive\Documents\INF4420a\utilitaireTP1\Codage> python trans3.py 0000 0000
❖❖❖❖
PS C:\Users\mathe\OneDrive\Documents\INF4420a\utilitaireTP1\Codage> python trans3.py 0000 0000
!!❖❖❖❖.❖❖
PS C:\Users\mathe\OneDrive\Documents\INF4420a\utilitaireTP1\Codage> python trans3.py 0000 0000
t
❖❖❖❖
```

- `(kali@kali) - [~/INF4420A/TP1/utilitaireTP1/Codage]`
`$ python trans3.py 1234 1234 > 3.bin`

- `(kali@kali) - [~/INF4420A/TP1/utilitaireTP1/Codage]`
`$ python recep3.py < 3.bin`
 Delai de transmission suspect, operation annulee

- `(kali@kali) - [~/INF4420A/TP1/utilitaireTP1/Codage]`
`$ python trans3.py 1234 1234 | python recep3.py`
 1234 1234

Pour l'encodage 3, c'est la même chose que le deuxième. L'attaque par table de correspondance n'est plus possible puisque un même NIP n'est pas toujours encodé de la même manière comme on peut le voir dans la première capture. Les attaques de rejeu sont également plus possible puisqu'il y a maintenant un timestamp pour vérifier le temps d'envoi du message et invalidé ceux qui sont trop vieux comme on peut le voir dans la deuxième capture. La troisième attaque est également rendue plus difficile grâce au deux bits de parités comme précisé plus haut.

Question F

Premièrement, le premier algorithme de codage est celui qui produit une sortie avec la plus grosse entropie. Effectivement, dans le cas du premier, 48 bits sont générés de manière aléatoires alors que c'est 16 pour le deuxième et 0 pour le dernier. Le timestamp quant à lui augmente moins l'entropie puisqu'un attaquant peut savoir le moment auquel il a envoyé un nip se faire encoder et ainsi connaître, du moins en partie, le timestamp. Similairement, le fait de mettre l'ancien nip comme avec le dernier encodage est une technique d'encodage peu sécuritaire puisqu'elle n'augmente pas l'entropie. Effectivement, c'est déterministe et un attaquant peut connaître ces données s'il peut faire plusieurs attaques de suite. Par contre, le premier encodage n'a pas de timestamp ce qui rend possible les attaques de rejeu. Ce genre d'attaque ne sont pas possible avec l'encodage 2 puisqu'il y a un timestamp. Sur ce, on dirais que l'encodage 2 est le meilleur encodage, puisqu'il augmente l'entropie suffisamment (plus que l'encodage 3) tout en évitant les attaques de rejeu (ce que l'encodage 1 ne fait pas).

Question 2

Question A

La différence entre HTTP et HTTPS est qu'avec HTTPS le trafic est chiffré en utilisant TLS. Ainsi un attaquant qui écoutait le réseau ne pourrait pas lire en clair le trafic.

Question B

Un certificat à clé publique est un mécanisme permettant de vérifier l'identité d'un site internet en utilisant des méthodes cryptographiques. Dans le cas de Desjardins, un tiers de confiance a signé le certificat afin de garantir que le site web est bien géré par Desjardins. Nous pouvons vérifier l'identité avec la clé publique.

Desjardins utilise l'algorithme de chiffrement RSA avec le hachage SHA-256.

Source: Wikipedia, Wikimedia Foundation. (2022) Certificat électronique. [En ligne]. Disponible: https://fr.wikipedia.org/wiki/Certificat_%C3%A9lectronique

Question C

Version	Version 3
Numéro de série	15:2E:EB:1A:B5:05:5A:85:47:82:9A:C0:2A:44:07:3E
Algorithme de signature de certificat	PKCS #1 SHA-256 avec le chiffrement RSA
Émetteur	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
Validité: Pas avant	2022-04-26 13 h 12 min 49 s HAE
Validité: Pas après	2023-04-25 13 h 12 min 49 s HAE
Objet	CN = www.desjardins.com O = Mouvement Desjardins L = Montreal ST = Quebec C = CA
Algorithme de clé publique du sujet	Chiffrement RSA PKCS #1
Clé public du sujet	Modulus (2048 bits):

	<p>C1 F3 8A 98 49 61 A0 F3 15 ED AE 99 02 07 8D B0 52 55 7D 90 A5 9A 32 21 A9 52 01 25 0A 6E 14 9B C9 31 C7 57 2B 1D 72 23 D7 F5 BE 9A 19 88 B5 65 28 C8 01 70 2B 2B 8F 69 41 0A 0B DF 30 86 24 CD AA 69 0E 5E 81 51 46 E1 16 13 4C 0F B8 A4 34 44 7F 38 9A B9 DD 5E C9 60 4D F0 48 56 3B 57 AD C9 9A E6 D1 DC B3 42 E6 E6 0E 6E 35 BC 34 58 AD 67 2C 82 E8 5F A1 53 DC 01 AB 46 9D 8E F7 CC DE 15 59 86 97 31 2B 71 4C 24 A3 79 F3 AD 07 68 D7 84 7E 85 D0 7D D4 62 37 FF F4 1A CF 21 D1 7D D1 42 2A 0B 81 98 FB 5C 7B 06 A6 DE 94 DA F5 05 4B FB A8 C6 A8 42 3A 1A 45 48 A0 D6 37 8A 67 29 76 A8 34 A2 74 53 92 E1 88 FE A5 76 26 E3 52 37 12 A7 7F 4D 82 72 62 E1 95 6E CB E7 89 14 DE F2 6C 5A B1 D6 32 85 20 D3 6E C7 C8 B0 E1 AF 9B 09 3F B3 0A F6 A3 91 45 B0 14 D3 85 4E AF CC B2 A6 70 6B</p> <p>Public Exponent (17 bits): 01 00 01</p>
Contrainte de base du certificat	<p>Critique N'est pas une autorité de certification</p>
ID de clé du sujet du certificat	<p>Non critique ID de clé : AB 32 84 41 E0 5D 2D 68 9B 93 16 4D E3 5B A7 AA 51 03 8F B8</p>
ID de clé de l'autorité de certification	<p>Non critique ID de clé : 82 A2 70 74 DD BC 53 3F CF 7B D4 F7 CD 7F A7 60 C6 0A 4C BF</p>
Accès aux informations de l'autorité	<p>Non critique Répondeur OCSP : URI : http://ocsp.entrust.net Émetteurs de l'autorité de certification : URI : http://aia.entrust.net/l1k-chain256.cer</p>
Point de distribution de la liste de révocation des certificats	<p>Non critique URI : http://crl.entrust.net/level1k.crl</p>
Autre nom de l'objet du certificat	<p>Non critique Nom DNS : www.desjardins.com Nom DNS : desjardins.com Nom DNS : zoneforus.com Nom DNS : www.mycard.desjardins.com Nom DNS : www.zoneforus.com Nom DNS : www.wscu.desjardins.com Nom DNS : www.posting.desjardins.com Nom DNS : www.mycard.desjardins.com Nom DNS : www.mfmc.formateurs.desjardins.com Nom DNS : www.mfmc.desjardins.com Nom DNS : www.mfmc-admin.desjardins.com Nom DNS : www.mastation-desjardins.com Nom DNS : www.lacabanedesjardins.fr Nom DNS : www.labfinance.desjardins.com</p>

	Nom DNS : www.jeunesse.desjardins.com
	Nom DNS : www.fondsdesjardins.com
	Nom DNS : www.financing.desjardins.com
	Nom DNS : www.femmesenmouvement.com
	Nom DNS : www.enflorideavecdesjardins.com
	Nom DNS : www.employeurd.com
	Nom DNS : www.desjardinsbank.com
	Nom DNS : www.desjardins.tv
	Nom DNS : www.desjardins.coop
	Nom DNS : www.dcrdesjardins.com
	Nom DNS : www.creditplus.desjardins.com
	Nom DNS : www.cooperathon.global
	Nom DNS : www.coastcapital.desjardins.com
	Nom DNS : www.caissedegatineau.com
	Nom DNS : www.bonusdollars.ca
	Nom DNS : www.bonidollars.com
	Nom DNS : www.bonidollars.ca
	Nom DNS : www.affichagemouvement.desjardins.com
	Nom DNS : www.accordd.desjardins.com
	Nom DNS : wscu.desjardins.com
	Nom DNS : webchapel.desjardins.com
	Nom DNS : tv.desjardins.ca
	Nom DNS : troussesfl.ca
	Nom DNS : tableau.mouvement.desjardins.com
	Nom DNS : static.mouv.desjardins.com
	Nom DNS : static.mouv.desjardins.ca
	Nom DNS : static.mouv.acadie.com
	Nom DNS : static.desjardins.com
	Nom DNS : savingsgoals.accesd.mouv.desjardins.com
	Nom DNS : savingsgoals.accesd.mouv.acadie.com
	Nom DNS : rrsp.desjardins.com
	Nom DNS : reer.desjardins.com
	Nom DNS : pvmobile.desjardins.com
	Nom DNS : prod-author-aem-www.desjardins.com
	Nom DNS : prod-aem-www.desjardins.com
	Nom DNS : posting.desjardins.com
	Nom DNS : portal.monetico.ca
	Nom DNS : portail.monetico.ca
	Nom DNS : partagedocuments.desjardins.com
	Nom DNS : paie.desjardins.com
	Nom DNS : owa-cld-teamconnect.desjardins.com
	Nom DNS : owa-cld-teamconnect-bi.desjardins.com
	Nom DNS : mysecurespace.desjardins.com
	Nom DNS : mycard.desjardins.com
	Nom DNS : mon.desjardins.ca
	Nom DNS : mobile.desjardins.com
	Nom DNS : mobile.desjardins.ca
	Nom DNS : mfmc.formateurs.desjardins.com
	Nom DNS : mfmc.desjardins.com
	Nom DNS : mfmc-admin.desjardins.com
	Nom DNS : mesprojets.accesd.mouv.desjardins.com
	Nom DNS : mesprojets.accesd.mouv.acadie.com
	Nom DNS : maboitesecurisee.desjardins.com
	Nom DNS : m.desjardins.com
	Nom DNS : lacabanedesjardins.fr
	Nom DNS : labfinance.desjardins.com
	Nom DNS : jeunesse.desjardins.com
	Nom DNS : itdm-sso.desjardins.com
	Nom DNS : gestion.troussesfl.ca
	Nom DNS : gestion.dfsinkit.ca
	Nom DNS : gestion.dfsikit.ca

	Nom DNS : fondsdesjardins.com Nom DNS : financing.desjardins.com Nom DNS : femmesenmouvement.desjardins.com Nom DNS : femmesenmouvement.com Nom DNS : enflorideavecdesjardins.com Nom DNS : empoweringwomen.desjardins.com Nom DNS : employeurd.com Nom DNS : dsf-dfs.ca Nom DNS : documentsharing.desjardins.com Nom DNS : dfsinkit.ca Nom DNS : dfsikit.ca Nom DNS : dfs-invest.mouv.desjardins.com Nom DNS : desjardins.tv Nom DNS : decs.desjardins.com Nom DNS : dcrdesjardins.com Nom DNS : creditplus.desjardins.com Nom DNS : cooperathon.global Nom DNS : coop.desjardins.com Nom DNS : cobrowse.desjardins.com Nom DNS : caissedegatineau.com Nom DNS : bonusdollars.ca Nom DNS : bonidollars.com Nom DNS : bonidollars.ca Nom DNS : blogues.desjardins.com Nom DNS : accordd.desjardins.com
Utilisation clé du certificat	Critique Signature Chiffrement de la clé
Utilisation amélioré de la clé	Non critique Authentification du serveur TLS WWW (OID.1.3.6.1.5.5.7.3.1) Authentification de client TLS WWW (OID.1.3.6.1.5.5.7.3.2)
Stratégies de certificat	Non critique OID.2.16.840.1.114028.10.1.5 : Pointeur d'instructions sur l'emploi de la certification : https://www.entrust.net/rpa OID.2.23.140.1.2.2
OID.1.3.6.1.4.1.1129.2.4.2	Non critique 04 82 01 6A 01 68 00 76 00 55 81 D4 C2 16 90 36 01 4A EA 0B 9B 57 3C 53 F0 C0 E4 38 78 70 25 08 17 2F A3 AA 1D 07 13 D3 0C 00 00 01 80 66 DC 88 93 00 00 04 03 00 47 30 45 02 21 00 FC E6 C6 F3 F5 E1 44 A8 AD 80 17 4A 0D 3F F9 0D 6A 2E AE 5F A4 32 23 1E A6 63 7E 9C 12 30 76 60 02 20 21 55 13 07 24 6C 19 C5 50 33 1B 11 A3 B1 57 CE 56 FD D7 5D C0 E9 62 6B 9C B8 86 63 AE 2F B6 00 00 76 00 B3 73 77 07 E1 84 50 F8 63 86 D6 05 A9 DC 11 09 4A 79 2D B1 67 0C 0B 87 DC F0 03 0E 79 36 A5 9A 00 00 01 80 66 DC 88 98 00 00 04 03 00 47 30 45 02 21 00 D6 9E 52 FE 50 A4 71 F5 90 44 2C 73 52 E7 D2 1A 56 40 32 67 46 4C 77 B0 A5 17 5B FF 71 6A F8 D7 02 20 13 C5 B1 DB 00 E7 99 88 BF D9 83 6F 0F 56 F3 F2 EF A4 EE 9C DD A8 87 2F 1F B7 C9 71 6D 52 DB D3 00 76 00 E8 3E D0 DA 3E F5 06 35 32 E7 57 28 BC 89 6B C9 03 D3 CB D1 11 6B EC EB 69 E1 77 7D 6D 06 BD 6E 00 00 01 80 66 DC 88 DB 00 00 04 03 00 47 30 45 02 21 00 B4 A6 35 3F 05 FA 99 C1 22 EC 71 01 D7 54 DB A9 F7 12 26 28 20 CA E0 4D DC 4A E6 72 C6 EE FB 2A 02 20 20 F3

	D2 99 43 43 A0 B4 E1 7C D5 DE E4 E3 A6 6E 57 91 D5 04 0B 5A 4B 13 AE 3D D4 0F B1 7F 25 E2
Valeur de la signature du certificat	3B E3 18 5C 4D E2 C1 53 86 91 29 CD DE 86 61 56 B3 D9 99 D0 05 C5 0A 07 19 E3 54 3E 13 3B B7 FC 11 2E 32 5A BC F3 26 5B 66 01 A0 FB FD 2B 94 39 2C D3 1B 23 0D 17 68 0A 38 31 1B A5 AD 0D 3E 46 58 5D 76 3D 96 8C C7 F4 B7 08 AC 63 F6 13 35 0F 6D B3 7A F4 55 C5 F8 5C 86 3A FC 9F C3 05 5F EE 7B 7F 84 7E 29 97 CB 10 AF EE E7 9D 75 5B 9F EB A9 93 D3 F2 FA 1E 41 BB FC FC 98 10 0D 41 00 7C A2 03 60 4D EF 18 6C 76 46 5F 8F 2B 97 9F 7B 06 DD 69 CB 7B 03 A4 6F 97 EA 3B 91 EC 1A CB 9A F8 27 EE 58 30 56 B0 C8 FC CD 0F 3E 12 29 9E F0 3E 3B CF 69 E4 E9 1C 72 59 C8 DB EF A3 9F 89 A8 FC 0B E6 09 65 AF 6E 2F B9 E5 37 06 44 E3 01 49 3D C1 F8 D9 51 FA 10 C8 A2 E4 60 08 7B CA D0 EB 02 D1 AD 25 57 57 8C E5 09 52 9A 7B 74 74 DF B8 2A DB 97 58 7E 97 98 AB 36 4F 74 91 28 DC 75 9E E6
Empreinte SHA-256	76 F3 74 10 0E AC 91 CF 15 8A 80 C3 AF 9E EE AC 46 39 29 AD A5 C4 97 C8 98 8E 26 92 59 1A EF C0
Empreinte SHA-1	E0 E3 E9 83 2D D7 60 31 5D F4 4E 31 1D 75 B8 A9 31 9B AF D3

Question D

Le navigateur va vérifier l'intégrité du certificat avec la clé publique qui lui est associée. Si les informations concordent, le certificat est valide et l'identité du site est vérifiée.

Source: Wikipedia, Wikimedia Foundation. (2022) Certificat électronique. [En ligne]. Disponible: https://fr.wikipedia.org/wiki/Certificat_%C3%A9lectronique

Question E

Un Certificat Authority (CA) est un tiers de confiance qui se charge de vérifier l'identité du site. Dans le cas de Desjardins, les deux CA du certificat sont Entrust Root Certification Authority - G2 et Entrust Certification Authority - L1K.

Il y a 2 CA pour avoir une plus grande chaîne de confiance, cela permet de vérifier aussi l'identité du 2ème CA. En haut de la hiérarchie on retrouve l'ancrage de confiance. Cela permet à l'utilisateur de vérifier plus rigoureusement la chaîne des certificats pour avoir plus de confiance dans le système.

Source: SSL.com. (2021) What Is a Certificate Authority (CA)?. [En ligne]. Disponible: <https://www.ssl.com/faqs/what-is-a-certificate-authority/>

Question F

Oui il peut être risqué d'accepter un CA dans notre navigateur car un CA frauduleux pourrait usurper la confiance d'un tiers de confiance en validant l'identité d'un site alors que c'est une usurpation.

Question G

Un certificat auto-signé, est signé directement par le site lui-même par opposition à un certificat émis par un tiers de confiance. Ainsi il ne garantit pas l'identité du système, n'importe qui peut en créer un.

Source: Wikipedia, Wikimedia Foundation. (2022) Certificat autosigné. [En ligne]. Disponible: https://fr.wikipedia.org/wiki/Certificat_autosign%C3%A9

Question 3

Question A

L'image chiffrée n'apporte pas de sécurité supplémentaire, en effet il est encore possible de lire clairement le mot de passe sur celle-ci. Cela s'explique par le fait qu'avec le mode ECB, un même bloc est chiffré de la même manière. Ainsi l'ensemble des pixels verts formant le texte dans l'image en clair ont été chiffrés de la même manière dans l'image chiffrée.



Question B

Avec le mode CBC, il n'est plus possible de lire le mot de passe dans l'image chiffrée. En effet, le vecteur d'initialisation permet d'éviter le problème abordé dans la question A. Ainsi un même pixel sera chiffré d'une façon différente (avec très haute probabilité).



Question C

Le mode d'opération est un facteur très important de sécurité pour les algorithmes de chiffrement par bloc. L'administrateur pensait avoir protégé ses mots de passe car il utilisait un algorithme de chiffrement robuste (AES 256) mais l'utilisation de celui-ci ne protégeait en fait pas ses mots de passe. Il est donc important d'utiliser un mode d'opération qui maximise la sécurité. Par ailleurs, le choix du vecteur d'initialisation est aussi important, il doit être généré avec une entropie maximale et doit changer pour chaque message.

Question 4

Question A

Le fichier `/etc/passwd` gère les informations des comptes sur le système. Le fichier contient un champ de mot de passe mais la valeur de celui-ci est `x` ce qui indique que le mot de passe chiffré est stocké dans `/etc/shadow`.

Le fichier possède les permissions de lecture pour tout le monde mais seul le propriétaire du fichier (root) peut y écrire. La lecture est autorisée car le fichier contient de nombreuses informations utiles (nom d'utilisateur, mot de passe, identifiant utilisateur, identifiant du groupe, répertoire personnel, etc.) sur les comptes.

```
(kali@kali)~$ cat /etc/passwd
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
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-network:x:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:103:110:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:104:111::/nonexistent:/usr/sbin/nologin
tss:x:105:113:TPM software stack,,,:/var/lib/tpm:/bin/false
strongswan:x:106:65534::/var/lib/strongswan:/usr/sbin/nologin
tcpdump:x:107:114::/nonexistent:/usr/sbin/nologin
usbmux:x:108:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:x:109:65534::/run/ssh:/usr/sbin/nologin
dnsmasq:x:110:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
avahi:x:111:117:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
rtkit:x:112:118:RealtimeKit,,,:/proc:/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: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
colord:x:119:127:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
mysql:x:120:128:MySQL Server,,,:/nonexistent:/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:122:130::/var/lib/geoclue:/usr/sbin/nologin
Debian-snmpp:x:123:131::/var/lib/snmpp:/bin/false
ssllh:x:124:132::/nonexistent:/usr/sbin/nologin
ntpsec:x:125:135::/nonexistent:/usr/sbin/nologin
redsocks:x:126:136::/var/run/redsocks:/usr/sbin/nologin
rwhod:x:127:65534::/var/spool/rwho:/usr/sbin/nologin
iodine:x:128:65534::/run/iodine:/usr/sbin/nologin
miredo:x:129:65534::/var/run/miredo:/usr/sbin/nologin
statd:x:130:65534::/var/lib/nfs:/usr/sbin/nologin
postgres:x:131:138:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
inetsim:x:132:140::/var/lib/inetsim:/usr/sbin/nologin
king-phisher:x:133:142::/var/lib/king-phisher:/usr/sbin/nologin
kali:x:1000:1000:::/home/kali:/usr/bin/zsh
```

Question B

Après la création de l'utilisateur INF4420A_TP1, la ligne suivante est ajoutée dans /etc/passwd : INF4420A_TP1:x:1001:100::/home/INF4420A_TP1:/bin/bash

Et la ligne suivante est ajoutée dans /etc/shadow :
INF4420A_TP1:!:19269:0:99999:7:::

Les 2 fichiers ont été modifiés, le fichier /etc/passwd afin de créer le nouvel utilisateur et le fichier /etc/shadow afin de contenir le futur mot de passe.

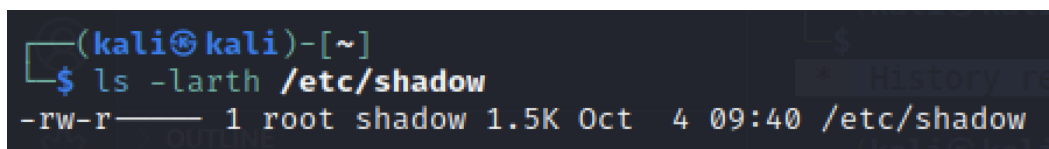
Question C

Après avoir changé le mot de passe du compte INF4420A_TP1 avec le mot de passe password la ligne correspondante dans le fichier /etc/shadow a été changé pour :

```
INF4420A_TP1:$y$j9T$TQpyBOVOE3CFVDCPYajr.$3EUux/AcS1biYpJAjWpQJIRUyL.YHmKHtYkZVWIQjh/:19269:0:99999:7:::
```

C'est le fichier /etc/shadow qui contient réellement l'information sur le mot de passe. Le fichier /etc/passwd n'a pas été modifié car il ne contient que des informations sur le compte qui n'ont pas changé.

Les permissions du fichier /etc/shadow autorisent la lecture uniquement pour l'utilisateur root et pour son groupe et l'écriture uniquement pour l'utilisateur root. Ainsi seuls des utilisateurs autorisés par l'administrateur peuvent consulter le fichier et seul l'administrateur peut le modifier.



```
(kali㉿kali)-[~]  
$ ls -l /etc/shadow  
-rw-r----- 1 root shadow 1.5K Oct  4 09:40 /etc/shadow
```

Question D

Après avoir changé le mot de passe du compte INF4420A_TP1 pour le même mot de passe password la ligne correspondante dans le fichier /etc/shadow a été changé pour

```
INF4420A_TP1:$y$j9T$vtoc7O9C1QKVpgQpJqg97.$p1GKTXjE2FXIH05YbMGj71NYk1o5Lorz/suINyuTak/:19269:0:99999:7:::
```


Les mots de passe sont hachés avec l'algorithme yescrypt (\$y\$) avec les paramètres j9T. La 3ème partie (séparées par \$) représente le sel du mot de passe. Ce sel est différent à chaque changement ce qui explique que les valeurs soient différentes bien que le mot de passe soit identique. En effet, lors du premier changement, le sel était TQpyBOVOE3CFVDCPYajr. et lors du deuxième c'est vtoc7O9C1QKVpgQpJqg97.

Question E

Après création de l'utilisateur INF4420A_TP1_e et remplacement du mot de passe par défaut par la valeur du champ mot de passe de l'utilisateur INF4420A_TP1 il est possible de se connecter au compte INF4420A_TP1_e avec le mot de passe de INF4420A_TP1.

En effet, la valeur du champ de mot de passe ne contient aucune information strictement propre à l'utilisateur. Lors de la connexion avec INF4420A_TP1_e, le système a cherché la valeur du sel dans /etc/shadow, a haché le mot de passe entré avec le sel correspondant et a bien retrouvé le hash stocké dans /etc/shadow.

Question F

Après suppression du compte INF4420_TP1_e, les lignes correspondantes sont supprimées dans les fichiers /etc/passwd et /etc/shadow.

Question 5

Question A

En utilisant le logiciel “John The Ripper” avec le dictionnaire `rockyou.txt` sur le fichier `Password_File.txt` on retrouve les mots de passe suivants :

```
simple:123456789  
brian:sunshine  
action:monkey  
vladimir:liverpool
```

Question B

Pour l'alphabet $[a-zA-Z]$, l'entropie maximale est $\log_2(2 \times 26) = 5.70$

Pour l'alphabet $[a-zA-Z0-9]$, l'entropie maximale est $\log_2(2 \times 26 + 10) = 5.95$

Pour l'ensemble de la table ASCII, l'entropie maximale est $\log_2(256) = 8$

Question C

Une grande entropie est un critère important pour qu'un mot de passe soit fort.

Question D

- Une grande taille de mot de passe (afin d'augmenter l'entropie générale)
- Éviter les mots de passe communs (dictionnaires, information liée à l'utilisateur, mots de passe classiques, etc.)
- Éviter d'avoir des séquences de caractères qui se répètent dans le mot de passe

Question E

Si on utilise le même mot de passe partout, cela simplifie la tâche de l'attaquant car s'il casse notre mot de passe il a accès à toutes nos informations de connexion sur différents systèmes.

Partie C

Question 1

Question A

Ève peut facilement effectuer une attaque à texte clair choisi. En effet, elle dispose de la clé publique et il n'y a que 26 possibilités par chaque caractère chiffré.

Ève peut donc intercepter les messages chiffrés (Man in The Middle) puis pour chaque caractère chiffré, elle essaye de chiffrer toutes les lettres (A à Z) et si une lettre chiffrée est égale au caractère intercepté alors elle sait à quelle lettre celui-ci correspond.

Pour optimiser les déchiffrements futurs, Ève peut même construire une table de correspondance qui lui permettra de déchiffrer les futurs messages chiffrés en temps constant.

Question B

La clé publique RSA associé au matricule 1984533 est ($e = 449, N = 105419$). Pour chaque symbole du texte {90592, 46646, 87476, 98298, 10420}, on essaie de chiffrer toutes les lettres de A à Z à la recherche d'une correspondance. Avec cette technique, nous avons déchiffré le message en clair qui est *RITES*.

```
e = 449
N = 105419
C = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
m = [90592, 46646, 87476, 98298, 10420]
for c in m:
    for x in range(26):
        if (x ** e) % N == c:
            print(x, C[x])
```

```
(kali@kali) - [~/INF4420A/TP1]
$ /bin/python /home/kali/INF4420A/TP1/RSA.py
17 R
8 I
19 T
4 E
18 S
```

Question C

Un caractère chiffré 0 correspond à la lettre A en effet $0^e \bmod N = 0$ et un caractère chiffré 1 correspond à la lettre B en effet $1^e \bmod N = 1$, et ce quelque soit la clé publique.

Pour assurer une bonne sécurité, il faudrait choisir un meilleur codage pour les lettres afin d'éviter l'exploitation de certaines propriétés mathématiques pour déchiffrer le texte.

Il faudrait aussi considérer le chiffrement par bloc afin de complexifier (voire de rendre techniquement non envisageable) l'attaque à texte clair choisi.

Question 2

Le texte à déchiffrer correspondant au matricule 1984533 est :

RFOBGJFBATGFVISSFLMDGUFOMFSGTJFDAUFKMWKSICGFLGAKGFKMWOUAKOFASSNAWKGRF
GROAESNRBFKMXGUKGFAWCFOMFCMFASSFMOBGUFAKORFAWCFOBW@RFDNBKBFNWCGLGWC
GWOFROAOGRFXAJFMVFUN@BOFCMFFFAWCFVMUFOBGFRIILLMUOFMVFOBNRFCGKSA

En analysant les fréquences de chaque symbole, le symbole le plus fréquence est “F” avec une fréquence relative de 18.5%. En anglais le symbole “espace” est 1.07 fois plus fréquent que la lettre “E” avec une fréquence de 13.59% ($1.07 * 12.702$). Nous posons donc l’hypothèse que le symbole “F” représente le symbole “espace”.

En analysant à nouveau les fréquences sans l’espace nous obtenons les tableaux suivants :

Symbole	Fréquence	Symbole	Fréquence
G (E)	0.1104	N (I)	0.0429
O (T)	0.0982	L (P)	0.0307
A (A)	0.092	V (F)	0.0245
M (O)	0.0798	J (Y)	0.0184
K (C)	0.0675	I (U)	0.0184
R (S)	0.0613	D (W)	0.0184
B (H)	0.0613	X (M)	0.0184
S (L)	0.0613	T (V)	0.0123
W (N)	0.0613	@ (G)	0.0123
C (D)	0.0552	E (B)	0.0061
U (R)	0.0491		

Tableau 1 : Fréquences relatives des symboles dans le texte chiffré

Digramme	Fréquence	Digramme	Fréquence
OB (TH)	0.0394	AK (A.)	0.0236
WC (ND)	0.0394	KG (.E)	0.0236
CG (DE)	0.0315	GR (E....)	0.0236
AW (AN)	0.0315	BN (H....)	0.0236
BG (HE)	0.0236	LM	0.0157
SS	0.0236	OM (TO)	0.0157
GU (.....)	0.0236	MW	0.0157
KM (.O)	0.0236	WK	0.0157

Tableau 2 : Fréquences relatives des digrammes dans le texte chiffré

Trigramme	Fréquence	Trigramme	Fréquence
OBG	0.0326	ROA	0.0217
AWC	0.0326	OBN	0.0217
KMW	0.0217	WCG	0.0217
AKO	0.0217	BGJ	0.0109
ASS	0.0217	BAT	0.0109

Tableau 3 : Fréquences relatives des trigrammes dans le texte chiffré

La fréquence relative du symbole “G” (11.04%) est similaire à celle du symbole “E” en anglais (12.702%). Il en est de même pour les symboles “O” (9.82%) et “T” (9.056%), “A” (9.2%) et “A” (8.167%) ainsi que “M” (7.98%) et “O” (7.507%).

En analysant les fréquences relatives des digrammes et trigrammes, le digramme le plus fréquent est “OB” (3.94%) et le trigramme le plus fréquent est “OBG” (3.26%). Sachant que “O” correspond à “T” et “G” à “E” on peut déduire que “B” correspond à “H”. Leur fréquences relatives correspondent (6.13% pour “B” et 6.094% pour “H”).

En analysant les fréquences relatives des digrammes et trigrammes, les digrammes “WC” (3.94%) et “AW” (3.15%) ainsi que le trigramme “AWC” (3.26%) ont des fréquences relatives élevées. En sachant que le symbole “A” correspond à au symbole “A”, on pose l’hypothèse que “W” correspond à “N” (6.13% et 6.749%) et “C” correspond à “D” (5.52% et 4.253%).

En analysant le texte chiffré et en appliquant les substitutions déjà trouvées nous remarquons que le mot "OTHEU" ressemble fortement au mot anglais "OTHER". Nous supposons ainsi que "R" (5.987%) est chiffré en "U" (4.91%).

En analysant le texte chiffré et en appliquant les substitutions déjà trouvées nous remarquons que le mot "THEJ" ressemble fortement au mot anglais "THEY". Nous supposons ainsi que "J" (1.84%) est chiffré en "Y" (1.974%).

En analysant le texte chiffré et en appliquant les substitutions déjà trouvées nous remarquons que les mots "THNN@R" et "RN@HT" ressemblent fortement aux mots anglais "THINGS" et "RIGHTS". Nous supposons ainsi que "I" (6.966%) est chiffré en "N" (4.29%) et que "G" (2.015%) est chiffré en "@" (1.23%).

La fréquence du symbole "R" (6.13%) est très proche de celle du symbole "S" (6.327%) et on retrouve beaucoup ce symbole en fin de mots. On peut donc supposer que "S" est chiffré en "R".

Avec les substitutions déjà établies nous obtenons l'extrait "AND TO DO ASS OTHER", nous pouvons donc supposer que "S" (6.13%) correspond à "L" (4.025%). De même pour l'extrait "INDELENDENT STATES" où "L" (3.07%) correspond sûrement à "P" (1.929%).

En analysant le texte chiffré et en appliquant les substitutions déjà trouvées nous remarquons que le mot "SIPPORT" ressemble fortement au mot anglais "SUPPORT". Nous supposons ainsi que "U" (2.758%) est chiffré en "I" (1.84%).

En analysant le texte chiffré et en appliquant les substitutions déjà trouvées nous remarquons que le mot "ALLIANKES" ressemble fortement au mot anglais "ALLIANCES". Nous supposons ainsi que "C" (2.782%) est chiffré en "K" (6.75%). Nous observons une divergence statistique significative mais la validité de la substitution est confirmée par une amélioration de la sémantique du texte.

En analysant le texte chiffré et en appliquant les substitutions déjà trouvées nous remarquons que le mot "ESTAEELISH" ressemble fortement au mot anglais "ESTABLISH". Nous supposons ainsi que "B" (1.492%) est chiffré en "E" (0.61%).

L'extrait "AND VOR THE SUPPORT OV THIS DECLA" laisse deviner une correspondance entre "V" (2.45%) et "F" (2.228%).

En suivant une logique similaire d'analyse du texte avec les différentes substitutions nous obtenons des correspondances entre "D" (1.84%) et "W" (2.360%) avec "DHICH", "X" (1.84%) et "M" (2.406%) avec "COXXERCE" et "T" (1.23%) et "V" (0.978%) avec "HATE".

Nous avons trouvé toutes les substitutions et le texte déchiffré a un sens donc nous pensons avoir trouvé la clé de chiffrement (table de substitution).

Le texte est un extrait de la Déclaration d'Indépendance des États-Unis signée le 4 juillet 1776 :

S THEY HAVE FULL POWER TO LEVY WAR CONCLUDE PEACE CONTRACT ALLIANCES
ESTABLISH COMMERCE AND TO DO ALL OTHER ACTS AND THINGS WHICH
INDEPENDENT STATES MAY OF RIGHT DO AND FOR THE SUPPORT OF THIS
DECLA