Devoir 1

Notes

Le code présenté est retrouvable sur ssh://bigblase.xyz:/srv/git/crypto1
mot de passe: `FDS8EbKiDNoJh2QN`
Le code a été testé sous linux, kernel 6.5 et librairie à jour (sept 2023), en utilisant gcc 13.2.1. Pour compiler, il faut d'abord créer le dossier *build* à la racine du projet. Pour tester, il faut faire `make run part=<NUMERO PARTIE>`. Il *devrait* être portable (C99, POSIX compliant)

I) Chiffre de César

On lance le code si dessus 3 fois: (`make run part=1`) et l'on observe les sorties. On obtient quelque chose comme:

```
-------PART 1-------
       key: 5
msg before: coding w/ freebsd style :)
  cypher: htinsl b/ kwjjgxi xydqj :)
 msg after: coding w/ freebsd style :)
       key: 5
msg before: ceciestlemessageclairadechiffre
  cypher: hjhnjxyqjrjxxfljhqfnwfijhmnkkwj
 msg after: ceciestlemessageclairadechiffre

-------PART 1-------
       key: 15
msg before: ceciestlemessageclairadechiffre
  cypher: rtrxthiatbthhpvtrapxgpstrwxuugt
 msg after: ceciestlemessageclairadechiffre

-------PART 1-------
       key: 3
msg before: ceciestlemessageclairadechiffre
  cypher: fhflhvwohphvvdjhfodludghfkliiuh
 msg after: ceciestlemessageclairadechiffre
```

J'ai laissé le premier message pour monter que l'on peut quand même envoyer des symboles. Je l'ai enlevé après par soucis de place / redondance.

De leurs côtés, Alice et Bob voient la clef key, le message msg et le cryptogramme cypher.

I) a)

Dans cette situation, Eve recevrait uniquement le cypher. On voit que quand on a des symboles autres que des lettres, il devient facile de voir des motifs, des mots. De plus, une analyse fréquentielle nous montreraient qu'il peut s'agir de texte français. Comme Eve n'a pas la clef, elle pourrait alors éssayer

de bruteforce toutes les clefs du chiffrement de césar, ce que lui retournerait une solution rapidement (il n'y a que 26 clefs possible).

II) OTP

II) a)

Comme on veut coder un message en char (1 octet) sur 512 bits (64 octets), j'ai pris la liberté de tester avec un message de cette taille, plutot que 32 octets.

Pour faire l'OTP, on fait un `XOR` entre le message et une chaine de 512 bits aléatoire. En faisant ceci, on doit faire attention de bien terminer la string avec un nullbyte, pour que notre programme fonctionne bien. Cela nous laisse donc 63 octets pour faire un message.

J'ai représenté dans mon programme les valeurs des cryptogrammes et messages en clair ainsi que leur représentation en hexa, pour pouvoir comparer et montrer que nous avons bien fait un `XOR`. Je n'ai pas utilisé de forme binaire, car il n'existe pas de façon portable de le faire en C, et le faire à la main retournait des blocks de taille variable: certaines valeurs étaient comprises comme des `int` plutôt que des `char`

```
-------PART 2-------
key: 9@5@wCttyy@6@@j5~q3F@@@J 2#hjnFn@V□@
                                                key as hex: 89 39 13 ED DD 35 DE 70 77 7F 43
B5 74 74 79 79 E6 03 36 C6 D2 06 DB 78 6A 35 A8 7E 71 7F 33 FA B9 46 E7 96 7B C5 07 F2
44 4A A7 B9 BE 20 32 A4 23 68 6A F6 6E 46 6E D8 7B 17 56 EC 96 89 E6
msg before: ceciestlemessageclairadechiffreilcomporte64charcestplutotlongxd
msg as hex: 63 65 63 69 65 73 74 6C 65 6D 65 73 73 61 67 65 63 6C 61 69 72 61 64 65 63
68 69 66 66 72 65 69 6C 63 6F 6D 70 6F 72 74 65 36 34 63 68 61 72 63 65 73 74 70 6C 75
74 6F 74 6C 6F 6E 67 78 64
V@pher: @pF&@oWg  ]
  u!|@A@@{9@
cypher as hex: EA 5C 70 84 B8 46 AA 1C 12 12 26 C6 07 15 1E 1C 85 6F 57 AF A0 67 BF 1D
09 5D C1 18 17 0D 56 93 D5 25 88 FB 0B AA 75 86 21 7C 93 DA D6 41 40 C7 46 1B 1E 86 02
33 1A B7 0F 7B 39 82 F1 F1 82
msg after: ceciestlemessageclairadechiffreilcomporte64charcestplutotlongxd

-------PART 2-------
key: @s@~@
key as hex: B8 88 19 D5 CC 04 8D 73 F2 8F C8 AA F7 A3 7E C4 42 BA A9 07 08
msg before: ceciestlemessageclairadechiffreilcomporte64charcestplutotlongxd
msg as hex: 63 65 63 69 65 73 74 6C 65 6D 65 73 73 61 67 65 63 6C 61 69 72 61 64 65 63
68 69 66 66 72 65 69 6C 63 6F 6D 70 6F 72 74 65 36 34 63 68 61 72 63 65 73 74 70 6C 75
74 6F 74 6C 6F 6E 67 78 64
cypher: @zw@@!@nza@S*r×@R@@g□@@)0U@b
cypher as hex: DB ED 7A BC A9 77 F9 1F 97 E2 AD D9 84 C2 19 A1 21 D6 C8 6E 7A 61 EB 4E
B7 88 1D 53 05 2A FB 72 8C D4 9F C1 CC 11 52 DA 68 DE 6C 67 E3 B6 BB AD F4 01 A1 EA 1E
11 B1 29 30 55 14 C9 F5 62 A7
msg after: ceciestlemessageclairadechiffreilcomporte64charcestplutotlongxd

-------PART 2-------
key: "K4*_aBx@n@d 0cT@V?HjJ@@I
                               Q;F@!@
                                     key as hex: 91 01 22 4B AD 34 2A 1A 01 5F 9A 87 61
42 78 E1 02 6E EB 55 64 C1 11 20 8D 30 91 81 63 54 F2 F4 56 14 3F 03 48 6A 1D 4A C9 B7
D1 2A F9 49 0C FB B7 F7 51 1B B9 62 3B 46 92 CD C8 F6 21 BA EA
msg before: ceciestlemessageclairadechiffreilcomporte64charcestplutotlongxd
```

```
msg as hex: 63 65 63 69 65 73 74 6C 65 6D 65 73 73 61 67 65 63 6C 61 69 72 61 64 65 63
68 69 66 66 72 65 69 6C 63 6F 6D 70 6F 72 74 65 36 34 63 68 61 72 63 65 73 74 70 6C 75
74 6F 74 6C 6F 6E 67 78 64
cypher: ?A"?^vd2?#a<uE??&:wPn8o>?(~%k?O)桧F
cypher as hex: F2 64 41 22 C8 47 5E 76 64 32 FF F4 12 23 1F 84 61 02 8A 3C 16 A0 75 45
EE 58 F8 E7 05 26 97 9D 3A 77 50 6E 38 05 6F 3E AC 81 E5 49 91 28 7E 98 D2 84 25 6B D5
17 4F 29 E6 A1 A7 98 46 C2 8E
msg after: ceciestlemessageclairadechiffreilcomporte64charcestplutotlongxd
```

Dans ces simulations, Alices voit le message, la clé ainsi que le cryptogramme qu'elle génère.

Eve ne voit passer que le cryptogramme.

Bob connait la clef, voit le cryptogramme arriver et peut donc ainsi déchiffrer le message.

Dans cette situation, Eve ne connait pas la clef, et voit passer du "texte" pseudo random ($x \oplus y$ est random si x ou y sont random)

III) Chiffrement de Vignère (César modifié)

On prend un chiffrement de césar, et l'on associe au message une clef de la meme taille que le message. On remplace chaque lettre du message par la lettre à la même position de la clef, ceci nous donne le cryptogramme.

III) a)

```
                              -------PART 3-------
                         key: bsgwzacemwtninxuivfczfjltdqufmw
key as hex: 62 73 67 77 7A 61 63 65 6D 77 74 6E 69 6E 78 75 69 76 66 63 7A 66 6A 6C 74
                                  64 71 75 66 6D 77
                      msg before: ceciestlemessageclairadechiffre
msg as hex: 63 65 63 69 65 73 74 6C 65 6D 65 73 73 61 67 65 63 6C 61 69 72 61 64 65 63
                                  68 69 66 66 72 65
                        cypher: dwiedsvpqixfandykgfkqfmpvkyzkda
cypher as hex: 64 77 69 65 64 73 76 70 71 69 78 66 61 6E 64 79 6B 67 66 6B 71 66 6D 70
                                  76 6B 79 7A 6B 64 61
                       msg after: ceciestlemessageclairadechiffre

                              -------PART 3-------
                         key: yzaofkhwqqhnyroxdchwyxjfimvgbpf
key as hex: 79 7A 61 6F 66 6B 68 77 71 71 68 6E 79 72 6F 78 64 63 68 77 79 78 6A 66 69
                                  6D 76 67 62 70 66
                      msg before: ceciestlemessageclairadechiffre
msg as hex: 63 65 63 69 65 73 74 6C 65 6D 65 73 73 61 67 65 63 6C 61 69 72 61 64 65 63
                                  68 69 66 66 72 65
                        cypher: adcwjcahuclfqrubfnhepxmjktdlggj
cypher as hex: 61 64 63 77 6A 63 61 68 75 63 6C 66 71 72 75 62 66 6E 68 65 70 78 6D 6A
                                  6B 74 64 6C 67 67 6A
                       msg after: ceciestlemessageclairadechiffre

                              -------PART 3-------
                         key: ythzjprrtodekptxpmaslwezmdqfjkm
key as hex: 79 74 68 7A 6A 70 72 72 74 6F 64 65 6B 70 74 78 70 6D 61 73 6C 77 65 7A 6D
                                  64 71 66 6A 6B 6D
                      msg before: ceciestlemessageclairadechiffre
msg as hex: 63 65 63 69 65 73 74 6C 65 6D 65 73 73 61 67 65 63 6C 61 69 72 61 64 65 63
                                  68 69 66 66 72 65
                        cypher: axjhnhkcxahwcpzbrxaacwhdokykobq
```

```
cypher as hex: 61 78 6A 68 6E 68 6B 63 78 61 68 77 63 70 7A 62 72 78 61 61 63 77 68 64
                                  6F 6B 79 6B 6F 62 71
                        msg after: ceciestlemessageclairadechiffre
```

Comme d'habitude, Alice voit le message, la clé ainsi que le cryptogramme qu'elle génère.

Eve ne voit passer que le cryptogramme.

Bob connait la clef, voit le cryptogramme arriver et peut donc ainsi déchiffrer le message.

Cependant, ici une analyse fréquentielle ne marche plus, mais ce n'est pas le seul moyen de détecter ce genre de cryptosystème. De plus, il est très fragile, il ne faut alors pas l'utiliser pour de réeles applications.

C'est aussi moins éfficace que celui de la question 2, car maintenant le texte est beaucoup moins aléatoire (la taille de clef est aussi beaucoup plus petite)

III) a) i) Eve's drop

```
                        -------PART3C-------
                   key: vwzxsofxcakcshixrqupivteyrcamhr
                msg before: ceciestlemessageclairadechiffre
                  cypher: xabfwgyigmoukhobtbuxzvwiaykfryv
cypher tempered by Eve: xabfwgyigmoukhobtbuxzvwiaykfryv
               msg after: ceciestlemessageclairadechiffre

                        -------PART3C-------
                   key: xagobxaskkpgmakdbcjgrogwkdfxzvp
                msg before: ceciestlemessageclairadechiffre
                  cypher: zeiwfptdowtyeaqhdnjoiojamkncemt
cypher tempered by Eve: zeiwfptdowtyeaqhdnjoiojamkncemt
               msg after: ceciestlemessageclairadechiffre

                        -------PART3C-------
                   key: kszazggktxfkhcmhuccvkpbsdzlsdso
                msg before: ceciestlemessageclairadechiffre
                  cypher: mwbidyzvxjjczcslwncdbpewfgtxijs
cypher tempered by Eve: mwbidyzvxjjczcslwnldbpewfgtxijs
               msg after: ceciestlemessagecljiradechiffre
```

Dans le premier cas, le cryptogramme n'avait pas de 'C', donc rien n'a changé. Le deuxieme en a $1 \leq 3$, ainsi rien ne change pour lui. En revanche, le troisième (j'ai triché pour en trouver un, j'ai fait plus que 3 éssais) en a 3: Bob voit alors une lettre changer. Bob peut de ce fait penser que le message a été manipulé, mais c'est assez négligeable que l'on pourrait penser qu'un bit a pu s'inverser au cours du trajet (par exemple, si l'on n'a pas de vérification de checksum de paquets, ie si l'on utilise UDP).

On voit, par contre, qu'Eve ne change que dans un nombre minime de cas. En effet, on veut 3 C ayant chacun une probabilité de $\frac{1}{26}$ d'apparaitre, pour 32 lettres total: $P = \left(\frac{1}{26}\right)^3 * \left(\frac{25}{26}\right)^{29} = 1.82 \cdot 10^{-5}$