Stéganographie : cacher des messages dans du son

Dans quelle mesure des modifications de fichiers restent-elles inaudibles tout en étant utilisables en cryptologie ?

17 juin 2024

B. GOURANTON, S. OZEROV, U. THOMAS

Introduction

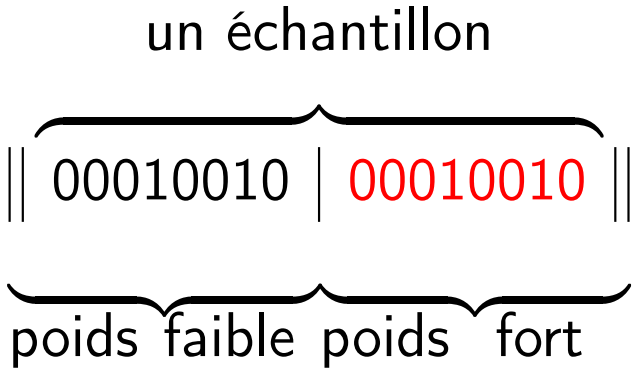
But

Principe

Méthode

Table des matières :

I - Principe



Entrées :

message (m)

fichier de départ (dep)

fichier vide (fv)

Recopiage de l'entête

pour *i allant de 1 à longueur(dep)* **faire**

  lire un échantillon E

  si $i > longueur(m)$ **alors**

    écrire E dans fv

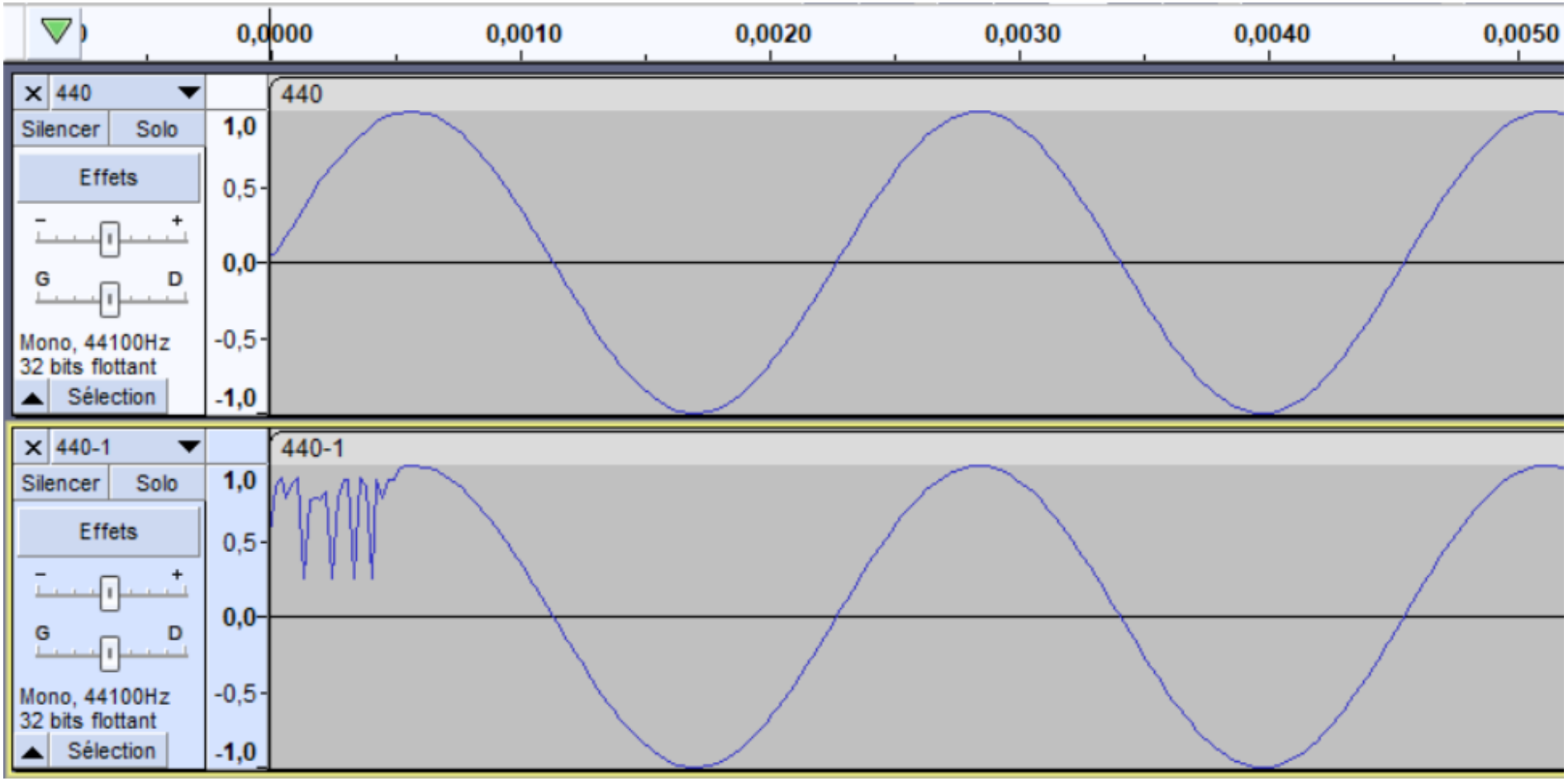
  sinon

    écrire $(E[0], m[i])$ dans fv

  fin

fin

I - Résultats


0,0000
0,0010
0,0020
0,0030
0,0040
0,0050
440
Silencer
Solo
Effets
G
D
Mono, 44100Hz
32 bits flottant
Sélection
1,0
0,5
0,0
-0,5
-1,0
440-1
Silencer
Solo
Effets
G
D
Mono, 44100Hz
32 bits flottant
Sélection
1,0
0,5
0,0
-0,5
-1,0

II - Principe

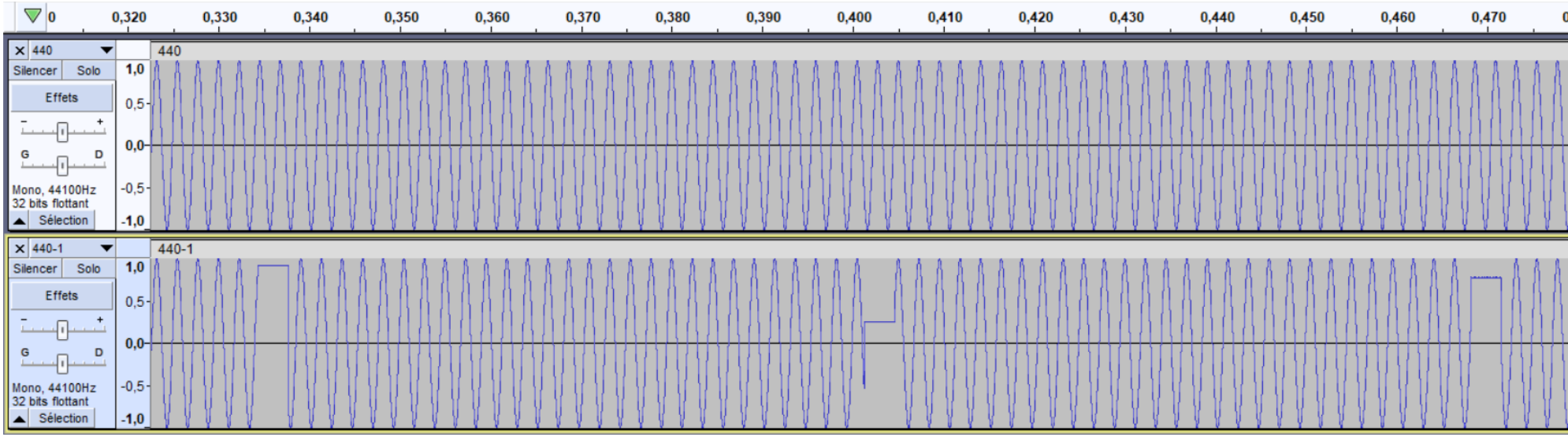
Entrées : message (m), nombre de répétitions de chaque caractère (rep), espacement des caractères (esp), fichier de départ (dep), fichier vide (fv)

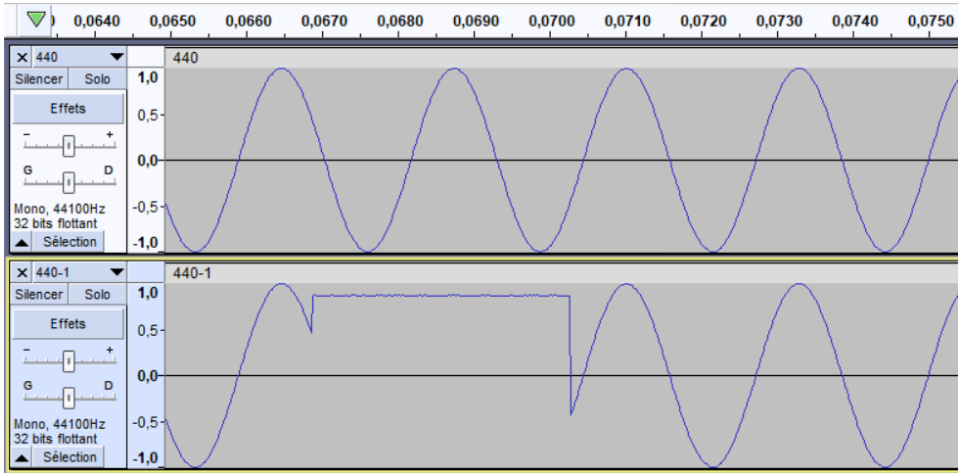
Recopiage de l'entête

pour i *allant de* 1 *à* $longueur(dep)$ **faire**
- **si** $i < longueur(m)$ **alors**
  - **répéter**
    - lire un échantillon E
    - écrire $(E[0], m[i])$ dans fv
  - **jusqu'à** rep;
  - **répéter**
    - lire un échantillon E
    - écrire E dans fichier vide
  - **jusqu'à** esp;
- **sinon**
  - lire un échantillon E puis écrire E dans fv
- **fin**

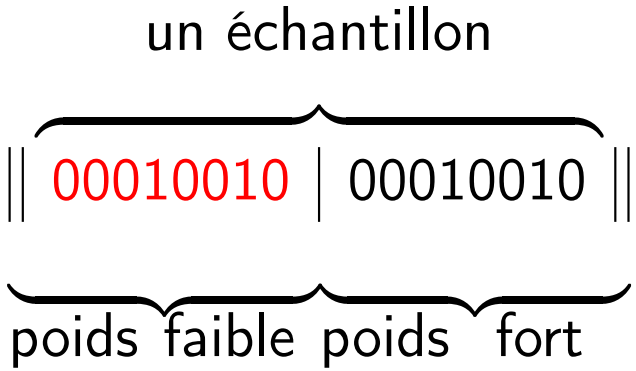
fin

II - Résultats





Principe



Entrées :

message (m)

fichier de départ (dep)

fichier vide (fv)

Recopiage de l'entête

pour *i allant de 1 à longueur(dep)* **faire**

  lire un échantillon E

  si $i > longueur(m)$ **alors**

    écrire E dans fv

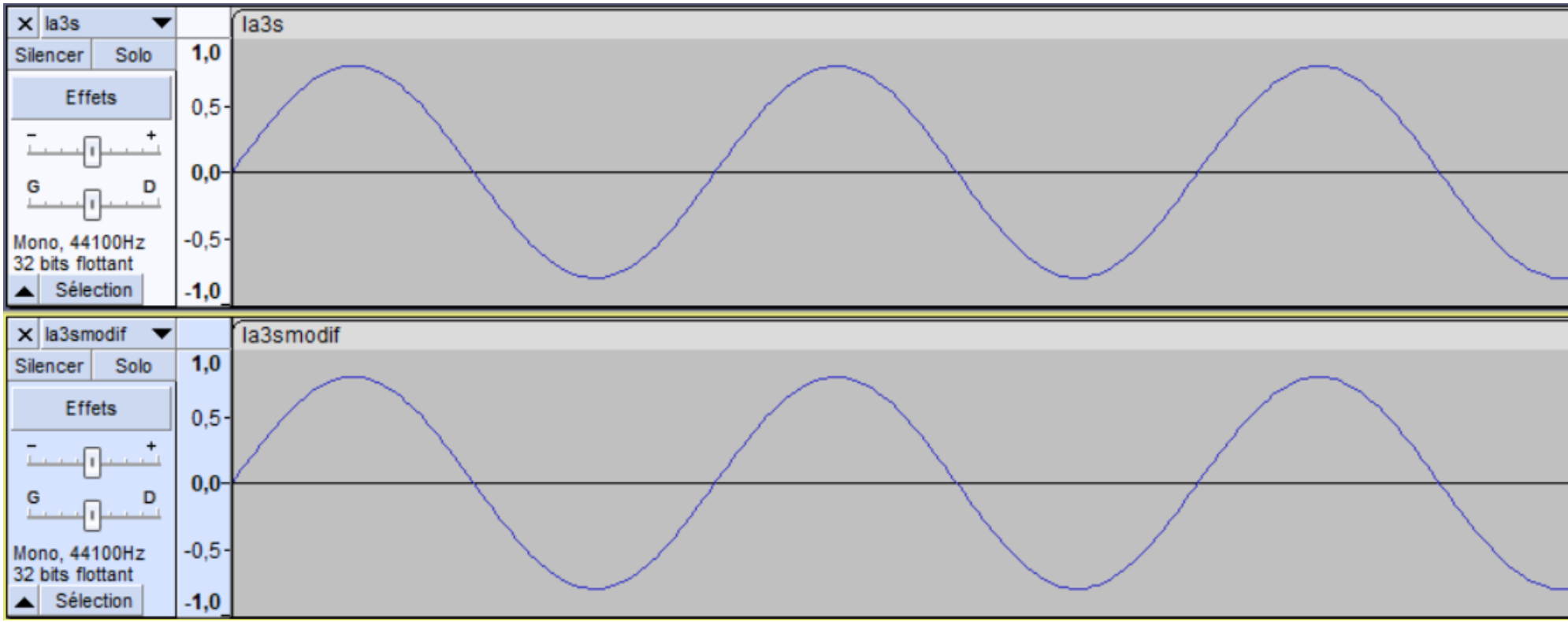
  sinon

    écrire $(m[i], E[1])$ dans fv

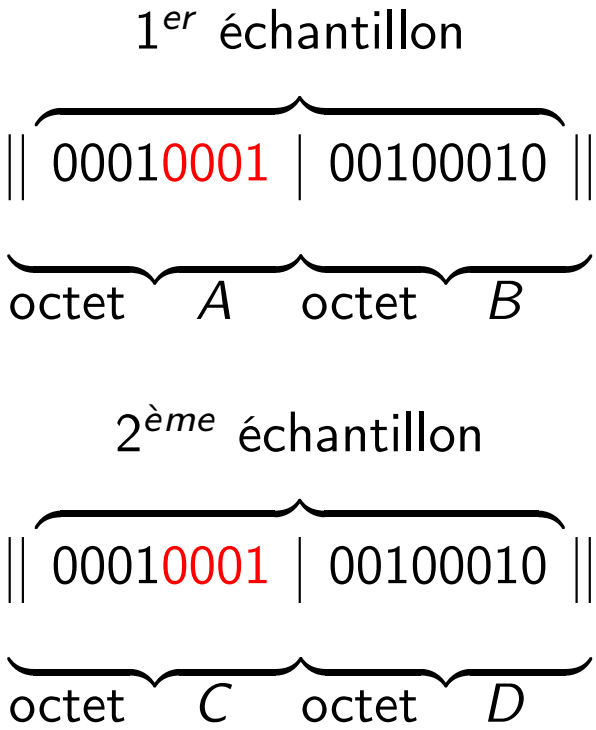
  fin

fin

Résultats


la3s
Silencer
Solo
Effets
G
D
Mono, 44100Hz
32 bits flottant
Sélection
1,0
0,5
0,0
-0,5
-1,0
la3smodif
Silencer
Solo
Effets
G
D
Mono, 44100Hz
32 bits flottant
Sélection
1,0
0,5
0,0
-0,5
-1,0

Principe : sur les bits de poids fort de l'octet de poids faible

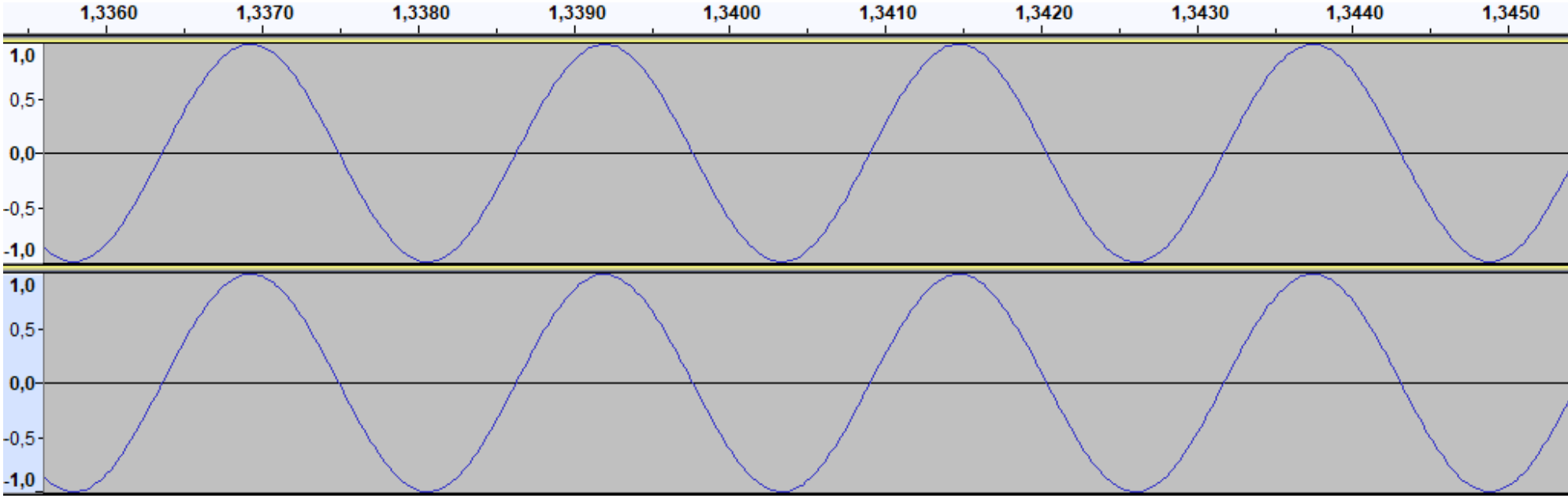


Entrées : message (m), nombre de répétitions de chaque caractère (rep), fichier de départ (dep), fichier vide (fv)

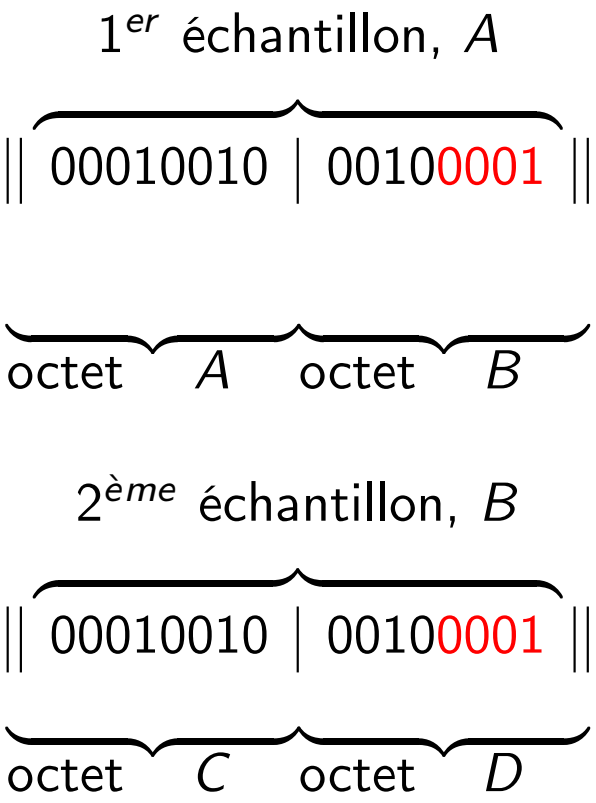
$n \leftarrow$nombre d'échantillons de dep
$k \leftarrow 0$

tant que $k < n//(2 * rep)$ **faire**
  $lettre \leftarrow m[k \bmod longueur(m)]$
  répéter
    lire deux échantillons A, B, C et D
    $l \leftarrow$ordinal ASCII de $lettre$
    $début \leftarrow l//16$
    $fin \leftarrow l \bmod 16$
    écrire dans fv $(début * 16,\ B\ ,\ fin * 16,\ D)$
    $k \leftarrow k + 1$
  jusqu'à rep;
fin

Résultats


1,3360
1,3370
1,3380
1,3390
1,3400
1,3410
1,3420
1,3430
1,3440
1,3450
1,0
0,5
0,0
-0,5
-1,0
1,0
0,5
0,0
-0,5
-1,0

Principe : sur les bits de poids faible de l'octet de poids fort

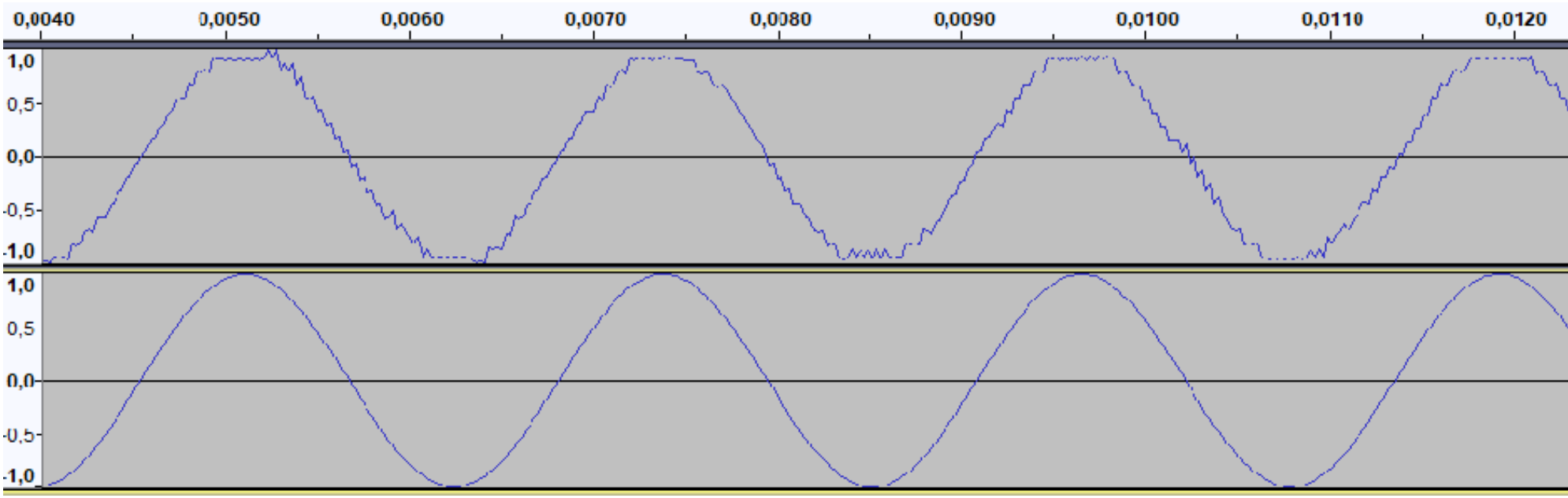


Entrées : message (m), nombre de répétitions de chaque caractère (rep), fichier de départ (dep), fichier vide (fv)

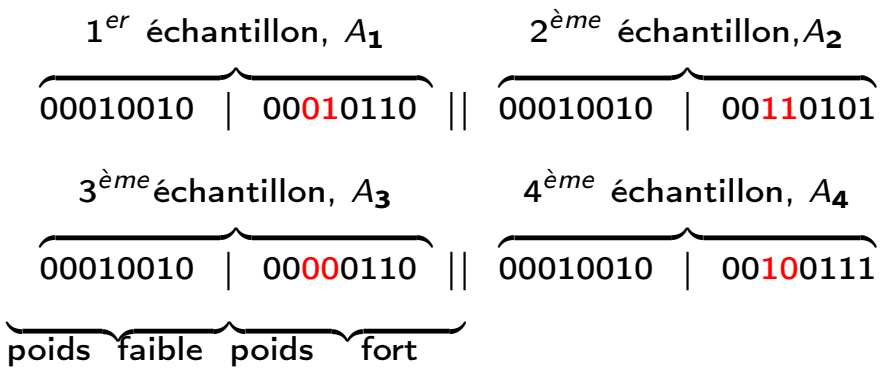
$n \leftarrow$nombre d'échantillons de dep
$k \leftarrow 0$

tant que $k < n//(2 * rep)$ **faire**
  $lettre \leftarrow m[k \bmod longueur(m)]$
  répéter
    lire deux échantillons A, B, C et D de dep
    $l \leftarrow$ordinal ASCII de $lettre$
    $début \leftarrow l//16$
    $fin \leftarrow l \bmod 16$
    écrire $(A, (B//16) * 16 + deb, C, (D//16) * 16 + fin)$ dans fv
    $k \leftarrow k + 1$
  jusqu'à rep;
fin

Résultats


0,0040
0,0050
0,0060
0,0070
0,0080
0,0090
0,0100
0,0110
0,0120
1,0
0,5
0,0
-0,5
-1,0
1,0
0,5
0,0
-0,5
-1,0

Principe : sur des bits de poids plus fort

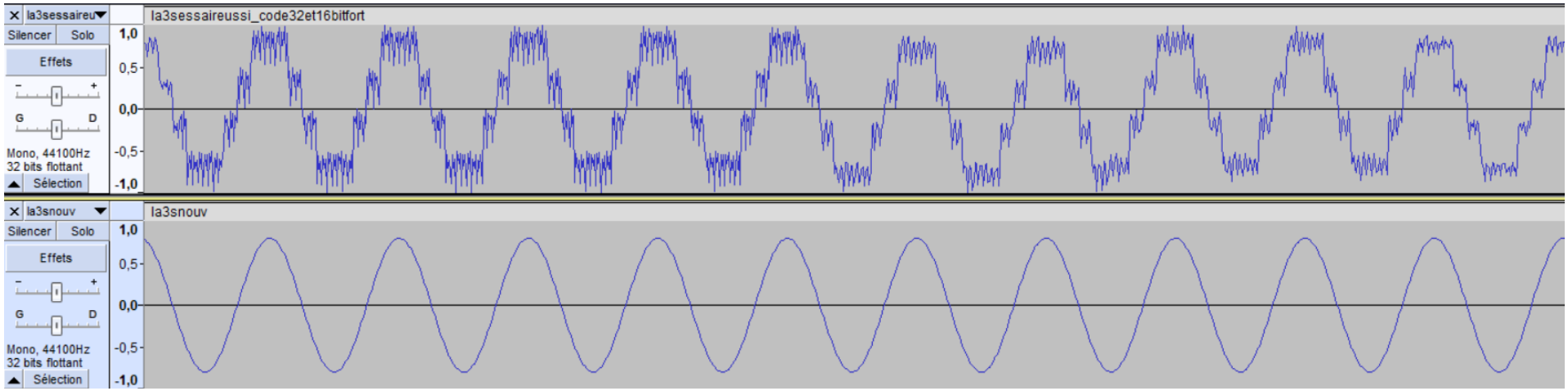


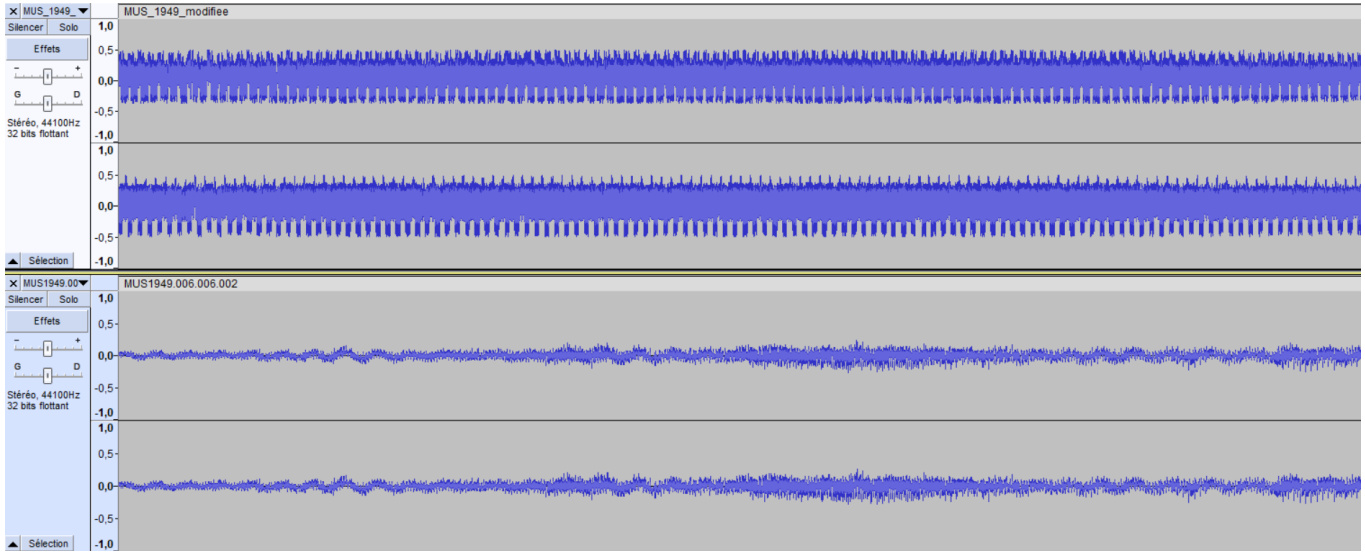
où 01110010 correspond à un caractère
poids fort en tête

Entrées : message (m), fichier de départ (dep), fichier vide (fv), nombre de répétitions de chaque caractère (rep)

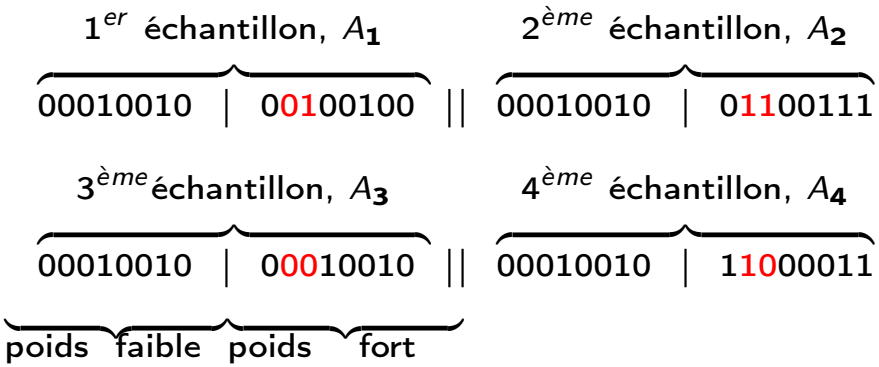
écriture de l'entête
pour *i allant de 0 à longueur(dep) // (4*rep)* **faire**
  pour *j allant de 1 à rep* **faire**
    lettre $\leftarrow m \mod$ *longueur*(*rep*)
    transformer *lettre* en octet
    couper *lettre* en 4 groupes de longueur 2 a_1, a_2, a_3 et a_4
    $A_1, A_2, A_3, A_4 \leftarrow$ lire 4 échantillons de *dep*
    pour chaque *échantillon* A_i **faire**
      transformer $A_i[1]$ *en un octet*
      $A_i[1] \leftarrow A_i[:2] + a_i + A_i[4:]$
      écrire $A_i[0]$ *dans fv*
      écrire $A_i[1]$ *dans fv*
    fin
  fin
fin

Résultats


la3sessaireu
Silencer
Solo
Effets
G
D
Mono, 44100Hz
32 bits flottant
Sélection
1,0
0,5
0,0
-0,5
-1,0
la3sessaireussi_code32et16bitfort
la3snouv
Silencer
Solo
Effets
G
D
Mono, 44100Hz
32 bits flottant
Sélection
1,0
0,5
0,0
-0,5
-1,0
la3snouv


MUS_1949_
Silencer
Solo
Effets
G
D
Stéréo, 44100Hz
32 bits flottant
Sélection
1,0
0,5
0,0
-0,5
-1,0
MUS_1949_modifiee
MUS1949.00
Silencer
Solo
Effets
G
D
Stéréo, 44100Hz
32 bits flottant
Sélection
1,0
0,5
0,0
-0,5
-1,0
MUS1949.006.006.002

Principe : sur des bits de poids encore plus forts

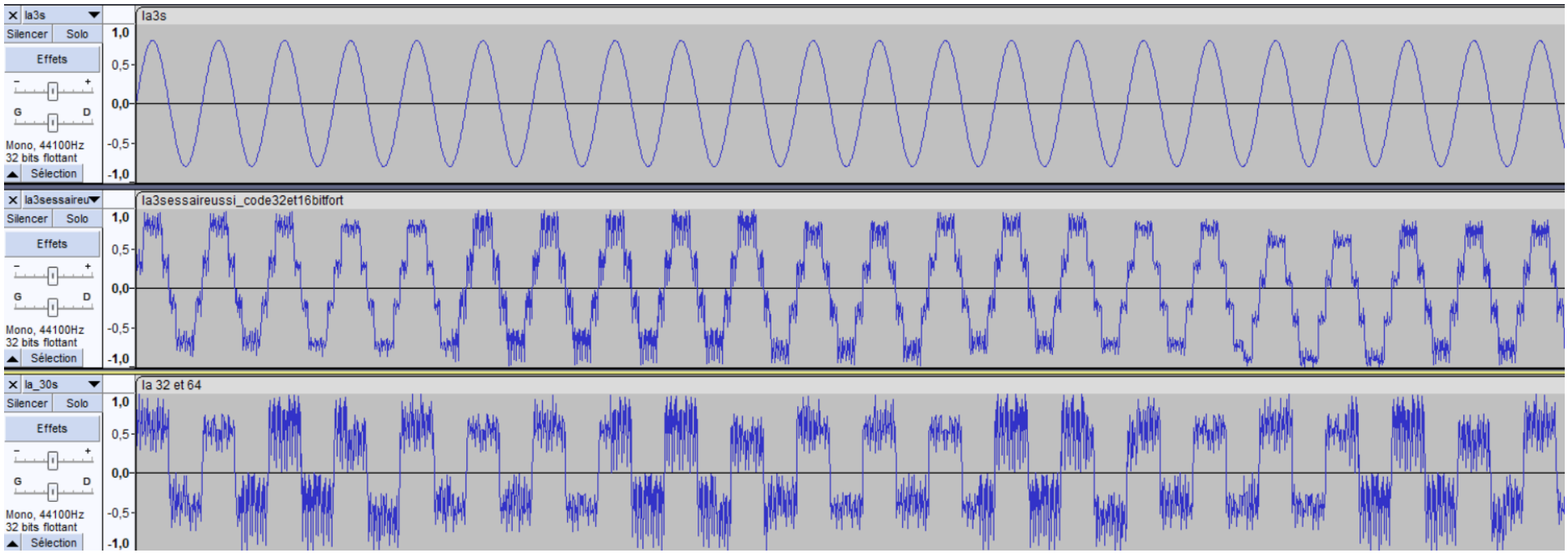


où 01110010 correspond à un caractère, poids fort en tête

Entrées : message (m), fichier de départ (dep), fichier vide (fv), nombre de répétitions de chaque caractère (rep)

écriture de l'entête
pour *i allant de 0 à longueur(dep) // (4*rep)* **faire**
  pour *j allant de 1 à rep* **faire**
    lettre $\leftarrow m \mod$ *longueur*(rep)
    transformer *lettre* en octet
    couper *lettre* en 4 groupes de longueur 2
    a_1, a_2, a_3 et a_4
    $A_1, A_2, A_3, A_4 \leftarrow$ lire 4 échantillons de *dep*
    pour chaque *échantillon* A_i **faire**
      transformer $A_i[1]$ en un octet
      $A_i[1] \leftarrow A_i[:1] + a_i + A_i[3:]$
      écrire $A_i[0]$ dans fv
      écrire $A_i[1]$ dans fv
    fin
  fin
fin

Résultats


la3s
Silencer
Solo
Effets
G
D
Mono, 44100Hz
32 bits flottant
Sélection
1,0
0,5
0,0
-0,5
-1,0
la3sessaireu
la3sessaireussi_code32et16bitfort
la_30s
la 32 et 64

Pour le programme n°1, avec une fonction auxiliaire

Entrées : nombre de répétitions de chaque caractère (*rep*), fichier codé (*f*)
Sorties : message

$i \leftarrow 0$
$n \leftarrow$nombre d'échantillons de f
tant que $i < n - (2 * rep)$ **faire**
    $lettre \leftarrow$ ""
    répéter
        lire 2 échantillons A_1A_2 et B_1B_2
        $début \leftarrow (A_2 \bmod 16) * 16$
        $fin \leftarrow B_2 \bmod 16$
        $lettre \leftarrow chr(debut + fin)$
        $i \leftarrow i + 2$
    jusqu'à *rep*;
    vraieLettre $\leftarrow$ auxiliaire (*lettres*)
    ajouter *vraieLettre* à *message*
fin
renvoyer *message*

Entrées : chaîne de caractères *lettres*
Sorties : un caractère *vraieLettre*
$d \leftarrow dictionnairevide$
pour chaque *l élément de lettre* **faire**
    si *l dans d* **alors**
        $d[l] \leftarrow d[l] + 1$
    sinon
        $d[l] \leftarrow 1$
    fin
fin
$max \leftarrow 0$
$vraieLettre \leftarrow$ ""
pour chaque *l dans d* **faire**
    si $d[l] > max$ **alors**
        $max \leftarrow d[l]$
        vraieLettre $\leftarrow l$
    fin
fin
renvoyer *vraieLettre*

Conclusion

Limites

Poursuites possibles

Fin de la présentation.