Méthodes de resynchronisation pour le tatouage audio

Leandro de C. T. Gomes[1], Emilia Gómez[2], Madeleine Bonnet[1], Nicolas Moreau[3]

[1]Université René Descartes – Crip5 / InfoCom
45 rue des Saints-Pères, 75270 Paris Cedex 06, France

[2]Pompeu Fabra University – Audiovisual Institute
Passeig de Circumval·lació, 8, 08003 Barcelona, Spain

[3]École Nationale Supérieure des Télécommunications – TSI
46 rue Barrault, 75634 Paris Cedex 13, France

{tgomes,bonnet}@math-info.univ-paris5.fr, moreau@tsi.enst.fr, emilia.gomez@iua.upf.es

Résumé – Pour plusieurs applications, les systèmes de tatouage audio doivent résister à des attaques de piratage. Les attaques qui désynchronisent la détection du tatouage sont particulièrement difficiles à neutraliser. Nous décrivons ici des méthodes de resynchronisation basées sur l'utilisation de suites d'apprentissage. Ces méthodes rendent le système robuste face à une large classe d'attaques désynchronisantes.

Abstract – Depending on the application, audio watermarking systems must resist piracy attacks. Attacks that desynchronize watermark detection are particularly difficult to neutralize. In this paper, we introduce resynchronization methods based on the use of training sequences. These methods reverse the effect of a large class of desynchronization attacks.

1 Introduction

Les signaux audio numériques peuvent être dupliqués facilement sans distorsion, ce qui crée un scénario favorable au piratage. Les méthodes de tatouage ont été proposées pour protéger les droits d'auteur. Une marque, le tatouage, insérée dans un signal audio, produit le signal tatoué. La puissance du tatouage est réglée de façon à ce que le signal tatoué soit auditivement identique au signal original.

Sur la figure 1, le tatouage est l'information à transmettre et le signal audio correspond à un bruit beaucoup plus fort que le tatouage à cause de la condition d'inaudibilité [1, 2].

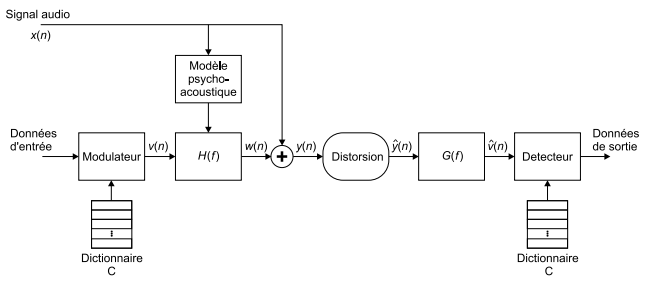

Fig. 1 – *Tatouage vu comme un canal de communication.*

L'information est codée à l'aide d'un dictionnaire $C = \mathbf{u_k} = [u_k(0) \cdots u_k(N-1)]$ associant à chaque symbole un vecteur de longueur N. Les vecteurs sont orthogonaux et gaussiens. Le modulateur reçoit en entrée une suite de symboles $\mathbf{s} = [s_0 \cdots s_{M-1}]$ et produit $v(n)$ par concaténation des vecteurs correspondants : $v(mN+n) = u_{s_m}(n)$.

Afin de garantir le caractère inaudible, $v(n)$ subit une mise en forme spectrale selon un seuil de masquage fourni par un modèle psychoacoustique [3]. C'est le rôle du filtre $H(f)$, dont la réponse en amplitude suit le seuil de masquage. Le signal résultant $w(n)$, ajouté au signal audio $x(n)$, fournit le signal tatoué $y(n)$.

L'observation $\hat{y}(n)$ est filtrée par $G(f)$, filtre de Wiener destiné à accroître le rapport tatouage à signal. Le détecteur reçoit $\hat{v}(n)$, estimation de $v(n)$, et fournit une suite de symboles détectés grâce à des mesures de corrélation.

2 Désynchronisation

Pour la plupart des applications, le tatouage doit résister à des opérations licites : codage MPEG, filtrage, re-échantillonnage etc. Pour la protection du copyright, le système doit résister à des actes de piratage visant à rendre le tatouage indétectable ; c'est le cas d'ajout de bruit, d'insertion ou de suppression d'échantillons.

La détection du tatouage nécessite la synchronisation de l'émetteur et du récepteur. Beaucoup d'opérations peuvent provoquer une désynchronisation. C'est le cas d'une compression/décompression qui peut introduire un délai en début de signal. Un pirate peut aussi tenter d'enlever ou d'ajouter des échantillons. Des simulations ont montré que, pour un signal échantillonné à 32 kHz, un échantillon sur 2500 peut être supprimé ou ajouté en moyenne sans distorsion perceptible [4]. Des modifications de la durée du signal (time warp et time-stretching) peuvent aussi être cause de désynchronisation.

Un filtrage passe-tout, bien que ne modifiant ni le début ni la fin des symboles, réduit la corrélation entre le signal

et le vecteur correspondant du dictionnaire, et peut ainsi introduire des erreurs de détection.

On s'intéresse ici aux attaques modifiant la localisation des symboles et non leur durée. La robustesse au filtrage passe-tout et au codage MPEG est aussi examinée.

3 Méthodes de resynchronisation

Une technique de synchronisation classique consiste à émettre des suites d'apprentissage intercalées entre les données utiles. Cette technique, appliquée au tatouage [4, 5], présente deux inconvénients majeurs. Pendant l'émission des suites d'apprentissage, le tatouage ne contient pas d'information utile, ce qui peut réduire significativement le débit. De plus ces suites peuvent subir des attaques.

Nous présentons ici deux méthodes qui remédient aux inconvénients précédents. Elles consistent à étaler les suites d'apprentissage, qui coexistent avec les données utiles, tout au long du signal tatoué permettant ainsi au détecteur de retrouver la synchronisation de façon continue.

3.1 Tatouage de synchronisation

La suite d'apprentissage peut être étalée dans le temps grâce à un tatouage $\dot{w}(n)$ qui n'est utilisé que pour la synchronisation. Un autre tatouage $\ddot{w}(n)$ contient l'information utile. Afin d'éviter toute interférence entre eux, ces deux tatouages sont construits à partir de dictionnaires orthogonaux. Le tatouage résultant $w(n) = \dot{w}(n) + \ddot{w}(n)$ doit évidemment être inaudible. Une attaque désynchronisante aura exactement le même effet sur les deux parties du tatouage puisqu'elles sont superposées. Donc, si le détecteur est capable de resynchroniser la partie $\dot{w}(n)$, il en sera de même pour $\ddot{w}(n)$ (figure 2).

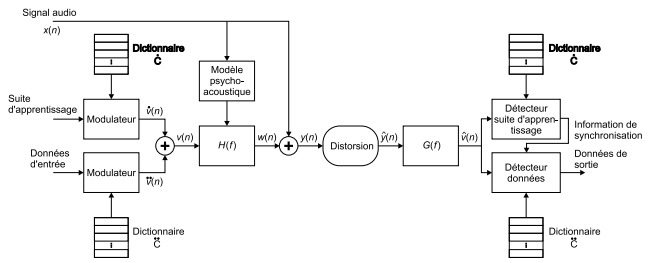

FIG. 2 – *Resynchronisation grâce à un second tatouage.*

Soit $\dot{\mathbf{C}}$ le dictionnaire utilisé pour construire le tatouage de synchronisation $\dot{w}(n)$. Il contient $\dot{K}$ vecteurs $\dot{\mathbf{u}}_k = [\dot{u}_k(0) \cdots \dot{u}_k(N-1)]$ ($k \in [0,\dot{K}-1]$) associés à $\dot{K}$ symboles. La suite d'apprentissage $\mathbf{z} = [z_0 \cdots z_{M-1}]$ est obtenue par $z_m = m \bmod \dot{K}$, où z_m est le m-ème symbole de la suite d'apprentissage ($m \in [0,M-1]$).

Le dictionnaire $\ddot{\mathbf{C}}$, utilisé pour construire $\ddot{w}(n)$, contient $\ddot{K}$ vecteurs $\ddot{\mathbf{u}}_k = [\ddot{u}_k(0) \cdots \ddot{u}_k(N-1)]$ ($k \in [0,\ddot{K}-1]$) associés à $\ddot{K}$ symboles. La suite de symboles $\mathbf{s} = [s_0 \cdots s_{M-1}]$ représente la véritable information à insérer dans le signal audio. Le tatouage $w(n)$ est construit par concaténations successives des vecteurs associés aux symboles des suites $\mathbf{z}$ et $\mathbf{s}$. Une opération de filtrage assure l'inaudibilité :

$$\begin{aligned} w(mN+n) &= \dot{w}(mN+n) + \ddot{w}(mN+n) \\ &= [\dot{v}(mN+n) + \ddot{v}(mN+n)] * h(n) \\ &= [\dot{u}_{z_m}(n) + \ddot{u}_{s_m}(n)] * h(n) \end{aligned}$$

où n est l'indice de temps dans la fenêtre d'analyse ($n \in [0,N-1]$) et $h(n)$ la réponse impulsionnelle du filtre obtenu à partir du seuil de masquage.

Pendant la phase de détection, une fenêtre glissante est utilisée pour calculer N corrélations pour chacun des M symboles de $\dot{w}(n)$ et pour chacun des $\dot{K}$ vecteurs de $\dot{C}$:

$$\dot{r}(\lambda,k,m) = \left| \sum_{n=0}^{N-1} \hat{v}(mN+n+\lambda)\dot{u}_k(n) \right|$$

où $\lambda \in [-\Lambda,\Lambda-1]$ est le décalage de la fenêtre glissante ($\Lambda = N/2$) et $\hat{v}(n)$ le tatouage estimé. Ensuite, grâce à une maximisation selon k, deux matrices $\mathbf{A} = \{\alpha_{\lambda,m}\}$ et $\mathbf{B} = \{\beta_{\lambda,m}\}$ sont construites. Les lignes correspondent aux décalages λ et les colonnes à la position des symboles de la suite :

$$\begin{aligned} \alpha_{\lambda,m} &= \max_k \dot{r}(\lambda,k,m) \\ \beta_{\lambda,m} &= \arg\max_k \dot{r}(\lambda,k,m). \end{aligned}$$

Ainsi, $\mathbf{A}$ contient les corrélations les plus élevées pour chaque décalage et chaque position dans la suite de symboles et $\mathbf{B}$ contient les symboles correspondants de $\dot{\mathbf{C}}$.

Un algorithme de programmation dynamique (cf. paragraphe 3.3) fournit un chemin optimal dans $\mathbf{A}$ et $\mathbf{B}$. L'opération d'optimisation tient compte de l'ordre attendu des symboles et des corrélations. Il en résulte un ensemble de M valeurs possibles pour λ, $[\hat{\lambda}_0 \cdots \hat{\lambda}_{M-1}]$, une pour chaque position dans la suite d'apprentissage.

La détection est alors effectuée pour $\ddot{w}(n)$. Des mesures de corrélation sont faites pour chaque symbole de $\mathbf{s}$, en utilisant les valeurs de décalage précédentes :

$$\ddot{r}(k,m) = \left| \sum_{n=0}^{N-1} \hat{v}(mN+n+\hat{\lambda}_m)\ddot{u}_k(n) \right|$$

La suite de symboles détectés $\hat{\mathbf{s}}$ est extraite en choisissant, pour chaque m, le symbole du dictionnaire $\ddot{\mathbf{C}}$ qui correspond au maximum de la mesure de corrélation :

$$\hat{s}_m = \arg\max_k \ddot{r}(k,m)$$

où $\hat{s}_m$ représente le m-ème symbole détecté.

3.2 Enchaînement de dictionnaires

Une autre méthode utilise plusieurs dictionnaires orthogonaux pour coder l'information. Ces dictionnaires sont utilisés consécutivement, leur enchaînement jouant le rôle d'une suite d'apprentissage.

On définit P dictionnaires $\mathbf{C}_p$ ($p \in [0 \cdots P-1]$) contenant K vecteurs $\mathbf{u}_{p,k} = [u_{p,k}(0) \cdots u_{p,k}(N-1)]$ ($k \in [0 \cdots K-1]$) associés à K symboles. Les symboles qui correspondent au même indice k dans les dictionnaires représentent la même information, mais ils sont associés à des vecteurs différents. Le détecteur connaît donc le dictionnaire duquel provient chaque symbole détecté (figure 3).

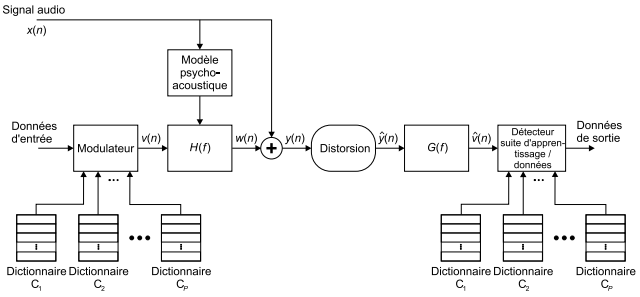

FIG. 3 – *Enchaînement de dictionnaires.*

L'enchaînement de dictionnaires $\mathbf{z} = [z_0 \cdots z_{M-1}]$ est obtenu par $z_m = m \bmod P$ où z_m est le m-ème dictionnaire dans la suite. L'enchaînement qui en résulte doit être retrouvé lorsqu'il y a synchronisation.

L'information à insérer est représentée par une suite de symboles $\mathbf{s} = [s_0 \cdots s_{M-1}]$. Le tatouage $w(n)$ est construit par concaténation des vecteurs associés à ces symboles, en respectant l'enchaînement de dictionnaires, suivie d'une opération de filtrage qui assure l'inaudibilité : $w(mN + n) = u_{z_m,s_m}(n) * h(n)$; n correspond au temps dans la fenêtre d'analyse courante ($n \in [0,N-1]$) et $h(n)$ à la réponse impulsionnelle de $H(f)$.

Pendant la détection, une fenêtre glissante est utilisée pour calculer N corrélations pour les M symboles du tatouage et pour les K vecteurs des P dictionnaires C_p :

$$r(\lambda,p,k,m) = \left| \sum_{n=0}^{N-1} \hat{v}(mN + n + \lambda) u_{p,k}(n) \right|$$

où λ et $\hat{v}(n)$ sont définis comme dans le paragraphe précédent. Alors, en maximisant $r(\lambda,p,k,m)$ selon p et k, on construit trois matrices $\mathbf{A} = \{\alpha_{\lambda,m}\}$, $\mathbf{B} = \{\beta_{\lambda,m}\}$ et $\mathbf{\Gamma} = \{\gamma_{\lambda,m}\}$ dont les lignes correspondent aux décalages λ et les colonnes à la position m des symboles dans la suite :

$$\alpha_{\lambda,m} = \max_{p,k} r(\lambda,p,k,m)$$
$$\beta_{\lambda,m} = \arg_p \max_{p,k} r(\lambda,p,k,m)$$
$$\gamma_{\lambda,m} = \arg_k \max_{p,k} r(\lambda,p,k,m).$$

Ainsi, $\mathbf{A}$ contient les mesures de corrélation les plus élevées pour chaque décalage et chaque position dans la suite de symboles, $\mathbf{B}$ contient les dictionnaires correspondants et $\mathbf{\Gamma}$ les symboles détectés.

Un algorithme de programmation dynamique est utilisé pour trouver le chemin optimal dans $\mathbf{A}$ et $\mathbf{B}$ (cf. paragraphe 3.3). L'optimisation prend en compte l'enchaînement attendu des dictionnaires et les mesures de corrélation. La suite de symboles détectés $\hat{\mathbf{s}}$ est directement obtenue en suivant ce chemin optimal dans $\mathbf{\Gamma}$.

3.3 Procédure d'optimisation

L'algorithme de programmation dynamique, utilisé pour déterminer les décalages de la fenêtre glissante qui correspondent le mieux au début des symboles, minimise une fonction coût dépendant de $\mathbf{A} = \{\alpha_{\lambda,m}\}$ et $\mathbf{B} = \{\beta_{\lambda,m}\}$. Il en résulte un ensemble de M valeurs pour le décalage λ, $[\lambda_0 \cdots \lambda_{M-1}]$, qui définit un chemin dans $\mathbf{A}$ et $\mathbf{B}$ à partir duquel les symboles détectés peuvent être obtenus.

Le coût $c(\lambda,\lambda',m)$ pour passer du nœud $[\lambda',m-1]$ au nœud $[\lambda,m]$ est composé de trois termes :

$$c(\lambda,\lambda',m) = c_1(\lambda,\lambda',m) + c_2(\lambda,\lambda',m) + c_3(\lambda,m).$$

Le premier, lié au respect de la suite d'apprentissage, est

$$c_1(\lambda,\lambda',m) = \begin{cases} \epsilon(\beta_{\lambda,m} - \beta_{\lambda',m-1} - 1) \text{ si } \beta_{\lambda,m} \geq \beta_{\lambda',m-1} \\ \epsilon(\beta_{\lambda,m} - \beta_{\lambda',m-1} - 1 + P) \text{ sinon} \end{cases}$$

où ϵ est une constante positive. Si l'ordre de la suite d'apprentissage est respecté, ce coût est nul ; sinon, il est proportionnel au décalage par rapport à la suite d'apprentissage. Cette définition est justifiée par le fait que le pirate ne peut enlever ou ajouter un grand nombre d'échantillons consécutifs à cause de la contrainte d'inaudibilité.

Le deuxième terme pénalise des changements dans le décalage λ lorsqu'on passe d'un nœud $[\lambda',m-1]$ à un nœud $[\lambda,m]$. Ceci a pour but de maintenir le chemin optimal sur la même ligne en l'absence de désynchronisation :

$$c_2(\lambda,\lambda',m) = \eta_{m-1}(\lambda - \lambda')^2$$

À cause du carré, la pénalité croît rapidement lorsque λ s'écarte de λ' (ceci est toujours justifié par le fait qu'un grand nombre d'échantillons consécutifs ne peuvent être enlevés ou ajoutés). Le facteur η_m est défini par

$$\eta_m = \begin{cases} \eta_{m-1} + \kappa_1 \text{ si } \lambda_m \neq \lambda_{m-1} \\ \max(\eta_{m-1} - \kappa_2, \eta_0) \text{ sinon} \end{cases}$$

avec κ_1 et κ_2 constantes positives (généralement $\kappa_1 > \kappa_2$), λ_m le numéro de la ligne correspondant à la colonne m sur le chemin courant et η_0 initialisée à une valeur positive. Cette définition a pour but d'éviter des zigzags dans le chemin qui feraient croître le paramètre η_m.

Le troisième terme est lié aux mesures de corrélation :

$$c_3(\lambda,m) = \rho \left(1 - \frac{\alpha_{\lambda,m}}{\max_{\tilde{\lambda}} \alpha_{\tilde{\lambda},m}} \right)$$

avec ρ constante positive. Cette définition pénalise des décalages λ conduisant à de faibles valeurs de corrélation.

On définit le coût accumulé $C(\lambda,m)$ comme étant le coût minimal pour atteindre le nœud $[\lambda,m]$ à partir d'un nœud de la première colonne. Il est initialisé à 0 pour $m = 0$ et pour tout λ. L'algorithme d'optimisation est :

Pour $m = 1 \cdots M-1$
  Pour $\lambda = -\Lambda \cdots \Lambda - 1$
    $\bar{\lambda} = \arg\min_{\lambda'} [C(\lambda',m-1) + c(\lambda,\lambda',m)]$
    $C(\lambda,m) = C(\bar{\lambda},m-1) + c(\lambda,\bar{\lambda},m)$
    $I(\lambda,m) = \bar{\lambda}$
$\hat{\lambda}_{M-1} = \arg\min_{\tilde{\lambda}} [C(\tilde{\lambda},M-1)]$
Pour $m = M-2 \cdots 0$
  $\hat{\lambda}_m = I(\lambda_{m+1},m+1)$.

On obtient ainsi l'ensemble de décalages $[\hat{\lambda}_0 \cdots \hat{\lambda}_{M-1}]$ correspondant au chemin optimal.

4 Simulations

Quatre signaux (4,8 secondes chacun, mono canal, fréquence d'échantillonnage de 32 kHz, quantification sur 16 bits) sont utilisés : "svega" ("Tom's diner", version a cappella, par Suzanne Vega), "violin" (un morceau de violon), "baron" (un extrait de musique des Caraïbes de Baron) et "queen" (de la musique pop). Après avoir été tatoué, chaque morceau subit une opération de suppression/ajout aléatoire d'un échantillon sur 2.500 en moyenne et un filtrage passe-tout.

La longueur des fenêtres est $N = 512$, avec un décalage maximal de $\Lambda = 256$. Le traitement est effectué pour des groupes de $M = 50$ fenêtres. Le débit est de 125 bit/s. Les valeurs des constantes (obtenues expérimentalement) sont $\epsilon = \Lambda$, $\eta_0 = 1$, $\kappa_1 = 5$, $\kappa_2 = 1$ et $\rho = 10\Lambda$.

Les seuils de masquage sont obtenus à partir du modèle psychoacoustique numéro 1 de MPEG-2. Les rapports signal à tatouage sont toujours supérieurs à 15 dB, ce qui correspond généralement à la limite d'audibilité.

En l'absence d'attaques, les taux d'erreur sont inférieurs à 0,005 pour tous les signaux tests ; après attaque, sans resynchronisation, le taux d'erreur approche 0,5.

Avec la méthode d'addition d'un tatouage de synchronisation, les dictionnaires $\dot{\mathbf{C}}$ et $\ddot{\mathbf{C}}$ contiennent chacun $\dot{K} = \ddot{K} = 4$ vecteurs distribués normalement. Les deux parties du tatouage, de synchronisation et de données, ont la même puissance. Dans la méthode d'enchaînement de dictionnaires, $P = 4$ dictionnaires ont été utilisés, chacun contenant $K = 4$ vecteurs distribués normalement.

La table 1 fournit les taux d'erreur pour les signaux tests. En plus de la suppression/addition d'échantillons et d'un filtrage passe-tout, les signaux ont subi une compression/décompression selon la norme MP3 (couche 3, mono, 128 kbps), ce qui ajoute un bruit de quantification avec un rapport signal à bruit d'environ 10 dB. Comme on peut le remarquer, ces deux méthodes fournissent des taux d'erreur significativement plus bas que ceux obtenus sans resynchronisation ($\approx 0{,}5$). La réduction est plus forte pour la seconde méthode puisque, lorsque deux tatouages sont simultanément présents, leurs puissances individuelles doivent être réduites afin de garantir l'inaudibilité.

Signal	Première méthode		Seconde méthode	
	RST	TEB	RST	TEB
svega	16,2 dB	0,035	17,6 dB	0,010
violon	16,5 dB	0,080	18,0 dB	0,037
baron	16,2 dB	0,098	17,2 dB	0,030
queen	16,6 dB	0,100	17,6 dB	0,020

TAB. 1 – *Rapport signal à tatouage (RST) et taux d'erreur binaire (TEB) pour la première méthode (tatouage de resynchronisation) et pour la seconde méthode (enchaînement de dictionnaires).*

La figure 4 montre les taux d'erreur en fonction du rapport signal à tatouage pour le signal "svega" pour les deux méthodes. Après introduction d'un bruit simulant une attaque (rapport signal à bruit de 20 dB après mise en forme spectrale), le taux d'erreur augmente avec l'affaiblissement de la puissance du tatouage. Cette expérience confirme la meilleure performance obtenue par la deuxième méthode.

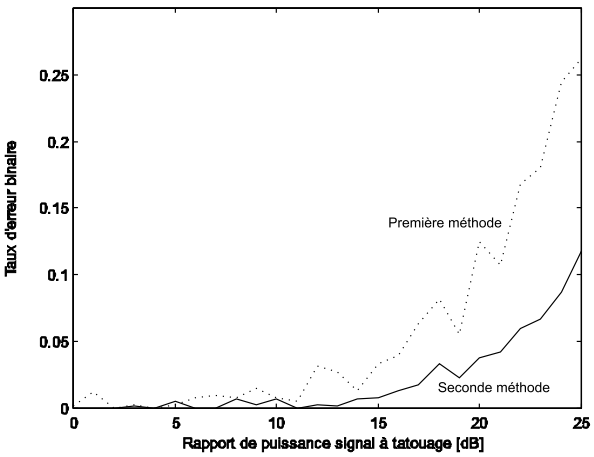

FIG. 4 – *Taux d'erreur binaire en fonction du rapport signal à tatouage pour le signal "svega".*

5 Conclusions

Les méthodes de resynchronisation présentées permettent de résister à une large classe de piratage. Ces méthodes étalent la suite d'apprentissage sur tout le tatouage. Les résultats de simulation montrent que ces méthodes réussissent à contrecarrer les attaques désynchronisantes qui consistent à supprimer ou ajouter des échantillons au signal tatoué.

Des recherches supplémentaires sont nécessaires pour augmenter la résistance à des attaques qui modifient significativement la durée des symboles (time warp et time stretching).

Références

[1] R.A. Garcia, *Digital watermarking of audio signals using a psychoacoustic auditory model and spread spectrum theory,* 107th AES Convention, New York, September 1999.

[2] L. de C.T. Gomes, M. Mboup, M. Bonnet, and N. Moreau, *Cyclostationarity-based audio watermarking with private and public hidden data,* 109th AES Convention, Los Angeles, September 2000.

[3] M. Perreau Guimarães, *Optimisation de l'allocation des ressources binaires et modelisation psychoacoustique pour le codage audio,* Thèse de Doctorat, Université Paris V, Paris, 1998.

[4] E. Gómez, *Tatouage de signaux de musique (méthodes de synchronisation),* rapport technique (DEA ATIAM), Université de la Méditerranée / ENST, Paris, juillet 2000.

[5] N. Moreau, P. Dymarski et L. de C.T. Gomes, *Tatouage audio : une réponse à une attaque désynchronisante,* CORESA, Poitiers, octobre 2000.