Décodage EM du code de Tardos pour le fingerprinting

Ana CHARPENTIER[1], Caroline FONTAINE[2,1], Teddy FURON[3]

[1]INRIA, Centre Rennes - Bretagne Atlantique
Campus de Beaulieu, 35042 Rennes cedex, France

[2]CNRS-IRISA, Campus de Beaulieu, 35042 Rennes cedex, France

[3]Thomson Security Lab
Cesson Sévigné, France

Ana.Charpentier@irisa.fr, Caroline.Fontaine@irisa.fr
Teddy.Furon@thomson.net

Résumé – Cet article porte sur les codes traçants, qui ont pour objectifs d'identifier des utilisateurs impliqués dans la création de fausses copies de documents, par exemple dans le contexte de la vidéo à la demande. Nous y présentons une amélioration de la phase d'accusation des codes de Tardos. Plus spécifiquement nous montrons comment l'optimiser en fonction de la stratégie d'attaque des pirates. Nous proposons également des moyens d'estimer à partir d'un faux le nombre d'attaquants qui se sont rassemblés pour le forger, ainsi que la stratégie qu'ils ont employée. Notre solution s'appuie sur un algorithme itératif *a la* EM, dans lequel une meilleure estimation de la stratégie permet une meilleure détection des attaquants, qui permet à son tour une meilleure estimation de la stratégie, etc..

Abstract – This paper presents our recent work on multimedia fingerprinting, also known as traitor tracing. We focus on deriving a better accusation process for the well known Tardos codes. It appears that Tardos orginal decoding is very conservative: its performances are guaranteed whatever the collusion strategy. Indeed, major improvements stem from the knowledge of the collusion strategy. Therefore, this paper investigates how it is possible to learn and adapt to the collusion strategy. Our solution is based on an iterative algorithm *a la* EM, where a better estimation of the collusion strategy yields a better tracing of the colluders, which in return yields a better estimation of the collusion strategy etc.

1 Introduction

Cet article porte sur le *multimedia fingerprinting*, connu aussi sous les noms de *transactional watermarking*, *traitor tracing*, *copy serialization*. Le problème est le suivant : un serveur multimédia distribue des copies d'un même contenu numérique à n acheteurs différents. Ces copies sont personnalisées et contiennent chacune un message qui les identifie, caché à l'aide d'une technique de tatouage. Des utilisateurs malhonnêtes, appelés *colluders*, utilisent leurs copies pour créer un faux qu'ils vont redistribuer de façon illégale. L'objectif est alors pour le distributeur de contenus de retrouver l'identité de ces *colluders* par un processus d'accusation utilisant le tatouage extrait de la copie pirate. Pour que cette accusation soit efficace, le schéma doit s'appuyer sur une technique de tatouage robuste et un bon code anti-collusion. C'est à ce code que nous nous intéressons ici.

En 2003, G. Tardos publie des codes traçants binaires reposant sur des probabilités [2]. Très intéressants par leur ordre de longueur minimal et faciles à implémenter, ils permettent un bon contrôle des probabilités d'erreur d'accusation (risque d'accuser un innocent, risque de n'identifier aucun pirate). B. Skoric *et al* ont proposé une version symétrique de ces codes et ont étendu l'application à un alphabet q-aire [3]. Les deux articles utilisent des fonctions d'accusation qui sont les mêmes pour tous les cas d'attaques. Récemment, T. Furon *et al* [1] ont démontré que ces fonctions sont optimales dans un contexte général, mais aussi que l'on peut trouver des fonctions d'accusation plus efficaces si on a des informations sur l'attaque qui a été réalisée. Ici, nous continuons dans cette direction, proposant des mécanismes d'estimation de la stratégie des pirates ainsi que de leur nombre, et proposant des fonctions d'accusation optimisées pour cette stratégie. Ces résultats ont été présentés en conférence en janvier 2009 [4].

2 Limites des études précédentes

Avant d'aller plus loin dans l'explication de notre solution, nous allons présenter brièvement les codes binaires de Tardos symétriques [3]. Pour plus de détails, nous renvoyons le lecteur vers les articles cités. Soit n le nombre total d'utilisateurs, et m la longueur du code. Les mots de code distribués forment une matrice $n \times m$ $\mathbf{X}$, L'utilisateur j correspond au mot binaire $\mathbf{X}_j = (X_{j1}, X_{j2}, \ldots, X_{jm})$. Pour générer cette matrice, m nombres réels $p_i \in [0,1]$ sont tirés au hasard selon une fonction de densité de probabilité $f(p) = \frac{1}{\pi\sqrt{p(1-p)}}$. On note $\mathbf{p} = (p_1, \ldots, p_m)$. Chaque élément de la matrice

$\mathbf{X}$ est ensuite indépendamment tiré en suivant la probabilité $\mathbb{P}(X_{ji} = 1) = p_i$. Chacun de ces n mots est caché dans la copie délivrée à l'utilisateur associé. Lors de la phase d'accusation, on extrait la séquence $\mathbf{Y}$ de la copie pirate. Afin de savoir si l'utilisateur j est impliqué dans la production du faux, on calcule un score d'accusation S_j. Si ce score est supérieur à un certain seuil Z, alors on considère l'utilisateur j comme coupable. Le calcul des scores repose sur quatre fonctions d'accusation, qui évaluent la corrélation entre la séquence $\mathbf{X}_j$, associée à l'utilisateur j, et la séquence extraite $\mathbf{Y}$:

$$S_j = \sum_{i=1}^{m} U(Y_i, X_{ji}, p_i), \tag{1}$$

avec les fonctions d'accusation

$$U(1,1,p) = g_{11}(p), \qquad U(0,0,p) = g_{00}(p),$$
$$U(0,1,p) = g_{01}(p), \qquad U(1,0,p) = g_{10}(p).$$

Dans [3], un certain nombre de contraintes sont imposées, menant aux relations $g_{11}(p) = g_{00}(1-p) = -g_{01}(p) = -g_{10}(1-p) = \sqrt{\frac{1-p}{p}}$.

Comment estimer la pertinence de l'accusation ? Dans le processus original, chaque utilisateur j est testé indépendamment et on n'a donc pas besoin de calculer tous les scores pour rendre un verdict. Cet utilisateur est considéré comme un *colluder* si son score est supérieur au seuil Z. Le seuil est choisi pour garantir une certaine probabilité de fausse alarme. Ainsi, lorsqu'on accuse un utilisateur on a une borne supérieure de la probabilité d'erreur, mais on a pas d'estimation précise de cette erreur. On adopte ici une autre stratégie, en calculant les scores de tous les utilisateurs et accusant celui qui a le plus grand. On peut noter que bien que cet utilisateur soit celui qui est le plus sûrement coupable, il n'y a pas de garantie de ne pas accuser un innocent. Contrairement à la version de G. Tardos, il n'y a pas ici de seuil ou de longueur de code qui garantit théoriquement la probabilité de fausse alarme. Néanmoins, T. Furon *et al* ont montré qu'il est possible d'accuser le plus grand score tout en estimant empiriquement la probabilité d'avoir d'erreur [5].

3 Une estimation itérative de la stratégie

En suivant les pas de T. Furon *et al*[1], notre objectif est d'optimiser les fonctions d'accusation en fonction de la stratégie des *colluders*. Nous procédons en deux étapes : d'abord nous estimons la stratégie, puis nous optimisons les fonctions d'accusation. Ce procédé est itéré, chaque itération tirant profit d'une nouvelle estimation de l'ensemble des *colluders via* Expectation-Maximization (EM).

On modélise la stratégie de la collusion par l'ensemble des probabilités $\{\mathbb{P}(Y_i = 1|\Sigma_i = \sigma_i), \sigma_i = 0..c\}_{i=1..m}$; la variable aléatoire $\Sigma_i = \sum_{j\in\mathcal{C}} X_{ij}$ correspond aux nombre de mots des *colluders* ayant un 1 à la position i. On admet que la même stratégie a été utilisée pour chaque position $1 \leq i \leq m$. Afin d'alléger les formules, nous omettrons l'indice i qui indique la position considérée et appellerons $\boldsymbol{\theta}$ le modèle de collusion : $\boldsymbol{\theta} = \{\mathbb{P}(Y = 1|\Sigma = \sigma), \sigma = 0..c\}$. Nous décrivons maintenant les étapes du processus d'estimation itérative en détails, en utilisant les notations allégées.

S1. Initialisation : nous calculons tous les scores d'accusation avec les fonctions d'accusation de B. Skoric *et al* : $g_{11}^{(S)}(p) = g_{00}^{(S)}(1-p) = -g_{01}^{(S)}(p) = -g_{10}^{(S)}(1-p) = \sqrt{\frac{1-p}{p}}$. Ces scores sont placés dans le vecteur $\mathbf{S}$.

S2. Décodage EM : la séquence $\mathbf{S}$ contient un mélange de scores d'innocents et de *colluders*. Un algorithme EM classique estime le statut, « innocent » ou « colluder », de chaque utilisateur. EM prend en entrée le vecteur $\mathbf{S}$ et les moyennes et variances théoriques des scores des innocents et *colluders*. En sortie, on a le vecteur $\hat{\mathbf{T}}$; $\hat{T}_j$ correspond à l'estimation de la probabilité que le score S_j de l'utilisateur j soit celui d'un *colluder*.

S3. Avec $\hat{\mathbf{T}}$ et $\mathbf{S}$, on estime la taille de la collusion, notée $\hat{c}$, ainsi que sa stratégie, notée $\hat{\boldsymbol{\theta}}$.

S4. En considérant $\hat{\boldsymbol{\theta}}$ on optimise les fonctions d'accusation $g_{00}(p, \hat{\boldsymbol{\theta}})$, $g_{11}(p, \hat{\boldsymbol{\theta}})$, $g_{10}(p, \hat{\boldsymbol{\theta}})$, et $g_{01}(p, \hat{\boldsymbol{\theta}})$.

S5. On calcule les nouveaux scores, conservés dans le vecteur $\mathbf{S}$. On revient alors à l'étape S2 pour itérer.

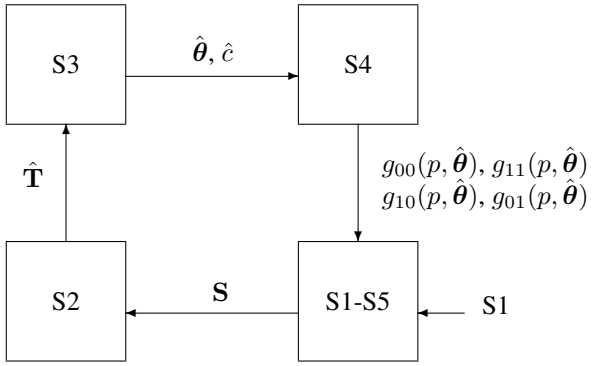

FIG. 1 – Calcul des nouvelles fonctions d'accusation

Les fonctions d'accusation optimisées obtenues dans [1] sont obtenues sous l'hypothèse que les scores des innocents ne sont pas corrélés entre eux. De fait, pour une valeur de m assez grande, les scores sont considérés comme suivant des distributions gaussiennes (somme de variables aléatoires i.i.d.). C'est pourquoi les scores des innocents sont indépendants. C'est une condition nécessaire pour appliquer l'algorithme EM à l'étape 2. Cette affirmation est fausse pour les scores des *colluders*, mais nos résultats expérimentaux montrent que cela n'empêche pas la convergence de cet algorithme pour un petit nombre de *colluders* $c \ll n$.

Nous allons maintenant détailler les étapes clés de notre algorithme.

3.1 Estimation de la stratégie : étape S3

On considère les scores $\mathbf{S}$ et les probabilités associées dans le vecteur $\hat{\mathbf{T}}$; $\hat{T}_j = 1$ signifie que l'utilisateur j est un *colluder*, $\hat{T}_j = 0$ signifie qu'il est innocent. Nous estimons la taille de la collusion par $\hat{c} = \lceil \Sigma_{j=0}^{n} \hat{T}_j \rceil$. On considère ensuite les $\hat{c}$ utilisateurs correspondant aux plus grandes probabilités $\hat{T}_j$, et on utilise leurs séquences pour calculer le modèle de collusion $\hat{\boldsymbol{\theta}}$. Pour chaque position i, on calcule σ_i comme somme des X_{ji} pour chaque utilisateur j dans l'ensemble estimé des *colluders*. Pour chaque valeur possible de $0 \leq \sigma_i \leq c$, on calcule la moyenne des éléments de $\mathbf{Y}$ correspondants.

3.2 Optimisation de l'accusation : étape S4

Les nouvelles fonctions d'accusation sont obtenues par une optimisation sous contraintes, pour une collusion estimée donnée $\hat{\boldsymbol{\theta}}$. On note μ_{Inn} et ν_{Inn} (resp. μ_{Coll} et ν_{Coll}) l'espérance et la variance de la distribution des scores des utilisateurs innocents (resp. *colluders*), et $\kappa(S_j, S_k)$ la covariance entre les scores des utilisateurs j et k. De par la construction du code, les symboles sont i.i.d. d'un index à l'autre. Ceci implique, avec (1), que les statistiques des scores sont linéaires en m :

$$\mu_{Inn} = m\tilde{\mu}_{Inn}, \qquad \nu_{Inn} = m\tilde{\nu}_{Inn}, \tag{2}$$

$$\mu_{Coll} = m\tilde{\mu}_{Coll}, \qquad \nu_{Coll} = m\tilde{\nu}_{Coll}. \tag{3}$$

On résume les contraintes principales :
- Les scores des innocents sont centrés : $\tilde{\mu}_{Inn} = 0$,
- Les scores des innocents sont normalisés : $\tilde{\nu}_{Inn} = 1$,
- deux utilisateurs innocents ont des scores indépendants, ce qui se traduit sous les affirmations gaussiennes, par $\kappa(S_j, S_k) = 0$.

Ce sont les mêmes contraintes que dans T. Furon *et al* [1], excepté qu'ici on ne contraint pas la variance des scores des *colluders*.

La distance de Kullback-Leibler mesure la « distance » entre les distributions des scores des *colluders* et des innocents. La théorie de la détection nous dit qu'elle doit être la plus grande possible afin d'assurer une bonne séparation entre les innocents et les *colluders*, afin d'assurer des verdicts fiables. Comme nous avons déjà considéré que les scores des *colluders* suivent une distribution normale $\mathcal{N}_{Coll}$, et que les scores des utilisateurs innocents une distribution normale $\mathcal{N}_{Inn}$, la distance de Kullback-Leibler entre deux distributions normales satisfaisant $\tilde{\mu}_{Inn} = 0$ et $\tilde{\nu}_{Inn} = 1$ est la suivante :

$$D_{KL}\left(\mathcal{N}_{Coll}, \mathcal{N}_{Inn}\right) = \frac{1}{2}\left(m\tilde{\mu}_{Coll}^2 - \log(\tilde{\nu}_{Coll}) + \tilde{\nu}_{Coll} - 1\right). \tag{4}$$

Comme m est très grand, le terme prépondérant de la somme est $m\tilde{\mu}_{Coll}^2$. Notre objectif est de maximiser $\tilde{\mu}_{Coll}$ sous les contraintes $\mu_{Inn} = 0$, $Cov(S_j, S_k) = 0$, et $\tilde{\nu}_{Inn} = 1$.

Theorem 1 *Considérant ces conditions, les fonctions qui maximisent $\tilde{\mu}_{Coll}$ sont*

$$g_{11}(p, \boldsymbol{\theta}) = \frac{1}{2\lambda}\frac{1-p}{q(p,\boldsymbol{\theta})}A(p,\boldsymbol{\theta}), \quad g_{00}(p,\boldsymbol{\theta}) = \tfrac{1}{2\lambda}\tfrac{p}{1-q(p,\boldsymbol{\theta})}A(p,\boldsymbol{\theta}),$$

$$g_{10}(p,\boldsymbol{\theta}) = -\frac{p}{1-p}g_{11}(p,\boldsymbol{\theta}), \qquad g_{01}(p,\boldsymbol{\theta}) = -\tfrac{1-p}{p}g_{00}(p,\boldsymbol{\theta}) \tag{5}$$

avec

$$\lambda = \tfrac{1}{2}\sqrt{\mathbb{E}_p\left[A^2(p,\boldsymbol{\theta})\frac{p}{q(p,\boldsymbol{\theta})}\frac{1-p}{1-q(p,\boldsymbol{\theta})}\right]}, \tag{6}$$

$$q(p,\boldsymbol{\theta}) = \mathbb{P}(Y=1|P=p,\boldsymbol{\theta}), \tag{7}$$

$$A(p,\boldsymbol{\theta}) = \mathbb{P}(Y=1|X=1,P=p,\boldsymbol{\theta}) - \mathbb{P}(Y=1|X=0,P=p,\boldsymbol{\theta}). \tag{8}$$

Les résultats nous permettent de calculer l'expression du μ_{Coll} maximisé :

$$\tilde{\mu}_{Coll} = \sqrt{\mathbb{E}_p\left[A^2(p,\boldsymbol{\theta})\frac{p}{q(p,\boldsymbol{\theta})}\frac{1-p}{1-q(p,\boldsymbol{\theta})}\right]}. \tag{9}$$

Preuve :
On maximise cette expression en utilisant un Lagrangien $J(g_{11}, g_{00}) = \tilde{\mu}_{Coll} - \lambda(\tilde{\nu}_{Inn} - 1)$. voir les détails dans [4]. □

3.3 Résultats expérimentaux

Lors des expériences, nous considérons différentes stratégies utilisées par les *colluders* pour créer $\mathbf{Y}$:

Uniforme les *colluders* choisissent au hasard un symbole parmi leurs copies : $\mathbb{P}(Y=1|\Sigma=\sigma) = \sigma/c$;

Majorité les *colluders* choisissent le symbole le plus fréquent : $\mathbb{P}(Y=1|\Sigma=\sigma) = 1$ si $\sigma > c/2$, 0 sinon ;

Minorité les *colluders* choisissent le symbole le moins fréquent : $\mathbb{P}(Y=1|\Sigma=\sigma) = 0$ si $\sigma > c/2$, 1 sinon ;

All1 si ils ont au moins un '1', les *colluders* mettent un '1' : $\mathbb{P}(Y=1|\Sigma=\sigma) = 1$ si $\sigma \neq 0$;

All0 si ils au moins un '0', les *colluders* mettent un '0' : $\mathbb{P}(Y=1|\Sigma=\sigma) = 0$ si $\sigma \neq c$.

De plus, par la « marking assumption » on a toujours $\mathbb{P}(Y=1|\Sigma=0) = 0$ et $\mathbb{P}(Y=1|\Sigma=c) = 1$.

On calcule le ratio $m\tilde{\mu}_{Coll}/\sqrt{\tilde{\nu}_{Inn}}$ obtenu avec les fonctions optimisées du théorème 1, et on les compare dans le tableau 1 aux précédents résultats de T. Furon *et al* [1], pour lequels $\tilde{\nu}_{Coll}$ était contrainte à être égale à 1. On voit que nos fonctions d'accusation sont, comme espéré, plus efficaces pour la stratégie des *colluders* pour laquelle elles ont été calculées. On voit aussi qu'elles sont plus efficaces que celles de [1] dans tous les cas, y compris lorsque les stratégies ne coïcident pas.

Première expérience. Nous supposons d'abord que le décodeur connaît c. Les performances sont inégales selon la stratégie de la collusion comme on le voit sur la figure 3.3. De bons résultats sur 'All1', 'All0', 'Majorité' et 'Minorité', mais l'amélioration par rapport à la version de B. Skoric *et al* est mitigée en ce qui concerne la stratégie 'Uniforme'. Cela se comprend quand on regarde la table 1. Nos nouvelles fonctions ont

TAB. 1 – Valeurs de $m\tilde{\mu}_{Coll}/\sqrt{\tilde{\nu}_{Inn}}$ obtenues après optimisation des fonctions d'accusation pour $m = 100$, $c = 3, 4, 5$. Entre parenthèses sont données celles obtenues par T. Furon *et al* [1]. Rappelons qu'avec les fonctions de B. Skoric *et al* [3] on a 64 dans tous les cas.

		Stratégie des *colluders*				
c	accusation	Uniforme	Majorité	Minorité	All1	All0
3	Uniforme	**98 (71)**	106 (80)	100 (53)	97 (66)	97 (66)
	Majorité	96 (67)	**110 (84)**	100 (34)	95 (59)	95 (59)
	Minorité	81 (50)	59 (38)	**112 (75)**	89 (56)	89 (56)
	All1	83 (69)	88 (73)	88 (62)	**114 (68)**	84 (68)
	All0	83 (69)	88 (73)	88 (62)	84 (68)	**114 (68)**
4	Uniforme	**98 (71)**	106 (80)	105 (44)	99 (62)	99 (62)
	Majorité	96 (67)	**110 (84)**	105 (17)	97 (50)	97 (50)
	Minorité	61 (34)	25 (15)	**128 (91)**	88 (53)	88 (53)
	All1	79 (65)	83 (63)	88 (72)	**121 (67)**	87 (67)
	All0	79 (65)	83 (63)	88 (72)	87 (67)	**121 (67)**
5	Uniforme	**98 (71)**	110 (83)	110 (33)	100 (58)	100 (58)
	Majorité	94 (63)	**120 (93)**	113 (-22)	98 (35)	98 (35)
	Minorité	37 (19)	-20 (-17)	**155 (121)**	82 (52)	82 (52)
	All1	77 (59)	83 (47)	90 (90)	**128 (69)**	90 (69)
	All0	77 (59)	83 (47)	90 (90)	90 (69)	**128 (69)**

vraiment de plus grandes espérances (augmentation de 100%) pour 'All1', 'All0', 'Majorité' et 'Minorité', alors que l'amélioration n'est pas aussi importante pour la stratégie 'Uniforme'.Le décodage originel de B. Skoric *et al* se comporte de la même façon quelque soit la stratégie, masquant le fait que les stratégies déterministes sont bien moins dangereuses que les probabilistes.

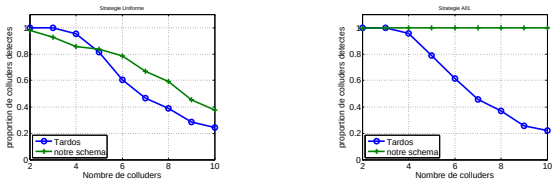

FIG. 2 – Comparaison de notre schéma avec le décodage de Tardos/Skoric [3] pour les stratégies 'Uniforme' (à gauche) et 'All1' (à droite). Les courbes donnent la proportion de *colluders* détectés en fonction de la taille de la collusion pour les paramètres $m = 1000$, $c \in \{2, \ldots, 10\}$, $n = 5000$.

Seconde expérience. Maintenant, le décodeur ignore *a priori* la valeur de c. C'est l'algorithme EM qui va en fournir une estimation. La précision de cette estimation apparaît comme fortement liée au quotient c/n. Nous avons remarqué que si c est trop petit, l'algorithme EM échoue et la stratégie estimée $\hat{\boldsymbol{\theta}}$ n'est pas du tout précise. On peut vérifier ici que le procédé d'accusation de Tardos/Skoric est indépendant de la stratégie des *colluders* alors que notre algorithme y est fortement lié.

La figure 3.3 montre que quand la taille de la collusion est grande, l'algorithme EM joue son rôle avec précision et les performances de notre décodage sont bien meilleures que le décodage de Tardos, mais elles sont très dépendantes de la stratégie de la collusion.

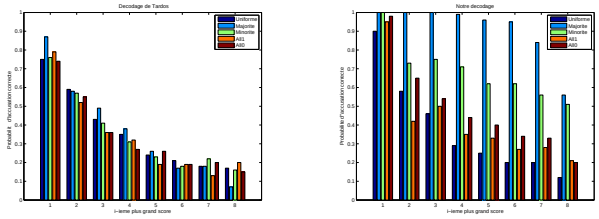

FIG. 3 – Comparaison de notre schéma (à droite) avec le décodage de Tardos/Skoric [3] (à gauche) pour les stratégies 'Uniforme', 'Majorité', 'Minorité', 'All0' and 'All1'. On donne la probabilité d'accuser avec raison le k-ième plus grand score, pour les paramètres $k \in \{1, \ldots, 8\}$. $m = 1000$, $c = 8$, $n = 5000$.

4 Conclusion

Nous avons exhibé de nouvelles fonctions d'accusation, meilleures que les fonctions originales lorsque l'on identifie correctement la stratégie de la collusion. Le gain de performance est très inégal selon les stratégies. La question qui reste à résoudre est de trouver pourquoi certaines stratégies de collusion sont plus dures à contrer que d'autres pour une taille de collusion donnée c, et ainsi identifier la pire d'entre elles. La structure itérative de notre décodeur nous permet d'estimer la taille et la stratégie de la collusion. Cependant, son efficacité est montrée expérimentalement seulement lorsque c est grand, tandis que de petites tailles de collusion sont sources d'imprécisions. Là encore, il reste deux questions fondamentales : « Y a t-il des stratégies de collusion plus dures à identifier que d'autres ? » et « Quel est le meilleur estimateur de stratégie ? »

Références

[1] T.Furon, A.Guyader et F.Cerou. *On the design and optimization of Tardos probabilistic fingerprinting codes*. Information Hiding, 2008

[2] G.Tardos. *Optimal probabilistic fingerprint codes*. Proc. of the 35th annual ACM symposium on theory of computing, 2003

[3] B.Skoric, S.Katzenbeisser et M.Celik. *Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes*. Designs, Codes and Cryptography, 2008.

[4] A.Charpentier, F.Xie, C.Fontaine et T.Furon *Expectation Maximization decoding of Tardos probabilistic fingerprinting code*. SPIE, 2009.

[5] T.Furon, A.Guyader et F.Cerou *Experimental assessment of the reliability for watermarking and fingerprinting schemes*. EURASIP J. on Information Security, 2008.