Attaques et évaluation des filigranes numériques

Fabien A. P. Petitcolas

University of Cambridge, Computer Laboratory
Pembroke Street, Cambridge CB2 3QG, UK
fapp2@cl.cam.ac.uk

Résumé : La facilité avec laquelle les documents multimédias peuvent être copiés a conduit beaucoup de chercheurs à s'interroger sur de nouvelles méthodes pour protéger la propriété intellectuelle dans le monde numérique. Les filigranes numériques ont été proposés comme un outil miracle pour contrôler l'usage de ces documents. Nous montrons que ce n'est pas le cas. D'une part il est encore trop facile de contourner beaucoup de systèmes existants. D'autre part, malgré le grand nombre de publications récentes dans ce domaine, il y a un manque évident de méthodes pour évaluer et comparer de tels systèmes.

Mots clé : Filigranes numériques, tatouage, attaques, ban d'essai, évaluation.

1. INTRODUCTION

Au cours des dernières années, beaucoup de nouvelles techniques ont été proposées pour cacher des notices de copyright et des numéros de série personnalisés dans les documents multimédias de manière à empêcher ou pour le moins réduire, les copies illégales. Nous sommes partis du principe que des progrès utiles pouvaient être réalisés en essayant d'attaquer toute cette première génération de méthodes de marquage [1]. Dans le domaine de la cryptographie, les progrès ont été itératifs : des algorithmes ont été proposés, des attaques ont été trouvées, de meilleurs algorithmes ont vu le jour, et ainsi de suite. Finalement, la théorie a fait son entrée et nous aide à comprendre ce qui fait la force d'un chiffre bien mieux qu'auparavant. Nous pensons qu'il en va de même pour les filigranes numériques.

Beaucoup de systèmes récemment proposés sont qualifiés de « robustes » par leur inventeur. Malheureusement les critères utilisés pour démontrer cette robustesse varient d'un système à l'autre, et les attaques récentes [1, 3, 4, 5] montrent que les critères utilisés pour démontrer cette robustesse sont parfois inadéquats. Compression J.P.E.G., bruit additif, filtrage passe-bas, changement de taille ou émargement sont pris en compte par la plupart des systèmes mais les transformations géométriques, même très simples, sont rarement évoquées [6, 7]. Dans certains cas, le système est dit seulement « résistant aux procédés usuels de traitement du signal et aux déformations géométriques sur certaines images standard ».

2. ATTAQUES

Nous avons déjà présenté [1] différentes attaques mettant en évidence des limitations sérieuses de plusieurs outils de marquage dont PictureMarc 1.51 [8], SysCoP [9], SureSign [10], JK_PGS, EIKONAmark [11], Echo Hiding [17], Giovanni et la méthode NEC [12]. Il va sans dire que des systèmes utilisant les mêmes techniques sont susceptibles d'être attaqués de la même manière.

L'attaque de base partait du constat que beaucoup de systèmes de marquage utilisent de façon plus ou moins déguisée la technique d'étalement de spectre. Cette dernière est très robuste à l'ajout de bruit ou aux distorsions de l'amplitude du signal mais supporte très mal les erreurs de synchronisation. Une méthode très simple pour briser cette synchronisation consiste simplement à effacer quelques échantillons. Dans le cas des images, quelques colonnes de pixels suffisent. Bien qu'extrêmement simple cette attaque fonctionne sur les prototypes naïfs qui ne prennent en compte que le bruit additif.

2.1 Attaques générales

Certains systèmes de marquage d'images supportent également des manipulations simples que quiconque peut faire avec des outils de traitement d'image disponibles dans le commerce : rotation, redimensionnement, émargement, retournement horizontal et compression J.P.E.G. (Ceci est confirmé par les résultats de tests résumés dans le Tableau 1). Malheureusement, des combinaisons de celles-ci suffisent généralement à mettre en défaut le système de marquage.

C'est ce qui a motivé la mise en œuvre de StirMark, initialement programmé par Markus G. Kuhn. Nombre d'améliorations ont été ajoutées depuis, et notamment la possibilité d'utiliser cet outil comme base d'un banc d'essai.

La version originale de StirMark applique de simples déformations bilinéaires aléatoires. Si *A*, *B*, *C* et *D* sont les sommets de l'image, un point *M* de ladite image peut être exprimé de la façon suivante : $M = \alpha(\beta A + (1-\beta)D) + (1-\alpha)(\beta B + (1-\beta)\mathrm{C})$ où $0 \le \alpha,\ \beta \le 1$ sont les coordonnées de *M* par rapport aux sommets de l'image. La déformation est appliquée en déplaçant légèrement et aléatoirement les sommets dans les deux directions. Les nouvelles coordonnées de *M* sont recalculées grâce à la formule précédente en gardant (α, β) constantes. L'avant dernière ligne du Tableau 1 montre que certains systèmes de marquage supportent les déformations.

Davantage de déformations—toujours invisibles—peuvent être appliquées à une image. En plus de la transformation bilinéaire précédente, les nouvelles versions de StirMark dévient légèrement chaque pixel de façon non uniforme : quelques 0,1% des dimensions de l'image au centre et quasiment rien sur les bords. Dans la version actuelle, la forme de cette « bosse » est simplement une fonction sinus : si (x, y), avec $0 \le x \le X$ et $0 \le y \le Y$, sont les coordonnées d'un pixel dans l'image après la déformations bilinéaire, alors ses nouvelles coordonnées sont : $x' = x + \lambda \sin(\pi y / Y)$ et $y' = y + \lambda \sin(\pi x / X)$. À cela est ajouté un déplacement de plus grande fréquence de la forme $\delta = \lambda \sin(\omega_x x)\sin(\omega_y y)(1 + n(x, y))$ où *n* est un nombre aléatoire : $x'' = x' + \delta_1$ et $y'' = y' + \delta_2$. Pour une bonne rapidité de traitement, le rééchantillonage utilise un algorithme d'approximation quadratique par B-spline [14] et, pour de meilleurs résul-

tats une légère compression J.P.E.G. est appliquée à la fin du processus. Un exemple d'image « attaquée » est donné dans la Figure 1.

Il existe aussi des méthodes générales pour attaquer les outils de marquage du son. Par exemple, les techniques de restauration de signaux sonores ont été étudiées en détail depuis de nombreuses années, et se sont montrées efficaces pour localiser et enlever les dégradations qui apparaissent dans les anciens enregistrements [15]. Ces mêmes méthodes peuvent être utilisées contre les outils de marquage du son. Notre attaque « reconstruit » simplement le signal bloc par bloc en utilisant une partie du

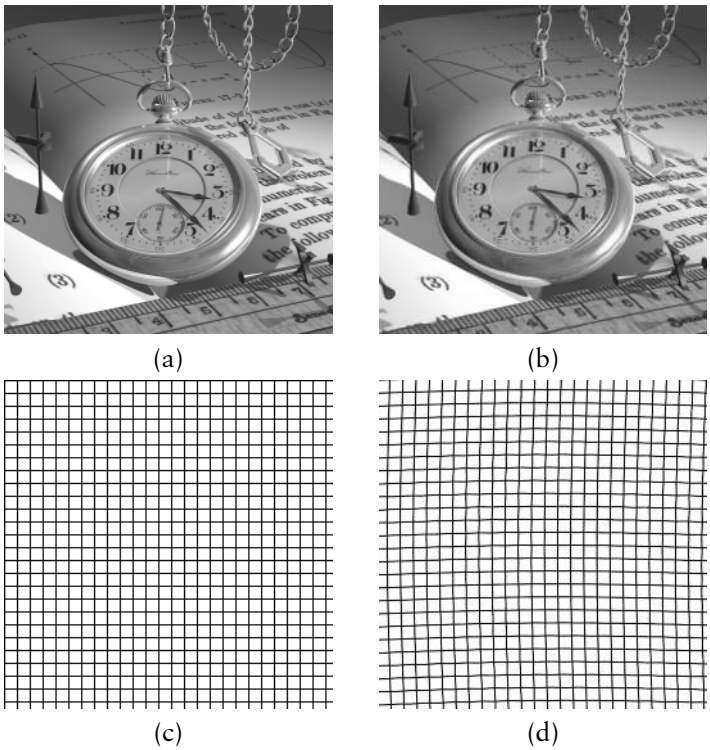

Figure 1—Lorsque StirMark est appliqué sur des images relativement complexes ou des photographies, les déformations introduites sont quasiment invisibles : *watch* avant (a) et après (b) StirMark (paramètres par défaut). Pour comparaison les mêmes déformations ont été appliquées à une grille (c et d). Image de synthèse : *Pocket Watch on a Gold Chain*. Copyright Kevin Odhner (jko@home.com)

	Digimarc 1.51	SureSign 3.0 Demo	EikonaMark 3.01	Giovanni 1.1.0.2	SysCoP 1.0R1
Conversion GIF	100	100	100	60	80
Échelle (0,5, 0,75, 0,9, 1,1, 1,5, 2)	70	100	0	63	0
Émargement (1, 2, 5, 10, 15, 20, 25, 50 %	100	100	0	15	0
Rotation (-2, -1, -0,5, 0,5, 1, 2 °)	82	58	0	10	0
J.P.E.G. (90, 85, 80, 75, 60, 50, 25, 10, 5)	56	72	90	12	58
Filtrage (médian 3×3, Gauss)	100	100	100	60	80
Retournement horizontal	100	100	0	0	0
StirMark 1.0	80	80	0	0	0
StirMark 2.2	0	0	0	0	0

Tableau 1—Test de robustesse pour cinq outils de marquage. Les valeurs sont des pourcentages de réussite. Pour chaque outil, cinq images de test *(lena, lunettes, fille, mercedes* et *babouin)* ont été utilisées. Chaque image a été tatouée avec les meilleurs paramètres n'introduisant aucun effet désagréable (à l'oeil). Bien que toute comparaison doive être effectuée avec le plus grand soin (tous ces outils de marquage n'ayant pas la même application), ce tableau confirme l'état de l'art dans le domaine.

signal marqué pour prédire chaque bloc. Un simple modèle auto-régressif est utilisé pour la prédiction, dont l'algorithme est détaillé dans [16].

2.2 Attaques spécifiques

Lorsque les méthodes générales ne permettent pas d'attaquer un système stéganographique, rien n'empêche un adversaire d'utiliser des méthodes spécifiques. C'est ce que nous avons fait pour la méthode *echo hiding* qui dissimule de l'information en introduisant des échos de très court délai, de l'ordre de la milliseconde, imperceptibles à l'oreille [17]. L'attaque évidente (détaillée dans [1]) consiste simplement à détecter les paramètres de l'écho en utilisant la même méthode que les inventeurs d'*echo hiding*, c'est-à-dire « l'alanyse cepstrale » de Bogert et al. [18]. Des essais sur des signaux aléatoires et sur de la musique montrent que notre méthode est relativement précise pour des échos entre 0,5 et 3 ms et permet d'extraire le signal caché, montrant ainsi une faille dans la méthode originale d'*echo hiding*.

Les faiblesses inhérentes au marquage en général peuvent aussi être utilisées malicieusement. Ceci est mis en évidence par une attaque générale contre les robots traqueurs, attaque à la propriété initiale remarquable que l'image marquée et l'image attaquée sont les mêmes. Ces robots, qui font partie d'un système de détection automatique basé sur le *Web*, téléchargent image après image et vérifient si elles contiennent une marque. L'attaque consiste simplement à découper l'image en plusieurs parties, telle une mosaïque, et à générer du code H.T.M.L. de telle façon qu'un fureteur « recolle » les morceaux (Figure 2). Cette attaque est relativement générale puisque toute méthode de marquage d'image requiert une taille minimale. Par conséquent, cette division de l'image empêche le détecteur de retrouver la marque.

À cette attaque très simple, il convient d'en ajouter d'autres, basées sur des appliquettes *Java* ou des contrôles *ActiveX*, qui peuvent être utilisés pour télécharger et afficher l'image à l'intérieur du fureteur ; ces objets peuvent éventuellement dé-brouiller l'image en temps réel si nécessaire. Déjouer de telles techniques, impliquerait de détecter les images dans la représentation de la page *Web* en mémoire et

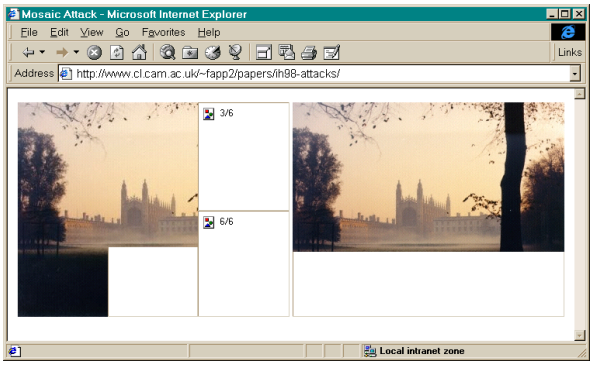

Figure 2—Copie de la fenêtre d'un fureteur en train de télécharger une image après une attaque par mosaïque. Cette attaque découpe simplement l'image en morceaux qui sont automatiquement « recollés » par le fureteur lors de la présentation du résultat. Un logiciel permettant d'automatiser le découpage et la génération du code H.T.M.L. correspondant est disponible [19]. Dans certains cas le chargement de la mosaïque est même plus rapide que l'image entière ! Pour cet exemple l'image (350×280 pixels) a été marquée avec PictureMarc 1.51 de Digimarc. Photographie : *Chapelle de King's College, Cambridge*. Copyright John Thompson, JetPhotographic, Cambridge.

d'essayer d'en extraire une marque. Les méthodes pour déjouer les robots traqueurs ne manquent donc pas et l'on est en droit de s'interroger sur l'utilité réelle de tels outils.

2.3 Attaque sur le protocole d'identification du propriétaire

La plupart des attaques « évidentes » tentent de filtrer, ou déformer l'image, pour atténuer le signal, ou empêcher la synchronisation du détecteur. Malheureusement cet aspect ne représente qu'une petite fraction des possibilités offertes. Chaque élément de la chaîne de gestion des copyrights peut être mis en défaut.

Par exemple, Craver et al. ont montré comment empêcher un photographe de prouver qu'il est bien le détenteur du copyright d'une image même après l'avoir marquée [20]. L'idée de base utilise le fait que beaucoup d'outils de marquage ne permettent pas de savoir quelle marque parmi plusieurs a été ajouté la première : le processus de marquage est souvent additif, ou tout au moins commutatif et inversible. Par conséquent, si un ayant droit possède le document d, y dissimule un filigrane f, publie la version marqué, c'est-à-dire $d + f$, et n'a pas d'autre preuve de propriété, un pirate qui a enregistré un filigrane f', peut très bien prétendre que le document est le sien et que la version originale est $d + f - f'$.

Craver et al. concluent qu'il faut utiliser des systèmes de marquage dont l'inverse ne peut être calculé facilement ; autrement dit, des systèmes dont la marque dépend de l'original. Cependant notre interprétation de cette attaque est que filigranes et empreintes doivent impérativement être utilisés dans des systèmes plus larges, mettant en œuvre des mécanismes de datage sécurisé et des « notaires » électroniques.

Les mécanismes d'enregistrement ont reçu peu d'attention et les publications qui abordent le problème [21–23] s'intéressent seulement à la protection du propriétaire et très peu aux droits des consommateurs qui peuvent très bien être abusés.

2.4 Problème de mise en œuvre

La robustesse des mécanismes de marquage et d'extraction n'est pas le seul problème à considérer. La plupart des attaques sur des outils de prestations cryptographiques proviennent de l'exploitation de failles découvertes accidentellement. La cryptanalyse s'est rarement révélée indispensable [24].

Il en est de même pour les outils de marquage. Une méthode (illégale) pour modifier directement le programme PictureMarc de Digimarc était déjà disponible en août 1997 [25]. Elle utilise un outil de décompilation pour modifier directement le binaire du logiciel afin de pouvoir ajouter une marque quelconque même sur une image déjà marquée. Si les logiciels résistant aux modifications de leur code [26] ne donnent pas entière satisfaction, il reste les mécanismes en ligne avec lesquels chaque utilisateur partage une clé avec un tiers de confiance.

Plus grave, nous avons déjà remarqué [1] qu'un utilisateur quelconque peut marquer très simplement une image à la place d'un autre, ouvrant la porte à un grand nombre d'abus. Une attaque similaire, due à Perrig, consiste à utiliser un détecteur/extracteur public comme « devin » : des modifications infimes peuvent être appliquées au signal marqué jusqu'à ce que le décodeur échoue. Chaque détection apporte théoriquement un bit d'information : y a-t-il ou pas de marque [23].

3. BANC D'ESSAIS

Toutes ces attaques débouchent sur la même question : comment évaluer et comparer différents outils de marquage. Très peu d'auteurs ont publié des résultats de tests intensifs sur leurs outils de marquage (e.g., [27]). Un banc d'essai est donc nécessaire pour mettre en évidence les domaines de recherche prometteurs et pour comparer rapidement les nouveaux algorithmes qui apparaissent régulièrement.

Aujourd'hui encore, chaque chercheur utilise sa propre batterie de tests, ses propres images et sa propre méthodologie. Par conséquent, toute comparaison est impossible sans reprogrammer la méthode, dans les cas où il n'existe pas de logiciel d'évaluation. Avec un banc d'essai commun—même imparfait—les avantages sont évidents : un tableau d'évaluation type pourrait être fourni avec chaque nouvel algorithme, permettant ainsi d'avoir une idée de sa robustesse sans perdre plusieurs jours pour comprendre et évaluer la méthode.

Une première tentative, basée sur StirMark est proposée dans [29]. Elle prend uniquement en compte les processus de marquage et d'extraction qui sont considérés comme des boîtes noires (cf. Tableau 2 pour des résultats). La procédure est très simple :

- Marquer avec les meilleurs paramètres les images fournies avec StirMark de telle façon que le P.S.N.R. (ou une autre mesure à définir) soit inférieur à 38 dB.
- Utiliser StirMark pour appliquer une série automatique de tests en une ligne de commande.
- Pour chaque image attaquée, tenter de détecter/extraire la marque (1 point en cas de succès ; 0 sinon).

Notons au passage que si l'outil de marquage offre une interface sous forme de ligne de commande, cette procédure peut être entièrement automatisée en utilisant des scriptes Unix, Perl ou DOS.

Ce schéma général contient encore quelques inconnues, et notamment le nombre de bits cachés par le marqueur et la mesure de qualité. Pour le premier, il semble que 70 bits soit raisonnable [30]. Pour le second, il reste à prouver que la mesure utilisée a une influence significative sur les résultats des tests. La plupart des outils de marquage récents utilisent des modèles basés sur le système humain de perception, si le P.S.N.R. est utilisé comme mesure de qualité, on voit difficilement comment une méthode pourrait être avantagée par rapport à une autre.

4. CONCLUSION

Nous avons expliqué brièvement pourquoi la plupart des outils de marquage sont vulnérables à différentes attaques relativement simples et notamment aux déformations géométriques aléatoires utilisées par StiMark ou aux méthodes de restauration du signal dans le cas de signaux sonores. Nous avons également montré la nécessité d'une méthode d'évaluation et d'un banc d'essai pour les outils de marquage de copyright.

Afin d'augmenter la résistance d'un système de marquage à différentes attaques, on peut essayer de prévoir les déformations possibles qu'un pirate peut utiliser : La marque pourrait alors être cachée dans l'espace de transformation inverse, ou dans un espace invariant à l'attaque. Ó Ruanaidh et Pun, par exemple, proposent d'utiliser la transformée de Mellin afin de résister aux rotations d'angle quelconques et aux changements de taille [6].

Dans le cas d'attaques plus générales comme StirMark, on peut remarquer que les déformations, bien que globalement aléatoires, sont quasiment linéaires sur une petite surface de l'image. Ainsi, en décomposant l'image en petits blocs, il devrait être possible d'augmenter, par exemple, la valeur de corrélation entre le signal reçu et le signal d'étalement.

L'étude de notre perception des déformations géométriques devrait également permettre de modéliser encore mieux les images et d'améliorer la résistance des marques à des attaques comme StirMark.

Enfin peu de recherches ont été entreprises pour étudier les effets des corrélations partielles, introduites par des attaques spécifiques et par certaines déformations, sur les systèmes à étalement de spectre.

	Digimarc	Unige	SureSign	SCMark
Filtrage				
Gauss	100	100	100	100
Médian	100	100	100	100
Rendre plus net	100	100	100	100
F.M.L.R.	100	67	100	100
Compression				
J.P.E.G.	65	63	87	100
GIF/Quantification des couleurs	100	1	100	20
Échelle				
Sans J.P.E.G. 90	81	86	97	0
Avec J.P.E.G. 90	72	83	83	0
Émargement				
Sans J.P.E.G. 90	100	83	94	2
Avec J.P.E.G. 90	98	83	91	2
Cisaillement				
X	50	38	42	0
Y	50	21	42	0
Rotation				
Auto-émargement	98	98	37	2
Auto-redimensionnement	97	98	51	26
Autres transformations géométriques				
Effacement de lignes et colonnes	100	83	89	7
Retournement horizontal	100	100	100	0
Déformations aléatoires (StirMark)	17	0	0	0

Tableau 2—Résumé des résultats d'un banc d'essai basé sur StirMark 3.0. Un tableau détaillé est disponible sur `www.cl.cam.ac.uk/~fapp2/watermarking/benchmark/`. Outils de marquage testés : Batch Embedding Tool 1.00.13 et ReadMarc 1.5.8 de Digimarc, outils de l'Université de Genève (version du 15 janvier 1999), SureSign Server 1.94 de Signum Technologies et un outil de marquage de l'Université de Californie du sud (version du 29 mars 1999). Les séparations verticales indiquent d'une part que les conditions expérimentales étaient légèrement différentes pour Signum, et d'autre part, que le type de marquage est différent pour SCMark puisque celui-ci est privé, en ce sens qu'il utilise l'image originale.

REMERCIEMENTS

L'auteur de cet article remercie vivement la Société Intel pour son soutien financier sous la forme d'une bourse de recherche « *robustness of information hiding systems* ». Certaines idées présentées ici ont été clarifiées avec Ross Anderson, Gabriela Csurka, Frédéric Deguillaume, Jean-Luc Dugelay, David Hilton, Shelby Pereira, Burt Perry and Thierry Pun. Remerciements particuliers au *Computer Vision Group* de l'Université de Genève, à la Société Digimarc, à la Société Signum Technologies et à Po-Chyi Su pour la fourniture de logiciels d'évaluation nécessaires à cette étude.

BIBLIOGRAPHIE

1 Fabien A. P. Petitcolas, Ross J. Anderson et Markus G. Kuhn. *Attacks on copyright marking systems*. In Aucsmith [2], pages 218–238, ISBN 3-540-65386-4.

2 David Aucsmith, éditeur. *Information hiding: second international workshop*, volume 1525 de *Lecture Notes in Computer Science*, Portland, Oregon, U.S.A., avril 1998. Springer Verlag, Berlin, Allemagne. ISBN 3-540-65386-4.

3 Jean-Paul M. G. Linnartz et Marten van Dijk. *Analysis of the sensitivity attack against electronic watermarks in images*. In Aucsmith [2], pages 258–272. ISBN 3-540-65386-4.

4 Maurice Maes. *Twin peaks: the histogram attack on fixed depth image watermarks*. In Aucsmith [2], pages 290–305. ISBN 3-540-65386-4.

5 Gerrit C. Langelaar, Reginald L. Lagendijk et Jan Biemond. *Removing spatial spread spectrum watermarks by non-linear filtering*. In *9th European Signal Processing Conference (EUSIPCO'98)*, pages 2281—2284, Île de Rhodes, Grèce, 8–11 septembre 1998. ISBN 960-7620-05-4.

6 Joseph J. K. Ó Ruanaidh et Thierry Pun. *Rotation, scale and translation invariant spread spectrum digital image watermarking*. Signal Processing, volume 66, numéro 3, pages 303–317, mai 1998. ISSN 0165-1684. *European Association for Signal Processing* (EURASIP).

7 Martin Kutter. *Watermarking resisting to translation, rotation, and scaling*. In *Proceedings of S.P.I.E. International Symposium on Voice, Video, and Data Communications*, volume 3528, pages 423–431, Boston, U.S.A., novembre 1998.

8 Geoffrey B. Rhoads. *Steganography methods employing embedded calibration data*. Digimarc Corporation. Brevet U.S.A. 5.636.292, 3 juin 1997.

9 E. Koch et J. Zhao. *Towards robust and hidden image copyright labeling*. In *Workshop on Nonlinear Signal and Image Processing*, pages 452–455, Neos Marmaras, Grèce, 20–22 juin 1995. I.E.E.E.

10 Signum Technologies – SureSign digital fingerprinting. `www.signumtech.com`, octobre 1997.

11 Ioannis Pitas. *A method for signature casting on digital images*. In *International Conference on Image Processing*, volume 3, pages 215–218, septembre 1996.

12 Ingemar J. Cox, Joe Kilian, Tom Leighton et Talal Shamoon. *A secure, robust watermark for multimedia*. In Anderson [12], pages 183–206. ISBN 3-540-61996-8.

13 Ross J. Anderson, éditeur. *Information hiding: first international workshop*, volume 1174 de *Lecture notes in Computer Science*, Newton Institute, Cambridge, Grande Bretagne. Springer Verlag, Berlin, Allemagne, mai 1996. ISBN 3-540-61996-8.

14 Neil A. Dodgson. *Quadratic interpolation for image resampling*. I.E.E.E. Transactions on Image Processing, volume 6, numéro 9, pages 1322–1326, septembre 1997. ISSN 1057-7149.

15 Simon J. Godsill, Peter J.W. Rayner et Olivier Cappé. *Digital audio restoration*. In Mark Kahrs et Karlheinz Brandenburg, éditeurs, *Applications of Digital Signal Processing to Audio and Electroacoustics*. Kluwer Academic Publishers, 1998.

16 Fabien A. P. Petitcolas et Ross J. Anderson. *Evaluation of copyright marking systems*. Présenté à *I.E.E.E. International Conference on Multimedia Computing & Systems*, Florence, Italie, 7–11 juin 1999.

17 Daniel Gruhl, Walter Bender et Anthony Lu. *Echo hiding*. In Anderson [12], pages 295–315. ISBN 3-540-61996-8.

18 Bruce P. Bogert, M.J.R. Healy et John W. Tukey. *The quefrency alanysis of time series for echoes: cepstrum, pseudo-autocovariance, cross-ceptstrum and saphe cracking*. In M. Rosenblatt, éditeur, *Symposium on Time Series Analysis*, pages 209–243, New-York, U.S.A., 1963. John Wiley & Sons, Inc.

19 Fabien A. P. Petitcolas. 2Mosaic. `www.cl.cam.ac.uk/~fapp2/watermarking/image_watermarking/2mosaic/`, octobre 1997.

20 Scott Craver, Nasir Memon, Boon-Lock Yeo et Minerva M. Yeung. *Resolving rightful ownerships with invisible watermarking techniques: limitations, attacks, and implications*. I.E.E.E. Journal of Selected Areas in Communications (J-SAC) – Numéro spécial sur la protection des copyright et de la vie privée, volume 16, numéro 4, pages 573–586, mai 1998. ISSN 0733-8716.

21 Marc Cooperman et Scott A. Moskowitz. *Steganographic method and device*. Société DICE. Brevet U.S.A. 5.613.004, 18 mars 1995.

22 Alexander Herrigel, Joseph J. K. Ó Ruanaidh, Holger Petersen, Shelby Pereira, et Thierry Pun. *Secure copyright protection techniques for digital images*. In Aucsmith [2], pages 170–191.

23 Adrian Perrig. *A copyright protection environment for digital images*. Rapport de *Diploma*, École Polytechnique Fédérale de Lausanne, Lausanne, Suisse, février 1997.

24 Ross J. Anderson. *Why cryptosystems fail*. Communications of the A.C.M., volume 37, numéro 11, pages 32–40, novembre 1994.

25 Anonyme (zguan.bbs@bbs.ntu.edu.tw). *Learn cracking IV – another weakness of PictureMarc*. `news://tw.bbs.comp.hacker`. Copie disponible sur `www.cl.cam.ac.uk/~fapp2/watermarking/image_watermarking/digimarc_crack.html`, août 1997.

26 David Aucsmith. *Tamper resistant software: an implementation*. In Anderson [12], pages 317–333. ISBN 3-540-61996-8.

27 Gordon W. Braudaway. *Results of attacks on a claimed robust digital image watermark*. In van Renesse [28]. ISBN 0-8194-2556-7.

28 Rudolf L. van Renesse , éditeur. *Optical Security and Counterfeit Deterrence Techniques II*, volume 3314, San Jose, Californie, U.S.A., 28–30 janvier 1998. La Société pour les sciences et techniques de l'image (I.S.&T.) et la Société internationale d'ingénierie optique (S.P.I.E.). ISSN 0277-786X. ISBN 0-8194-2556-7.

29 Martin Kutter et Fabien A. P. Petitcolas. *A fair benchmark for image watermarking systems*. In *proceedings of Electronic Imaging '99, Security and Watermarking of Multimedia Contents*, volume 3657, pages 226–239, San Jose, Californie, U.S.A., 25–27 janvier 1999. La Société internationale d'ingénierie optique (S.P.I.E.).

30 Jean-François Delaigle. *Common functional model*. Compte rendu AC019-UCL-TEL-DR-P-D12-b1, CEC, 29 mars 1996. Projet *tracing author's rights by labelling image services and monitoring access*.