

## Sécurisation d'Image par Crypto-Tatouage

William Puech, José Marconi Rodrigues

► **To cite this version:**

William Puech, José Marconi Rodrigues. Sécurisation d'Image par Crypto-Tatouage. CORESA: COMpression et REprésentation des Signaux Audiovisuels, May 2004, Lille, France. 9ème Conférence Nationale de Compression et Représentation des Signaux Audiovisuels, pp.215-218, 2004. <lirmm-00108802>

**HAL Id: lirmm-00108802**

**<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00108802>**

Submitted on 23 Oct 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Sécurisation d'image par crypto-tatouage

William PUECH

José Marconi RODRIGUES

Laboratoire LIRMM, UMR CNRS 5506, Université Montpellier II

161, rue Ada, 34392 MONTPELLIER CEDEX 05, FRANCE

william.puech@lirmm.fr, jose-marconi.rodrigues@lirmm.fr

## Résumé

*Cet article présente une nouvelle méthode combinant le chiffrement et le tatouage d'images pour le transfert sécurisé. Cette méthode est basée sur la combinaison d'algorithmes de chiffrement à clés publiques-privées et à clés secrètes, et de tatouage. L'algorithme de chiffrement d'image utilise une clé secrète. Nous chiffons ensuite cette clé secrète avec un algorithme asymétrique. Cette clé secrète chiffrée est alors insérée par tatouage dans l'image cryptée.*

## Mots clefs

Cryptages symétrique et asymétrique, tatouage d'image, transfert sécurisé.

## 1 Introduction

Le transfert ou l'archivage de données est encore actuellement peu sécurisé. Les algorithmes standards de chiffrement ne conviennent pas au cas particulier des données images. L'idéal serait de pouvoir appliquer sur les images des systèmes de chiffrement asymétriques afin de ne pas avoir de clé à transférer. Du fait de la connaissance partielle de la clé (clé publique), les systèmes asymétriques imposent l'utilisation de grands nombres supérieurs à 512 bits. Par conséquent, dans le cadre d'un transfert sécurisé d'images, le chiffrement d'images n'est pas envisageable avec l'algorithme RSA par exemple. Le fait d'utiliser des algorithmes symétriques impose d'avoir à transférer la clé secrète au récepteur. Les méthodes classiques de chiffrement d'images nécessitent le transfert de la clé secrète par un autre canal ou un autre moyen de communication [1, 2, 3, 4].

Le tatouage d'image peut également être une solution pour sécuriser le transfert d'images. L'objectif du tatouage est d'insérer une information dans l'image de manière invisible et indélébile. L'insertion du message peut s'effectuer dans le domaine spatial ou fréquentiel, ou dans une combinaison des deux domaines [5].

Dans cet article, nous proposons une méthode combinant cryptage symétrique d'images robuste au bruit et tatouage dans l'image cryptée de la clé secrète. Nous chiffrerons la clé secrète avant de l'insérer dans l'image en utilisant

un algorithme asymétrique. A la réception, seules des clés de type publique-privée seront nécessaires afin d'extraire et de déchiffrer une clé secrète de session qui permettra de rendre lisible l'image.

Le chiffrement de données peut être effectué par blocs ou par flux. Dans notre cas, les algorithmes de chiffrement par blocs présentent deux inconvénients. Premièrement, quand l'image contient des zones homogènes, tous les blocs identiques sont également identiques après chiffrement. Dans ce cas, l'image cryptée contient des zones texturées et l'entropie de l'image n'est pas maximale. Le second problème est que les méthodes de cryptage par blocs ne sont pas robustes au bruit. En effet, une erreur sur un bit chiffré va propager des erreurs importantes dans tout le bloc courant. Pour chiffrer l'image, nous avons choisi de développer un algorithme symétrique de chiffrement par flux afin d'être robuste à l'insertion dans l'image cryptée de la clé par tatouage. Pour insérer la clé secrète dans l'image, nous avons choisi une nouvelle méthode de tatouage basée sur la DCT [6]. En travaillant dans le domaine fréquentiel, il est également possible de résister durant le transfert à des taux d'erreur binaire (TEB) importants. Une application importante de ce travail concerne le transfert sécurisé d'images médicales [7, 8].

## 2 Chiffrement par flux

### 2.1 Principe du chiffrement par flux

Les algorithmes de chiffrement de flux peuvent être définis comme étant des algorithmes de chiffrement par blocs, où le bloc a une dimension unitaire (1 bit, 1 octet, etc) ou relativement petite. Leurs avantages principaux sont leur extrême rapidité et le changement de la clé de chiffrement pour chaque symbole du message clair. Ces algorithmes sont donc utilisés dans un environnement où les erreurs sont fréquentes car ils ont l'avantage de ne pas les propager [9]. Ils sont aussi utilisés lorsque l'information ne peut être traitée qu'avec de petites quantités de symboles à la fois, comme, par exemple, dans le cas où l'équipement n'a pas de mémoire physique ou une mémoire tampon très limitée. Ils appliquent des transformations simples selon un keystream donné. Le keystream est une séquence de bits utilisée en tant que clé qui peut être générée aléatoirement.

Avec un keystream choisi aléatoirement et utilisé qu’une seule fois, le message chiffré est excessivement sécurisé. En fait, les chiffrements de flux sont une approximation des propriétés théoriques de l’algorithme one time pad, appelé aussi chiffrement Vernam [10].

La génération du keystream peut être dépendante du message clair et du message chiffré (self-synchronizing stream cipher) ou indépendante, et dans ce cas appelée chiffrement de flux synchrone (synchronous stream cipher). Les chiffrements de flux les plus répandus sont synchrones.

Le Linear Feedback Shift Register (LFSR) est un mécanisme très souvent utilisé dans les chiffrements symétriques de flux. Il génère des séquences de bits pseudo-aléatoires. La série de bits appelée registre est initialisée par un vecteur d’initialisation qui est la plupart du temps la clef du chiffrement.

Le comportement du vecteur registre est défini par rapport à un compteur. A chaque itération de la boucle, le contenu du registre est décalé vers la droite d’une position et l’opération du *ou exclusif* est appliquée sur un sous-ensemble de bits (choisi selon l’algorithme), dont le résultat est placé à l’extrême gauche du registre, illustré Figure 1. A la fin de chaque itération, un bit de sortie est généralement gardé pour former le registre transformé résultant.

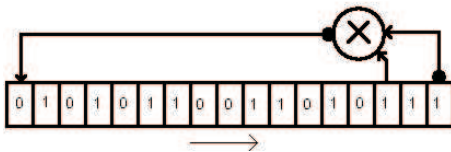


Figure 1 – Mécanisme très souvent utilisé dans les chiffrements symétriques de flux (Linear Feedback Shift Register (LFSR)).

Bien que moins nombreux que les algorithmes de chiffrement par blocs, la popularité des algorithmes de chiffrement par flux est croissante du fait de la quantité toujours grandissante d’informations circulant sur les réseaux. C’est dans le domaine des logiciels que les stream ciphers ont toute leur importance. Actuellement, les deux principaux sont les algorithmes RC4 [11] et SEAL. L’algorithme RC4 a été pensé par Ron Rivest en 1987 et développé pour la RSA Security. Il est basé sur les permutations aléatoires, avec des opérations sur des octets.

## 2.2 Chiffrement d’images par flux

De manière générale, la longueur de la clef d’un algorithme de chiffrement par flux peut être aussi longue que la longueur du message. Le principe de la méthode réside dans le fait que pour chaque pixel, le cryptage dépend de la valeur initiale du pixel, de la clef et des  $k$  pixels précédemment cryptés [12].

Pour chaque pixel  $p(n)$  de l’image originale, si  $k$  est la longueur de la clef, nous calculons la valeur du pixel  $p'(n)$

de l’image cryptée en utilisant l’équation :

$$p'(n) = p(n) + \sum_{i=1}^{i=k} \alpha(i)p'(n-i), \quad (1)$$

avec  $N$  le nombre de pixels de l’image,  $n \in [k, N]$ ,  $k \in [1, n]$  et  $\alpha(i)$  une séquence de coefficients générés par la clef secrète. L’équation (1) a donc un ordre de récurrence correspondant à la longueur  $k$  de la clef. Les coefficients  $\alpha(i)$  sont des entiers compris entre  $-2$  et  $+2$ , tels que :

$$\sum_{i=1}^{i=k} \alpha(i) = 0. \quad (2)$$

De plus, la probabilité d’apparition pour chacune des valeurs est uniforme. Le principe de chiffrement est illustré Figure 2. Une autre information apparaît également dans la clef. En effet, considérant que le cryptage d’un pixel tient compte des  $k$  pixels précédemment cryptés, il n’est pas possible de chiffrer les  $k$  premiers pixels de l’image de la même manière. Pour cela, dans la clef nous devons également coder un vecteur d’initialisation contenant  $k$  pixels virtuels  $p'(-i)$  avec  $i \in [1, k]$ . Nous avons montré avec cet algorithme qu’il était possible d’exprimer  $p'(n)$  uniquement avec les données initiales [12].

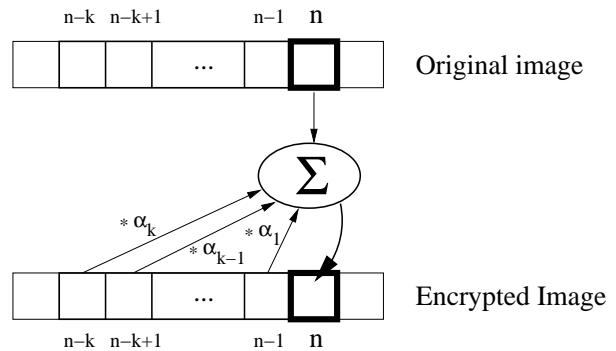


Figure 2 – Chiffrement d’un pixel par flux.

Plus la valeur de  $k$  sera importante, plus le système sera sécurisé. Supposons  $k = 64$ , comme 2 bits par coefficients  $\alpha(i)$  sont nécessaires, la longueur effective de la clef sera de 128 bits. Le vecteur d’initialisation, nécessaire pour le chiffrement des  $k$  premiers pixels, est alors construit avec la même clef de 128 bits par une opération particulière non cyclique. Les valeurs des  $k$  pixels dépendent donc également de la clef. Cette opération est basée sur une fenêtre coulissante lisant les bits de la clef. La fenêtre lit les 8 premiers bits de la clef pour le premier pixel virtuel, puis se décale sur la clef et lit à nouveau 8 bits pour le pixel virtuel suivant.

## 3 Description de la combinaison des méthodes

Nous avons développé une nouvelle méthode combinant un algorithme de chiffrement symétrique pour l’image, un

algorithme de cryptage asymétrique pour le chiffrement de la clé secrète, et une méthode de tatouage.

Prenons le cas d'une personne  $M$  souhaitant envoyer de manière sécurisée par réseau une image à une seconde personne  $S$ .  $M$  utilisera alors un algorithme rapide de chiffrement à clé secrète  $K$  pour crypter l'image. Dans notre méthode, nous avons utilisé un algorithme de chiffrement par flux, robuste au bruit, présenté Section 2.2. Pour transférer la clé  $K$ ,  $M$  devra alors crypter la clé  $K$  en utilisant un algorithme à clés publique-privée, comme RSA par exemple [13]. Soient  $pub_m$  et  $priv_m$ , les clés publique et privée de  $M$ , et  $pub_s$  et  $priv_s$ , les clés publique et privée de  $S$ . Dans un premier temps,  $M$  génère une clé secrète  $K$  pour cette session et chiffre l'image avec l'algorithme symétrique. Ensuite,  $M$  doit chiffrer la clé  $K$  avec l'algorithme RSA en utilisant sa clé privée,  $priv_m$ , afin d'obtenir une clé secrète chiffrée  $K'$ . Cette clé chiffrée  $K'$  est également chiffrée avec RSA en utilisant la clé publique  $pub_s$  de son correspondant  $S$  afin d'obtenir  $K''$ . Cette clé doublement chiffrée  $K''$  est alors insérée dans l'image cryptée par tatouage dans le domaine fréquentiel par DCT [6]. L'algorithme de tatouage modifie la composante DC de chaque bloc DCT de l'image afin de résister lui-même au bruit durant la transmission.

Finalement  $M$  envoie cette image à  $S$  comme présenté Figure 3.  $S$  reçoit alors l'image, extrait de l'image la clé secrète cryptée  $K''$ . Il peut à ce moment décrypter la clé  $K''$  par utilisation de sa clé privée  $priv_s$ , puis authentifier  $M$  en utilisant la clé publique  $pub_m$  de  $M$ . Avec la clé obtenue  $K$ ,  $S$  peut alors déchiffrer l'image et la visualiser.

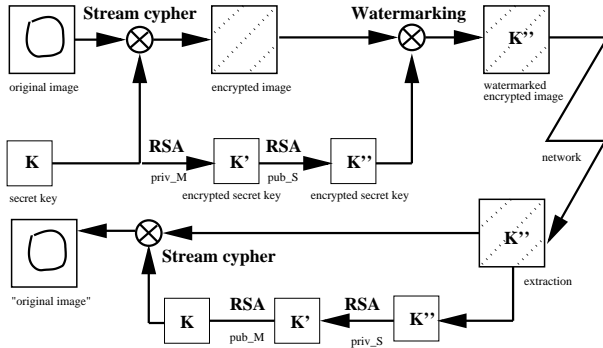


Figure 3 – Combinaison de cryptage à clé secrète, cryptage à clés publique-privée et tatouage d'image.

Si  $S$  souhaite envoyer une nouvelle image à  $M$ , il utilisera une nouvelle clé secrète  $K_1$  pour cette nouvelle session. La méthode sera ensuite identique, mais les clés publiques et privées ne seront pas utilisées dans le même ordre. Même si cinq clés sont nécessaires par session, la plupart d'entre elles sont transparentes pour les utilisateurs. En fait, les clés privées peuvent être associées au logiciel, et pour les deux correspondants, il n'est pas utile de connaître la clé secrète qui est insérée dans l'image. Cependant, pour chaque session, la valeur de la clé secrète  $K$  doit changer.

En effet, si la clé secrète n'était pas changée, toutes les personnes ayant le logiciel pourraient décrypter toutes les images.

## 4 Résultats

### 4.1 Chiffrement d'images par flux

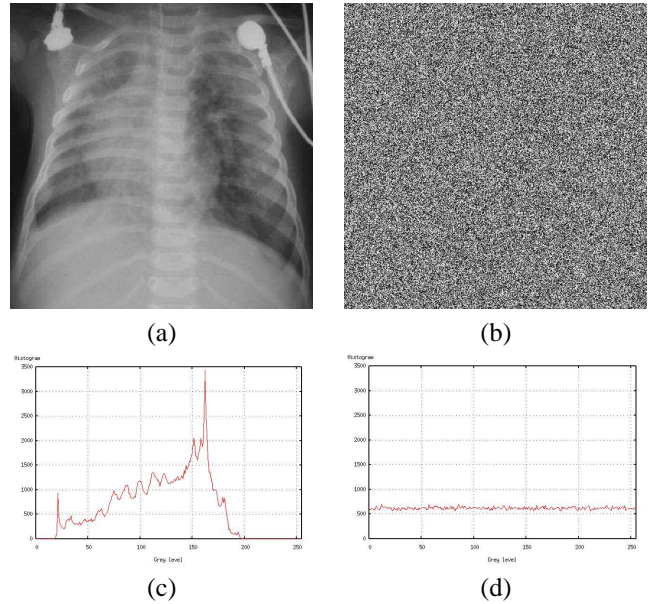


Figure 4 – a) Image originale, b) Image cryptée avec l'algorithme de chiffrement par flux, avec une clé de 128 bits, c) Histogramme de l'image originale, d) Histogramme de l'image cryptée.

A partir de l'image, Figure 4.a, nous avons appliqué l'algorithme de chiffrement par flux avec une clé de 128 bits. Figure 4.b, nous remarquons que l'information initiale n'est plus du tout visible. En comparant l'histogramme de l'image initiale, Figure 4.c, avec l'histogramme de l'image cryptée, Figure 4.d, nous remarquons que les probabilités d'apparition de chaque niveau de gris sont uniformément distribués. Par conséquent, l'entropie de l'image cryptée est très haute (proche de 8 bits/pixel).

### 4.2 Crypto-tatouage d'images

A partir d'une image originale, Figure 5.a, nous appliquons la méthode de chiffrement par flux à clé secrète pour obtenir une image cryptée, Figure 5.b. Si nous déchiffrons cette image, il n'y a aucune différence. La longueur de la clé est de 128 bits. Nous insérons ensuite la clé cryptée par RSA dans l'image cryptée, Figure 5.c. La différence entre l'image chiffrée et celle qui est tatouée est présentée Figure 5.d. Nous voyons les blocs de pixels où nous avons inséré notre message, le  $PSNR = 42.12 \text{ dB}$ . Après décryptage de l'image crypto-tatouée nous obtenons l'image de la Figure 5.e. La différence entre l'image originale et l'image décryptée est présentée Figure 5.f. Nous voyons,

Figure 5.f, que les différences entre les deux images se sont répandues dans l'image. Nous remarquons également que le bruit dû au tatouage dans le domaine crypté n'est pas augmenté pendant la phase de décryptage, avec un  $PSNR = 43.71 \text{ dB}$ . Au contraire, le bruit est atténué pendant cette phase de décryptage du fait de l'équation (2).

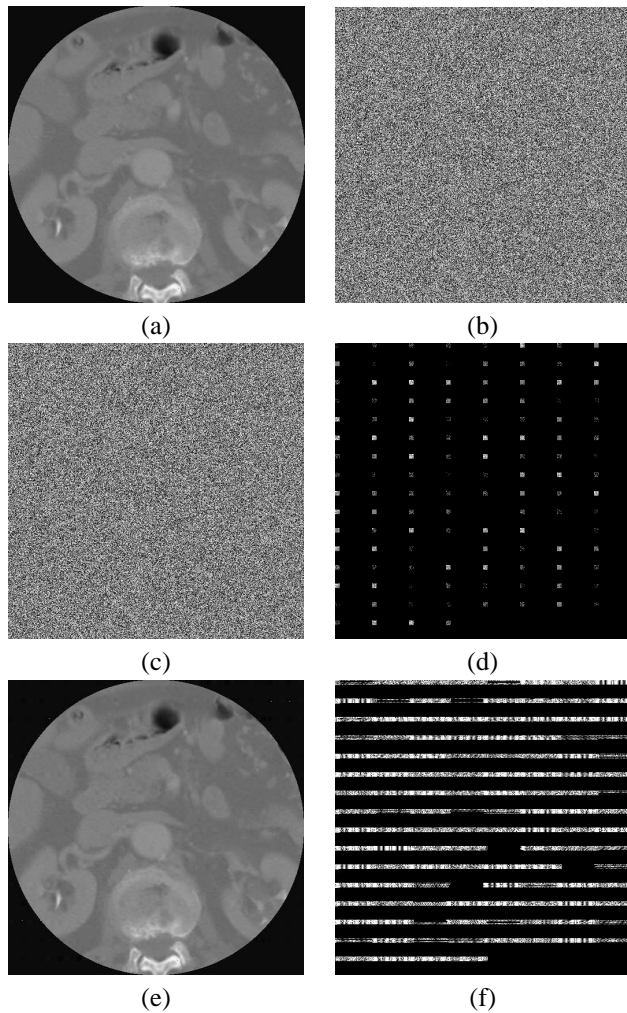


Figure 5 – a) Image originale, b) Image chiffrée, c) Image crypto-tatouée avec une clef de 128 bits, d) Différence entre l'image chiffrée et l'image crypto-tatouée, e) Décryptage de l'image crypto-tatouée, f) Différence entre l'image originale et l'image décryptée.

## 5 Conclusion

Dans cet article, nous avons présenté une nouvelle méthode de crypto-tatouage d'image. Cette méthode combinant cryptages symétrique et asymétrique et tatouage permet de sécuriser le transfert. Nous avons choisi de chiffrer l'image avec un algorithme symétrique, et de chiffrer la clef secrète de cet algorithme avec un algorithme asymétrique.

La méthode de chiffrement par flux d'image développée

est résistante à différents bruits pouvant intervenir après le cryptage. Pour insérer la clef secrète cryptée, nous avons utilisé une méthode de tatouage basée sur la DCT. Nous avons choisi d'insérer le message dans le domaine fréquentiel afin de résister au bruit pouvant intervenir durant le transfert de l'image cryptée.

Nous avons appliqué l'algorithme de chiffrement par flux sur une image en montrant que l'information initiale n'était plus visible, puis nous avons appliqué la méthode de combinaison complète sur une autre image médicale.

## Références

- [1] F. Li, J. Knipe, et H. Cheng. Image compression and encryption using tree structures. *Pattern Recognition Letters*, 18 :1253–1259, 1997.
- [2] K.L. Chung et L.C. Chang. Large encrypting binary images with higher security. *Pattern Recognition Letters*, 19 :461–468, 1998.
- [3] C.C. Chang, M.S. Hwang, et T-S Chen. A new encryption algorithm for image cryptosystems. *The Journal of Systems and Software*, 58 :83–91, 2001.
- [4] A. Sinha et K. Singh. A technique for image encryption using digital signature. *Optics Communications*, 218 :229–234, 2003.
- [5] F. Y. Shih et S. Y.T. Wu. Combinational image watermarking in the spatial and frequency domains. *Pattern Recognition*, 36 :969–975, 2003.
- [6] G. Lo-varco, W. Puech, et M. Dumas. Dct-based watermarking method using error correction codes. Dans *ICAPR'03, International Conference on Advances in Pattern Recognition, Calcutta, India*, pages 347–350, 2003.
- [7] J. Bernarding, A. Thiel, et A. Grzesik. A JAVA-based DICOM server with integration of clinical findings and DICOM-conform data encryption. *International Journal of Medical Informatics*, 64 :429–438, 2001.
- [8] R. Norcen, M. Podesser, A. Pommer, H.P. Schmidt, et A. Uhl. Confidential storage and transmission of medical image data. *Computers in Biology and Medicine*, 33 :277–292, 2003.
- [9] S. Guillem-Lessard. <http://www.uqtr.ca/~delisle/Crypto>. 2002.
- [10] G.S. Vernam. Cipher printing telegraph systems for secret wire and radio telegraphic communications. In *Journal of the American Institute of Electrical Engineers*, 45 :109–115, 1926.
- [11] RSA Security. <http://www.rsasecurity.com/>. 2003.
- [12] W. Puech, J.J. Charre, et M. Dumas. Transfert sécurisé d'images par chiffrement de Vigenère. Dans *NimesTic 2001, La relation Homme - Système : Complexe, Nîmes, France*, pages 167–171, Dec. 2001.
- [13] B. Schneier. *Applied cryptography*. Wiley, 1995.