Protection en temps réel des visages dans une séquence d'images par cryptage partiel et sélectif

J.M. Rodrigues[1] W. Puech[1,2]

[1] Laboratoire LIRMM, UMR CNRS 5506,
Université Montpellier II - France

[2] Centre Universitaire de Formation et de Recherche de Nîmes - France

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

Résumé

Dans cet article nous proposons une nouvelle approche de protection de visages pour des environnements surveillés par caméra vidéo. L'objectif est de chiffrer rapidement les visages contenus dans les images d'une séquence. Nous utilisons un algorithme classique pour la détection des visages et nous proposons une nouvelle méthode pour le cryptage partiel et sélectif. Cette approche originale est basée sur un codage à longueur variable et sur le cryptage d'une partie des codes de Huffman du codeur JPEG en utilisant l'algorithme AES en mode de chiffrement par flot. La méthode proposée permet également le déchiffrage partiel d'une région d'intérêt (RI) chiffrée. Cette méthode permet d'obtenir une réduction significative du temps de codage et de décodage en comparaison avec un chiffrement complet des données après compression. Elle fournit également un débit binaire constant en restant conforme aux normes du format du codeur JPEG.

Mots clefs

Cryptage sélectif et partiel, détection de visages, compression d'images, protection.

1 Introduction

La sécurisation des transferts de contenus multimédias peut se faire soit par cryptage total soit par cryptage sélectif ou partiel. Les applications militaires et relatives à la loi exigent un cryptage total. Néanmoins, pour de nombreuses applications, un cryptage sélectif ou partiel est suffisant. Ces approches réduisent le temps de calcul ainsi que les puissances informatiques dans un réseau hétérogène avec des dispositifs de différentes capacités [1]. Le cryptage partiel (CP) d'une image a pour but de ne crypter qu'une partie de l'information spatiale de l'image en s'appliquant uniquement sur des régions d'intérêt alors que le cryptage sélectif (CS) d'une image ne crypte qu'une partie précise des informations de toute l'image (les hautes fréquences par exemple).

Un CS peut être utilisé par exemple pour des images acquises par une caméra de surveillance. Pour des visualisations en temps réel, ces images doivent être rapidement transmises et le cryptage total n'est pas nécessaire. La sécurité d'un CS ou d'un CP est toujours inférieure à celle d'un chiffrage complet, mais le CS ou le CP diminue la quantité de données à chiffrer, et par conséquent le temps de calcul. De plus, un CS ou un CP peut-être intégré à l'intérieur même d'un codeur (JPEG, JPEG2000, H264/AVC) et donc le flux en sortie reste conforme aux normes du codeur. De ce fait, un décodeur classique aura accès à l'information basse résolution. Cependant, un décodeur adapté, muni d'une clef secrète pourra décoder correctement l'information cryptée.

Dans cet article nous proposons une nouvelle approche de cryptage partiel et sélectif (CPS) du codage de Huffman pour des séquences d'images comprimées avec JPEG. Dans notre approche nous utilisons l'algorithme AES (*Advanced Encryption Standard*) [2] en mode de chiffrement par flot OFB (*Output Feedback Block*).

Dans la section 2, nous passons en revue les travaux précédents dans le domaine. Dans la section 3, nous présentons la méthode proposée, l'algorithme de cryptage sélectif, ainsi que le processus de détection des visages. Dans notre expérimentation la caméra est fixe. Du fait d'un non déplacement de la caméra, la détection des visages dans la séquence d'images est fortement facilitée. Enfin, section 4, nous montrons les résultats expérimentaux sur une séquence d'images protégeant le visage de deux personnes.

2 Travaux précédents

De nombreuses méthodes de CS et CP ont été proposées pour des images comprimées par des algorithmes basés sur la transformée en cosinus discrète (DCT). Droogenbroeck et Benedett [3] ont proposé une méthode pour chiffrer une quantité limitée de coefficients AC. Dans leur méthode les coefficients DC ne sont pas chiffrés parce qu'ils sont fortement prévisibles et qu'ils portent une information évidente. Dans cette approche les étapes de compression et de chiffrage sont faites séparément et ceci conduit à un doublement du temps de calcul par rapport à une compression

simple. Fisch et al. [4] ont proposé une méthode de cryptage sélectif d'images où les données sont organisées sous une forme de flot de bits graduable. Ces flots binaires sont construits avec des coefficients DC et quelques coefficients AC de chaque bloc de l'image et puis arrangés dans des couches selon leur importance visuelle.
Tang [5] a proposé une technique appelée *permutation zigzag* applicable aux images et aux vidéos basées DCT. Bien que cette méthode offre plus de confidentialité, elle augmente le nombre de bits total. D'autres travaux proposent également des algorithmes de CS ou CP pour des vidéos basées DCT [6, 7, 8]. Par rapport aux méthodes existantes, le fait d'utiliser l'algorithme AES en mode de chiffrement par flot (au lieu d'une approche de chiffrement par bloc) et de l'avoir d'intégré au niveau du codage de Huffman, notre méthode méthode permet de conserver le taux de compression initial du codeur. Récemment, Said [9] a mesuré la robustesse des méthodes de CS. Il a montré que des attaques qui exploitent les informations des bits en clair (non cryptés) permettent un décryptage plus rapide de l'image.

2.1 Compression d'images basée DCT

Dans les images comprimés par DCT, le codage de Huffman est fait sur les coefficients quantifiés des blocs de 8×8 pixels, et sont codés par le couple $\{$(HEAD), (AMPLITUDE)$\}$. L'entête HEAD contient les contrôleurs obtenus par les tables de Huffman pour la compression et la décompression. Le paramètre AMPLITUDE est un entier signé correspondant à l'amplitude d'un coefficient AC non nul, ou dans le cas du coefficient DC de la différence entre deux coefficients voisins DC. La structure HEAD varie en fonction du type de coefficient. Pour les ACs, cette structure est composée de (RUNLENGTH, SIZE), alors que pour les DCs elle est composée seulement de la taille SIZE. Pour le codage du coefficient AC, les informations conjointes du RUNLENGTH et AMPLITUDE sont utilisées et appliqués dans les tables standards. La valeur RUNLENGTH correspond au nombre de coefficients AC égaux à zéro précédant une valeur non nulle dans la séquence en zigzag. La taille SIZE est la quantité nécessaire de bits pour représenter la valeur de l'amplitude. Il y a deux codes particuliers correspond à (RUNLENGTH, SIZE) égale à (0, 0) et (15, 0). Ils sont utilisés pour symboliser la fin d'un bloc (EOB) et la longueur d'une plage de zéros. Le symbole EOB est transmis après le dernier coefficient non nul du bloc quantifié. C'est ainsi le chemin le plus efficace pour coder la fin d'une plage de zéros. Le symbole EOB est omis dans le cas où l'élément final du vecteur est non nul. Le symbole ZRL est transmis quand la valeur du RUNLENGTH est plus grande que 15 et représente une longueur de plage de 16 zéros.

2.2 Le chiffrement par AES

L'algorithme AES (*Advanced Encryption Standard*) est le standard pour le cryptage à clef secrète. L'algorithme AES est composé d'un ensemble d'étapes répétées plusieurs fois, appelé ronde. La figure 1.a représente le schéma de chiffrement d'un texte clair X_i. L'algorithme AES peut supporter les modes de chiffrement suivants : CBC *Cipher Block Chaining*, ECB *Electronic Code Book*, CFB *Cipher Feedback*, OFB *Output Feedback* et CTR *Counter Mode*.
Même si l'AES est un algorithme de chiffrement par bloc, les modes OFB, CFB et CTR opèrent comme des chiffrements par flot. Chaque mode a différents avantages et inconvénients. Dans les modes ECB, OFB et CTR par exemple, tout changement dans le bloc du texte clair X_i provoque modification dans le bloc chiffré Y_i, mais les autres blocs chiffrés ne sont pas affectés. Avec les modes CBC ou CFB, si un texte clair du bloc X_i est changé alors le bloc crypté Y_i et tous les blocs chiffrés suivants seront affectés.

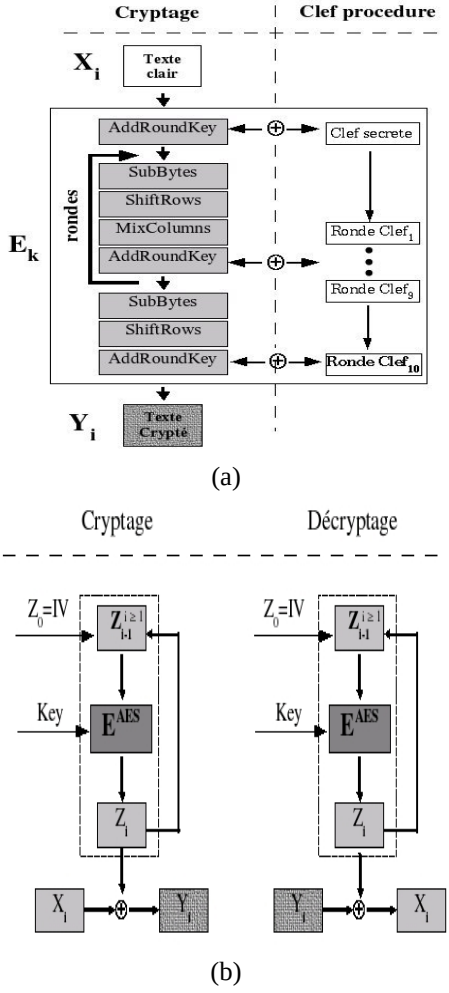

Figure 1 – *a) Schéma général de l'algorithme AES, b) Cryptage et décryptage en mode OFB.*

Les propriétés des modes CBC et CFB permettent de régler des problèmes d'authentification alors que celles des modes ECB, OFB et CTR permettent de traiter séparément chaque bloc. Enfin, contrairement aux algorithmes de chiffrement par bloc, les algorithmes de chiffrement par flot permettent de faire varier de manière graduable la quantité

de bits à crypter.
Afin de pouvoir traiter séparément les régions de l'image de manière graduable, notre algorithme est donc basé sur le mode OFB, qui est un mode de chiffrement par flot synchrone de l'algorithme AES. La figure 1.b montre le chiffrement en mode OFB où le bloc en clair X_i est chiffré avec la clef secrète k afin de produire le bloc chiffré Y_i pour tout $i \geq 1$:

$$\begin{cases} Z_i &= E_k(Z_{i-1}) \\ Y_i &= X_i \oplus Z_i \end{cases}, \quad (1)$$

où $\oplus$ est le ou exclusif.
Dans la figure 1.b présentant le mode OFB, il est important de noter que la fonction de cryptage $E_k()$ est utilisé pour la phase de cryptage mais également pour la phase de décryptage.

3 Méthode proposée

Soit $E_k(X)$ le cryptage d'un bloc X de n bits en utilisant la clef secrète k avec l'algorithme AES en mode OFB. Dans la description de la méthode, nous supposerons $n = 128$. Soit $D_k(Y)$ le décryptage d'un texte chiffré Y en utilisant la clef secrète k.

3.1 Cryptage sélectif d'une séquence d'images

Le cryptage de la méthode proposée est appliqué conjointement au processus de codage entropique durant la création du vecteur de Huffman du codeur JPEG. La méthode proposée est appliquée au niveau des blocs 8×8 pixels au moment du codage de Huffman des coefficients AC. Les trois étapes sont les suivantes :
– Construction du texte clair X_i à partir des coefficients AC non nuls du flux binaire de Huffman, des plus hautes fréquences vers les basses fréquences,
– Codage de X_i avec l'algorithme AES en mode OFB pour obtenir Y_i,
– Substitution du flux binaire de Huffman par l'information cryptée qui est de même taille.
Il est important de mentionner que ces opérations sont appliquées séparément à chaque bloc quantifié.

Construction du texte clair X. Pour construire le texte clair X_i, nous prenons les coefficients AC non nuls du bloc courant i en accédant au vecteur de Huffman de la fin vers le début afin de créer des paires {HEAD, AMPLITUDE}. De chaque entête HEAD nous obtenons la longueur de l'AMPLITUDE en bit. Ces valeurs sont calculées à partir de l'équation (2). Comme montré dans la vue générale de la méthode proposée (figure 2), seules les AMPLITUDE $(A_n, A_{n-1}...A_1)$ sont prises en compte pour construire le vecteur X_i. La longueur finale du message en clair L_{X_i} dépend à la fois de l'homogénéité ρ du bloc et de la contrainte donnée C :

$$f(\rho) < L_{X_i} \leq C, \quad (2)$$

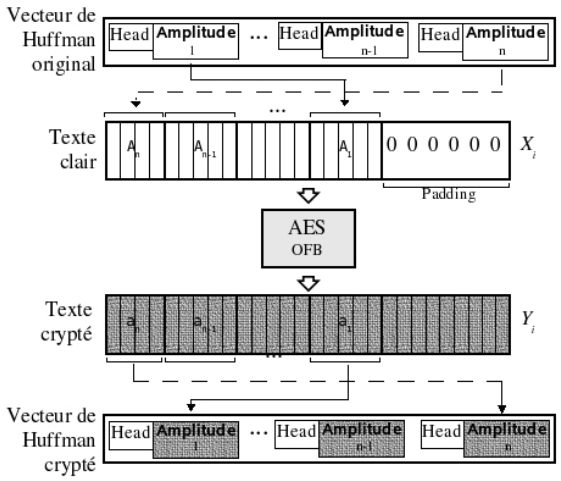

Figure 2 – *Présentation générale de la méthode proposée.*

où $f(\rho) = 0$ pour $\rho \to \infty$ et $C \in \{128, 64, 32, 16, 8\}$ bits. Cette contrainte C spécifie la quantité maximale de bits qui doit être prise en compte dans chaque bloc. La valeur de C doit donc être choisie en fonction de l'homogénéité ρ de l'image mais aussi du pourcentage de bits de l'image que l'on souhaite chiffrer. C'est donc par l'intermédiaire de C que nous graduons l'importance du cryptage. Comme l'homogénéité dépend du contenu de l'image, dans l'équation (2) celle-ci spécifie la quantité minimale de bits. Cela signifie qu'un bloc avec un grand ρ va produire un petit L_{X_i}. Le vecteur de Huffman est donc traité tant que $L_{X_i} \leq C$ et que le coefficient DC n'est pas atteint. Ensuite, nous appliquons la fonction de remplissage (padding) $p(j) = 0$, où $j \in \{L_{X_i} + 1, \ldots, 128\}$, afin de remplir si nécessaire avec des zéros le vecteur X_i.

Chiffrement de X avec AES en mode OFB. Dans l'étape de chiffrement, la clef dynamique Z_{i-1} est utilisée comme entrée pour le cryptage par AES afin d'obtenir une nouvelle clef dynamique Z_i. Pour la première itération, le vecteur d'initialisation IV ($Initialization\ Vector$) est créé à partir de la clef secrète k avec la stratégie suivante : la clef secrète k est utilisée comme une semence pour un générateur de nombres pseudo-aléatoire (GNPA). Ce k est divisé en 16 portions de 8 bits chacun. Le GNPA produit 16 nombres aléatoires qui définissent l'ordre de formation du IV. Ensuite chaque Z_i est additionnée par un ou exclusif avec le texte en clair X_i pour générer le bloc chiffré Y_i.

Substitution du flux binaire de Huffman. L'étape finale est la substitution de l'information initiale par l'information chiffrée dans le vecteur de Huffman. Comme dans la première étape (construction du texte clair X_i), le vecteur de Huffman est lu depuis la fin vers le début mais le vecteur chiffré Y_i est lu du début vers la fin. Connaissant la longueur en bits de chaque AMPLITUDE $(A_n, A_{n-1}...A_1)$, nous commençons par couper ces portions dans Y_i pour remplacer l'AMPLITUDE dans le vec-

teur de Huffman. La quantité totale de bits doit être L_{X_i}. Cette procédure est faite pour chaque bloc. Les blocs homogènes ne sont pas ou peu chiffrés. L'utilisation du mode OFB pour le chiffrement permet une génération de clef dynamique Z_i indépendante. Il est important de noter que l'algorithme de CS utilisé au niveau le vecteur de Huffman n'augmente pas la taille finale du vecteur binaire comprimé. Ceci vient du fait de l'utilisation d'un mode de chiffrement par flot.

3.2 Procédure de décryptage

La procédure de décryptage en mode OFB fonctionne de la manière suivante. Comme pour la phase de cryptage, la clef dynamique Z_{i-1} est utilisée en entrée du cryptage par AES afin d'obtenir une nouvelle clef dynamique Z_i. Dans la phase de décryptage, la différence est que la clef dynamique Z_i est additionnée par un ou exclusif avec le bloc chiffré Y_i afin de régénérer le texte en clair X_i comme illustré figure 1.b. Le vecteur résultat du texte en clair X_i est coupé en parties de la fin vers le début afin de remplacer les AMPLITUDE dans le chiffré de Huffman pour générer le vecteur de Huffman.

3.3 Détection des visages

La méthode de CS présentée section 3.1 est appliquée sur une séquence d'images afin de masquer les visage des personnes. Dans cette section, nous présentons l'étape de détection des visages. Les visages détectés constitueront des régions d'intérêts (RIs) dans lesquelles le CS sera appliqué. Nous obtenons alors un CP (uniquement les RIs) combiné avec un CS (les plus hautes fréquences).
La première étape de l'algorithme consiste en une transformation couleur de l'espace RGB à l'espace couleur YUV. Comme dans le codeur JPEG, nous calculons ensuite la transformée en cosinus discrète pour chaque composante (Y, U et V) par bloc afin de générer les coefficients DC et AC. Nous employons les coefficients DC des composantes U et V pour produire deux imagettes qui sont utilisées pour détecter la couleur peau à partir de l'équation (3). A partir d' une image de $M \times N$ pixels, nous obtenons alors deux imagettes DC_U et DC_V. Un bloc est considéré comme un bloc contenant une partie de visage si la couleur peau est détectée :

$$\sqrt{(DC_U/8 - U_p)^2 + (DC_V/8 - V_p)^2} < S, \quad (3)$$

où U_p et V_p sont les couleurs de peau de référence fournis dans l'espace YUV, S le seuil de détection. Pour un facteur de qualité $FQ = 100\%$ nous avons $DC_U/8$ et $DC_V/8$ qui correspondent aux valeurs moyennes des blocs correspondants.
Le résultat de l'étape de détection est généralement une image binaire bruitée. Afin de filtrer cette image binaire, nous appliquons une fermeture suivie d'une ouverture morphologique [10]. Dans notre approche la caméra est fixe, ceci facilite la détection des RIs dans la séquence d'images. L'image binaire filtrée indique les blocs constituant les RIs qui doivent être chiffrées dans la séquence. Chaque pixel blanc dans l'image binaire correspond à un bloc dans l'image originale. La méthode de CS décrite dans la section 3.1 est alors appliquée sur le vecteur de Huffman de la composante Y.

4 Résultats expérimentaux

Pour nos expériences, nous avons sélectionné quatre images ($\#083, \#123, \#135$ et $\#147$) (figures 3.a) d'une séquence de 156 images de taille 640×480 pixels. Pour la détection des visages nous avons utilisé comme couleurs de référence de la peau $U_p = 120$ et $V_p = 140$ et le seuil $S = 5$. Pour le chiffrement, nous avons employé l'algorithme AES en mode de chiffrement par flot OFB avec une clef de longueur 128 bits. Cependant, notre méthode peut être employée avec d'autres valeurs de longueur pour la clef et pour les blocs. Pour l'utilisation de l'équation (2), nous avons fixé C à 128 bits.
Pour chacune des quatre images sélectionnées présentées figures 3.a, nous sommes passés de l'espace couleur RGB à l'espace YUV. Dans l'espace YUV nous obtenons les imagettes constituées des composantes DC, illustrées figures 4 pour l'image #083. À partir des coefficients DC des composantes U et V nous avons appliqué l'équation (3) pour générer les masques binaires. Nous avons ensuite utilisé l'algorithme d'érosion et dilatation sur les imagettes binaires bruitées afin d'obtenir les figures 3.b. Les pixels blancs représentent des blocs considérés comme représentant des visages. En effet chaque pixel blanc de l'image binaire est un bloc de peaux dans l'image originale. Sur ces blocs nous avons alors appliqué l'algorithme de CS décrit précédemment pour produire des images partiellement et sélectivement chiffrées présentées figures 5.

	Total chiffré			
Image	#. Blocs	Coeffi cients	Bits	% Bloc
#083	79	2547	10112	1.65
#123	113	3042	14464	2.35
#135	159	4478	20352	3.31
#147	196	5396	25088	4.08

Tableau 1 – *Résultats du CS employé dans la séquence d'images.*

Le tableau 1 indiquent les résultats obtenus pour chaque image. Pour l'image #083 nous avons détecté 79 blocs de peaux. Dans cette image 2547 coefficients AC (10112 bits) ont été chiffrés. Le nombre de blocs chiffrés correspond à 1.6 % du nombre total de blocs de l'image originale. Pour l'image #123 nous avons crypté 113 blocs, du fait que les visages sont plus grands que dans l'image #083. Nous avons alors chiffré 3042 coefficients AC, ce qui représente 14464 bits et 2.35 % des blocs chiffrés. Dans la séquence d'images utilisée, la quantité de blocs à chif-

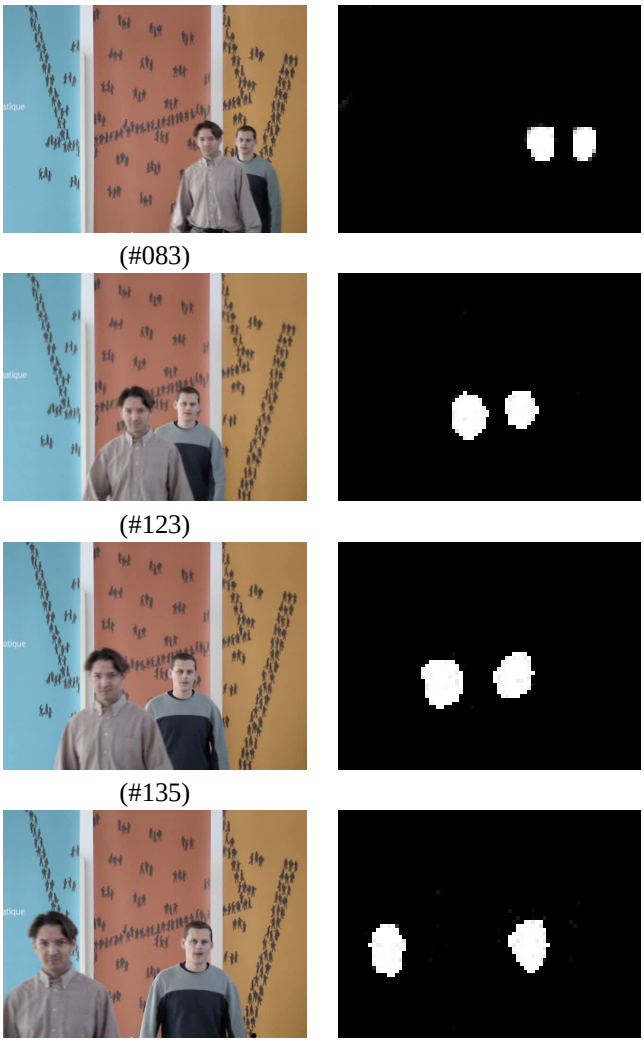

Figure 3 – *a) Séquence des images originales, b) Séquence des images binaires.*

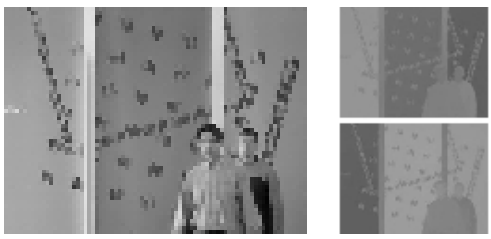

Figure 4 – *Imagettes des coefficients DC de l'image #083 pour les plans YUV.*

Figure 5 – *Séquence d'images cryptée partiellement et sélectivement.*

frer augmente du fait que les deux personnes s'approchent de la caméra. Cependant, comme le montre le tableau 1, la quantité de bits chiffrés est très faible par rapport la taille de l'image. Ceci fait que notre méthode est applicable dans des environnements de faible puissance de calcul comme, par exemple, des vidéos issues de caméras portables. La figure 5 montre les résultats de détection des visages et du cryptage partiel et sélectif dans cette séquence d'images. Afin de montrer plus clairement nos résultats, nous avons agrandit, figures 6, une zone de 216×152 pixels de l'image #123.

Il convient de noter que la sécurité d'un cryptosystème est liée à la capacité de deviner les valeurs des données chiffrées. Par exemple, il est préférable de chiffrer les bits qui sont les plus aléatoires possible. Cependant, la sécurité d'un CS ou d'un CP est toujours plus faible que celle d'un cryptage complet. La raison la plus importante d'accepter ce schéma est la réduction importante du temps de calcul par rapport à un cryptage total. Cependant, en pratique, une attaque est plus difficile sur les coefficients ACs non nuls d'une séquence d'images que sur ses coefficients DC qui sont fortement prévisibles [3, 11].

Figure 6 – *Région de* 216×152 *pixels de l'image #123.*

5 Conclusion

Dans cet article, nous avons proposé un nouveau schéma de cryptage partiel et sélectif pour des séquences d'images codées avec JPEG en utilisant le cryptage AES en mode par flot OFB. Les avantages de notre méthode sont la portabilité, un taux de compression conservé par rapport à une compression standard, une compatibilité avec le codeur JPEG, un cryptage sélectif réglable en quantité et un décryptage partiel par région d'intérêt. En perspectives nous envisageons d'intégrer notre approche dans des séquences vidéos H264/AVC.

Références

[1] X. Liu et A. Eskicioglu. Selective Encryption of Multimedia Content in Distribution Networks :Challenges and New Directions. Dans *IASTED Communications, Internet & Information Technology (CIIT), USA*, November, 2003.

[2] J. Daemen et V. Rijmen. AES Proposal : The Rijndael Block Cipher. Rapport technique, Proton World Int.l, Katholieke Universiteit Leuven, ESAT-COSIC, Belgium, 2002.

[3] M. Van Droogenbroeck et R. Benedett. Techniques for a Selective Encryption of Uncompressed and Compressed Images. Dans *Proceedings of Advanced Concepts for Intelligent Vision Systems (ACIVS) 2002, Ghent, Belgium*, Sept. 2002.

[4] M. M. Fisch, H. Stgner, et A. Uhl. Layered Encryption Techniques for DCT-Coded Visual Data. Dans *European Signal Processing Conference (EUSIPCO) 2004, Vienna, Austria*, Sep., 2004.

[5] L. Tang. Methods for Encrypting and Decrypting MPEG Video Data Efficiently. Dans *ACM Multimedia*, pages 219–229, 1996.

[6] W. Zeng et S. Lei. Efficient Frequency Domain Video Scrambling for Content Access Control. Dans *ACM Multimedia, Orlando, FL, USA*, pages 285–293, Nov. 1999.

[7] H. Cheng et X. Li. Partial Encryption of Compressed Images and Videos. *IEEE Transactions on Signal Processing*, 48(8) :2439–2451, 2000.

[8] J. Wen, M. Severa, W. Zeng, M. Luttrell, et W. Jin. A Format-Compliant Configurable Encryption Framework for Access Control of Video. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(6) :545–557, 2002.

[9] A. Said. Measuring the Strength of Partial Encryption Scheme. Dans *ICIP 2005, IEEE International Conference in Image Processing, Genova, Italy*, volume 2, pages 1126–1129, 2005.

[10] J. Serra. *Image Analysis and Mathematical Morphology, vol. 2*, volume 2. London : Academic Press, 1988.

[11] T. Lookabaugh. Selective encryption, information theory, and compression. Dans *38th ASILOMAR Conference on Signals, Systems and Computers*, volume 1, pages 373–376, 2004.