Recherche Tabou pour le Développement d'une Nouvelle Approche de Cryptage d'Images

I. Souici, H. Seridi
LabSTIC, Université de Guelma
Guelma, Algérie
souici.ismahane@yahoo.fr
seridihamid@yahoo.fr

Abstract— ***L'article présente une nouvelle approche basée sur une métaheuristique de recherche taboue pour la résolution du problème de cryptage d'images. L'idée est de considérer les nombres d'occurrences des valeurs de pixels, informations non utiles pour les cryptanalystes, pour maximiser la différence entre l'image originale et sa version chiffrée correspondante. L'algorithme développé et appelé Tabu-Crypt a été testé sur des images de différentes natures et tailles où il a montré un bon niveau confusionnel grâce à l'aspect non déterministe innové à travers l'algorithme proposé et un temps de calcul raisonnable. Une étude de la résistibilité de Tabu-Crypt contre les attaques les plus avancées et une comparaison entre cet algorithme et les plus connus des standards cryptographiques conclura ce travail.***

Mots clés- Métaheuristiques ; recherche tabou ; optimisation ; sécurité ; cryptage ; confusion, attaques avancées.

I. INTRODUCTION

Dès que les hommes ont appris à communiquer, ils ont trouvé des moyens d'assurer la confidentialité d'une partie de leurs communications : l'origine de la cryptographie remonte sans doute aux origines de l'homme. En effet, le mot cryptographie est un terme générique désignant l'ensemble des techniques permettant de chiffrer des messages c'est-à-dire de les rendre inintelligibles sans une action spécifique [1].

Mais ce n'est qu'à l'avènement de l'informatique et d'Internet que la cryptographie prend tout son sens. Les efforts conjoints d'IBM et de la NSA conduisent à l'élaboration du DES (Data Encryption Standard) [2], l'algorithme de chiffrement le plus utilisé au monde durant le dernier quart du XXe siècle. À l'ère d'Internet, le nombre d'applications civiles de chiffrement (banques, télécommunications, cartes bleues…) explose [3]. Le besoin d'apporter une sécurité accrue dans les transactions électroniques fait naître les notions de signature et authentification électroniques [4, 5]. La première technique de chiffrement à clef publique sûre (intimement liée à ces notions) apparaît : le RSA [6].

Donc, ce sont les conséquences liées à la survenance de ces risques qui introduisent le besoin de protection de l'information. C'est d'ailleurs l'objet de notre présent travail à travers lequel nous cherchons à ramener et à modéliser le problème de cryptage comme un problème d'optimisation.

Dans ce travail, nous nous intéressons à l'exploitation d'une métaheuristique à trajectoire pour la résolution du problème de cryptage après les bons résultats obtenus suite à l'application d'une métaheuristique à population qui est celle des algorithmes évolutionnaires [7, 8, 9]. Il s'agit de la recherche tabou. Cette dernière s'appuie sur une recherche locale combinée à un mécanisme de prévention des cycles, grâce à un système de mémoire des mouvements précédemment appliqués ou des configurations visitées (la liste tabou) [10].

De manière générale, une recherche locale démarre d'une solution initiale possible et essaie de l'améliorer, en cherchant une solution meilleure dans le voisinage courant [11]. Un voisinage d'une certaine solution correspond à des éléments adjacents à cette solution dont chacun est atteint par un changement dans la configuration courante. Le processus de recherche est réitéré jusqu'à ce qu'aucune amélioration dans la solution courante ne puisse être faite.

Nous présentons, après la description de l'approche proposée, les résultats numériques obtenus afin de les comparer aux résultats de certaines des méthodes de la littérature.

II. MOTIVATION

La recherche Tabou est une métaheuristique basée sur des idées simples, mais reste néanmoins efficace. Cette méthode combine une procédure de recherche locale avec un certain nombre de règles et de mécanismes lui permettant de surmonter l'obstacle des extremums locaux, tout en évitant les problèmes de cycles.

L'originalité de la méthode de recherche tabou, par rapport aux autres méthodes locales, réside dans le fait que l'on retient le meilleur voisin, même si celui-ci est plus mauvais que la solution dont il est le voisin direct. Pour cela, en autorisant les dégradations de la fonction objective f l'algorithme évite, au mieux, d'être piégé dans un minimum local, mais il induit un risque de répétitions cycliques. En effet, lorsque l'algorithme a quitté un minimum quelconque par acceptation de la dégradation de la fonction objective, il peut revenir sur ses pas aux itérations suivantes.

Pour pallier à ce problème, l'algorithme utilise une mémoire pour conserver pendant un moment la trace des dernières meilleures solutions déjà inspectées. Ces solutions sont déclarées *taboues*, d'où le nom de la méthode. Elles sont stockées dans une liste d'une certaine longueur, appelée *liste Tabou*. Une nouvelle solution n'est acceptée que si elle n'appartient pas à cette liste Tabou. Ce critère d'acceptation d'une nouvelle solution évite le rebouclage de l'algorithme, durant la visite d'un nombre de solutions au moins égal à la

longueur de la liste Tabou, et il dirige l'exploration de la méthode vers des régions du domaine de solutions non encore visitées.

III. Tabu-Crypt : ALGORITHME PROPOSE

À partir d'une solution initiale, le principe général de la recherche tabou est celui résumé à travers l'algorithme 1.

Le schéma général de l'algorithme finalement implémentant les étapes de l'approche de cryptage tabou proposé est celui donné par l'algorithme 2.

Présentons maintenant, plus ou moins en détails, les différentes étapes de l'algorithme développé.

A. Création de la solution initiale

Une image soumise au chiffrement par *Tabu-Crypt* sera codée en calculant le nombre d'occurrences des 256 valeurs possibles des composantes R, V et B codant les pixels dans cette image. Ainsi, le codage proposé sera celui résumé par la figure 1.

Algorithme 1 *Schéma général d'un algorithme tabou*

Engendrer une configuration initiale s
$s^* \leftarrow s$
$T \leftarrow \emptyset$ liste tabou
Tant que *Condition d'arret non satisfaite* **faire**
  $m \leftarrow$ meilleur mouvement parmi ceux non tabou
  Ou ceux vérifiant un critère d'aspiration ;
  Modifier s en effectuant le mouvement m ;
  Mettre T à jour ;
  Si $f(s) < f(s^*)$ **alors**
    $s^* \leftarrow s$;
  Fin
Fin
Retourner s^*

Algorithme 2 *Tabu-Crypt*

Étape 1 : Générer une solution initiale de gain total G_0.
Étape 2 : Initialiser
- le nombre d'itération iter (à 1),
- la table de Hachage H (ajouter la solution initiale à H),
- la liste tabou T (vide).

Répéter les étapes de 3 à 11 jusqu'à iter = iterMax.
  Étape 3 : Initialiser le compteur d'éléments (composants d'une solution)
  Répéter les étapes de 4 à 6 jusqu'à
    Étape 4 : Sélectionner aléatoirement un ensemble de voisins des éléments non tabous, puis choisir le meilleur de l'ensemble.
    Étape 5 : Déclarer comme tabou le voisin choisi (voisin ajouté à T).
    Étape 6 : Déplacer l'élément courant et le voisin choisi vers leurs positions mutuelles.
    Étape 7 : Evaluer la nouvelle solution calculée pour mesurer son gain G_{iter}.
    Étape 8 : ***Si :*** la solution calculée est déjà contenue dans H *alors :* Solution ignorée, ***Sinon :*** Ajouter la solution à H
    Étape 9 : Si $G_{iter} > G_{iter-1}$ *alors* continuer le calcul à partir de $Solution_i$ *Sinon* Continuer le calcul à partir de la $solution_{iter-1}$
    Étape 10 : Vider T.
Étape 11 : iter + +.

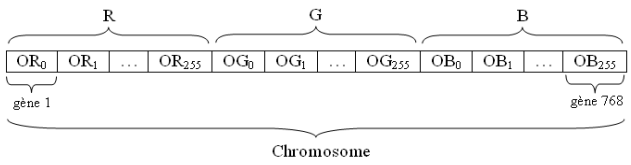


Figure 1. Codage de données sous Tabu-Crypt.

Où :
OR_i est le nombre d'occurrence des valeurs de la matrice composante R qui égale à *i*,
OG_i est le nombre d'occurrence des valeurs de la matrice composante G qui égale à *i*,
OB_i est le nombre d'occurrence des valeurs de la matrice composante B qui égale à *i*.

B. Choix des éléments à déplacer

Les solutions sont représentées soit, par un vecteur englobant 768 éléments. *Tabu-Crypt* cherche à réarranger aléatoirement le vecteur correspondant à la solution initiale codant l'image originale en permutant les éléments entre eux.

C. Génération de voisinage

Le voisinage est le responsable sur le fait de pousser l'algorithme à l'amélioration de la qualité des résultats. Dans notre cas, il est généré de la manière suivante :
Pour chacun des éléments du sous-ensemble construit dans l'étape précédente, nous générons aléatoirement un ensemble de voisins candidats au déplacement avec l'élément en question. Lors de l'élection d'un voisin, les conditions suivantes doivent être vérifiées :

- Un voisin ne doit pas être élu plus qu'une fois, c'est-à-dire, tous les voisins de l'ensemble de voisins sont différents les uns des autres.
- Les voisins élus n'appartiennent pas dans la liste tabou.

Une fois l'ensemble des voisins construit, nous choisissons celui qui diffère le plus de l'élément à déplacer. Formellement, soit E_V l'ensemble de n voisins V, et E_i l'élément candidat au déplacement. La sélection du voisin v avec lequel l'élément E_i sera permuté se fait comme suit :

$$v = V_j \,/\, j = \overline{1,n} : Max\left(\left|E_i - V_j\right|\right) \qquad (1)$$

D. Mise à jour de la liste tabou

Dans notre cas et à une itération donnée, la liste tabou regroupe les voisins qui ont participé à des déplacements au cours des itérations précédentes avec l'un des éléments du sous-ensemble construit dans l'étape 2 (voir section III.B). Ainsi, la mise à jour de cette liste (liste tabou) se fait à chaque fois qu'un élément sera permuté avec un voisin choisi suivant l'étape 3 (voir section III.C) en ajoutant ce dernier à la liste des éléments tabous. A la fin de chaque itération, le contenu de la liste tabou sera effacé.

E. Calcul de solutions

Au cours d'une itération donnée, le calcul de la solution proposée à cette itération se base sur la solution calculée à l'itération précédente. En effet, des éléments du vecteur présentant la solution à l'itération précédente échangeront leurs positions avec celles des voisins désignés pour chacun d'eux. Cette nouvelle dispersion des éléments forme le nouveau vecteur solution.

F. Mise à jour de la table de hachage et évaluation des solutions

A chaque itération la solution calculée est ajoutée à la table de hachage puis elle sera examinée selon les trois critères suivants:

1) Si la solution S_i obtenue n'est pas une nouvelle solution alors elle sera rejetée et les étapes de 2 à 6 sont à refaire.

2) Si la solution S_i obtenue est une nouvelle solution mais plus mauvaise que celle de l'itération *i-1* (S_{i-1}), de même elle est rejetée, mais sauvegardée dans la table de hachage, et les étapes de 2 à 6 sont à refaire.

3) Si la solution S_i obtenue est une nouvelle solution mais, cette fois ci, est meilleure que celle de l'itération (i-1), alors elle sera retenue pour continuer la recherche à partir de la prochaine itération et, ainsi, elle servira à calculer la solution suivante S_{i+1}. En même temps, elle sera mémorisée dans la table de hachage.

Dans notre cas, l'évaluation se fait suivant la fonction d'évaluation illustrée par la formule suivante, notant que S_0 soit la solution initiale (voir étape 1).

$$F(S_i) = \sum_{j=1}^{768} \left| S_{i_j} - S_{0_j} \right| \qquad (2)$$

G. Critère d'arrêt

Pour toute recherche itérative un critère d'arrêt est obligatoire pour éviter le problème de boucles infinies. Ce dernier qui doit être fixé à l'avance permet de déterminer le nombre de fois que la tâche à exécuter sera répétée. Pour notre algorithme, nous avons choisi de fixer, expérimentalement, le nombre d'itérations. Ce choix influe directement sur longueur du temps de calcul et de la qualité de la solution construite.

H. Mécanismes avancés en recherche tabou

1. Intensification / Exploitation

L'intensification consiste à approfondir la recherche dans certaines régions de l'espace, identifiées comme susceptibles de contenir un optimum global.
Pour notre cas, les mouvements globaux en plus des mouvements élémentaires serviront à mieux intensifier la recherche. En effet, les mouvements élémentaires traduisent les déplacements des éléments d'un vecteur solution vers les positions de leurs voisins et vise versa. Ces voisins sont les éléments optimaux (les meilleurs voisins) de l'ensemble des voisins candidats au déplacement (voir étape 3). Donc, nous cherchons à exploiter les qualités des éléments. De même, les mouvements globaux cherchent à exploiter les qualités des solutions (les vecteurs d'éléments) en n'acceptant de poursuivre la recherche qu'à partir des solutions améliorantes lors du passage d'une itération à l'itération qui suit.

2. Diversification / Exploration

La diversification vise à utiliser des mouvements encore jamais réalisés afin d'explorer de nouvelles régions de l'espace de recherche et des régions éloignées du voisinage actuel. Dans notre cas, cette caractéristique est accomplie par la sélection aléatoire d'un nouveau voisinage non tabou pour chaque élément.

3. Critère d'aspiration

Le critère d'aspiration autorise un mouvement tabou sous certaines conditions. Il consiste à tester si la solution produite de statut tabou présente un coût inférieur ou de qualité meilleure que ceux de la meilleure solution trouvée jusqu'à présent. Si c'est le cas, le statut tabou de la solution est levé.

Dans notre cas, la notion de tabou est exploitée lors de la construction d'une certaine solution en interdisant les mouvements élémentaires vers les éléments voisins avec

lesquels les éléments du sous-ensemble construit comme l'indique l'étape 2 ont échangé de positions. Ainsi, le mécanisme d'aspiration n'a pas de place dans notre méthode du fait que toute libération d'un élément de la liste tabou entraîne une modification des éléments des vecteurs et, par conséquent la modification des caractéristiques servira à calculer ultérieurement la donnée déchiffrée, tandis que notre but est, plutôt, modifier la répartition des éléments des vecteurs.

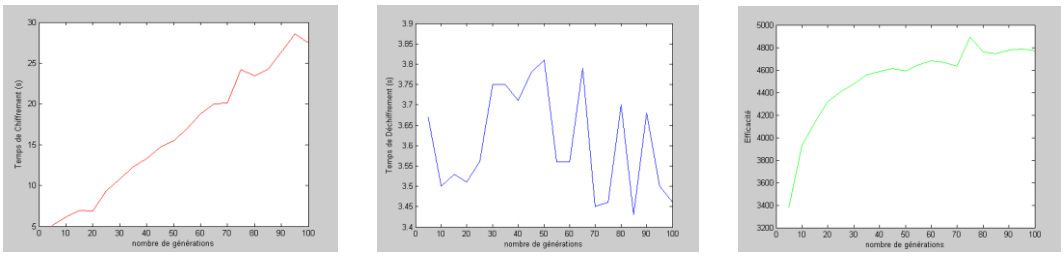
I. Déchiffrement

L'opération inverse au chiffrement est le déchiffrement qui permet de regénérer l'image originale précédemment chiffrée par *Tabu-Crypt*. Pour ce faire, ce processus exploite une information secrète calculée lors du chiffrement à coté de l'image chiffrée. Il s'agit d'une clé de session secrète. Elle représente les permutations des positions des 768 éléments du vecteur codant une certaine image à travers les différentes itérations. Une fois elle parviendra sans modification au destinataire approprié, elle servira à calculer la donnée originale.

J. Paramétrage et résultats

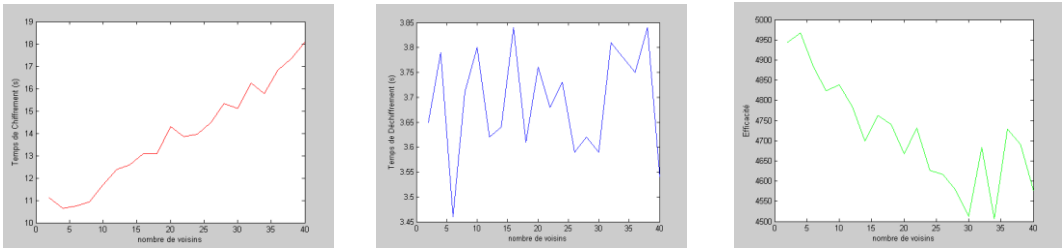
Les principaux paramètres de *Tabu-Crypt* concernent le nombre de générations ou d'itérations de l'algorithme et le nombre de voisins [12].

A ce niveau nous présentons les résultats de chiffrement par *Tabu-Crypt* de certaines images tests. Deux versions chiffrées de chacune des images tests sont présentées. Le tableau II résume les temps de chiffrement et de déchiffrement ainsi que l'efficacité de chacun des exemples.



(a) (b) (c)

Figure 2. Résultats obtenus suivant les différentes valeurs de test de nombre d'itérations (générations) : (a) Temps de chiffrement, (b) Temps de déchiffrement, (c) Efficacité.



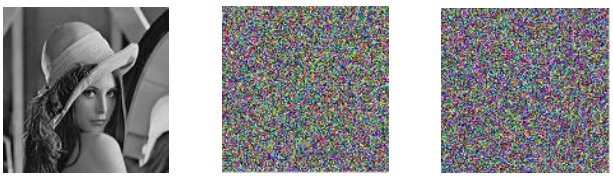
(a) (b) (c)

Figure 3. Résultats obtenus suivant les différentes valeurs de test de nombre de voisins : (a) Temps de chiffrement, (b) Temps de déchiffrement, (c) Efficacité.

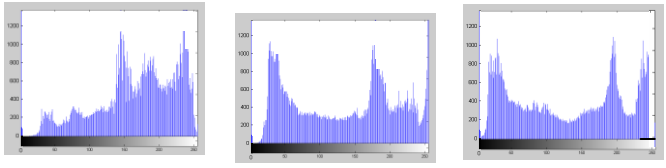
Le tableau suivant (tableau I) résumé les valeurs de paramètres adoptés par *Tabu-Crypt.*

TABLE I. VALEURS ADOPTEES POUR LES PARAMETRES DE Tabu-Crypt.

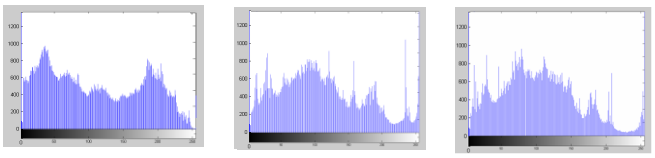
	Valeurs des paramètres de *Tabu-Crypt*
Nombre de voisins	4
Nombre de générations	45



(a) (b) (c)



(d)



(e)

Figure 4. (a) Image test Lena, (b) première version chiffrée, (c) deuxième version chiffrée, (d) Histogrammes de la première version chiffrée, (e) Histogrammes de la deuxième version chiffrée.

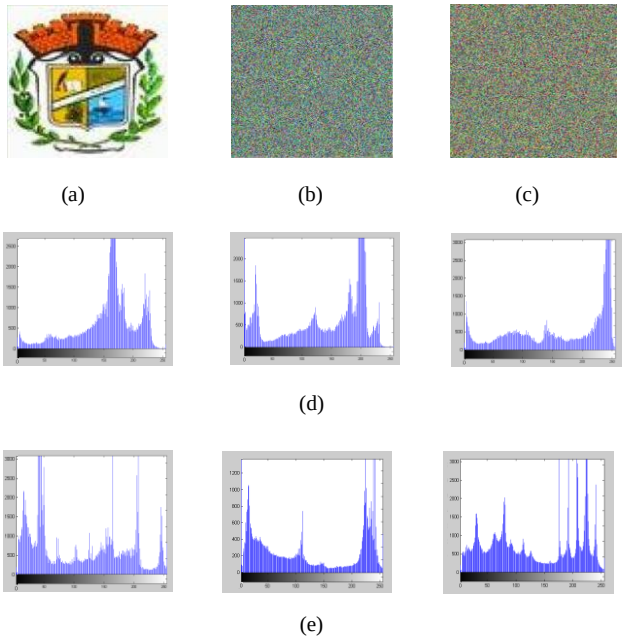


Figure 5. (a) Image test Logo, (b) première version chiffrée, (c) deuxième version chiffrée, (d) Histogrammes de la première version chiffrée, (e) Histogrammes de la deuxième version chiffrée.

Le tableau II présente une récapitulation des résultats de chiffrement des données tests présentées ci-dessus.

La source de notre motivation d'application de la recherche tabou qui est une métaheuristique était l'adaptabilité et l'efficacité d'application d'une telle méthode dans le domaine de cryptage à cause de sa large utilisation de l'aléatoire. L'algorithme ainsi développé, *Tabu-Crypt*, sera doté d'un grand pouvoir confusionnel pour compliquer le plus possible la tâche de cryptanalyse.

En effet, l'aspect non déterministe innové à travers notre approche proposée et traduit par le fait que le chiffrement d'une même image sur différentes instances donne lieu à des versions chiffrées différentes (voir les résultats sur les figures 4 et 5) ; cet aspect représente l'un des points forts de *Tabu-Crypt* leur assurant, ainsi, une résistibilité contre les attaques différentielles. Ceci est assuré par une élection aléatoire, sur l'ensemble des générations, des voisins des éléments codant une solution intermédiaire pour calculer une autre solution intermédiaire ou une solution finale qui soit satisfaisante.

De même, l'attaque statistique sera presque impossible à appliquer grâce à ce dernier point en plus du fait que les images originales et leurs version chiffrées, seront presque toutes différentes comme le montre le tableau III résumant les valeurs d'application des mesures de similarité NPCR, MAE et MSE où les valeurs sont proches de l'optimum qui vaut 1 dans le cas de la mesure NPCR, par exemple.

TABLE III. NIVEAUX DE CONFUSION DE Tabu-Crypt.

	NPCR	**MAE**	**MSE**
Lena-version chiffrée 1	0.8927	0.97	0.68
Lena-version chiffrée 2	0.9104	1.05	0.71

$$NPCR = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} D(i,j) \qquad \text{(III.39)}$$

$$D(i,j) = \begin{cases} 0 Si\ \mathrm{Im}_O(i,j) = \mathrm{Im}_C(i,j) \\ 1 Si\ \mathrm{Im}_O(i,j) \neq \mathrm{Im}_C(i,j) \end{cases} \qquad \text{(III.40)}$$

$$MAE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} \frac{\left|\mathrm{Im}_O(i,j) - \mathrm{Im}_C(i,j)\right|}{255} \qquad \text{(III.41)}$$

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} \frac{(\mathrm{Im}_O(i,j) - \mathrm{Im}_C(i,j))^2}{255^2} \qquad \text{(III.42)}$$

De son tour, l'attaque exhaustive sera mise à l'écart grâce à la taille de la clé générée par *Tabu-Crypt* qui est largement sécurisée et qui est de taille égale à 7680 bits. Cette taille est clairement meilleure que celle des clés générées par les algorithmes DES, 3DES [1, 2] ou même AES [13] qui génèrent des clés de tailles égalent à 56 bits, 128 bits et 256 bits respectivement.

TABLE II. RESULTATS OBTENUS PAR Tabu-Crypt.

			Taille image (pixels)	**Taille clé (bits)**	**Efficacité**	**Temps chiffrement (s)**	**Temps déchiffrement (s)**
Images	Lena	Version-Chiff1	131 X 131	7680	139362.0	14.62	3.7
		Version-Chiff2			134074.0	14.21	4.01
	Logo	Version-Chiff1	420 X 395	7680	246426.0	15.06	3.11
		Version-Chiff2			225688.0	15.28	3.81

IV. CONCLUSION

Dans ce travail, nous avons traité la sécurisation des données images, qui sont considérées comme des données particulières en raison de leurs tailles et de leurs informations qui sont de natures bidimensionnelles et redondantes [14]. Ces particularités des données rendent les algorithmes développés dans la littérature inutilisables sous leurs formes classiques, à cause des contraintes de la vitesse et de la perte de l'information. Ainsi, nous avons pensé à développer un nouvel

algorithme de chiffrement d'images fondé sur la métaheuristique de recherche tabou.

En effet, les détails des différentes étapes du processus développé ont été explicités et testés sur différentes images test. Donc, notre objectif qui se résume en l'exploitation d'un voisinage en recherche locale de type taboue pour la manipulation de données images, est atteint dans certains cotés vu les qualités de la méthode tabou (simplicité du principe, performance…) en donnant lieu à des résultats de bon niveau de confusion.

REFERENCES

[1] G. Ganteaut, et F. Lévy, La cryptologie moderne. L'Armement, 73:76_83, 2001.

[2] D. Stinson, Cryptographie, théorie et pratique, International Thomson Publishing, France, 1996.

[3] S. Ghernaouti-Hélie,. Sécurité informatique et réseaux, Editions Dunod, 2004.

[4] S. Katzenbeisser and F. A. P. Petitcolas, Information Hiding Techniques for Steganography and Digital Watermarking. Artech House. London, 2000.

[5] D. Kahn, The Codebreakers - The Story of Secret Writing. Scribner. New York, 1996.

[6] A.J. Menezes, P.C. Oorschot and S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

[7] I. Souici, K. Benhamza et H. Seridi, Nouvel Algorithme Crypto-évolutionnaire (ACEO) : Conception et Evaluation. CDROM, 6ème Conférence sur le Génie Electrique, CGE'06, 13-14 Avril, 2009. Ecole Militaire Polytechnique, Bordj El Bahri, Algérie, 2009.

[8] I. Souici, H. Seridi et H. Akdag, Images evolutionary encryption. Premier Congrès International sur les modèles, Optimisation et Sécurité des Systèmes, ICMOSS'2010, 29-31 Mai, 2010. Tiaret, Algérie.

[9] I. Souici, H. Seridi et H. Akdag, Images Encrypton by the Use of Evolutionary algorithms. Analog Integrated Circuits and Signal Processing, Springer, Volume 69, Issue 1, pp 49-58, ISSN 1573-1979, Springer, 2011.

[10] F. Glover and M. Laguna, Tabu Search. Kluwer, Boston, 1997.

[11] J. Ayas et M. André Viau, La recherche Tabou. Notes de cours, 2004.

[12] Z. Michalewicz and M. Schmidt, Parameter Control in Practice, Studies in Computational Intelligence (SCI) 54. 2007, 277–294, Springer-Verlag Berlin Heidelberg.

[13] F. Leprévost, Les standards cryptographiques du XXIe siècle : AES et IEEE-P1363. 2000, Gazette des Mathématiciens - n°85.

[14] J.M.M. Rodrigues, Transfert sécurisé d'images par combinaison de techniques de compression, cryptage et marquage, thèse de doctorat, 2006 Université de Montpellier II.