

Cryptographie évolutionniste

Application des algorithmes évolutionnistes à la cryptographie

Fouzia Omary* –Abderrahim Tragha** –Aboubakr Lbekkouri*

*Département de mathématiques et informatique
faculté des sciences-Rabat
université MohammedV-Maroc
omaryfouzia@yahoo.fr

**Département de mathématiques et informatique
faculté des sciences Ben Msik
Université HassanII-Mohammedia Casablanca
a.tragha@univh2m.ac.ma

.....
RESUME. Vu le grand succès des algorithmes évolutionnistes dans les problèmes d'optimisations, nous les exploitons dans la phase principale de la cryptographie soit: le chiffrement. Dans cet article nous avons conçu et réalisé un algorithme de chiffrement évolutionniste. Pour commencer, nous avons codé le problème de chiffrement en le simulant (ou rapprochant) à un problème d'ordonnement (notamment permutations). Après, nous avons créé un codage adapté pour nos chromosomes. Ensuite nous avons défini la fonction d'évaluation adéquate. Quand aux opérateurs génétiques nous avons utilisé ceux travaillant sur les permutations. Pour donner de la force et de résistance à notre algorithme nous lui avons attribué des clés de différentes natures, certaines sont secrètes et symétriques, autre de session et peut être d'un usage symétrique ou asymétrique.

ABSTRACT. Seen the big success of the evolutionist algorithms in the problems of optimizations, we exploit them in the main phase of the cryptography is: the ciphering. In this article we conceived and achieved an evolutionist ciphering algorithm. To begin, we coded the problem of ciphering while simulating it (or bringing closer) to a problem of scheduling (notably permutations). After, we have created a coding adapted for our chromosomes. Then we defined the adequate function of evaluation. When to the genetic operators we used those working on the permutations. To give strength and resistance to our algorithm us assigned him keys of different natures, some are secret and symmetrical, other of session and can be of a symmetrical or asymmetric use.

Mots-clés: algorithmes-génétiques, évolutionnistes, cryptographie, chiffrement, clés.

Key-words: genetic-algorithms, evolutionist, cryptography, ciphering, keys .

.....

Cryptographie évolutionniste

1. Introduction

De nos jours, la cryptographie s'est imposée dans la vie courante notamment pour la protection (ou sécurité) des transactions. Malgré son antiquité (plus de 3000 ans) elle est toujours à l'état embryonnaire. En effet, les vrais algorithmes de chiffrement sont comptés sur les bouts des doigts (DES [3], IDEA (1977) [12], RSA (1970) [4], ...). Et actuellement, le cryptosystème dominant est PGP (1992) [7] et [11],

Parallèlement, les domaines d'application des algorithmes évolutionnistes n'ont cessé de s'élargir grâce à leur grande efficacité (performance, rapidité,...). Leur succès a atteint son apogée dans le domaine de l'intelligence artificielle et recherche opérationnelle pour résoudre des problèmes d'optimisation. Certes, concevoir et réaliser un bon algorithme évolutionniste pour chiffrement ne peut qu'être couronné de succès. C'est le but de notre travail [10]. Pour commencer nous avons ramené le problème de chiffrement d'un message M à un problème d'optimisation. Après, nous avons codé ce problème d'une manière spéciale permettant de nous ramener aux problèmes d'ordonnancement (en particulier permutations) dont la résolution par algorithmes génétiques est une vraie réussite. Ensuite, nous avons bâti notre algorithme en définissant soigneusement la fonction d'évaluation, en précisant la nature des individus aptes à la sélection et en choisissant les opérateurs génétiques adaptés à notre problème. Pour achever, nous avons présenté les moyens de déchiffrement et nous avons entamé une discussion pour illustrer les avantages de notre algorithme.

2. Description de notre algorithme de chiffrement

Soit M le message à chiffrer. M est une suite de n caractères. Soit C une clé secrète formée de p caractères pris dans $\{0,1\}$. Après un brouillage⁽¹⁾ initial, permettant de masquer les caractères de M , nous décomposons ce dernier en m blocs B_1, B_2, \dots, B_m de même taille N .

Note . cette décomposition n'aura lieu que pour les messages de grandes tailles.

Nous distribuons les m blocs sur m processeurs fonctionnant en parallèle . Chacun des processeurs applique notre algorithme décrit dans 2.2.3. Dorénavant B désigne le bloc courant affecté à l'un des processeurs et O.T.L notre algorithme.

2.1. Schématisation du problème

Soient a_1, a_2, \dots, a_l les différents caractères de B ($l \leq N$). Désignons par L_i ($1 \leq i \leq l$) la liste des différentes positions du caractère a_i dans B avant le chiffrement et par $\text{card}(L_i)$ le nombre des occurrences de a_i dans B .

Le bloc B peut alors être représenté par le tableau ci-dessous :

(a_1, L_1)	(a_2, L_2)	(a_l, L_l)
--------------	--------------	-----------	--------------

Le but de notre algorithme est de créer le maximum de désordre dans les positions des caractères de B . Pour cela, nous changeons itérativement la répartition des listes L_i ($1 \leq i \leq l$) sur les différents caractères de B (sans modifier le contenu des listes) de telle manière que la différence entre le cardinal de la nouvelle liste affectée à chaque caractère a_i et le cardinal de la liste L_i d'origine soit maximale. Nous sommes donc devant un problème d'optimisation et nous pouvons faire appel aux algorithmes évolutionnistes, notamment ceux appliqués dans les problèmes d'ordonnancement [2]. Ces derniers ont plusieurs versions la plus utilisée est celle décrite ci-dessous.

2.2 Squelette de l'algorithme

Etape 0 : Définir un codage du problème

Etape 1 : Créer une population initiale P_0 de q individus $\{X_1, X_2, \dots, X_q\}$

$i := 0$;

Etape 2 : Evaluation des individus.

Soit F la fonction d'évaluation. Calculer $F(X_i)$ pour chaque individu X_i de P_i

Etape 3 : Sélection

Sélectionner les meilleurs individus (au sens de F) et les grouper par paire.

Etape 4 : Application des opérateurs génétiques

1-Coisement : Appliquer l'opération de croisement aux paires sélectionnées

2-Mutation : Appliquer la mutation aux individus issus du croisement

Ranger les nouveaux individus obtenus (de 1 et 2) dans une nouvelle génération P_{i+1}

Répéter les étapes 2,3 et 4 jusqu'à l'obtention du niveau de performance souhaité .

2.3 Notre algorithme O.T.L

Etape 0 : Codage

Un individu (ou chromosome) est un vecteur de taille l . Les gènes sont les listes L_{ki} ($1 \leq i \leq l$). Le $j^{\text{ème}}$ gène L_{kj} contient les nouvelles positions que prendra le caractère a_j .

Etape 1 : Création de la population initiale P_0 composée de q individus $:X_1, X_2, \dots, X_q$.

Nous appelons Ch-initial le chromosome initial dont les gènes sont (avec respect d'ordre) : L_1, L_2, \dots, L_l .

Nous appliquons q bonnes⁽²⁾ permutations sur Ch-initial afin d'obtenir q individus distincts formant ainsi la population initiale constituée de q solutions potentielles du problème.

$i:=0$.

Etape 2: Evaluation des individus

Soit X_k un individu de P_i dont les gènes sont: $L_{k_1}, L_{k_2}, \dots, L_{k_l}$.

Nous définissons la fonction d'évaluation F sur l'ensemble des individus X_k par:

$$F(X_k) = \sum_{i=1}^l |card(L_{k_i}) - card(L_i)|$$

étape3: Sélection des meilleurs individus

Nous utilisons la méthode classique de la roulette [5], permettant de retenir les individus les plus forts. Rappelons le processus :

On affecte à chaque individu X_i une probabilité d'apparition $p(X_i)$, (ou force relative) par:

$$p(X_i) = \frac{F(X_i)}{\sum_{k=1}^q F(X_k)}$$

La sélection d'un individu se fait de la manière suivante.

Soit $q_i = \sum_{k=1}^i p(X_k)$ la probabilité d'apparition cumulée d'un individu X_i et

soit r un nombre aléatoire compris entre 0 et 1, l'individu retenu est:

$$\begin{cases} X_1 & \text{si } q_1 \leq r \quad \text{Ou} \\ X_i & \text{si } q_{i-1} < r \leq q_i \quad \text{pour } 2 \leq i \leq q \end{cases}$$

(q étant le nombre d'individu dans la population)

Ce processus est répété q fois. Avec ce principe, un individu fort peut être sélectionné plusieurs fois. Par contre un individu faible a moins de chance d'être sélectionné.

Et nous introduisons une fonction de contrôle qui va éliminer les individus pour lesquels seulement une minorité de gènes ont changé de valeurs par rapport au chromosome initial : Ch-initial.

Puisque nous nous sommes ramenés à un problème de permutations avec contraintes, nous appliquons alors les opérateurs génétiques adaptés à ce genre de problème.

Etape 4: Application des opérateurs génétiques

Croisement MPX (Maximal Preservative X) :

Ce croisement a été proposé par Müllhenbein [8] et [9] pour le problème du voyageur de commerce. L'idée de cet opérateur est d'insérer une partie du chromosome d'un parent dans le chromosome de l'autre parent de telle façon que le croisement résultant soit le plus proche possible de ses parents. C'est un croisement à deux points. Les deux fils sont obtenus de manière symétrique. Par un exemple nous illustrons le fonctionnement.

Exemple :

Parent 1 : a b c d e f g h i j k l

Parent 2 : c b a g h i j l k f d e

La zone de croisement est comprise entre les positions 5 et 9. La première étape consiste à recopier la zone de croisement du parent 1 sur le fils 1. Ensuite, les gènes du fils qui ne sont pas dans la zone de croisement sont complétés de la façon suivante: Le 1^{er} gène du parent2 est recopié sur le 1^{er} gène du fils 1 si cette copie respecte les contraintes(ne crée pas une tournée incohérente). Sinon, le 1^{er} gène du parent1 est recopié sur le 1^{er} gène du fils 1 si cette copie ne crée pas de doublons.. Si les deux cas précédents ne peuvent pas s'appliquer, le 1^{er} gène du fils 1 reçoit un gène de la zone de croisement du parent 2 qui respecte les contraintes(premier non pris).

Fils 1 : c b a d e f g h i j k l

Fils 2 : a b c d h i j l k f e g

Ce croisement est appliqué aux individus sélectionnés avec un taux bien précis. D'après [6], le meilleur taux est de l'ordre de 60% à 100%.

Mutation de transposition :

Nous choisissons la mutation qui consiste à permuter aléatoirement deux gènes d'un chromosome. Cet opérateur est appliqué aux individus issus du croisement avec un taux adapté, de préférence de 0.1% à 5% [6].

Placer la nouvelle progéniture dans une nouvelle population P_{i+1} .

Répéter les étapes 2, 3 et 4 jusqu'à un critère d'arrêt.

Définir la condition d'arrêt :

La fonction F est bornée car $0 \leq F(X) \leq 2 * l$ pour tout individu X.

En effet :

$$\begin{aligned} \sum_i |card(L_{k_i}) - card(L_i)| &\leq \sum_i (card(L_{k_i}) + card(L_i)) \\ &\leq \sum_i card(L_{k_i}) + \sum_i card(L_i) \leq 2 * l \end{aligned}$$

Théoriquement et mathématiquement parlant, la fonction F admet un maximum Max puisqu'elle est bornée. En pratique, nous ne sommes pas sûrs que F converge vers Max. Cependant, d'après certains résultats de recherches [1], la convergence est assurée mais peut-être vers une valeur proche de Max, déterminée expérimentalement.

PHASE FINALE DE L'ALGORITHME :

A partir de la meilleure solution Ch-final obtenue par O.T.L nous constituons le bloc chiffré correspondant. Ensuite nous concaténons ces blocs chiffrés (obtenus par les différents processeurs). Ceci nous donne le message chiffré M' du message initial M.

(1) Brouillage: ce brouillage peut être effectué par combinaison de plusieurs méthodes simples comme les substitutions[3][12] ou les transpositions matricielles[3] Là, nous devons disposer de clés secrètes symétriques.

(2) Bonnes permutations: comme les permutations qui inversent l'ordre des éléments (par exemple (a b c d e) devient (e d c b a)). Et en général, toutes les permutations pour lesquels aucun élément n'a gardé sa position initiale.

3. Déchiffrement

Le déchiffrement doit commencer par la recherche de la réciproque de la dernière opération du chiffrement . Ainsi le message M' sera décomposé de nouveau en m blocs B_i de même taille l .

3.1. Déchiffrement d'un bloc

Une fois la meilleure solution Ch-final est donnée par l'algorithme, nous identifions alors la permutation qui, partant de Ch-initial aboutit à Ch-final. Cette permutation servira de clé de session pour obtenir le chromosome initial Ch-initial. Ainsi à la fin de l'algorithme nous ajoutons une fonction dont le rôle est d'établir cette identification.

3.2. Déchiffrement du brouillage

Toutes les fonctions ou opérations utilisées dans le brouillage ⁽¹⁾ admettent des réciproques permettant ainsi facilement le déchiffrement grâce aux clés symétriques.

4. Discussion.

Etant donné que la plupart des processus utilisés dans les algorithmes évolutionnistes sont aléatoires, ceci donne de la résistance à notre algorithme de chiffrement. En effet, l'aléatoire est un ennemi des cryptanalystes. D'autant plus, nous avons exploité ces processus pour déduire la clé de session en plus de clés secrètes symétriques utilisées initialement. Ces dernières renforcent la sécurité davantage. Cela dit, notre algorithme jouit de tous les avantages d'un algorithme évolutionniste, à savoir: simplicité, rapidité, coût faible des opérateurs génétiques et performance. Il suffit de choisir des paramètres (taux de population, taux des opérateurs,...) adaptés.

Pour bien situer notre travail deux cas se présentent:

1-Si la clé de session peut être acheminée à travers un canal sûr (comme dans n'importe quel chiffrement symétrique à clé de session) alors nous avons établi un algorithme de chiffrement symétrique.

2-Si la clé de session est chiffrée par un algorithme asymétrique ensuite envoyée accompagnée du message chiffrée nous pouvons dire que nous avons établi un système de chiffrement hybride (comme PGP).

5. Conclusion

Les domaines d'application des algorithmes évolutionnistes s'étendent de plus en plus et l'avenir est à eux. Mais en dépit de leur simplicité, les algorithmes évolutionnistes ne sont pas évidents dans leur application. En fait, concevoir et réaliser un bon algorithme évolutionniste demande une bonne connaissance du problème, une bonne compréhension des mécanismes évolutionnistes et beaucoup de créativité. Nous avons vécu tout cela avant de franchir les trois obstacles de base dans un algorithme évolutionniste soient:

-Le codage du problème en essayant de le ramener dans notre cas à un problème d'ordonnement .

-Le codage des chromosomes (ou individus).

-La définition de la fonction d'évaluation .

Mais cela en vaut la peine vu les avantages mentionnés dans la discussion plus haut. En contre partie, les études théoriques sur la vitesse de convergence et de l'influence des jeux de paramètres sont jusque là un vrai chantier de travaux mathématiques rigoureux.

6. Bibliographie

- [1] Catherine Khan Phang. "Algorithmes heuristiques et évolutionnistes"
Thèse de doctorat université de Lille. Octobre 1988.
- [2] Christophe Caux- Henri Pierreval- Marie-Claude Portmann
" Les algorithmes génétiques et leur application aux problèmes
d'ordonnement "
APII. Volume 29-N° 4-5/1995 pages 409 à 443.
- [3] Florin.G et Natkin.S
" les techniques de la cryptographie "
CNAM 2002.
- [4] Ghislaine Labouret
"Introduction à la cryptographie ".2001.
www.hsc.fr/ressources/cours/crypto/index.html
- [5] Goldberg D.E.
" Genetic algorithms in search optimisation & Machine Learning "
(Addison-Wesley. Publishing Company, Inc) 1989.

- [6] Grenfenslette,j.j "optimisation of control parameters for genetic algorithms"
 IEEE translation on systems Man, and cybernetics
 Vol 16 N°1 pp122-128.1986.
- [7] Michel Arboi
 FAQ de sci.crypt "Cryptographie à clé publique".
<http://michel.arboi.free.fr/crypt> FAQ/
- [8] Mühlenbein H.,
 "Evolutionary Algorithms: Theory and applications" (Wiley) 1993.
- [9] Mühlenbein H., and Schlierkamp-Voosen D.
 "Predictive Models for the Breeder Genetic Algorithm-I, continuous
 Parameter Optimization. Evolutionary computation,1(1),25-49". 1993
- [10] Omary.F, Lbekkouri.A et Tragha.A
 "Extension des applications des algorithmes évolutionnistes " .Rapport
 interne N° 7 / 01/ 2004 .Département de mathématiques et informatique
 Faculté des sciences -Rabat.
- [11] PGP. "Personal Privacy pour windows 95, 98 et NT"
 NAI (traduction française:news.misc-cryptologie,1998)
- [12] Renaud Tisserand.,
 "Etat de l'art sur la cryptographie". 2000
http://www.mines.u-nancy.fr/~tisseran/cours/cryptographie_Renaud.pdf

7.Biographie :

Fouzia Omary.

Enseignante chercheur au département de mathématiques et informatique à la faculté des sciences Université Mohammed V Rabat depuis 1984. Mathématicienne de formation, elle a poursuivi ses études de 3^{ème} cycle en informatique à la même université. Le sujet de la thèse se situait dans les différents domaines soient: analyse syntaxique, incrémentalité et parallélisme .Elle a été professeur responsable de l'informatique théorique pendant cinq ans au sein du même département. Ses domaines de recherche actuels sont les algorithmes évolutionnaires, la cryptographie et la complexité des algorithmes.