Un système numérique de cryptographie basé sur les propriétés des signaux chaotiques discrets

V. GUGLIELMI, P-Y. BESNARD, D. FOURNIER-PRUNARET,
P. PINEL, AK. TAHA, L. BENETEAU.

LESIA, INSA 135 avenue de Rangueil 31077 Toulouse Cedex 04
veronique.guglielmi@insa-tlse.fr

Résumé – Nous présentons ici un cryptosystème numérique. Les algorithmes de cryptage proposés reposent sur l'utilisation des propriétés de suites chaotiques qui sont générées par des récurrences non inversibles de dimension 2. L'implémentation de ces algorithmes a été réalisée sur DSP. Cette réalisation effective du système nous a permis tout d'abord une première validation expérimentale, via la détermination de temps de calcul prohibitifs pour les attaques à force brute. Elle nous a aussi permis de mettre en évidence le problème de la représentation du chaos sur des machines discrètes, et de dégager certaines caractéristiques nécessaires aux récurrences utilisées afin de pouvoir continuer plus avant l'analyse de leur utilisation en cryptographie.

Abstract *– We present here a digital cryptosystem. The proposed algorithms are based on the properties of chaotic signals, which are generated by non inversible two-dimensional maps. The implementation of these algorithms has been realized on DSP. This realization has demonstrated first that the cryptosystem works properly, and then has shown very important computing times for brute force attacks. It has also brought to the fore the problem of representation of chaos within discrete systems, and the different characteristics needed for the chaotic maps to be used in digital cryptography.*

1. Contexte

L'utilisation du chaos pour sécuriser les transmissions est un sujet d'études depuis plusieurs années [1]. Le chaos est obtenu à partir de systèmes non linéaires ; il correspond à un comportement non divergent, apériodique et éventuellement borné, de ces systèmes, ce qui le fait apparaître comme du bruit pseudo-aléatoire. Il peut donc être utilisé pour masquer les informations dans une transmission sécurisée : il suffit de le « mélanger » de manière appropriée au message qu'Alice souhaite envoyer confidentiellement à Bob. Dans notre communication, nous présentons un système de cryptographie numérique reposant sur la prise en compte des propriétés de signaux chaotiques issus de récurrences discrètes non linéaires [2].

La grande sensibilité du chaos fait que la même récurrence permet de générer, en modifiant légèrement les valeurs de ses conditions initiales ou de ses paramètres, des suites non périodiques de points aux distributions analogues mais qui pourtant ne prendront en fait jamais les mêmes valeurs. Un « pirate » voulant s'attaquer au décryptage du message envoyé par Alice pour Bob, ne pourra donc pas se servir de la connaissance du système chaotique s'il ne connaît pas les valeurs exactes ayant été utilisées pour initialiser la transmission. Il lui sera alors difficile de s'affranchir de ce « bruit » mélangé au message avant transmission. en particulier à cause du caractère non inversible (car non biunivoque) des opérations concernées..

2. Algorithmes de cryptage

Un principe de cryptage bien connu consiste à crypter des messages grâce à une opération mathématique simple (comme un OU exclusif logique) avec une suite binaire pseudo-aléatoire construite par exemple à partir de registres à décalages. Notre première idée a été de constituer la suite de cryptage à partir d'un signal chaotique, dont l'inhérente non-linéarité devait permettre d'atteindre des performances supérieures [3]. Malheureusement, cette méthode n'amène pas à des systèmes robustes puisque la connaissance ponctuelle de bits de la suite de cryptage permet de trouver d'autres bits de cette suite.

De ce fait, nous avons modifié l'algorithme de cryptage [4][5] ; nous avons de plus essayé non seulement de prendre en compte le plus complètement possible les caractéristiques de nos signaux chaotiques discrets, mais aussi d'arriver à une réalisation effective de notre algorithme sans perte de robustesse liée à l'implémentation.

Afin d'exploiter plus efficacement les propriétés du chaos, nous avons cherché une loi de cryptage dont la sécurité repose directement sur la difficulté à distinguer des suites de points issues d'une même récurrence chaotique.

Pour cela, supposons que le message en clair $\{p_n\}$ soit sous forme binaire, et considérons une récurrence non inversible amenant à des séquences chaotiques $\{s_n\}$. Pour l'implémentation de notre système, les récurrences paraissant

pour l'instant les plus appropriées sont des récurrences de dimension égale au minimum à 2, comme par exemple la récurrence cubique ci-dessous :

$$\begin{cases} x_{n+1} = y_n \\ y_{n+1} = a\left(-{x_n}^3 + x_n\right) + b\left(-{y_n}^3 + y_n\right) \end{cases} \quad (1)$$

où (x,y) sont les variables réelles, et a et b des paramètres réels.

Nous prenons alors pour la séquence chaotique $\{s_n\}$ une seule des deux coordonnées (x_n , y_n), et nous notons: $s_n = F^n(s_0)$, où s_0 représente la condition initiale de la récurrence.

Considérons maintenant deux séquences différentes $\{s_n\} = F^n(s_0)$ et $\{t_n\} = F^n(t_0)$. La n-ième valeur cryptée c_n sera soit s_n soit t_n, selon la valeur du n-ième bit correspondant p_n du message en clair :

$$c_n = s_n \text{ si } p_n = 0 \; ; \; c_n = t_n \text{ si } p_n = 1$$

Nous utilisons donc deux trajectoires du même attracteur chaotique pour les séquences de chiffrement $\{s_n\}$ et $\{t_n\}$, où s_0 et t_0 sont deux conditions initiales différentes choisies dans le bassin d'attraction de l'attracteur. De ce fait, par définition du chaos, les valeurs réelles prises par les suites $\{s_n\}$ et $\{t_n\}$ peuvent être très proches, et distribuées de la même manière. Mais, en même temps, s_n n'est jamais égal à t_n !

Puis, avant la transmission, et pour augmenter encore la sécurité du système, nous utilisons à nouveau les propriétés du chaos, mais cette fois-ci en nous servant de ses ressemblances avec du bruit pseudo-aléatoire pour dissimuler l'information à transmettre. Nous considérons ainsi une troisième trajectoire de la même récurrence amenant au chaos, ce qui nous amène en définitive à trois suites : $s_n = F^n(s_0)$, $t_n = F^n(t_0)$ et $z_n = F^n(z_0)$, où s_0, t_0 et z_0 représentent trois conditions initiales différentes. La n-ième valeur transmise est finalement :

$$A_n = [s_n + z_n] \text{ si } p_n = 0 \; ; \; B_n = [t_n + z_n] \text{ si } p_n = 1$$

où p_n est le n-ième bit du message en clair.

Le récepteur déchiffre le message après soustraction de z_n, par comparaison des valeurs du texte chiffré c_n avec celles des deux séquences s_n et t_n :

$$p_n = 0 \text{ si } c_n = s_n \; ; \; p_n = 1 \text{ si } c_n = t_n$$

Pour que ce déchiffrement soit possible, il est clair qu'il faut pouvoir reconstruire exactement les trois suites $s_n = F^n(s_0)$, $t_n = F^n(t_0)$ et $z_n = F^n(z_0)$ au niveau du récepteur. Les caractéristiques du signal chaotique (paramètres ou conditions initiales, en supposant donc l'équation de récurrence connue de tous) sont donc les clés de notre cryptosystème, et doivent être échangées par un canal sécurisé. Ceci permettra de résoudre le problème de la synchronisation du chaos entre émetteur et récepteur sans diminuer la sécurité.

3. Implémentation

Une première étude de la faisabilité, ainsi que de la robustesse de notre algorithme, avait déjà été réalisée par le passé via des simulations sous le logiciel Matlab. Nous sommes maintenant passés à la réalisation effective de notre système. Le système proposé étant numérique et non analogique, le problème se ramène à effectuer, à un coût demeurant raisonnable, un nombre conséquent de calculs avec le plus de précision possible et le plus vite possible (beaucoup plus vite que les pirates menant leurs différentes attaques successives !). De ce fait, nous nous sommes orientés vers une réalisation basée sur des DSP (Digital Signal Processors), microprocesseurs dédiés au calcul discret.

Cette réalisation sur DSP nous a permis de mettre en évidence certaines faiblesses de l'algorithme, et d'y remédier. Tout d'abord, comme nous travaillons sur des systèmes numériques, les valeurs réelles s_n, t_n, z_n, A_n et B_n ne peuvent être calculées et transmises qu'avec une précision finie, ce qui soulève le problème de l'existence et de la conservation des propriétés théoriques des suites chaotiques, qui sont toujours définies dans l'ensemble des réels, après « quantification » pour passer dans un ensemble fini de valeurs possibles (ensemble déterminé par la précision nécessairement limitée de la représentation des nombres réels sur une machine numérique) [6].

Par ailleurs, nous avons remarqué que l'inhérente non-linéarité de F, même associée à l'importante sensibilité aux conditions initiales et aux paramètres, ne garantit pas à elle seule la sécurité du système : la connaissance d'une certaine portion d'une suite chaotique peut, sous certaines conditions, permettre à un « pirate » de reconstituer l'ensemble de la séquence, puis éventuellement de « casser » totalement le code.

Nous avons alors décidé de transmettre, pour chaque valeur réelle A_n ou B_n codée en représentation binaire sur le DSP, non pas la totalité des bits de la représentation mais seulement certains bits, choisis selon un « masque » de cryptage (notons que ce masque sur les bits peut être considéré comme faisant lui aussi partie de la clé de communication entre Alice et Bob, au même titre que les conditions initiales ou les paramètres de la récurrence).

Ceci nous a permis non seulement d'augmenter encore la sécurité de notre système, mais aussi de poser différemment notre problème de représentation du chaos avec la précision finie d'une machine numérique. Il s'agit désormais de déterminer le codage binaire « optimal » des points de nos suites chaotiques : pour optimiser à la fois la sécurité du cryptage et le taux de transmission des données, les bits effectivement transmis, pour chaque valeur réelle A_n

ou B_n, doivent représenter la quantité minimale d'information nécessaire au récepteur pour déterminer à quelle trajectoire chaotique appartient le réel correspondant (et donc pour être capable de mener à bien le déchiffrement du message).

Appliquer ainsi un « masque » de cryptage pour ne transmettre que certains bits de la représentation binaire des réels, n'a été rendu possible que par l'implémentation de notre algorithme de cryptage sur DSP. En effet, comme le logiciel Matlab ne permet pas de manipuler les différents bits de la représentation « machine » d'un nombre réel, les différentes simulations faites avec ce logiciel ne prenaient que partiellement en compte ce niveau supplémentaire de cryptage (via par exemple la transmission uniquement de certains chiffres après la virgule, ou encore la suppression de la partie entière de chaque réel). En langage C sur DSP, il est au contraire bien possible d'accéder à chacun des bits qui codent en machine un nombre réel donné.

4. Performances

Nous avons donc implémenté notre algorithme en langage C sur un DSP Motorola de type 56824. Le compilateur C utilisé (Dev-C++ pour Windows) nous a permis de simuler un codage des réels en virgule flottante sur 32 bits (selon la norme IEEE-754 simple précision), mais la représentation interne des réels sur le DSP est un codage sur 32 bits en virgule fixe, moins favorable à une bonne possible quant à la représentation des réels.

De ce fait, les différentes performances mesurées, que ce soit en termes de temps de calcul ou de problèmes de précision, ne sont certainement pas optimales et pourraient sans doute être nettement améliorées avec l'utilisation d'un DSP plus adapté à nos calculs (nous avons d'ailleurs récemment pu obtenir, via une collaboration avec l'antenne toulousaine de Motorola, le prêt d'un DSP beaucoup plus rapide et travaillant effectivement en virgule flottante, et nous sommes en train de porter notre algorithme de cryptographie vers ce nouveau DSP.)

Néanmoins, grâce à l'étude des performances de notre système, nous sommes maintenant en mesure d'associer à plusieurs récurrences discrètes non inversibles de dimension 2 des représentations binaires satisfaisantes. Nous pouvons donc proposer un système numérique de cryptographie semblant relativement sûr.

Nous avons implémenté et testé de manière satisfaisante pour l'instant deux récurrences différentes : la récurrence de type cubique donnée par l'équation (1), et la récurrence suivante, dite « DPCM » :

$$\begin{cases} x_{n+1} = y_n \\ y_{n+1} = a_2\left(x_n + Q(s - x_n)\right) + a_1\left(y_n + Q(s - y_n)\right) \end{cases} \quad (2)$$

Cette seconde récurrence apparaît dans les systèmes de télécommunications de type DPCM (Differential Pulse Code Modulation). Elle nous a semblé elle aussi intéressante pour notre cryptosystème, et ce au vu d'études théoriques antérieures des modulateurs DPCM [7][8][9].

Dans l'équation de récurrence (2), la partie non linéaire Q représente le quantificateur du système ; elle peut être modélisée par une tangente hyperbolique :

$$Q(e) = th(pe) \quad (3)$$

p est le gain de compression du quantificateur, a_1 et a_2 sont les deux coefficients du filtre linéaire du second ordre qui joue un rôle de prédicteur dans le modulateur DPCM, et s est le signal d'entrée .

En ce qui concerne le masque de cryptage à appliquer aux bits codant un nombre réel avant transmission, nous l'avons réalisé sous la forme d'un ET logique entre les bits de la représentation binaire des réels et les bits du masque. Ceci permet de forcer à 0 les bits que l'on souhaite masquer sans modifier les autres.

Nous avons alors pu tester tout d'abord le bon fonctionnement du cryptosystème : qu'il s'agisse de la récurrence (1) ou de la récurrence (2), pour des valeurs de paramètres et de conditions initiales correspondant à un attracteur chaotique, les problèmes de précision quant au codage des réels sur une machine numérique ne sont jamais venus bloquer le fonctionnement des algorithmes de cryptage et de décryptage, et ce quel que soit le nombre de caractères à crypter. Il s'agissait là du premier point à vérifier expérimentalement ! Rappelons en effet que la représentation du chaos avec la précision finie d'une machine numérique soulève les problèmes de l'existence et de la conservation des propriétés des suites chaotiques, problèmes qui ne sont pas aujourd'hui théoriquement résolus.

Nous avons ensuite réalisé des attaques de type « à force brute ». Elles ont permis la détermination de temps de calcul très largement prohibitifs pour les différentes attaques de ce type envisageables.

Ces attaques reposent sur le fait que les caractéristiques du signal chaotique (paramètres ou conditions initiales) appartiennent à un sous-ensemble fini de l'ensemble des réels : comme nous travaillons sur des machines numériques, la précision, nécessairement limitée, de la représentation « machine » des nombres réels amène à un nombre fini de valeurs différentes possibles.

Nos attaques à force brute ont donc consisté à réaliser des balayages de l'ensemble des valeurs possibles pour les caractéristiques des signaux chaotiques, et à tester, pour chaque valeur possible, s'il s'agit ou non de la valeur correspondant au signal chaotique utilisé. Nous avons réalisé différents balayages, pour les paramètres ou pour les

conditions initiales, en nous donnant à chaque fois des hypothèses supplémentaires sur les intervalles possibles pour les caractéristiques (afin de limiter les balayages à des temps de calcul mesurables).

Par exemple, avec la récurrence cubique donnée par l'équation (1), en supposant connus les paramètres et en ne recherchant donc que les conditions initiales (x_0 , y_0), et en supposant de plus que les deux conditions initiales sont choisies égales ($x_0 = y_0$), nous avons mesuré les temps de calcul suivants avant d'arriver à retrouver la valeur exacte de la condition initiale utilisée pour le signal chaotique : 1 mn 29 s en restreignant le nombre de valeurs possibles à 10, 15 mn 35 s en le restreignant à 100, 2h 35 mn 50 s pour 1000 valeurs possibles, et 26h 01 mn pour 10 000 valeurs.

Nous avons donc pu vérifier que le temps passé est une fonction linéaire du nombre de clefs possibles. Ceci permet alors d'extrapoler ces temps de calcul au fonctionnement nominal du cryptosystème sans plus limiter le nombre de clefs : en prenant une condition initiale dans [0,1], où il y a environ 2 milliards de possibilités, un couple (x_0 , y_0) représente 4.10^{18} combinaisons possibles ; par règle de 3 par rapport aux temps mesurés, cela amène à un temps de calcul approximatif de 10^{16} heures soit environ 1140 milliards d'années pour l'attaque à force brute !

5. Conclusions

L'étude du cryptosystème numérique que nous venons de présenter a été à la fois pratique (via l'implémentation sur DSP des algorithmes et le lancement de différentes attaques du système) et théorique.

Ceci nous a permis une première validation expérimentale concluante du système. D'une part, les caractéristiques chaotiques des récurrences définies sur l'ensemble des réels demeurent en numérique suffisamment fortes pour que notre cryptosystème fonctionne effectivement. D'autre part, la détermination des temps de calcul pour les attaques à force brute a amené à des temps très largement prohibitifs.

En même temps, nous avons pu dégager certaines propriétés souhaitables pour les récurrences chaotiques utilisées (comme par exemple la non-linéarité à la fois par rapport aux conditions initiales et aux paramètres, ou la forte indépendance entre les différentes dimensions de la récurrence).

La poursuite de notre travail s'oriente alors désormais à la fois vers une implémentation sur des DSP plus performants en terme de précision numérique, et vers des attaques non plus à force brute mais supposant davantage de connaissances a priori sur les données cryptées de la part de l'attaquant.

Références

[1] Kocarev L., « Chaos-based cryptography: a brief overview », *IEEE Circuits & Systems Magazine*, vol. 1, no. 3, 2001.

[2] Fournier-Prunaret D. & Guglielmi V., « Bifurcations and attractors in two-dimensional maps of cubic type », *NOLTA99*, Hawaii (USA), 29 Nov. - 2 Dec. 1999

[3] Bénéteau L., Fournier-Prunaret D., Gaudron G., Guglielmi V., Pinel P., Rouabhi S., Taberly A., Taha A.K., « Applications de la Dynamique Chaotique à la Cryptographie », *Journées Automatique et Télécommunications*, Bordeaux, 13-14 mars 2001

[4] Bénéteau L., Fournier-Prunaret D, Guglielmi V., Pinel P., Rouabhi S., Taha A.K., « Two encryption schemes using the chaotic dynamics of two-dimensional noninvertible maps », *NDES'02 The Nonlinear Dynamics of Electronic Systems*, Izmir, Turquie, 21-23 juin 2002

[5] Bénéteau L., Fournier-Prunaret D, Guglielmi V., Pinel P., Taha A.K., « Chaotic dynamics applied to cryptography », *NOLTA'02 The Nonlinear Theory and Applications*, Xian, Chine, 7-10 octobre 2002.

[6] Robert F., « Les systèmes dynamiques discrets », *Springer*, 1995.

[7] Uhl C. & Fournier-Prunaret D., « Chaotic phenomena in an order 1 DPCM transmission system », *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, vol. 5, no. 4, pp. 1033-1070, 1995.

[8] Taralova I. & Fournier-Prunaret D., "Dynamical study of a second order DCPM transmission system modeled by a piece-wise linear function", *IEEE Transactions on Circuits and systems, I Fundamental theory and applications*, nov. 2002, vol 49, n°11, pp1592-1609.

[9] Fournier-Prunaret D., «Some complex situations in a (Z1, Z3, Z1) two-dimensional map», *Grazer Math. Ber.,* n°339, 1999, pp141-148.