Sur les propriétés structurelles des systèmes dynamiques pour le chiffrement

Phuoc Vo-Tan, Gilles Millérioux, Jamal Daafouz

Centre de Recherche en Automatique de Nancy (CRAN)
Nancy Université - CNRS UMR 7039, France

phuoc.votan,gilles.millerioux@esstin.uhp-nancy.fr,jamal.daafouz@ensem.inpl-nancy.fr
http://cran.uhp-nancy.fr

Résumé— **L'objectif principal de cet article est d'étudier la connexion entre les techniques de "brouillage" de l'information basées sur l'utilisation de dynamiques complexes et les algorithmes standards de chiffrement symétrique. Le but de cette étude est d'être en mesure de sélectionner les algorithmes pertinents de "brouillage" et de pouvoir à terme transposer aisément les techniques standard de cryptanalyse à ces algorithmes en vue de leur validation. On montre comment les propriétés structurelles des systèmes dynamiques, à savoir l'inversibilité, la platitude ou l'identifiabilité paramétrique jouent un rôle fondamental pour établir la connexion. On montre également pourquoi cette étude est constructive dans le sens où elle révèle une nouvelle technique de synthèse de chiffreurs symétriques autosynchrones inédite à ce jour. Cette technique est particularisée pour la classe des systèmes linéaires par morceaux.**

Mots-clés— **Dynamiques complexes, chiffrement, inversibilité, platitude**

I. Introduction

La sécurité des réseaux, reposant en partie sur la confidentialité des échanges d'informations, est devenue une réelle nécessité de par le développement considérable des nouvelles technologies de communication. Ainsi, la cryptographie joue un rôle prépondérant car les informations sont devenues accessibles en transitant au travers de réseaux publics.

De multiples techniques de "brouillage" de l'information utilisant des systèmes dynamiques non linéaires ont été proposées depuis le début des années 90. En effet, une spécificité de ces systèmes réside dans la nature de leurs régimes permanents. Parmi l'ensemble des régimes permanents autres que ceux de type stationnaire ou périodique, les dynamiques complexes se distinguent par la nature erratique des signaux associés et la grande sensibilité aux conditions initiales. Il s'avère que ces caractéristiques sont étroitement liées aux notions de diffusion et de confusion usuellement rencontrées en cryptographie. Cependant, une attention insuffisante a été portée sur les principes de base auxquels ces techniques de "brouillage" devaient se soustraire pour garantir des niveaux de sécurité exigés en pratique.

L'objectif principal de cet article est d'étudier, après avoir rappelé le cadre général développé dans [1], la connexion entre les techniques de "brouillage" basées sur l'utilisation de dynamiques complexes et les algorithmes standards de chiffrement symétrique. Le but de cette étude est d'être en mesure de sélectionner les algorithmes pertinents de "brouillage" et de pouvoir à terme transposer aisément les techniques standard de cryptanalyse à ces algorithmes en vue de leur validation.

Cet article est organisé comme suit. En Section II sont rappelés les principaux types de chiffrement standards dits par flot. Deux algorithmes particuliers connus sous le nom de SSS et Moustique sont ensuite détaillés et les différences structurelles sont mises en exergue. La Section III s'attache à présenter les principaux algorithmes de "brouillage", proposés depuis les années 90, basés sur l'utilisation de systèmes non linéaires exhibant des dynamiques complexes. Une comparaison avec les algorithmes de chiffrement par flot est menée. On montre comment les propriétés structurelles des systèmes dynamiques, à savoir l'inversibilité, la platitude ou l'identifiabilité paramétrique jouent un rôle fondamental à la fois pour établir une connexion ainsi que pour élaborer une nouvelle technique de synthèse de chiffreurs symétriques autosynchrones. En Section IV, cette nouvelle technique de synthèse est particularisée pour la classe des systèmes dynamiques linéaires par morceaux.

II. Chiffreurs symétriques usuels par flot

Une communication sécurisée (cf. [2] pour plus de détails) obéit de manière générale au schéma représenté sur la Figure 1. On distingue les parties émetteur (ou chiffreur) et récepteur (ou déchiffreur) avec leur générateur de clé respectif. m désigne le message clair et c le message chiffré. Côté émetteur, une entité caractérisée par une fonction e dépendant à la fois du message clair m et de la clé k^e assure le chiffrement. Côté récepteur, le déchiffrement est assuré par une fonction d dépendant de la clé k^d et du message chiffré c. Le message chiffré c est véhiculé au travers d'un canal supposé public et peut donc être intercepté par une tierce partie appelée pirate. Dans un procédé de chiffrement symétrique, le cadre de notre étude, les clés des parties émetteur et récepteur doivent être identiques, c'est-à-dire $k^e = k^d$, pour retrouver correctement l'information et assurer $\hat{m} = m$. Pour le chiffrement symétrique par flot, le message clair (*resp.* message chiffré) est constitué d'une séquence de symboles clairs m_k (*resp.* symboles chiffrés c_k). Notons que la terminologie message clair (*resp.* message chiffré) lorsqu'on se réfère au symbole m_k (*resp.* symbole c_k) est souvent abusivement conservée. La clé k^e du chiffreur (*resp.* k^d du déchiffreur) est remplacée par une séquence (ou flot) de clés notées z_k pour le chif-

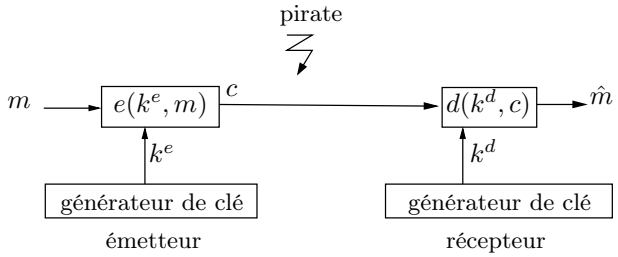

Fig. 1. Principe général d'un schéma de chiffrement/déchiffrement

frement (*resp.* $\hat{z}_k$ pour le déchiffrement). Conformément au principe du chiffrement symétrique, les quantités z_k et $\hat{z}_k$, appelées clés courantes, doivent être égales à chaque instant pour retrouver correctement le message clair et donc assurer $\hat{m}_k = m_k$ pour tout k. En d'autres termes les séquences de clés courantes doivent être synchronisées. Les générateurs côté émetteur et récepteur produisant les séquences de clés courante dépendent d'un paramètre θ qui constitue la clé secrète. Deux types de chiffrement par flot se distinguent par leur générateurs de clés : synchrones et autosynchrones.

A. Chiffreurs par flot synchrones

Un chiffreur par flot synchrone obéit aux équations (1), le schéma-bloc associé est représenté sur la Figure 2.

$$\begin{cases} z_k = \sigma^s(z_{k-1}) \\ c_k = e(z_k, m_k) \end{cases} \quad (1)$$

La clé courante z_k est générée par un système dynamique

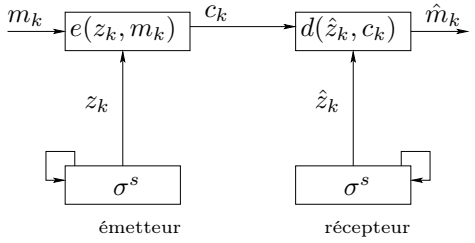

Fig. 2. Schéma de chiffrement/déchiffrement par flot synchrone

autonome caractérisé par une fonction σ^s. Afin de pouvoir récupérer correctement le message clair m_k, les générateurs de clé côté émetteur et récepteur doivent être parfaitement synchronisés. Etant donné qu'il n'y a aucun couplage entre eux, la seule possibilité pour garantir leur synchronisation est de les initialiser à la même valeur $z_0 = \hat{z}_0$. Ainsi, la condition initiale des générateurs $z_0 = \hat{z}_0$ constitue la clé secrète θ.

B. Chiffreurs par flot autosynchrones

Un chiffreur par flot autosynchrone obéit aux équations (2), le schéma-bloc associé est représenté sur la Figure 3.

$$\begin{cases} z_k = \sigma_\theta^{ss}(c_{k-l}, \ldots, c_{k-l-M}) \\ c_k = e(z_k, m_k) \end{cases} \quad (2)$$

Dans ce cas, le générateur de clé n'est pas un système dynamique autonome. Il est caractérisé par une fonction statique σ_θ^{ss} dépendant d'un nombre fini de textes chiffrés précédents c_{k-i} et paramétrée par θ qui joue le rôle de clé secrète. La propriété essentielle de ce type de chiffrement est

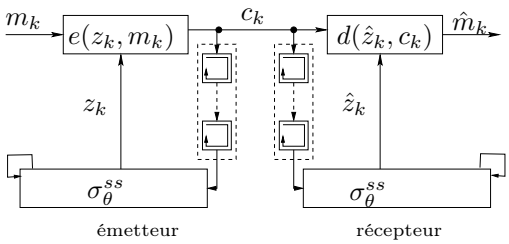

Fig. 3. Schéma de chiffrement/déchiffrement par flot autosynchrone

la capacité d'auto synchronisation des générateurs de clé. En effet, après un nombre fini d'itérations correspondant au nombre de chiffrés c_{k-i} dont dépend σ_θ^{ss}, les séquences des clés z_k et $\hat{z}_k$ sont identiques puisque σ_θ^{ss} dépend exactement des mêmes arguments. Ce schéma de chiffrement est intéressant pour ses propriétés d'auto-synchronisation. En effet, aucun protocole ou trame de resynchronisation n'est nécessaire en cas de désynchronisation impromptue. On y a recours dans les situations où on doit faire face à de fortes contraintes en terme de débit.

Les équations (2) ne sont en réalité que des formes canoniques, en particulier la fonction σ_θ^{ss}. Le générateur de clé peut être en fait un système dynamique mais aux propriétés telles qu'une réécriture via une fonction statique σ_θ^{ss} est possible comme cela est illustré au travers des deux exemples qui suivent.

C. Exemples de chiffreurs autosynchronisants

C.1 SSS

L'algorithme SSS a été proposé dans [3]. L'architecture générale est représentée sur le schéma-bloc de la Figure 4. Les opérations suivantes seront utilisées pour décrire l'algorithme.

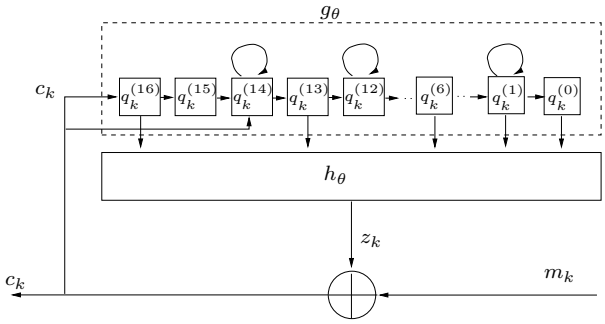

Fig. 4. Le schéma bloc de SSS côté émetteur

- $x >>> n$ est une rotation de n bits à droite du mot x
- $S_\theta(x) = SBOX_\theta(x_H) \oplus x$ avec x_H l'octet de poids fort de x est le OU exclusif entre x et le résultat de la fonction $SBOX_\theta$, combinaison de deux tables de substitution appelées Skipjack S-box et Q-box [3] paramétrées par la clé secrète θ

Le générateur de clé est un système dynamique caractérisé par une fonction de transition g_θ et de sortie h_θ.

$$\begin{cases} q_{k+1} &= g_\theta(q_k, c_k) \\ z_k &= h_\theta(q_k) \end{cases} \quad (3)$$

Le vecteur d'état q_k est de dimension $n = 17$ correspondant au nombre de registres à décalage (d'une longueur de 16

bits) auxquels sont associées chacune des composantes $q_k^{(j)}$. Les dynamiques notées g_θ^j des $q_k^{(j)}$ sont indépendantes et décrites par :

$$\begin{array}{lcl} q_{k+1}^{(16)} & = & c_k \\ q_{k+1}^{(j)} & = & q_k^{(j+1)} \ (j = 0, 2, \ldots, 11, 13, 15) \\ q_{k+1}^{(14)} & = & q_k^{(15)} + S_\theta(c_k >>> 8) \\ q_{k+1}^{(12)} & = & S_\theta(q_k^{(13)}) \\ q_{k+1}^{(1)} & = & q_k^{(2)} >>> 8 \end{array} \quad (4)$$

L'état initial du registre 16 vérifiant $q_1^{(16)} = c_0$, il est facile de voir qu'au bout de 16 itérations, l'état interne q_k ne dépend plus que des 16 valeurs précédentes du message chiffré c_k. Il existe donc pour tout $k \geq 16$ une fonction l_θ vérifiant

$$q_k = l_\theta(c_{k-16}, \ldots, c_{k-1}) \quad (5)$$

La fonction de sortie h_θ délivrant le flot de clés courantes z_k est définie par :

$$z_k = h_\theta(q_k) = A_\theta >>> 8 \oplus q_k^{(0)} \quad (6)$$

avec $A_\theta = S_\theta\big(S_\theta(q_k^{(0)} + q_k^{(16)}) + q_k^{(1)} + q_k^{(6)} + q_k^{(13)}\big)$
Finalement, en combinant l'équation (6) et (5), le générateur de clé peut être réécrit sous la forme canonique (2) :

$$\begin{array}{ll} z_k & = h_\theta(l_\theta(c_{k-16}, .., c_{k-1})) \\ & = \sigma_\theta^{ss}(c_{k-16}, .., c_{k-1}) \end{array} \quad (7)$$

La fonction de chiffrement e est un ou exclusif entre le message clair m_k et la clé courante z_k :

$$c_k = z_k \oplus m_k \quad (8)$$

La fonction de déchiffrement d est donc également un ou exclusif entre le message chiffré c_k et la clé courante $\hat{z}_k$

$$\hat{m}_k = c_k \oplus \hat{z}_k = m_k \text{ si } \hat{z}_k = z_k \quad (9)$$

C.2 Moustique

Le chiffreur autosynchrone Moustique [4] est la dernière version de l'algorithme original KNOT [5]. C'est un algorithme de chiffrement binaire. Son principe est dérivé de la méthode de synthèse proposé pour la première fois par [6] qui consistait à remplacer les registres à décalage des générateurs de clés par une association (série, parallèle) d'automates à mémoire finie.
Pour Moustique, le générateur de clé est un système dynamique caractérisé par une fonction de transition g_θ et fonction de sortie h.

$$\left\{ \begin{array}{lcl} q_{k+1} & = & g_\theta(q_k, c_k) \\ z_k & = & h(q_k) \end{array} \right. \quad (10)$$

Le vecteur d'état q_k est de dimension $n = 96$. Moustique se différencie de SSS à la fois au niveau de la fonction de transition et de la fonction de sortie.
Concernant la fonction de transition g_θ, chaque composante $q_k^{(j)} \in \{0, 1\}$ obéit à une dynamique g_θ^j de la forme :

$$q_{k+1}^{(j)} = g_\theta^j(q_k^{(j-1)}, q_k^{(j-2)}, ..., q_k^{(1)}, c_k) \ j = 1, \ldots, n \quad (11)$$

Ainsi, la $j^{ème}$ composante de q_{k+1} ne dépend plus d'une seule composante de q_k (comme c'est le cas avec les registres à décalage dans SSS), mais dépend de plusieurs composantes de q_k, plus précisément des composantes $q_k^{(l)}$ avec $l < j$. Ainsi, la fonction g_θ est « triangulaire » ; cela garantit que q_k ne dépend plus de la condition initiale q_0 après n itérations. De même que pour SSS, il existe donc une fonction l_θ qui permet de réécrire (11) sous une forme strictement équivalente pour $k \geq n$ et ne dépend plus que des messages chiffrés c_{k-i} précédents

$$q_k = l_\theta(c_{k-n}, ..., c_{k-1}) \quad (12)$$

A la différence de SSS, la fonction de sortie de Moustique n'est pas une simple fonction statique. Elle est en réalité implémentée sous la forme d'un « pipeline » (Fig. 5). Le calcul de la clé courante s'opère de façon séquentielle en impliquant des étages successifs. Le « pipeline » de Moustique comprend $b_s = 9$ étages. Chaque étage correspond à une fonction spécifique s_i $(i = 0, \ldots, b_s - 1)$ dont l'argument d'entrée est le résultat de la sortie de l'étage précédent. L'argument d'entrée de la fonction s_0 est q_k et on a $s_0(q_k) = q_k$. Notons qu'aucune fonction s_i ne dépend de la clé secrète θ. La fonction de sortie h (qui n'est donc pas, à la différence de SSS, paramétére par θ) s'écrit comme une composition de b_s fonctions et la clé courante calculée à partir d'un état q_k n'est disponible qu'à l'instant $k + b_s$:

$$z_{k+b_s} = s_8(s_7(...(s_0(q_k)))) = h(q_k) \quad (13)$$

En combinant (12) et (13), on obtient la fonction σ_θ^{ss}

$$\begin{array}{ll} z_{k+b_s} & = h(l_\theta(c_{k-n}, ..., c_{k-1})) \\ & = \sigma_\theta^{ss}(c_{k-n}, ..., c_{k-1}) \end{array} \quad (14)$$

On comprend alors l'intérêt matériel du pipeline : une augmentation de la complexité de la fonction de sortie mais qui nécessite un seul cycle d'horloge pour générer la clé dynamique, chaque fonction s_i étant calculée en parallèle. Le prix à payer est un délai b_s entre le message clair et le message chiffré correspondant. La question du temps de calcul est abordée dans [7][4].
La fonction de chiffrement e est un ou exclusif entre le message clair m_k et la clé courante z_{k+b_s} :

$$c_{k+b_s} = z_{k+b_s} \oplus m_k \quad (15)$$

La fonction de déchiffrement d est donc également un ou exclusif entre le message chiffré c_{k+b_s} et la clé courante $\hat{z}_k$ et vérifie

$$\hat{m}_{k+b_s} = c_{k+b_s} \oplus \hat{z}_k = m_k \text{ si } \hat{z}_k = z_{k+b_s} \quad (16)$$

III. Procédés de "brouillage" et comparaison structurelle

On appelle procédés de "brouillage" les algorithmes qui ont été proposés dans la littérature depuis les années 90 basés sur l'utilisation des dynamiques complexes dans le contexte de la sécurité des communications. Un description exhaustive est donnée dans [1]. On ne retient ici que les deux principaux pour notre étude comparative.

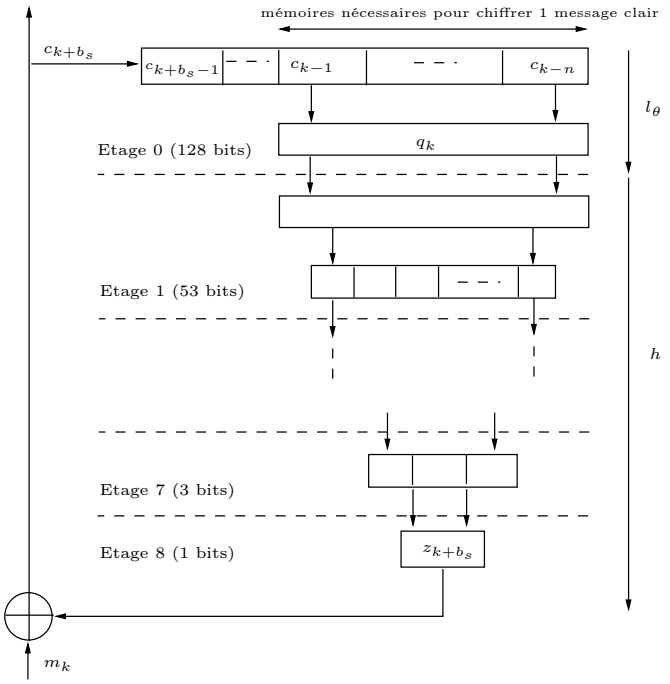

Fig. 5. Le schéma bloc de Moustique côté récepteur

A. Masquage additif

Le masquage additif a été le premier schéma de "brouillage" proposé en 1993 [8][9]. Ses équations de fonctionnement et le schéma-bloc associé sont donnés respectivement par (17) et la Figure 6.

$$\begin{cases} x_{k+1} &= f_\theta(x_k) \\ y_k &= h_\theta(x_k) + m_k \end{cases} \tag{17}$$

Le système dynamique exhibant une dynamique complexe

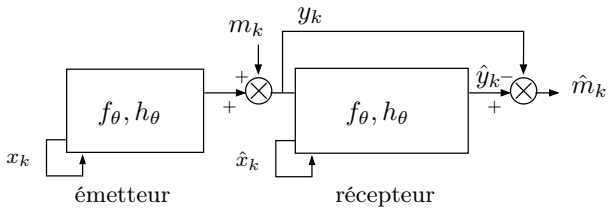

Fig. 6. Masquage additif

est caractérisé par son vecteur d'état $x_k \in \mathbb{R}^n$, sa nonlinéarité $f : \mathbb{R}^n \mapsto \mathbb{R}^n$ et sa fonction de sortie $h : \mathbb{R}^n \mapsto \mathbb{R}$. Les fonctions f et h sont paramétrées par θ qui constitue la clé secrète.

Le principe du "brouillage" est basé sur l'addition entre le signal de sortie $h_\theta(x_k)$ du système dynamique émetteur et le message clair m_k. Le résultat noté y_k constitue le message chiffré. La récupération du message clair nécessite un système dynamique identique côté récepteur capable de se synchroniser parfaitement avec le système émetteur afin que les deux vecteurs d'état respectifs x_k et $\hat{x}_k$ soient égaux. Le message clair est alors reconstitué par opération de soustraction $\hat{m}_k = y_k - \hat{y}_k$.

Dans le schéma proposé par [8][9], une synchronisation parfaite des séquences x_k et $\hat{x}_k$ est attendue sans que les systèmes dynamiques soient initialisés à la même valeur. Cependant l'ajout de m_k au niveau de la sortie côté émetteur constitue une perturbation agissant sur le canal de transmission et celle-ci ne peut en général pas être totalement rejetée. La synchronisation des vecteurs d'état qui jouent en réalité le rôle de clé courante ne peut donc pas être parfaitement garantie. Ce schéma apparaît comme une version dégradée du chiffrement par flot synchrone et ne présente donc aucun intérêt.

B. Inclusion

La technique par inclusion obéit aux équations (18) et le schéma-bloc associé est représenté sur la Figure 7.

$$\begin{cases} x_{k+1} = f_\theta(x_k, m_k) \\ y_k = h_\theta(x_k, [m_k]) \end{cases} \tag{18}$$

Cette technique consiste à injecter le message clair m_k

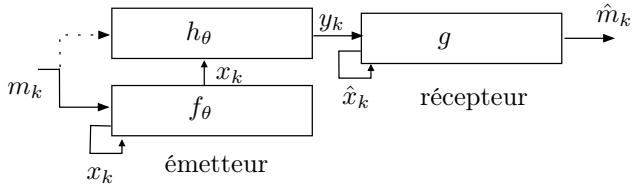

Fig. 7. Schéma de "brouillage" par inclusion

dans la dynamique du système. La sortie y_k du système dynamique constitue le message chiffré. Notons que m_k peut également être injecté au niveau de la sortie. Cette éventualité est caractérisée par la présence des crochets [.] dans l'équation de sortie de (18). La récupération du message clair m_k nécessite une inversion dynamique g côté récepteur. Cette inversion, dans la théorie du contrôle, est appelée "inversion à gauche". De la même manière qu'un lien a été établi entre le masquage additif et le chiffrement par flot synchrone, la recherche d'un lien éventuel entre la technique par inclusion et le chiffrement par flot s'impose.

C. Une nouvelle approche pour la synthèse de chiffreurs autosynchrones

Trois propriétés caractérisant un système dynamique jouent un rôle fondamental en vue d'établir une connexion entre les techniques de "brouillage" et les algorithmes de chiffrement par flot. Elle sont rappelées ci-dessous et peuvent être trouvées dans [10][11].

Definition 1: Le **degré relatif** d'un système dynamique est le nombre minimum r d'itérations de la sortie y_k requises afin que la sortie y_{k+r} à l'instant $k+r$ dépende de manière explicite de l'entrée m_k. Lorsqu'un système possède un degré relatif fini ($r < \infty$), il existe une fonction notée $h^{(r)}$ qui lie la paire (x_k, m_k) à y_{k+r}

$$y_{k+r} = h^{(r)}(x_k, m_k) \tag{19}$$

et qui vérifie l'implication suivante :

$$\exists x_k,\ \exists m_k,\ \exists m'_k \text{ tq } m_k \neq m'_k \Rightarrow h^{(r)}(x_k, m_k) \neq h^{(r)}(x_k, m'_k)$$

Definition 2: Un système dynamique est dit **inversible à gauche** si l'entrée m_k peut être déterminée de manière unique à partir d'un nombre fini de sorties retardées y_{k+i} $(i = 1, \ldots, R)$ pour toute condition intiale x_k.

Pour une introduction à la platitude on peut se référer à [12].

Definition 3: Une sortie y_k d'un système dynamique est dite **plate** si toutes les autres variables du système peuvent être exprimées au travers de fonctions dépendant uniquement des retards ou avances de y_k. En particulier il existe une fonction notée $\mathcal{F}$ et deux entiers relatifs t_1 et t_2 tel que :

$$x_k = \mathcal{F}(y_{k+t_1}, \cdots, y_{k+t_2}) \quad (20)$$

La Proposition suivante est le résultat central de notre étude.

Proposition 1: Le chiffrement par inclusion (18) est équivalent à un chiffrement par flot autosynchrone si le système dynamique (18) est inversible à gauche, plat et possède un degré relatif fini.

Preuve : si (18) est plat, alors (20) est vérifié. S'il possède un degré relatif fini, alors (19) est également vérifié. Enfin, s'il est inversible à gauche, pour tout vecteur d'état x_k, la fonction $h^{(r)}$ est inversible et m_k se déduit donc de manière unique à partir de y_{k+r}. En identifiant (20) et (19) avec les équations (2), on en déduit que le chiffrement par inclusion (18) est équivalent au chiffrement par flot autosynchrone avec :

- un générateur de clé $\mathcal{F}$
- une clé courante x_k
- un message chiffré y_{k+r} correspondant au message clair m_k (le délai correspondant à un pipeline virtuel à r étages)
- une fonction de chiffrement $h^{(r)}$

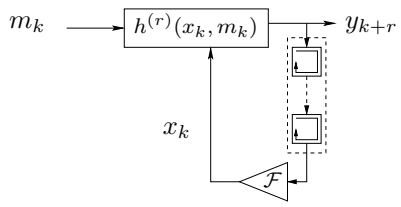

Fig. 8. Schéma équivalent autosynchrone du procédé par inclusion

Il est caractérisé par le schéma-bloc de la Figure 8. La preuve de ce théorème est constructive car elle révèle une nouvelle technique de synthèse de chiffreurs autosynchrones inédite à ce jour. Le recours à des systèmes dynamiques plats constitue en effet une alternative aux techniques existantes : registres à décalage (comme pour SSS) ou fonctions "triangulaires" (comme pour Moustique).

IV. Exemple

Le but de cette section est d'illustrer comment se traduit, pour la classe des systèmes linéaires par morceaux, la démarche générale de conception d'un algorithme de chiffrement par flot autosynchrone décrite au travers de la preuve de la Proposition 1. En particulier, on montre comment tester l'inversibilité à gauche et la platitude. Ces systèmes obéissent à :

$$\left\{ \begin{array}{lcl} x_{k+1} & = & A_{\sigma(k)} x_k + B_{\sigma(k)} m_k \\ y_k & = & C_{\sigma(k)} x_k + D_{\sigma(k)} m_k \end{array} \right. \quad (21)$$

où $x_k \in \mathbb{R}^n$, $m_k \in \mathbb{R}$ et $y_k \in \mathbb{R}$. Toutes les matrices $A_{\sigma(k)} \in \mathbb{R}^{n\times n}$, $B_{\sigma(k)} \in \mathbb{R}^{n\times 1}$, $C_{\sigma(k)} \in \mathbb{R}^{1\times n}$ et $D_{\sigma(k)} \in \mathbb{R}$ appartiennent à des ensembles finis $(A_j)_{1\le j\le J}$, $(B_j)_{1\le j\le J}$, $(C_j)_{1\le j\le J}$ et $(D_j)_{1\le j\le J}$. A chaque instant k, l'index j correspond au "mode" du système qui résulte d'une fonction de commutation $\sigma : k \in \mathbb{N} \mapsto j = \sigma(k) \in \{1, \ldots, J\}$.
On considère l'exemple suivant :

$$A_{\sigma(k)} = \begin{pmatrix} q^1_{\sigma(k)} & 1 \\ 0.5 & 0 \end{pmatrix} \quad B_{\sigma(k)} = \begin{pmatrix} 0 \\ q^2_{\sigma(k)} \end{pmatrix} \quad (22)$$
$$C_{\sigma(k)} = (1\ 0) \qquad D_{\sigma(k)} = 0$$

Degré relatif et inversibilité à gauche
On cherche une condition algébrique d'inversibilité à gauche pour (21). Dans cette perspective, écrivons l'expression de la sortie y_{k+i} en itérant (21)

$$y_{k+i} = C_{\sigma(k+i)} A^{\sigma(k+i-1)}_{\sigma(k)} x_k + \sum_{j=0}^{j=i} \mathcal{T}^{i,j}_{\sigma(k)} m_{k+j} \quad (23)$$

avec

$$\begin{array}{l} \mathcal{T}^{i,j}_{\sigma(k)} = C_{\sigma(k+i)} A^{\sigma(k+i-1)}_{\sigma(k+j+1)} B_{\sigma(k+j)} \text{ si } j \le i-1 \\ \mathcal{T}^{i,i}_{\sigma(k)} = D_{\sigma(k+i)} \end{array} \quad (24)$$

et avec la matrice de transition définie par :

$$\begin{array}{rcl} A^{\sigma(k_1)}_{\sigma(k_0)} & = & A_{\sigma(k_1)} A_{\sigma(k_1-1)} \ldots A_{\sigma(k_0)} \text{ si } k_1 \ge k_0 \\ & = & \mathbf{1}_n \text{ si } k_1 < k_0 \end{array}$$

$\mathbf{1}_n$ étant la matrice identité de dimension n.
Il en découle directement le résultat suivant. Le degré relatif r de (21) est

- $r = 0$ si $\mathcal{T}^{0,0}_{\sigma(k)} \neq 0$ pour tout k
- le plus petit entier $r < \infty$ tel que pour tout k

$$\begin{array}{l} \mathcal{T}^{i,j}_{\sigma(k)} = 0 \text{ pour } i = 0, \ldots, r-1 \text{ et } j = 0, \ldots, i \\ \mathcal{T}^{r,0}_{\sigma(k)} \neq 0 \end{array} \quad (25)$$

Si (21) a un degré relatif r, sa sortie au temps $k+r$ s'écrit :

$$y_{k+r} = C_{\sigma(k+r)} A^{\sigma(k+r-1)}_{\sigma(k)} x_k + \mathcal{T}^{r,0}_{\sigma(k)} m_k \quad (26)$$

et il est inversible à gauche puisque pour tout x_k, on peut retrouver m_k de manière unique à partir de y_{k+r}, la matrice $\mathcal{T}^{r,0}_{\sigma(k)}$ étant toujours inversible, (25) étant vérifié.
Application
Le degré relatif de (21) pour les matrices particulières (22) est $r = 2$ puisque $\mathcal{T}^{i,j}_{\sigma(k)} = 0$ pour $i = 0, 1$ et $j = 0, \ldots, i$, $\mathcal{T}^{2,0}_{\sigma(k)} = C_{\sigma(k+2)} A_{\sigma(k+1)} B_{\sigma(k)} \neq 0$ pour tout k. Il vient :

$$y_{k+2} = q^1_{\sigma(k+1)} y_{k+1} + 0.5 y_k + q^2_{\sigma(k)} m_k \quad (27)$$

Platitude
Pour tester si le système est plat, on cherche à exprimer chaque composante $x^{(i)}_k$ $(i = 1, 2)$ de l'état x_k en fonction uniquement de la sortie et éventuellement de ses itérés. La méthode proposée est basée sur une technique d'élimination. Il existe de nombreux algorithmes d'élimination (cf. [13] pour une étude comparative), notamment ceux dérivés de la théorie des résultants, des ensembles caractéristiques ou des bases de Gröbner. Mis à part certains cas particuliers, il est impossible d'affirmer qu'une méthode et meilleure qu'une autre. Nous avons choisi ici

d'utiliser le logiciel libre Maxima (disponible à l'adresse http ://maxima.sourceforge.net) qui est basé sur la théorie des résultants [13]. Trois étapes sont nécessaires pout tester la platitude :

- définir les équations d'état
- écrire les itérés successifs des équations d'état
- spécifier les variables à éliminer

Pour la composante $x_k^{(1)}$, la réponse est évidente et ne nécessite aucun calcul car on a :

$$x_k^{(1)} = y_k \tag{28}$$

Afin de comprendre le script Maxima correspondant à la recherche de l'expression de $x_k^{(2)}$, rappelons quelques notations utilisées. Toute variable $x_k^{(i)}$ (*resp.* y_k et m_k) est notée `xik` (*resp.* `yk` et `mk`), les itérés de rang l sont notés `xikl` (*resp.* `ykl` et `mkl`). Une équation doit toujours être labellée. Le label est de la forme `eqj` où `j` correspond à la j^{ieme} équation. Les composantes non constantes des matrices $A_{\sigma(k)}$ et $B_{\sigma(k)}$, à savoir $q_\sigma^1(k)$ et $q_{\sigma(k)}^2$ sont notées `q1k` et `q2k`; les itérés de rang l à savoir $q_{\sigma(k+l)}^1$ et $q_{\sigma(k+l)}^2$ notés `q1kl` et `q2kl`. Enfin, une équation est toujours comprise comme une relation égalant zéro. Ces quelques commentaires permettent de comprendre les scripts ci-dessous.
On spécifie tout d'abord les équations du système :

```
(%i1) eq1:x1k1-q1k*x1k-x2k;
(%i2) eq2:x2k2-0.5-q2k*mk;
(%i3) eq3:yk-x1k;
```

On écrit alors les équations d'état itérées.

```
(%i4) eq4:x1k2-q1k1*x1k1-x2k1
(%i5) eq5:x2k2-0.5-q2k1*mk1;
(%i6) eq6:yk1-x1k1;
(%i7) eq7:y1k2-x1k2;
```

Pour obtenir l'expression (si elle existe) de la composante $x_k^{(2)}$ de x_k en fonction uniquement de la sortie et éventuellement de ses itérés, on fait appel à la fonction `eliminate` de Maxima qui utilise la technique des résultants. On spécifie la liste des équations à utiliser et des variables à éliminer à savoir toutes les variables sauf `x1k` (qui n'est rien d'autre que `yk`) et `x2k`.

```
eliminate([eq1,eq2,eq3,eq4,eq5,eq6,eq7],
[mk,mk1,x1k1,x1k2,x2k1,x2k2]);
```

On obtient

$$x_k^{(2)} = y_{k+1} - q_{\sigma(k)}^1 y_k \tag{29}$$

D'après (28) et (29), les deux composantes $x_k^{(i)}$ $(i = 1, 2)$ du vecteur d'état ne dépendent pas de l'entrée mais uniquement de la sortie et de ses itérés donc le système est plat.
Conclusion : le système (21) possède un degré relatif fini $r = 2$, il est inversible à gauche et plat. Conformément à la Proposition 1, (21) est équivalent à un chiffreur autosynchrone avec :

- un générateur de clé où $\mathcal{F}$ est donnée par les équations (28) et (29)
- une clé courante x_k
- un message chiffré y_{k+2} correspondant au message clair m_k (le délai correspondant à un pipeline virtuel à 2 étages)
- une fonction de chiffrement $h^{(2)}$ donnée par (27)

V. Considérations cryptanalytiques et conclusion

Dans cet article une étude comparative entre les techniques de "brouillage" de l'information basées sur l'utilisation de dynamiques complexes et les algorithmes standards de chiffrement symétrique a été menée. On a montré et illustré comment les propriétés structurelles des systèmes dynamiques, à savoir l'inversibilité, la platitude jouent un rôle fondamental pour la synthèse de chiffreurs symétriques autosynchrones.
Une issue majeure pour évaluer la sécurité d'un algorithme de chiffrement est la complexité de recouvrement de la clé secrète. Tout algorithme de chiffrement doit pouvoir faire face au moins à l'attaque la plus basique, à savoir l'attaque gloutonne (ou recherche exhaustive). Cette attaque consiste à balayer toutes les valeurs possibles de clés appartenant à un ensemble de cardinalité fini (espace des clés). Le pirate est supposé disposer d'un grand nombre de paires message clair - message chiffré correspondant. Cette hypothèse est vérifiée dans une attaque à texte clair choisi. Afin que l'espace des clés ne puisse être réduit par cette approche, il ne faut pas que pour une même paire de messages clair - chiffré, plusieurs valeurs de clés conviennent. Si tel est le cas, des valeurs de clés sont redondantes et le cryptosystème peut être plus rapidement cassé par attaque gloutonne. Or, l'unicité de la clé correspond à la propriété d'identifiablité des paramètres θ du système dynamique. Ainsi en arrive-t-on à la conclusion essentielle non triviale car au premier abord paradoxale qu'une condition minimale pour la sécurité d'un cryptosystème est que les paramètres θ qui sont supposés jouer le rôle de clé secrète sont identifiables. La sécurité repose sur la complexité d'identification, tâche qui correspond dans un contexte cryptanalytique à une attaque algébrique. La recherche des conditions d'identifiabilité et des processus d'identifications pertinents pour le contexte cryptographique constituera la suite de ces travaux.

Références

[1] G. Millérioux, J. M. Amigó et J. Daafouz : A connection between chaotic and conventional cryptography. IEEE Trans. on Circuits and Systems I : Regular Papers, 55(6), July 2008.

[2] A. J. Menezes, P. C. Oorschot et S. A. Vanstone : Handbook of Applied Cryptography. CRC Press, October 1996.

[3] P. Hawkes, M. Paddon, G. G. Rose et W. V. Miriam : Primitive specification for sss. Rapport technique, e-Stream Project, 2004. Available at : http ://www.ecrypt.eu.org/stream/ciphers/sss/sss.pdf.

[4] J. Daemen et K. Paris : The self-synchronzing stream cipher moustique. Rapport technique, e-Stream Project, 2006. Available at : http ://www.ecrypt.eu.org/stream/p3ciphers/mosquito/mosquito_p3.pdf.

[5] J. Daemen, R. Govaerts et J. Vandewalle : On the design of high speed self-synchronizing stream ciphers. In Proc. of the ICCS/ISITA'92 conference, volume 1, pages 279–283, Singapore, November 1992.

[6] U. M. Maurer : New approaches to the design of self-synchronizing stream cipher. Advance in Cryptography, In Proc. Eurocrypt '91, Lecture Notes in Computer Science, pages 548–471, 1991.

[7] J. Daemen et Paris K. : The self-synchronizing stream cipher mosquito : estream documentation, version 2. Rapport technique, e-Stream Project, 2005. Available at :www.ecrypt.eu.org/stream/p3ciphers/mosquito/mosquito.pdf.

[8] Wu C. W. et Chua L. O. : A simple way to synchronize chaotic systems with applications to secure communications systems. International Journal of Bifurcation and Chaos, 3(6):1619–1627, 1993.

[9] Cuomo K. M., Oppenheim A. V. et Strogatz S. H. : Synchronization of lorenz-based chaotic circuits with applications to communications. IEEE Trans. Circuits. Syst. II : Anal. Digit. Sign. Process, 40(10):626–633, 1993.

[10] A. Isidori : Nonlinear control systems. Communications and control engineering series. Springer, 1995.

[11] M. Fliess et R. Marquez : Une approche intrinsèque de la commande prédictive linéaire discrète. Journal Européen des Systèmes Automatisés, 35:127–147, 2001.

[12] M. Fliess, J. Levine, P. Martin et P. Rouchon : Flatness and defect of non-linear systems : introductory theory and examples. Int. Jour. of Control, 61(6):1327–1361, 1995.

[13] D. Wang : Elimination theory, methods and practice. Mathematics and Mathematics-Mechanization, pages 91–137, 1991. available at http ://www-calfor.lip6.fr/ wang/.