Un nouvel algorithme de chiffrement

Joan DAEMEN et Vincent RIJMEN

Après vingt ans de loyaux services, le DES, l'algorithme de chiffrement à clé secrète le plus utilisé au monde, va céder sa place. L'administration américaine a choisi son successeur : c'est l'AES, que décrivent ses deux concepteurs belges.

En cet été 1998, un frisson parcourt l'échine des responsables de l'administration américaine : l'algorithme DES (*Data Encryption System* soit «système de chiffrement de données»), adopté en 1977 pour le chiffrement des informations non classifiées, vient d'être décrypté. L'attaque n'a nécessité que très peu de moyens : avec une somme dérisoire pour un État, pour une entreprise ou pour une organisation mafieuse – moins de 250 000 $ –, une simple association de particuliers a fait construire un processeur spécialisé, lancé avec succès à l'attaque du DES. La situation était d'autant plus préoccupante que l'usage du DES s'est largement étendu au secteur privé, notamment aux communications bancaires *(voir* La cryptographie à clé secrète *par Jacques Patarin,* dans ce dossier). Que s'est-il passé?

Avec l'amélioration des performances des ordinateurs depuis vingt ans, la taille de la clé secrète du DES, 56 bits, le rend aujourd'hui vulnérable en moins de trois jours aux attaques exhaustives. Il existe bien une version améliorée du DES, le triple-DES, qui double la taille de la clé, mais il n'est pas assez rapide. Il fallait donc trouver un successeur, et le NIST (*National Institute of Standards and Technologies*) a lancé un concours «ouvert» : n'importe qui pouvait se porter candidat et soumettre un code. Le 2 octobre 2000, notre algorithme, le Rijndael, a été choisi et a pris le nom d'AES (*Advanced Encryption Standard* ou standard de chiffrement avancé).

Un algorithme de chiffrement par bloc, «aussi sûr que le triple-DES, mais beaucoup plus performant» : tel était l'objectif fixé par le NIST pour l'AES. Ainsi, les longueurs possibles pour la clé secrète devaient être de 128, 192 et 256 bits – contre 112 bits pour le triple-DES –, et la longueur du bloc de chiffrement égale à 128 bits. Cette flexibilité dans la taille des clés est primordiale pour faire évoluer la sécurité de l'AES avec la puissance des ordinateurs. Une fois ces précautions prises, et écartés les rares cas ne remplissant pas les critères de sécurité mathématiques, la sûreté des algorithmes présentés était difficile à comparer. Restaient les performances, un facteur décisif pour le choix de l'AES.

LES DEUX LAURÉATS, Joan Daemen *(à gauche)* et Vincent Rijmen *(à droite).*

Le cahier des charges

La première d'entre elles est d'assurer à l'algorithme une large portabilité, c'est-à-dire la mise en œuvre facile sur un grand nombre de systèmes informatiques. Au vu du succès qu'a connu le DES, ces derniers risquent d'être variés : des cartes à puces, dotées généralement de processeurs 8-bits disposant de peu de mémoire, jusqu'aux processeurs spécialisés chiffrant et déchiffrant des communications avec des débits de l'ordre du gigabit par seconde, en passant par les ordinateurs de poche, les ordinateurs personnels, les stations de travail, les serveurs, etc.

Le chiffrement devait donc être rapide, sans surcharger l'application qui y fait appel : la taille des programmes, la mémoire nécessaire et la fraction du processeur utilisée ont été considérées. En particulier la

LE FONCTIONNEMENT DE L'AES

Le Rijndael ou AES est un chiffrement par bloc dont la longueur de bloc et la longueur de clef peuvent varier indépendamment entre 128 et 256 bits, par incréments de 32 bits. Ainsi, cet algorithme sera adaptable sur des supports aux capacités de calculs variés. Dans le cadre de l'AES, la longueur de bloc est fixée à 128 bits, et la longueur de clé à 128 bits pour ces prochaines années. Rien n'empêche toutefois, lorsque le besoin s'en fera sentir, d'utiliser des clés plus longues, de 192 ou 256 bits.

Comment fonctionne le chiffrement? Le message binaire est d'abord tronçonné en blocs de 128 bits. Chaque bloc est disposé dans un tableau d'octets de quatre lignes et de quatre colonnes (un octet représente 8 bits). Le bloc est chiffré par une suite de quatre transformations répétées dix fois: Rijndael est un algorithme à dix tours (ou itération).

D'abord, avant toute transformation, on effectue une addition de clé initiale secrète. En effet, toute transformation appliquée avant cet ajout et qui ne nécessiterait pas la connaissance de la clé, serait un tour inutile, aisément attaquable. Les transformations sont toujours faites dans l'ordre suivant:

1. Transformation non linéaire d'octets *(SubBytes)*: à chacun des seize octets du bloc est appliquée une même transformation non linéaire fixée nommée S. La transformation est appliquée indépendamment à chaque octet. Par exemple, l'octet b_{12} est l'image de l'octet $a_{1,2}$ par cette transformation ($b_{12} = S(a_{12})$).

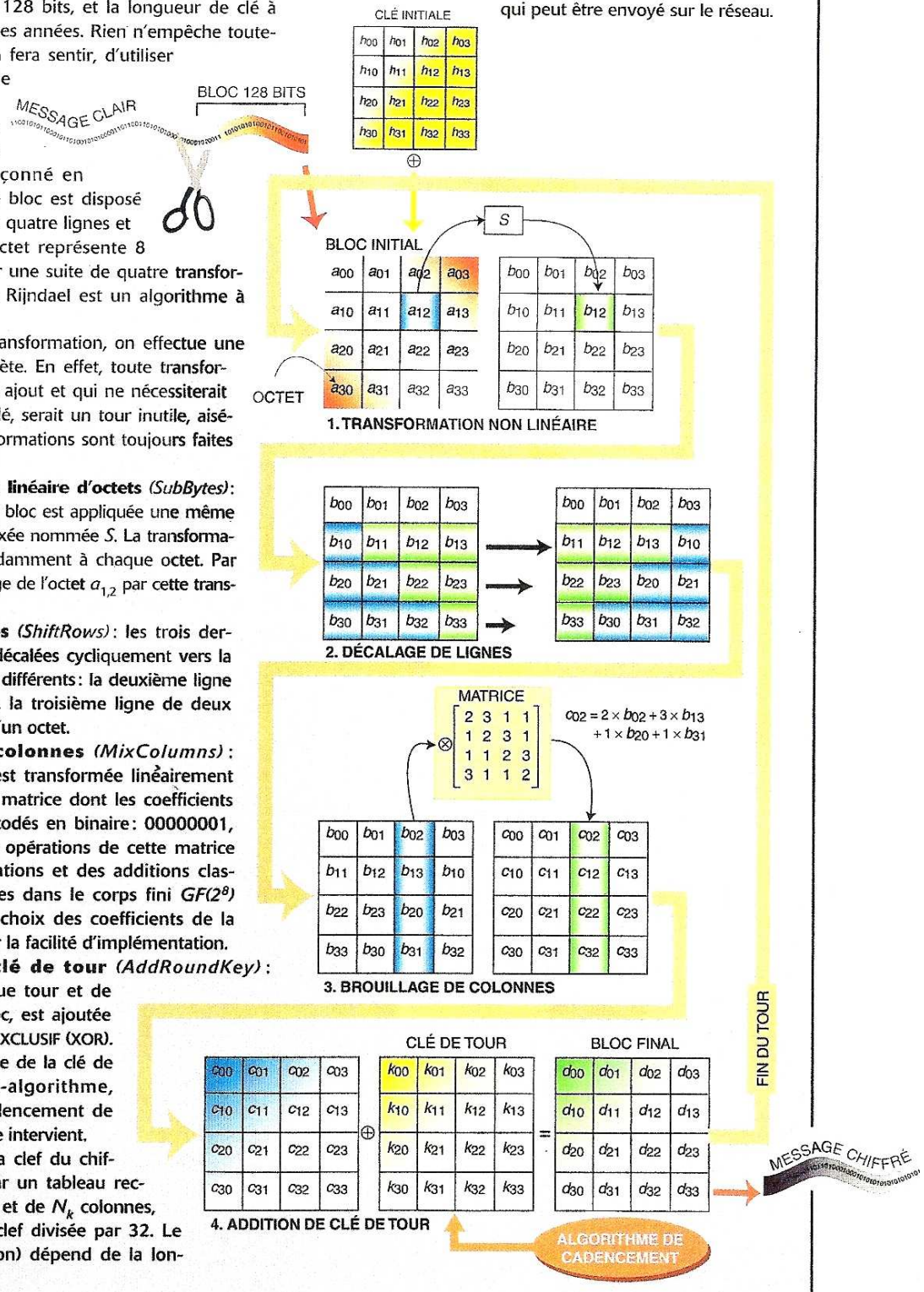
2. Décalage de lignes *(ShiftRows)*: les trois dernières lignes du bloc sont décalées cycliquement vers la gauche, avec des décalages différents: la deuxième ligne est décalée de trois octets, la troisième ligne de deux octets et la première ligne d'un octet.

3. Brouillage des colonnes *(MixColumns)*: Chaque colonne du bloc est transformée linéairement par la multiplication d'une matrice dont les coefficients sont soit 1, soit 2, soit 3, codés en binaire: 00000001, 00000010, 00000011. Les opérations de cette matrice ne sont pas des multiplications et des additions classiques, mais sont effectuées dans le corps fini $GF(2^8)$ *(voir l'autre encadré)*. Le choix des coefficients de la matrice a été déterminé par la facilité d'implémentation.

4. Addition de la clé de tour *(AddRoundKey)*: une clé, différente à chaque tour et de même longueur que le bloc, est ajoutée bit à bit au bloc par un OU EXCLUSIF (XOR). Cette clé de tour est dérivée de la clé de chiffrement par un sous-algorithme, nommé algorithme de cadencement de clé. C'est là que la clé secrète intervient.

De manière générale, la clef du chiffrement est représentée par un tableau rectangulaire de quatre lignes et de N_k colonnes, où N_k est la longueur de clef divisée par 32. Le nombre de tours (d'itération) dépend de la longueur de la clef, il vaut respectivement 10, 12 ou 14 pour 128, 192, ou 256 bits.

Chaque bloc qui a ainsi subit dix tours est retranscrit en nombres binaires. En mettant bout à bout les blocs chiffrés de 128 bits, on obtient le message chiffré qui peut être envoyé sur le réseau.



UNE ADDITION ET UNE MULTIPLICATION PARTICULIÈRES

L'AES est un chiffrement par bloc défini en termes d'opérations sur les octets. Il existe de nombreuses manières de combiner deux valeurs d'octet pour en donner une troisième : ce sont les lois de composition. Bien qu'elles soient généralement notées par des symboles courants comme «+», «-», «×», etc., leur définition peut être très éloignée de leur signification habituelle. C'est le cas dans l'AES, où les opérations se font dans un corps fini.

Qu'est-ce qu'un corps fini ?

Un corps est une structure algébrique plus large qu'un groupe. Un groupe est un ensemble muni d'une loi de composition, ayant trois propriétés simples : associativité, existence d'un élément neutre (0 pour l'addition d'entiers, 1 pour la multiplication), existence pour chaque élément de son symétrique (l'opposé pour l'addition, l'inverse pour la multiplication).

Un corps $<S, +, \times>$ est formé d'un ensemble S d'éléments et de deux lois de composition : l'«addition» et la «multiplication». Les lois d'addition et de multiplication forment chacune un groupe avec S (en excluant de S, pour la multiplication, l'élément neutre de l'addition), et la multiplication est distributive par rapport à l'addition : $a \times (b + c) = (a \times b) + (b \times c)$. Si le nombre d'éléments de S est fini, on parle de *corps fini*, ou corps de Galois noté GF (*Galois Field* en anglais), d'après le nom du mathématicien Évariste Galois.

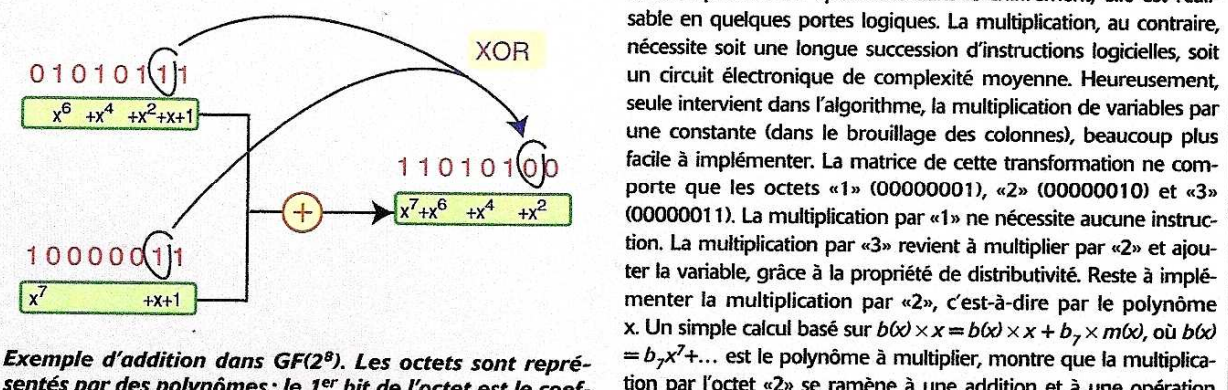
Le corps fini $GF(2^8)$

L'opération de brouillage des colonnes de l'algorithme, (3ème transformation de l'algorithme), effectue ses opérations dans le corps fini $GF(2^8)$, dont les caractéristiques sont les suivantes.

L'*ensemble S* contient les 256 valeurs que peut prendre un octet (2^8). Sa représentation est particulière : elle est polynomiale de degré maximal égal à 7, avec des coefficients binaires (0 ou 1).

La *loi d'addition* dans $GF(2^8)$ correspond à l'addition des polynômes, où les coefficients correspondants de même degré sont additionnés *modulo* 2. Ceci revient à effectuer un OU exclusif (XOR) bit à bit entre les deux octets : il donne 1 si les bits sont différents et 0 sinon. C'est en fait pour la multiplication que la représentation polynomiale prend tout son sens

La *loi de multiplication* dans $GF(2^8)$ correspond à la multiplication des polynômes *modulo* un polynôme irréductible $m(x)$, choisi dans l'AES pour sa simplicité. La division avec reste des polynômes se définit de la même manière que la division entre les entiers. De même, l'équivalent d'un nombre premier est un polynôme irréductible, qui n'est divisible que par 1 et par lui-même.



Exemple de multiplication dans $GF(2^8)$.

Exemple d'addition dans $GF(2^8)$. Les octets sont représentés par des polynômes : le 1er bit de l'octet est le coefficient du terme en x^7, le 2ème bit du terme en x^6, ... et le dernier bit est le terme constant.

D'autres lois, d'autres représentations que polynomiales, d'autres polynômes irréductibles étaient possibles. Toutefois, comme les corps finis de même nombre d'éléments sont isomorphes, c'est-à-dire qu'ils ont exactement la même structure logique, cela n'aurait rien changé aux transformations effectivement subies par les octets. Nous avons privilégié dans l'AES les choix qui facilitaient l'implémentation.

Implémentation des opérations

L'addition dans $GF(2^8)$ est implémentée avec l'opération OU EXCLUSIF (XOR), disponible comme instruction élémentaire sur la plupart des processeurs. Si ce n'est pas le cas, notamment pour certains processeurs spécialisés dans le chiffrement, elle est réalisable en quelques portes logiques. La multiplication, au contraire, nécessite soit une longue succession d'instructions logicielles, soit un circuit électronique de complexité moyenne. Heureusement, seule intervient dans l'algorithme, la multiplication de variables par une constante (dans le brouillage des colonnes), beaucoup plus facile à implémenter. La matrice de cette transformation ne comporte que les octets «1» (00000001), «2» (00000010) et «3» (00000011). La multiplication par «1» ne nécessite aucune instruction. La multiplication par «3» revient à multiplier par «2» et ajouter la variable, grâce à la propriété de distributivité. Reste à implémenter la multiplication par «2», c'est-à-dire par le polynôme x. Un simple calcul basé sur $b(x) \times x = b(x) \times x + b_7 \times m(x)$, où $b(x) = b_7x^7 + \ldots$ est le polynôme à multiplier, montre que la multiplication par l'octet «2» se ramène à une addition et à une opération de décalage des bits dans l'octet. Suivant le processeur, il est parfois préférable de simplement stocker les résultats dans une table.

génération de la clé ne devait pas nécessiter des calculs trop importants, car dans certaines applications – sur Internet par exemple –, elle change très fréquemment. Enfin, l'AES devant être largement diffusé, le NIST avait souhaité que l'algorithme soit simple à comprendre, mais aussi libre de droits d'auteur. Si tous les candidats ont accepté d'abandonner ces droits, il était envisageable que d'autres personnes revendiquent ultérieurement la paternité de certaines portions des algorithmes et réclament des royalties : ce risque de surcoût a donc été étudié pour chacun.

L'évaluation des algorithmes

Quinze équipes spécialisées, venant des États-Unis, de Grande-Bretagne, de Corée, de France, se sont portées candidates, avec des effectifs parfois modestes – moins de cinq personnes –,

ou plus important, comme celles sponsorisées par les sociétés IBM et NTT. Le NIST a testé statistiquement la bonne qualité des algorithmes en tant que générateurs pseudo-aléatoires, condition minimale pour être sélectionné. Il a également comparé les performances des solutions proposées sur un certain nombre d'architectures. Manquant des ressources nécessaires pour effectuer des évaluations de sécurité et de performances plus poussées sur toutes les propositions, le NIST a explicitement demandé l'assistance de la communauté cryptologique. Tous les résultats envoyés au NIST pouvaient être diffusés. Un forum de discussion public était ouvert sur Internet et deux conférences ont été organisées en mars 1999 et en avril 2000. Quelles sont les caractéristiques de l'AES, finalement retenu?

Dans l'AES, il y a quatre transformations successives, qui opèrent directement sur les octets *(voir le premier encadré)*. Ce sont dans l'ordre : une transformation non linéaire d'octet, un décalage de lignes, un brouillage de colonnes et une addition de clé.

L'algorithme ressemble à une préparation – un peu particulière – de pâtes aux épinards. Imaginons que le message soit une longue pâte. On commence par la couper en morceaux de même longueur. Ce sont les blocs. Chacun de ces morceaux est ensuite découpé en seize petits carrés (les octets) que l'on aligne en quatre colonnes. On prend chacun de ces petits carrés, on les pétrit, on les malaxe, puis on les remet à leur place (c'est la «transformation non linéaire»). Ensuite on décale certaines rangées de petits carrés, on en intercale, etc. (le «décalage de lignes»). Puis on prend les carrés de chaque colonne, on les coupe en tout petits morceaux, qu'on mélange pour former une nouvelle colonne, remise en place (le «brouillage de colonnes»). Enfin, on incorpore à la pâte un peu d'épinards (l'«addition de clé»). L'opération est recommencée dix fois de suite, jusqu'à l'obtention d'une pâte verte : le message brouillé.

Le procédé que nous venons de décrire est un exemple d'algorithme de chiffrement par blocs. Dans ce type de chiffrement, le message est fractionné en blocs d'égale longueur, qui seront chiffrés séparément. La plupart des blocs modernes sont conçus à partir de la même structure de base : une transformation simple itérée un certain nombre de fois. Chaque itération est nommée un tour.

Un tour se présente sous la forme d'une succession de transformations élémentaires non-linéaires, les «boîtes de substitution» *(S-boxes)*, qui agissent sur des subdivisions du bloc – les octets dans l'AES. Les sorties de ces boîtes de substitution sont ensuite mélangées, par des permutations de bits par exemple, afin qu'après un certain nombre d'itérations, chaque bit de sortie dépende d'une manière complexe et non linéaire de chaque bit d'entrée. Dans l'AES, toute la non-linéarité est concentrée sur la première des quatre transformations décrites plus haut.

Contrairement à ce qu'on pourrait penser, les opérations de base de l'AES sont assez classiques. Seule une des opérations intervenant dans la définition de l'algorithme Rijndael n'est pas utilisée de longue date en cryptographie, mais l'est néanmoins depuis des décennies dans les codes de correction d'erreurs (techniques utilisées par exemple sur les disques compacts pour ajouter de la redondance aux données qui y sont présentes, afin de détecter et de corriger en temps réel d'éventuelles erreurs). L'itération d'une transformation, même simple, suffit souvent à donner un chiffre solide.

Le compromis sécurité-performance

Pour un cryptographe expérimenté, la conception d'un codage qui satisfasse aux seules conditions de sécurité ne présente en effet pas de réelle difficulté : après un nombre suffisant de tours, la plupart des chiffres sont sûrs. En revanche, effectuer ce nombre de tours nécessaires prend beaucoup de temps. C'est donc quand on prend en compte les performances et les restrictions pratiques imposées par chaque utilisation particulière, que la tâche devient vraiment ardue. Quand nous avons élaboré le Rijndael, nous avons constamment gardé à l'esprit la performance lors de la mise au point du tour.

L'un de nos objectifs principaux lors de la conception de l'AES, a été de construire un algorithme de structure simple. De même, nous avons fait aussi peu d'hypothèses que possible concernant l'architecture utilisée. De cette manière, l'implémentation dans chaque architecture informatique est plus facilement optimisée par le programmateur. Par exemple, pour les processeurs 8-bits couramment employés, ainsi que pour d'autres environnements à accès restreint, le choix le meilleur est probablement d'implémenter directement les différentes transformations de base du Rijndael, puisque ce sont des transformations d'octets (8 bits). En revanche, sur des processeurs plus élaborés (32, 64 bits...), ceci conduirait à une utilisation non optimale de la puissance de calcul. Pour ces systèmes, avec une quantité raisonnable de mémoire cache (mémoire tampon à accès rapide, où le microprocesseur stocke les données auxquelles il fait le plus souvent appel), une amélioration significative des performances est obtenue en stockant, dans des tables, les résultats des quatre transformations élémentaires – exceptée l'addition de clé – combinées entre elles. Par ailleurs, nous nous sommes attachés à ce que les quatre transformations de base de chaque tour agissent le plus possible en parallèle, soit sur les lignes ou les colonnes, soit directement sur les seize octets. Si les transformations ont été combinées, il est également possible d'effectuer la consultation des tables en parallèle pour optimiser l'occupation des processeurs.

Notre algorithme est-il sûr ? Contre l'AES, on ne connaît pour l'instant que l'attaque exhaustive, qui essaie toutes les clés possibles. Si la vitesse de calcul des ordinateurs continue de doubler tous les dix-huit mois, comme elle l'a fait ces dernières décennies, même pour la plus petite longueur de clef (128 bits), aucun ordinateur au monde n'aura une puissance de calcul suffisante avant 70 ans. En revanche, si le nombre de tours est inférieur à celui recommandé – dix – des attaques sont possibles : il en existe une pour sept tours. Cette attaque suppose tout de même d'engendrer et d'envoyer 10^{38} blocs de texte vers l'algorithme pour y être chiffrés, et de réaliser un nombre de calculs équivalent à 10^{36} chiffrements, ce qui restera hors de portée des ordinateurs pendant plusieurs décennies.

Joan DAEMEN travaille pour la société *Proton World International* et Vincent RIJMEN pour la société *Cryptomathic*. Ils ont tous les deux effectué leur thèse dans le laboratoire COSIC (*Computer Security and Industrial Cryptography*) de l'Université de Louvain (Belgique).

J. DAEMEN, V. RIJMEN. *The design of Rijndael, AES – The Advanced Encryption Standard*. Springer-Verlag, 2002.