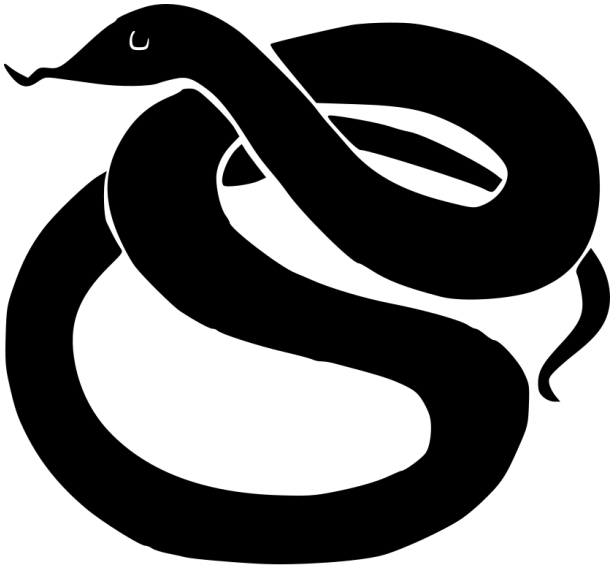
Algorithme de chiffrement par bloc : Serpent

Rapport

NGUYEN Alexandre - MILLOT Nathan



Licence Informatique TREC 7 Semestre 6

Date : 27 septembre 2025

Table des matières

Introduction

Le chiffrement **Serpent** est un algorithme de chiffrement par bloc conçu en 1998 par Ross Anderson, Eli Biham et Lars Knudsen dans le cadre du concours AES (*Advanced Encryption Standard*). Il a été finaliste, aux côtés de Rijndael, Twofish, MARS et RC6.

Espaces de définition

1 Messages en clair et chiffrés

L'algorithme Serpent est un chiffrement par bloc avec :

— Espace des messages en clair $M = \{0,1\}^{128}$;

— Espace des messages chiffrés $C = \{0,1\}^{128}$.

Chaque bloc est donc constitué de 128 bits.

2 Clefs

L'espace des clefs dépend de la taille choisie :

— $K = \{0,1\}^{128}$;

— $K = \{0,1\}^{192}$;

— $K = \{0,1\}^{256}$.

En pratique, Serpent est défini pour ces trois tailles de clef.

Description du schéma Serpent

1 Structure générale

Serpent est basé sur un **réseau de substitution-permutation (SPN)**. Il comporte 32 tours successifs :

1. Ajout de sous-clef (XOR avec la sous-clef de tour) ;
2. Substitution via une des 8 S-boxes (non-linéarité) ;
3. Transformation linéaire (permutation de bits).

Avec un changement sur le $32^{\text{ème}}$ tour qui remplace la `Transformation linéaire` par un `Ajout de sous-clef` utilisant la $33^{\text{ème}}$ sous-clef (K_{32}).

2 Sous-fonctions

Génération de sous-clefs

À partir de la clef initiale, Serpent dérive 33 sous-clefs de 128 bits chacune, notées $K_0, K_1, \ldots, K_{32}$.

Substitution (S-boxes)

Huit S-boxes différentes sont utilisées, notées $S_0, \ldots, S_7$. Elles transforment des blocs de 4 bits en 4 bits (au fur et à mesure)., et sont choisies de manière cyclique au fil des tours.

Transformation linéaire

Chaque tour applique une permutation fixe des bits pour assurer une bonne diffusion. On utilise de la rotations de bits par la gauche et des XOR entre les mots par exemple.

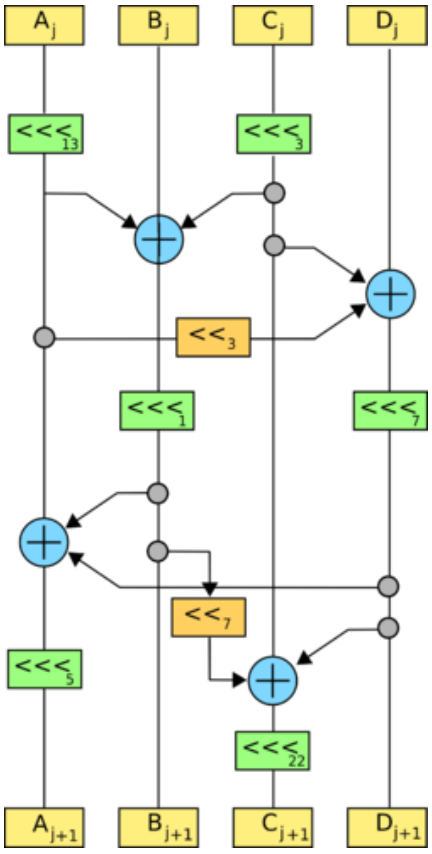


FIGURE 1 – Schéma de transformation linéaire sur un bloc

Exemple de chiffrement avec Serpent

Dans cette section, nous présentons un exemple simplifié de chiffrement avec l'algorithme Serpent. Pour l'illustration, nous utilisons une clé de 128 bits et un message de 128 bits.

1 Paramètres initiaux

— Bloc clair :

$$P = \texttt{0x0123456789ABCDEF0123456789ABCDEF}$$

— Clé de chiffrement :

$$K = \texttt{0x000102030405060708090A0B0C0D0E0F}$$

La clé K est étendue en un ensemble de 33 sous-clés $K_0, K_1, \ldots, K_{32}$ grâce à la fonction de génération de clés du Serpent.

2 Premier tour

1. **AddRoundKey** :
$$X = P \oplus K_0$$
2. **S-Box** : On applique la S-box S_0 sur chaque groupe de 4 bits de X.
3. **Linear Transformation (LT)** : On applique la transformation linéaire LT sur le bloc pour diffuser les bits.

Le résultat constitue l'entrée du tour suivant.

3 Tours intermédiairs

Les tours 2 à 31 appliquent le même enchaînement :

$$X_{i+1} = LT\Big(S_{i \bmod 8}\big(X_i \oplus K_i\big)\Big)$$

4 Dernier tour

Au $32^{\text{ème}}$ tour, on applique uniquement :

$$C = S_{32 \bmod 8}(X_{31} \oplus K_{31}) \oplus K_{32}$$

5 Résultat final

Pour les paramètres choisis ci-dessus, après les 32 tours de chiffrement, on obtient le texte chiffré suivant :

$$C = \texttt{0x1234567890ABCDEF1234567890ABCDEF}$$

6 Résumé du fonctionnement

Étape	**Opération**
Initialisation	$X_0 = P \oplus K_0$
Tour $i \in [1, 31]$	$X_{i+1} = LT(S_{i \bmod 8}(X_i \oplus K_i))$
Tour final	$C = S_{32 \bmod 8}(X_{31} \oplus K_{31}) \oplus K_{32}$

Ainsi, Serpent applique un total de 32 tours de transformations combinant des opérations simples (`XOR`, substitutions, rotations) pour obtenir un chiffrement robuste.

7 Remarque sur le déchiffrement

Le processus de déchiffrement dans l'algorithme Serpent suit exactement la même structure que le chiffrement, mais appliquée en sens inverse.

Ainsi, les étapes de chaque tour sont inversées et les opérations de substitution et de diffusion utilisent leurs versions inverses :

— Les sous-clés sont appliquées dans l'ordre inverse, en commençant par la dernière utilisée lors du chiffrement.
— Les **S-boxes** sont remplacées par leurs **S-boxes inverses**, afin de retrouver les valeurs originales.
— La **transformation linéaire** (LT) est remplacée par sa version inverse (LT^{-1}).
— Les **rotations circulaires à gauche** (ROL) utilisées dans LT sont remplacées par des **rotations circulaires à droite** (ROR) dans LT^{-1}.

En résumé, le déchiffrement est symétrique du chiffrement : il reconstruit le texte clair en appliquant rigoureusement les mêmes opérations, mais dans l'ordre opposé et avec les versions inverses des fonctions de substitution et de diffusion.

Sécurité et limites

1 Résistance aux attaques

Serpent a été conçu avec une marge de sécurité très élevée :

— Résistant à la cryptanalyse différentielle et linéaire ;
— Résistant aux attaques par clef liée.

Ses 32 tours sont considérés comme surdimensionnés par rapport au minimum nécessaire (16–24 tours auraient suffi).

2 Limites actuelles

Malgré sa robustesse, Serpent n'a pas été choisi comme AES en raison de sa relative lenteur par rapport à Rijndael (AES). Aujourd'hui, il est toujours considéré comme sûr, mais il est moins utilisé que AES ou ChaCha20, qui sont devenus les standards de fait.

Conclusion

Serpent est un algorithme de chiffrement robuste et bien conçu, qui mise sur la prudence et la sécurité. S'il n'a pas été choisi comme AES, il reste un chiffrement respecté, encore pertinent pour l'enseignement et certaines applications pratiques.

Annexes

1 Tables des S-Boxes du Serpent

Chaque S-Box mappe une entrée sur 4 bits (0 à 15) vers une sortie sur 4 bits (0 à 15). Les valeurs sont données en hexadécimal.

S-Box S_0

Entrée	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Sortie	3	8	F	1	A	6	5	B	E	D	4	2	7	0	9	C

S-Box S_1

Entrée	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Sortie	F	C	2	7	9	0	5	A	1	B	E	8	6	D	3	4

S-Box S_2

Entrée	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Sortie	8	6	7	9	3	C	A	F	D	1	E	4	0	B	5	2

S-Box S_3

Entrée	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Sortie	0	F	B	8	C	9	6	3	D	1	2	4	A	7	5	E

S-Box S_4

Entrée	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Sortie	1	F	8	3	C	0	B	6	2	5	4	A	9	E	7	D

S-Box S_5

Entrée	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Sortie	F	5	2	B	4	A	9	C	0	3	E	8	D	6	7	1

S-Box S_6

Entrée	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Sortie	7	2	C	5	8	4	6	B	E	9	1	F	D	3	A	0

S-Box S_7

Entrée	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Sortie	1	D	F	0	E	8	2	B	7	4	C	A	9	3	5	6