Whitebox

David Wong Jacques Monin Hugo Bonnin

9 avril 2014

Table des matières

1 Introduction **3**
1.1 Utilisation 3
1.2 Problèmes 3
1.3 Solution 3

2 DES **4**
2.1 Pourquoi DES ? 4

3 Concepts **5**
3.1 Tabularisation 5
3.1.1 Étape 1 à Étape 2 5
3.1.2 Étape 2 à Étape 3 6

4 Notre implémentation **7**
4.1 Les fonctions de base 7

5 Conclusion **8**

6 Bibliographie **9**

1 Introduction

1.1 Utilisation

1.2 Problèmes

l'attaquant à accès à la mémoire et donc il peut facilement récupérer la clef en analysant l'exécution du programme.
De base il peut aussi choisir ce qu'il envoie au programme et voir comment le programme l'encrypte ou le décrypte (chosen plaintext attack) (cas d'une blackbox)
Whitebox : l'attaquant a encore plus de possibilité puisqu'il contrôle l'environnement : accès à la mémoire, trace, breakpoints,...

1.3 Solution

Le but est de rendre l'extraction de clef impossible. Pour cela, nous allons prendre un algorithme connu : DES et y appliquer de la tabularisation ainsi que de la délinéarisation qui sont des concepts que nous expliquerons ultérieurement.

2 DES

2.1 Pourquoi DES ?

3 Concepts

3.1 Tabularisation

3.1.1 Étape 1 à Étape 2

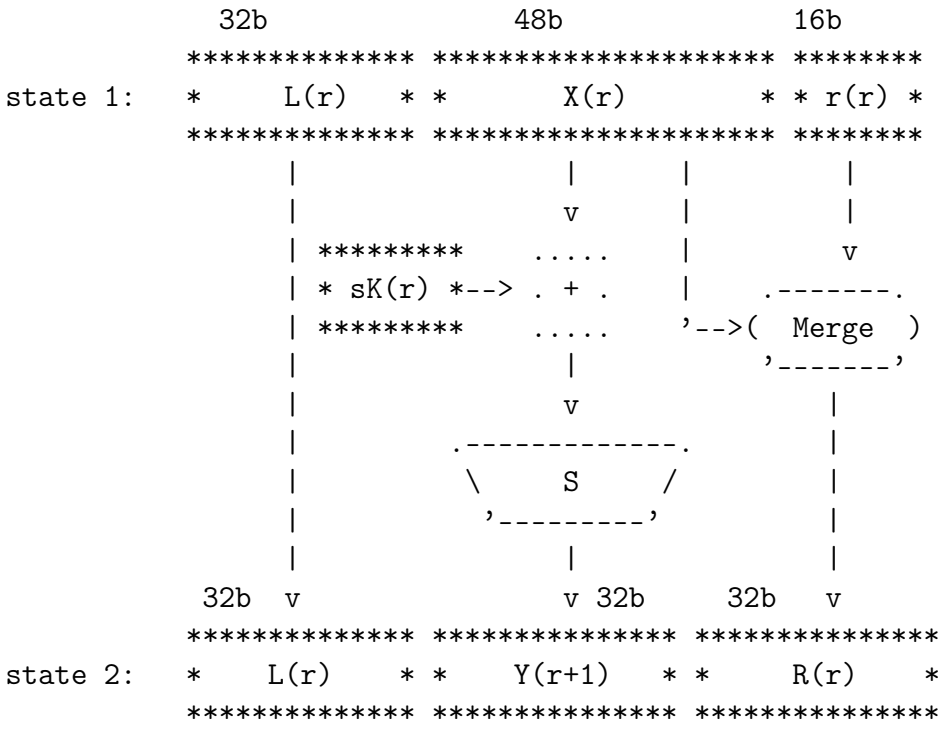

FIGURE 1 – Avant Tabularisation

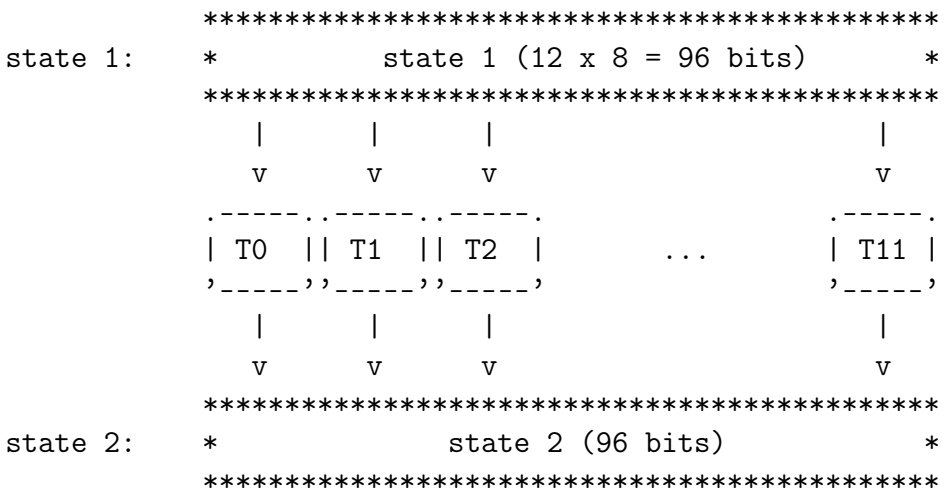

FIGURE 2 – Après Tabularisation

Pour ce faire, nous allons calculer 12 Look up Tables qui prendront 8 bits chacun ce qui recouvrira les 96 bits d'input.

Il y a :

8 look up tables non lineaires qui permettent le xor avec la clef et la substitution

4 look up tables lineaires qui nous serviront à by-passer les bits qui ne subissent pas de calculs.

3.1.2 Étape 2 à Étape 3

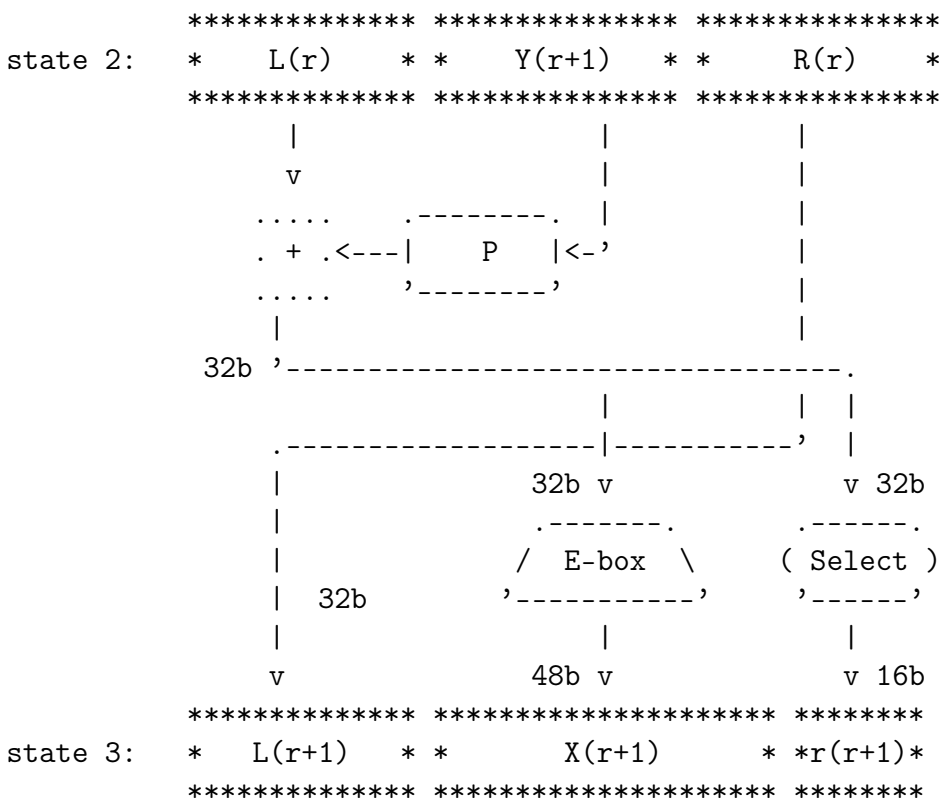

FIGURE 3 – Après Tabularisation

4 Notre implémentation

4.1 Les fonctions de base

5 Conclusion

6 Bibliographie