Chiffrement RC4 et fonction de hachage

Chassat Christophe

28 septembre 2023

Table des matières

Introduction

La partie cryptographie de cette SAE se divise en 2 parties indépendantes. Dans la première partie vous mettrez en oeuvre un algorithme de chiffrement pour votre application permettant le chiffrement des mots de passe d'accès à celle-ci. Puis dans la seconde partie vous devrez réaliser une recherche bibliographique afin d'expliquer un procédé de chiffrement.

1 Algorithme RC4

1.1 Introduction et contexte

En 1987, Ronald RIVEST met au point l'algorithme de chiffrememnt RC4 (Rivest Cipher 4). Il est également connu pour avoir mis au point en 1977 avec Adi SHAMIR et Len ADLEMAN le premier algorithme de chiffrement à clé publique, nommé RSA selon leurs initiales. RC4 est un cryptosystème à *chiffrement symétrique* c'est-à-dire que les clefs de chiffrement et déchiffrement peuvent se déduire l'une de l'autre. En pratique la clef utilisée pour le déchiffrement est identique à celle utilisée pour le chiffrement. Enfin on dit que RC4 fait partie des *chiffrements par flots (Stream Ciphers)* c'est-à-dire qu'ils opèrent sur le message clair par bit (ou quelquefois par petit groupement de bits).

1.2 Principe général

Le système RC4 fonctionne de la façon suivante : une permutation S des 256 octets est construite à partir d'une clé K en utilisant des opérations très simples sur le contenu de ce tableau S (échange de case, addition modulaire,...),une suite chiffrante est ensuite générée. L'algorithme de génération de la suite chiffrante K prend en en entrée le tableau S et retourne une suite de n octets $K = (z_1, \ldots, z_n) \in \{0, \ldots, 255\}^n$

1.3 Génération de la suite chiffrante K

L'algorithme permettant d'obtenir la suite chiffrante K est décrit dans le pseudo code ci-dessous. De plus la figure 1 page 2 vous décrit également le principe du chiffrement par flot.

Algorithm 1 Génération de la suite chiffrante de RC4 ou Pseudo Random Generator Algorithm PRGA

Entrée : S
Sortie : $K = (z_1, \ldots, z_n)$
$i \leftarrow 0$
$j \leftarrow 0$
while $j < n$ **do**
    $i \leftarrow (i + 1) \bmod 256$
    $j \leftarrow (j + S[i]) \bmod 256$
    $S[i] \leftrightarrow S[j]$ ▷ on échange les 2 valeurs $S[i] et S[j]$
    $z_i \leftarrow S[(S[i] + S[j]) \bmod 256\ ]$
return $K = (z_1, \ldots, z_n)$

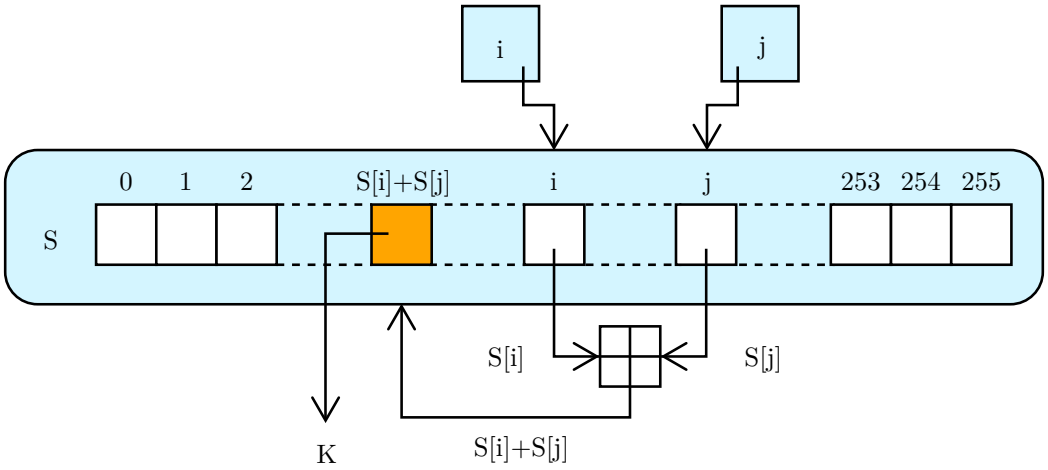

FIGURE 1 – Principe du chiffrement par flot RC4

Enfin la génération de la permutation S à partir de la clef K est réalisée à l'aide de l'algorithme ci-dessous. La longueur de la clé varie généralement de 1 à 256 bits. Elle est souvent choisie égale à $l = 16$ octets (i.e. 128 bits).

Algorithm 2 Génération de la permutation S ou Key Scheduling Algorithm KSA

Entrée : clé K
Sortie : chaîne S
for i de 0 à 255 **do**
    $S[i] \leftarrow i$
$j \leftarrow 0$
for i de 0 à 255 **do**
    $j \leftarrow (j + S[i] + K[i \bmod l]) \bmod 256$
    $S[i] \leftrightarrow S[j]$ ▷ on échange les 2 valeurs $S[i] et S[j]$
return S

1.4 Création du module de chiffrement/déchiffrement

Au sein de votre projet vous réaliserez un module permettant l'accès à votre application à l'aide d'un mot de passe chiffré. Ce module devra comporter un module de chiffrement du mot de passe s'appuyant sur le chiffrement RC4 décrit précédemment puis un module de déchiffrage du mot de passe encodé. On pourra stocker dans un premier temps les mots de passe chiffrés dans un fichier texte. Afin de vérifier le bon fonctionnement de votre module vous pourrez utilisez le tableau 1 comme test de votre module. La clé et le texte sont en ASCII et la suite chiffrante ainsi que le texte encodé sont en hexadecimal.

Clé/Key	*Suite chiffrante/Keystream*	*Texte/Plaintext*	*Texte chiffré/Ciphertext*
Key	EB9F7781B734CA72A719…	Plaintext	BBF316E8D940AF0AD3
Wiki	6044DB6D41B7…	pedia	1021BF0420
Secret	04D46B053CA87B59	Attack at dawn	45A01F645FC35B383552544B9BF5

Table 1 – Exemples de couple clé/texte et leurs valeurs chiffrés

2 Fonction de Hachage

Dans cette partie vous devrez définir ce qu'est une fonction de hachage cryptographique et les propriétés qu'une telle fonction doit disposer. Vous devrez dans un deuxième temps présenter le fonctionnement de la fonction de hachage MD5. Enfin vous expliciterez l'utilisation de telles fonctions dans le domaine de la cryptographie.