****

**SAE**

**Rapport du module de cryptographie**

Yanis Merabet

Samuel Charton

Hugo Chartier

Brandon Bui

Vivien Lampin

**Sommaire :**

1. Interfaçage du module avec le web
2. Explication du code avec commentaires

Interfaçage du module avec le site web

Pour le module de cryptographie, nous avons décidé de partir sur une page avec 2 champs, un pour le texte et l’autre pour la clé ainsi que sur un bouton de validation qui affichera le texte chiffré. Une fois le bouton de validation cliqué, le script JavaScript du module de cryptographie sera appelé avec comme paramètres les données des 2 champs.

Explication du code avec commentaires

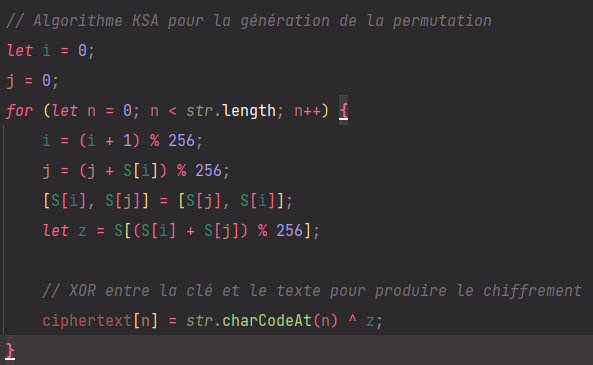
Le module de cryptographie était décomposé en 2 algorithmes bien distincts qui une fois « combinés » donne l’algorithme RC4.

Le premier algorithme s’appelle l’algorithme PRGA. Cet algorithme va se charger de générer une suite de caractères permettant de chiffrer notre message en tenant compte de notre clé de chiffrement.

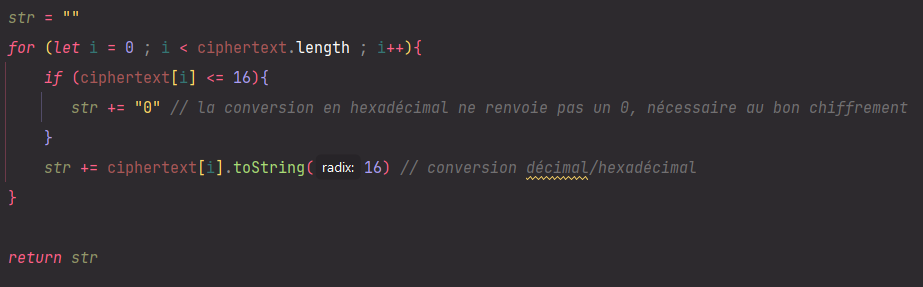


On crée donc ici un tableau qui contiendra les caractères générés à l’aide de la clé dans la 2ème boucle for (fonction key.charCodeAt)

Le deuxième algorithme utilisé dans l’algorithme RC4 s’appelle l’algorithme KSA. Celui-ci va se charger d’effectuer toutes les opérations nécessaires sur le tableau de caractères pour le chiffrement.



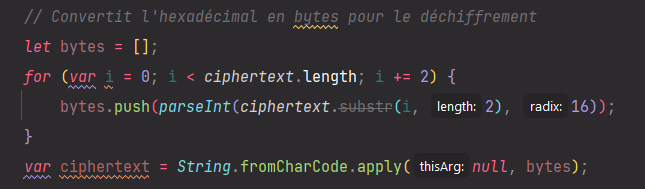
Pour finir, le texte final sera généré à partir du tableau de caractères sous forme décimal. Il faut les mettre sous forme hexadécimal. Pour cela, il faut utiliser une fonction de JavaScript permettant de transformer un texte en hexadécimal



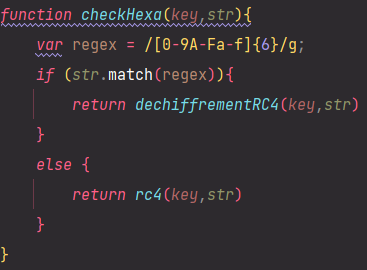
On met tous les caractères du tableau sous la forme d’un texte et on transforme ce texte en hexadécimal en oubliant pas de rajouter un 0 avant chaque nombre inférieur ou égal à 16 car l’hexadécimal ne rajoute pas un 0 (renvoie 9 au lieu de 09 par exemple), le 0 étant utilisé dans l’algorithme RC4.

Il faut maintenant faire le déchiffrement, on ne peut pas utiliser la même fonction pour le déchiffrement dans ce cas-là. Il faut donc faire une fonction permettant de déchiffrer (passer de l’hexadécimal à une chaîne de caractères déchiffrée)

Cette fonction va être très similaire à celle du chiffrement à ceci près qu’elle va devoir convertir notre message chiffré en bytes pour que le chiffrement puisse se faire



Le reste du code de cette fonction est exactement la même chose que le chiffrement RC4 sauf que l’on affiche une chaîne de caractères « normale » et non en hexadécimal.

Pour vérifier si la chaîne de caractères passée est en hexadécimal ou une simple chaîne de caractères, il faut faire une fonction qui vérifiera cela et qui appellera soit le chiffrement, soit le déchiffrement en fonction de si la chaîne est en hexadécimal ou non.