Cryptographie et Sécurité Informatique

Kasengedia Motumbe Pierre
Assisté par : Junior Kaningini

Edition : UNIKIN - L3 LMD Informatique

20 décembre 2023

QUESTION 1

Algorithme pour la génération des clés de Feistel

1. Entrée : La clé K de longueur 8
2. Appliquer la fonction de permutation H = 65274130
3. Diviser K en deux blocs de 4 bits : $K = k_1' || k_2'$
4. $k_1 = k_1' \oplus k_2'$ et $k_2 = k_2' \wedge k_1'$
5. Appliquer le décalage à gauche d'ordre 2 pour k_1 et le décalage à droite d'ordre 1 pour k_2
6. Sortie : Deux sous-clés (k_1 , k_2) de longueur 4.

Algorithme de chiffrement de Feistel

1. Entrée : Le bloc N de 8 bits
2. Appliquer la permutation $\pi = 46027315$
3. Diviser N en deux blocs de 4 bits : $N = G_0 || D_0$
4. Premier Round, calculer :
   - $D_1 = P(G_0) \oplus k_1$ et
   - $G_1 = D_0 \oplus (G_0 \vee k_1)$ où $P = 2013$ est la permutation
5. Deuxième Round, calculer :
   - $D_2 = P(G_1) \oplus k_2$ et
   - $G_2 = D_1 \oplus (G_1 \vee k_2)$
6. $C = G_2 || D_2$ (la concaténation)
7. Appliquer l'inverse de la permutation $\pi^{-1}(C)$
8. Sortie : Le texte chiffré C de longueur 8.

Algorithme de déchiffrement de Feistel

1. Entrée : Le bloc C de 8 bits
2. Appliquer la permutation $\pi = 46027315$
3. Diviser C en deux blocs de 4 bits : $C = G_2||D_2$
4. Premier Round, calculer :
   - $G_1 = P^{-1}(D_2 \oplus k_2)$ et
   - $D_1 = G_2 \oplus (G_1 \vee k_2)$ où $P = 2013$ est la permutation
5. Deuxième Round, calculer :
   - $G_0 = P^{-1}(D_1 \oplus k_1)$ et
   - $D_0 = G_1 \oplus (G_0 \vee k_1)$
6. $N = G_0||D_0$ (la concaténation)
7. Appliquer l'inverse de la permutation $\pi^{-1}(N)$
8. Sortie : Le texte clair N de longueur 8.

Question

En utilisant votre langage au choix entre :

- Python
- Java
- PHP

Implémenter ces trois algorithmes sachant que l'utilisateur pourra définir sa propre permutation de longueur 8 et aussi l'ordre de décalage.

Votre programme doit s'adapter à la permutation donnée ainsi qu'à l'ordre du décalage.

QUESTION 2

Soit $x^b(mod\ n)$, implémentez l'algorithme des carrés et des multiplications (Square & Multiply Algorithm) en laissant l'utilisateur le choix d'insérer les valeurs de x, b et n.

Information sur la Soumission

Chaque étudiant est prié de créer un compte github sur lequel il créera le **repository** nommé **Feistel cipher and Square & Multiply Algorithm** .

Pour créer un compte GitHub, vous pouvez cliquer ICI ou taper `https://github.com/login` à votre navigateur.

Vous pouvez suivre les differentes étapes de la création d'un compte github en cliquant ici.

N.B : Seul le lien de votre ***repository*** nous sera envoyé à l'adresse :

labcoursjk@gmail.com

avec comme **Objet *Lab3-L3LMD2023-PrenomNom***
ex : Lab3-L3LMD2023-AliceBob.

Exemple du lien d'un repository :
https://github.com/Junior-081/Bernouilli-Naive-Bayes

Date limite : 25 Décembre 2023 avant 23h59

⚠ Deux travaux similaires entrainent l'annulation.