FALL ABDOU AZIZ 21804685

EXPLICATION DE L'ATTAQUE ET L'IMPLÉMENTATION

(m1,c1) = (8a5c19, 7cc9ee) (m2,c2) = (cfdbd2,cc89f3)

pour l'implémentation de l'attaque, on a utilisé la structure de données main_keys qui permet de stocker la clé maitre et est composée d'un tableau 20 cages de type uint4_t. Ce dernier est une structure donnée composée d'une variable de 4 bits qui peut contenir des valeurs comprises entre [0..9,a..f].Le tableau permet par la suite de représenter les 80 bits de la clé maître.

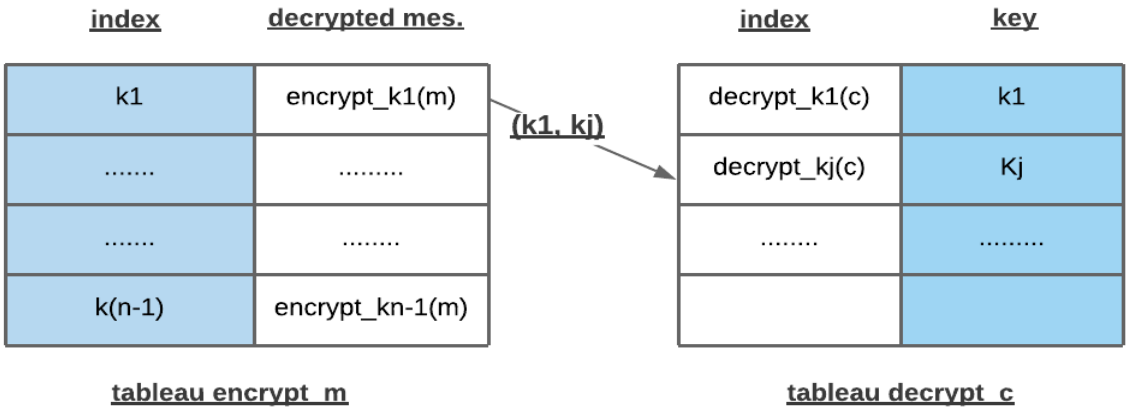
les inverses de la substitution et de la permutation, on les obtient comme suit :
par exemple si perm[1] = 6 alors l'inverse de la permutation de 6 est inv_perm[6] = 1, de même pour la substitution.

Pour stocker l'ensemble des clés générées on a utilisé la structure de données data_attack composée d'une variable m qui est le message et c qui est 2PRESENTk1,k2(m), un tableau encrypt_m qui contient tout les chiffrés de m en utilisant les clés k comprises [0, 0xffffff] et un tableau decrypt_c qui contient tous les clés qui ont permit de déchiffrés c et les messages déchiffrés sont les index du tableau et les valeurs du tableau sont les clés. Donc le tableau decrypt_c devient une table de hachage dans laquelle chaque entrée correspond à un couple de (déchiffré, clé).

Sur le schéma ci-dessous, la recherche de clés se résume comme suite :
Au cours de la génération des chiffrés et des déchiffrés, on remplit le tableau de decrypt_c par decrypt_cj[decrypt_ki(cj)] = ki et le tableau encrypt_mj[ki] = encrypt_ki(mj) avec (j appartient [1, 2]). Aprés dans la fonction **attacks**, pour chaque clé **ki**, On cherche dans decrypt_c la deuxième clé via cette instruction :
kj = decrypt_c1[encrypt_c1(ki)]
kj' = decrypt_c2[encrypt_c2(ki)]
où :

- i et j appartiennent [0, 0xffffff – 1].
- ki : est la clé qui a permit de chiffré le message m1 et m2
- encrypt_c[ki] : est le chiffré du message m1 et m2 par la clé ki
- kj , kj': est la clé qui a permit de déchiffré c1 et c2 respectivement.

Et si kj 2PRESENTki,kj(m1) = c1 et 2PRESENTki,kj'(m2) = c2 et kj = kj' on affiche le couple (ki, kj), sinon on continue la recherche.
Pour la compléxité pour génération des clés et la recherche de couple de clé est de O(N) avec N = 0xffffff le nombre de clé de 24 bits (2^24 – 1).



rechercher de couple de clé(k1, k2)