Cryptography

Jonas Wagner

Medium Task

Now that Heidi can authenticate her friends, they want to solve the problem of secure communication. Heidi decides that for her purposes, the substitution cipher might be adequate. There are two problems with it, though: (1) It is vulnerable to frequency attacks, and (2) given a plaintext and a ciphertext, it is trivial to find the key.

Heidi solves the first problem by requiring that each message contains at least one occurrence of every character. She thinks that this makes frequency attacks harder. She solves the second problem by encrypting every message twice. Can you prove to Heidi that her system is still weak?

Heidi's messages are written in an alphabet consisting of the underscore character and lowercase letters `a` to `z`. One round of the substitution cipher replaces each letter by another, according to the secret key. The letter `_` is replaced by the first character of the key, `a` by the second, and `z` by the twenty-seventh.

You have to implement a function `find_key` that receives as parameters a plaintext and the corresponding ciphertext. Both contain every letter of the alphabet, and `_`, at least once. The function must return a key such that applying the key twice to encrypt the plaintext results in the ciphertext. Any such key is OK.

Sample

Plaintext	`the_quick_brown_fox_jumps_over_the_lazy_dog`
Ciphertext	`uifarvjdlacspxoagpyakvnqtapwfsauifamb_zaeph`
Key	`nopqrstuvwxyz_abcdefghijklm`

After the first round of encryption, the intermediate ciphertext is `gvsndhwqynpebjantbknxh_cfnbisengvsnzomlnrbu` .

Tâche Moyenne

Maintenant que Heidi sait authentifier ses amis, elle se donne au problème de la sécurisation de la communication. Elle décide que le chiffrement par substitution est adéquat pour ses besoins. Une telle technique présente pourtant deux faiblesses : (1) Elle est vulnérable aux attaques fréquentielles et (2) la connaissance d'une paire de texte brut (plaintext) et sa version encryptée (ciphertext), permet de trouver la clé (key) trivialement.

Heidi résout ce premier problème en exigeant que chaque message contient chaque caractère au moins une fois. Elle pense que ceci rend les attaques fréquentielles plus difficiles. Et pour contraire le second problème, elle chiffre chaque message deux fois. Est-ce que tu montrer à Heidi que son système est toujours faible ?

Les messages de Heidi sont écris dans un alphabet formé par le tiret bas (`_`) et les lettres minuscules `a` à `z`. Chaque itération du chiffrement par substitution remplace chaque lettre par une autre, en accord avec la clé secrète. La lettre `_` est remplacée par la première lettre de la clé, `a` par la deuxième, et `z` par la vingt-septième.

Tu dois implémenter une fonction `find_key` qui reçoit comme paramètres le plaintext et le ciphertext correspondant. Les deux contiennent chaque lettre de l'alphabète et _ au moins une fois. La fonction doit retourner une clé, qui, appliquée deux fois sur le plaintext, génère le ciphertext. Toute telle clé est OK. Voir la première page pour un example.

Mittlere Aufgabe

Heidi erkennt ihre Freunde jetzt am Telefon, und möchte als nächstes das Problem der verschlüsselten Kommunikation lösen. Die monoalphabetische Substitution scheint ihr recht für diese Anwendung. Diese Technik weist jedoch zwei Schwachpunkte auf: (1) Sie kann durch Frequenzattacken geknackt werden, und (2) anhand von einem Klartext (plaintext) und dem zugehörigen Geheimtext (ciphertext), kann der Schlüssel (key) gefunden werden.

Heidi löst die erste Schwachstelle, indem sie erfordert, dass jede Botschaft jedes Textzeichen mindestens einmal enthält. Sie denkt, das mache Frequenzattacken bedeutend schwieriger. Das zweite Problem löst sie, indem sie jede Botschaft zweimal verschlüsselt. Kannst du Heidi beweisen, dass ihr System immer noch anfällig ist?

Heidis Alphabet für die Botschaften besteht aus dem Unterstrich (_), sowie den Kleinbuchstaben `a` bis `z`. Jeder Durchgang der monoalphabetischen Substitution ersetzt jedes Zeichen durch ein anderes, so wie es der Schlüssel erfordert. Das Zeichen _ wird durch den ersten Buchstaben des Schlüssels ersetzt, `a` durch den zweiten, und `z` durch den siebenundzwanzigsten.

Schreibe eine Funktion `find_key`, welche als Argumente einen Klartext und den dazugehörigen Geheimtext erhält. Beide enthalten jedes Zeichen des Alphabets mindestens einmal. Die Funktion muss einen Schlüssel zurückgeben, der nach zweifacher Anwendung auf den Klartext den Geheimtext ergibt. Jeder solche Schlüssel ist OK. Siehe erste Seite für ein Beispiel.