ESISAR

CS353 - Algorithmique

TP numéro 5
RAINBOW TABLE

Table des matières

1 Objectifs du TP numéro 5

Ce TP va permettre l'implémentation d'une RainbowTable (vu en TD).

Pour faire ce TP, vous avez besoin de :

- 3H de préparation avant le TP
- 1h30 avec l'enseignant en salle de TP
- une poignée de courage et un zeste de persévérance !

2 Les documents du TD

Relisez les documents vus en TD :
https://en.wikipedia.org/wiki/Rainbow_table
http://kestas.kuliukas.com/RainbowTables/
https://en.wikipedia.org/wiki/MD5

3 Exo 1 - La force brute ...

Votre ami Jack Le Hacker a réussi à s'introduire sur un site WEB de e-commerce et à voler la base de données du site.

A partir de cette base de donnée, votre ami vous a fourni une liste de login / mot de passe , de la forme suivante :

Alice	84E6A804E2069365DF19AB2D0157E818
Bob	C3A222A452FE95C956953C41F46CD334
Clara	7C436F13E2C4E8309A93D1E1C887A228
Dilbert	C63B2B0396DD870448894F1152320E67

La colonne 1 contient le login de l'utilisateur, la colonne 2 contient le hash MD5 du mot de passe de l'utilisateur

Sur ce site WEB, tous les mots de passe sont composés uniquement de chiffres, et ont toujours pour longueur 6. Exemples de mot de passe : 500123 , 456123 , … Attention, les mots de passe sont bien des chaînes de caractères, mais composés uniquement de chiffres.

Par une attaque de type brute force, déterminer le mot de passe de tous les utilisateurs.
Indiquez le temps de calcul de tous ces mots de passe par brute force.

4 Exo 2 - Calcul d'une chaîne

Faites une fonction **`CalculChaine(int px)`** qui :

- prend en entrée un entier quelconque entre 1 et 999999
- convertit ce nombre en une chaîne de 6 caractères (mot de passe PX, exemple « 012012 »)
- calcule le hash MD5 de la chaîne de caractère
- le hash MD5 est ensuite réduit par la fonction de réduction R0, on obtient le mot de passe P0
- on répète ensuite l'opération : calcul du hash MD5 de P0, puis réduction par R1, on obtient P1
- …
- calcul du hash MD5 de P998, puis réduction par R999, on obtient P999
- la fonctionne retourne ensuite P999

```
   Hash      R0      Hash   R1                    Hash      R999
PX   =>  H0  => P0   =>  H1  => P1 => ...   P998  =>  H998  => P999
```

Vous utiliserez la fonction de réduction suivante (pseudo code). num correspond au numéro de la fonction de réduction (num varie de 0 à 999).

```
int reduction(byte[] hash,int num)
{
	int res = num;

	int mult = 1;
	for (int i = 0; i < 4; i++)
	{
		res = res + mult*hash[i];
		mult = mult * 256;
	}

	if (res<0)
	{
		res = -res;
	}

	res = res  % 1000000;


	return res;
}
```

Par exemple si l'entrée de la fonction **`CalculChaine`** est 1, la sortie devra être `279947.`

5 Exo 3 - Calcul de la table complète et sauvegarde dans un fichier

Faites maintenant un programme qui calcule 10 000 chaînes et qui stocke pour chaque chaîne les valeurs (PX et P999) dans une table de hachage à adressage ouvert de taille 10 061.

L'algorithme sera le suivant

- faites une boucle de 1 à 10 000
- pour chaque pas de la boucle
  - calculer un nombre aléatoire
  - en déduire PX et P999
  - insérer le couple (PX,P999) dans la table de hachage, **P999 est la clé** et PX est une valeur satellite
  - si la clé est déjà existante dans la table de hachage (il y a déjà dans la table une chaîne se terminant par P999), alors le couple (PX,P999) n'est pas inséré (la chaîne que l'on vient de calculer est oubliée)

Détail sur la table de hachage : il est ici possible de faire quelque chose de très simple, car il n'est pas nécessaire de gérer les éléments supprimés.

Vous pouvez utiliser la structure suivante

```
struct Noeud
{
      int px ;
      int p999 ;
}
```

et la table sera un tableau

```
struct Noeud T[10061] ;
```

Un Noeud sera considéré comme vide si p999 = -1.

Questions :
1/Pourquoi calcule t on 10 000 chaînes ?
2/ Pourquoi utilise t on la valeur 10 061 pour la table de hachage ?
3/Combien y a t il réellement d'éléments dans la table ? Pourquoi ?

Une fois votre table de hachage calculée, sauvegardez la dans un fichier (il suffit d'écrire tout le tableau dans un fichier).

6 Exo 4 - Utilisation de la table pour « casser » un mot de passe

Faites maintenant un programme :

- charge votre table de hachage en mémoire
- demande en entrée un hash MD5 de mot de passe (le mot de passe est toujours 6 caractères numériques)
- retourne le mot de passe en clair

Indiquez le temps de calcul d'un mot de passe.

7 Exo 5 - Calcul d'une table pour 9 caractères numériques

Faites la même chose avec 9 caractères numériques.

Attention, vous devez changer une ligne de la fonction de réduction , celle ci devient

```
      int reduction(byte[] hash,int num)
      {
            int res = num;

            int mult = 1;
            for (int i = 0; i < 4; i++)
            {
                  res = res + mult*hash[i];
                  mult = mult * 256;
            }

            if (res<0)
            {
                  res = -res;
            }

            res = res  % 1000000000;   // LIGNE MODIFIEE !!!!!!!!!!
```

```
            return res;
    }
```