



كلية العلوم والتقنيات بطنجة
ⵜⴰⵎⴻⵔⴰⵏⵜ ⴰⵏ ⵜⴰⵎⴻⵔⴰⵏⵜ ⴰⵏ ⵜⴰⵎⴻⵔⴰⵏⵜ ⴰⵏ ⵜⴰⵎⴻⵔⴰⵏⵜ
Faculté des Sciences et Techniques de Tanger

10/29/2024

Création d'une Fonction de Hachage

Idriss Khattabi

BLOCKCHAIN & SECURITE APPLICATIVE

Introduction :

Une fonction de hachage est une fonction qui convertit une chaîne de texte en une valeur numérique ou alphanumérique fixe. Elle permet de représenter une grande quantité de données sous forme d'un nombre ou d'une chaîne de caractères de longueur fixe. Les fonctions de hachage sont essentielles pour garantir l'intégrité des données, vérifier leur authenticité et sécuriser les informations sensibles.

Dans ce rapport, nous allons détailler le processus de création d'une fonction de hachage simple en Python, qui produit un hachage d'une longueur fixe de 10 caractères.

Étapes de l'Algorithme :

La fonction 'hash10' est constituée des étapes suivantes :

1. Initialisation des Variables :

- La variable 'hash' est initialisé à 0 pour accumuler les valeurs de hachage.
- La longueur fixe du hachage est définie à 10 caractères.

2. Calcul de la Valeur de Hachage :

- Pour chaque caractère de la chaîne d'entrée 'text', on calcule la valeur ASCII du caractère multipliée par sa position dans la chaîne (index + 1).
- Ces valeurs sont cumulées pour produire une somme finale, stockée dans hash.

3. Transformation de la Somme en Liste de Caractères :

- La somme est convertie en une chaîne du caractère numérique.
- Chaque chiffre de cette chaîne est transformé en caractère ASCII en ajoutant 33 pour s'assurer que chaque chiffre soit un caractère imprimable.

4. Formation de la Chaîne Finale :

- Les caractères sont joints pour former une chaîne unique.
- Si le résultat est plus court que 10 caractères, il est complété par des zéros à droite pour obtenir une longueur fixe.

Le code en Python :

```
def hash10(text : str) -> str :  
    hash = 0  
    fixed_length = 10  
  
    for index, char in enumerate(text) :  
        hash += ord(char)*(index+1)  
  
    hash = str(hash)  
    hash = [chr(int(char)+33) for char in hash]  
    hash = ''.join(hash)  
  
    return hash.ljust(fixed_length, '0')
```

Exemple Pas à Pas :

On va appliquer la fonction **hash10** sur la chaîne "idriiss" pour observer le fonctionnement :

1. Initialisation :

- 'hash' est initialisé à 0.
- 'fixed_length' est fixé à 10.

2. Calcul de la Valeur de Hachage :

- Pour chaque caractère de "idriiss", nous effectuons les opérations suivantes :
 - Caractère 'i' : ASCII -> 105, Position -> 1, Multiplication $105 * 1 = 105$
 - Caractère 'd' : ASCII -> 100, Position -> 2, Multiplication $100 * 2 = 200$
 - Caractère 'r' : ASCII -> 114, Position -> 3, Multiplication $114 * 3 = 342$
 - Caractère 'i' : ASCII -> 105, Position -> 4, Multiplication $105 * 4 = 420$
 - Caractère 's' : ASCII -> 115, Position -> 5, Multiplication $115 * 5 = 575$
 - Caractère 's' : ASCII -> 115, Position -> 6, Multiplication $115 * 6 = 690$
- La somme de ces produits est 2332.

3. Transformation de la Somme en Liste de Caractères :

- La somme 2332 est convertie en chaîne "2332".
- Chaque chiffre de cette chaîne est transformé en caractère ASCII en ajoutant 33 :
 - 2 devient '#'
 - 3 devient '\$'
- La liste de caractères résultante est ['#', '\$', '\$', '#'].

4. Formation de la Chaîne Finale :

- Les caractères sont joints pour former la chaîne "\$\$\$#".
- Puisque la chaîne obtenue a une longueur de 4 caractères, elle est complétée par des zéros à droite pour atteindre une longueur fixe de 10 caractères : "\$\$\$#000000".

L'exécution :

```
hash10('idriss')
-> Char : i - ASCII Char : 105 - Position : 1 - Multiplication : 105
-> Char : d - ASCII Char : 100 - Position : 2 - Multiplication : 200
-> Char : r - ASCII Char : 114 - Position : 3 - Multiplication : 342
-> Char : i - ASCII Char : 105 - Position : 4 - Multiplication : 420
-> Char : s - ASCII Char : 115 - Position : 5 - Multiplication : 575
-> Char : s - ASCII Char : 115 - Position : 6 - Multiplication : 690
Total of Multiplication : 2332
List of Chars : ['#', '$', '$', '#']
Join the chars : $$$#
Complete text with 0 to reach 10 chars : $$$#
'$$$#000000'
```

Conclusion :

Cette fonction de hachage simple démontre une méthode de création d'un hachage de longueur fixe. Bien qu'elle ne soit pas adaptée aux applications de sécurité, elle illustre les principes de base d'un hachage et peut être utilisée dans des cas où la complexité cryptographique n'est pas nécessaire. Les méthodes plus avancées utilisent des algorithmes de hachage cryptographique tels que SHA ou MD5 pour des applications nécessitant une sécurité renforcée.