IUT ANNECY
UNIVERSITÉ SAVOIE MONT BLANC

Cryptographie et sécurité

SAé: sécuriser son site web

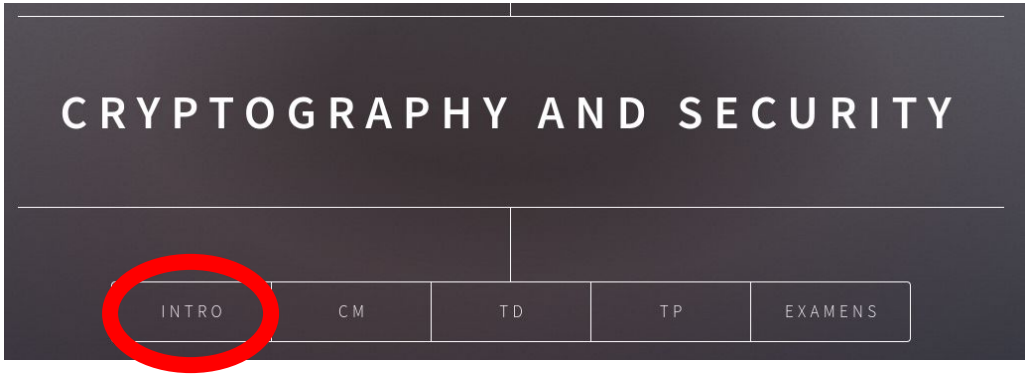
Mickaël Bettinelli
(mickael.bettinelli@univ-smb.fr)

Slides disponibles sur: *https://milowb.github.io/cryptographie*

Avant de commencer

Merci de répondre au questionnaire d'évaluation de ce cours :)

Lien sur mon site web: *https://milowb.github.io/cryptographie*




Retours sur le cours de cryptographie

Merci de donner votre avis sur le cours R3.09. Votre avis sera bien entendu anonyme.

bettinelli.mickael@gmail.com (not shared) Switch account

* Required

Evaluez votre investissement dans le travail personnel dans ce module. *

	1	2	3	4	5	6	
Minimum	○	○	○	○	○	○	Maximum

Evaluez votre niveau d'acquisition *

Sommaire

1. Délivrable cryptographie du SAé
2. Gestion des mots de passe en C#
3. Stockage des mots de passe

Délivrable attendu pour la cryptographie

Un compte rendu (1-3 pages)

Pour chaque méthodes de sécurité vous avez utilisé dans votre SAé:

- Expliquez en quoi elle consiste
- Justifiez son utilisation (quelle utilité)
- Discutez ses avantages et inconvénients (s'il y en a)
- Discutez les détails techniques (code, librairies, etc.) que vous jugez intéressants

Deadline 13/01/23

A rendre au lien suivant: https://forms.gle/bkt7myaG1JgK3nKs6

Gestion des mots de passe en C#

Pourquoi ? Pour éviter que les informations soient récupérées directement dans la mémoire (ex: les mdp).

Comment ? En C#: avec un type de string appelé ***SecureString***

Définition du SecureString:

- Un string qui doit être gardé confidentiel. Son contenu est supprimé de la mémoire de l'ordinateur lorsqu'il n'est plus utilisé.

Utiliser le SecureString

```
SecureString securePwd = new SecureString();
ConsoleKeyInfo key;

Console.Write("Enter password: ");
key = Console.ReadKey(true);

// Ignore any key out of range.
if (((int) key.Key) >= 65 && ((int) key.Key <= 90)) {
    // Append the character to the password.
    securePwd.AppendChar(key.KeyChar);
    Console.Write("*");
}
securePwd.Dispose();
```

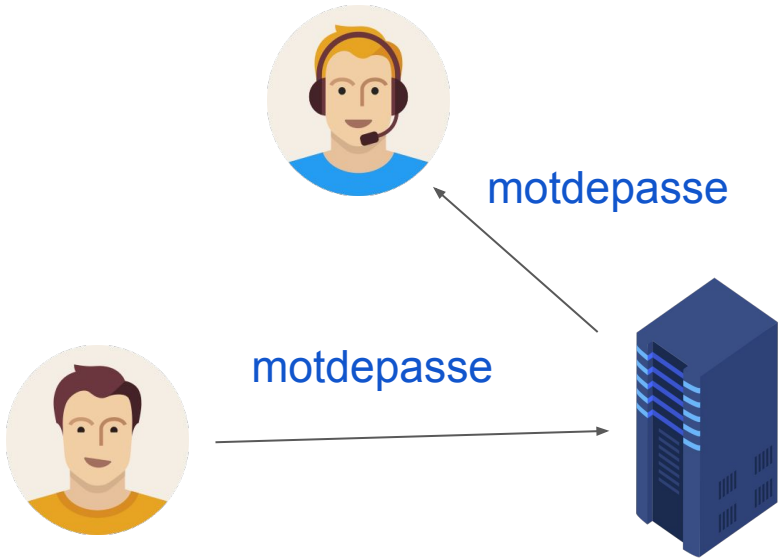
Stockage des mots de passes

Pourquoi ?

- Accéder à ses espaces personnels

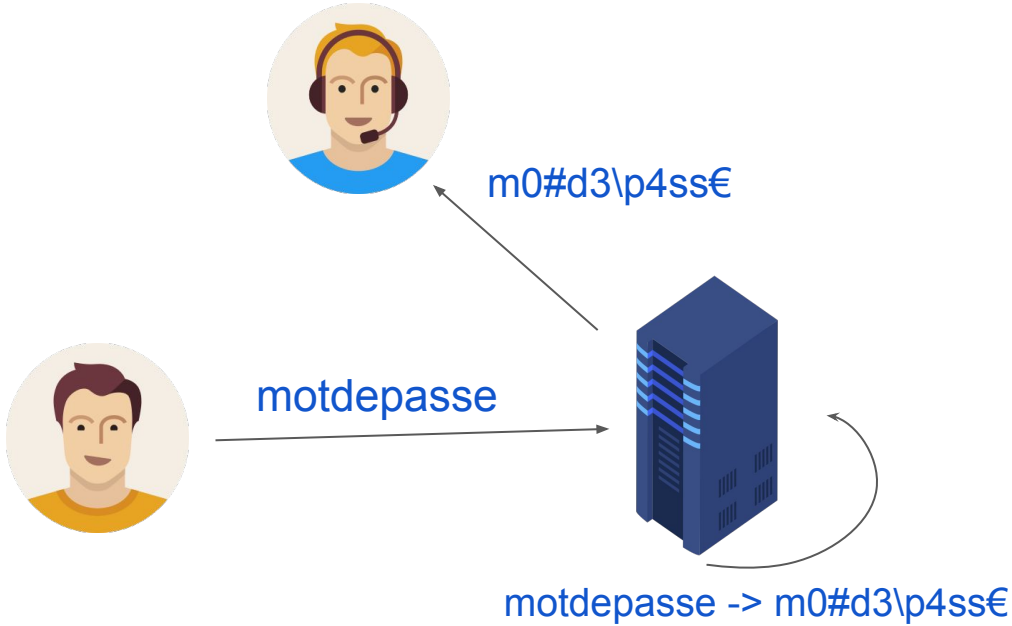
Comment ?

- Stockage des mots de passe en base de données
- Besoin de les chiffrer en cas de fuite



Problématique

Chiffrer les mots de passe pour les rendre inutilisable en cas de fuite



Les fonctions de hachage

Définition 1. Une fonction de hachage est une fonction particulière qui, **à partir d'une donnée fournie** en entrée, **calcule une empreinte numérique** servant à identifier rapidement la donnée initiale.

Définition 2. Le retour de la fonction est appelée un *hash* ou une *empreinte*.

Les propriétés des fonctions de hachage.

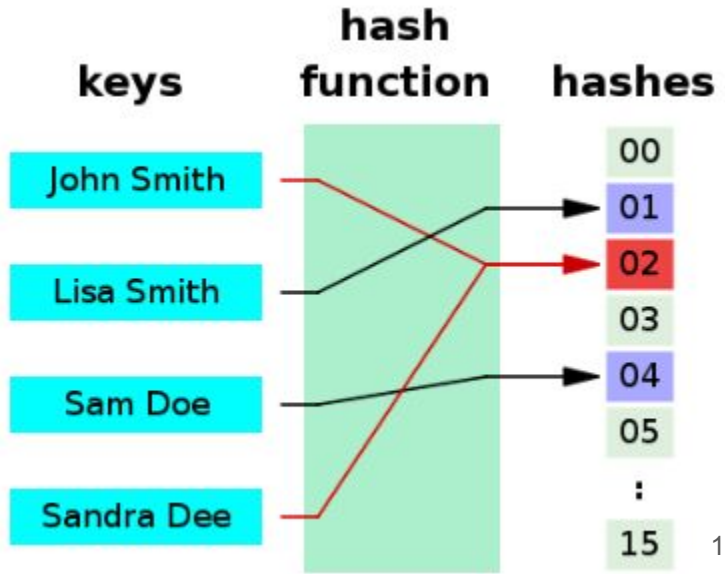
1. Fonction à sens unique: impossible de déchiffrer une empreinte
2. Déterminisme: hacher deux fois le même message conduit au même résultat
3. Unicité de la signature: un même message va correspondre une signature unique

Les fonctions de hachage

Exemple simple de fonction de hachage:

- Une fonction qui retourne le premier octet de votre mot de passe
  - Exemple:
    - motdepasse = 01101101
    - chat = 01100011

Cette fonction est très peu efficace !



En C# - créer l’empreinte

```
string sSourceData;
byte[] tmpSource;
byte[] tmpHash;

sSourceData = "MySourceData";
//Create a byte array from source data.
tmpSource = ASCIIEncoding.ASCII.GetBytes(sSourceData);

//Compute hash based on source data.
tmpHash = new SHA256CryptoServiceProvider().ComputeHash(tmpSource);
```

En C# - comparer les empreintes

```
bool bEqual = false;
if (tmpNewHash.Length == tmpHash.Length) {
    int i=0;
    while ((i < tmpNewHash.Length) && (tmpNewHash[i] == tmpHash[i])) {
        i += 1;
    }
    if (i == tmpNewHash.Length) {
        bEqual = true;
    }
}

if (bEqual)
    Console.WriteLine("The two hash values are the same");
else
    Console.WriteLine("The two hash values are not the same");
Console.ReadLine();
```

Intégrer dans une table SQL

```sql
CREATE TABLE Student
(
  Id INT IDENTITY,
  name VARCHAR(100),
  password BINARY
);
```

Ressources complémentaires

- SecureString : https://learn.microsoft.com/en-us/dotnet/api/system.security.securestring?view=net-7.0
- https://en.wikipedia.org/wiki/Hash_function
- Générer et comparer les haches : https://learn.microsoft.com/en-us/troubleshoot/developer/visualstudio/csharp/language-compilers/compute-hash-values