php

COURS

Franck LAMY – BTS SIO1 – Année 2022/2023

Partie théorique

Gestion des utilisateurs

comment stockés les mots de passe

Les mots de passe

Mots de passe en clair

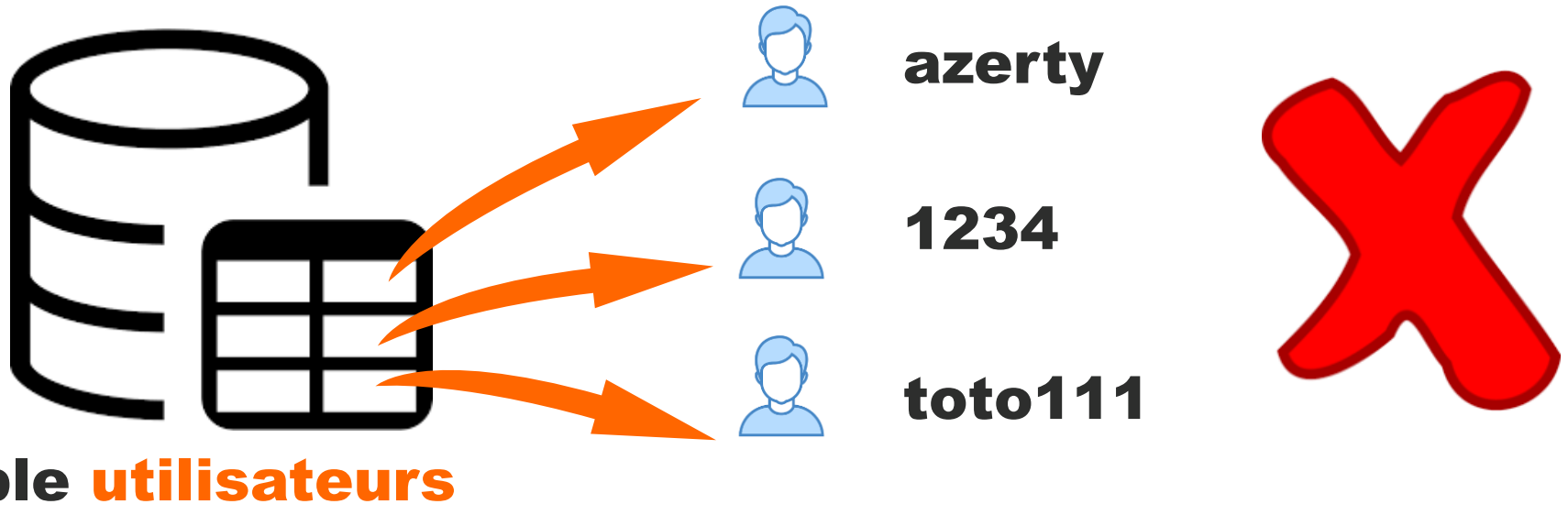


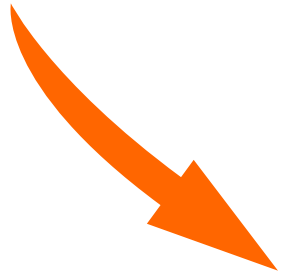
Table utilisateurs

Les mots de passe

Les mots de passe ne doivent jamais être stockés en clair dans la base de données !

Les mots de passe

Pour des besoins de sécurité, les mots de passe doivent être hachés



Fonction de hachage

Fonction de hachage

Une **fonction de hachage** est un **algorithme** permettant, à partir d'un **message**, de **générer** une **valeur** de **longueur fixe** appelée **hash**

Fonction de hachage

Fonction de hachage

Message

Texte
Fichier

f(message)

Hash
Empreinte

Longueur fixe par rapport à la fonction de hachage utilisée

5ebe2294ecd0e0f08eab7690d2a6ee69

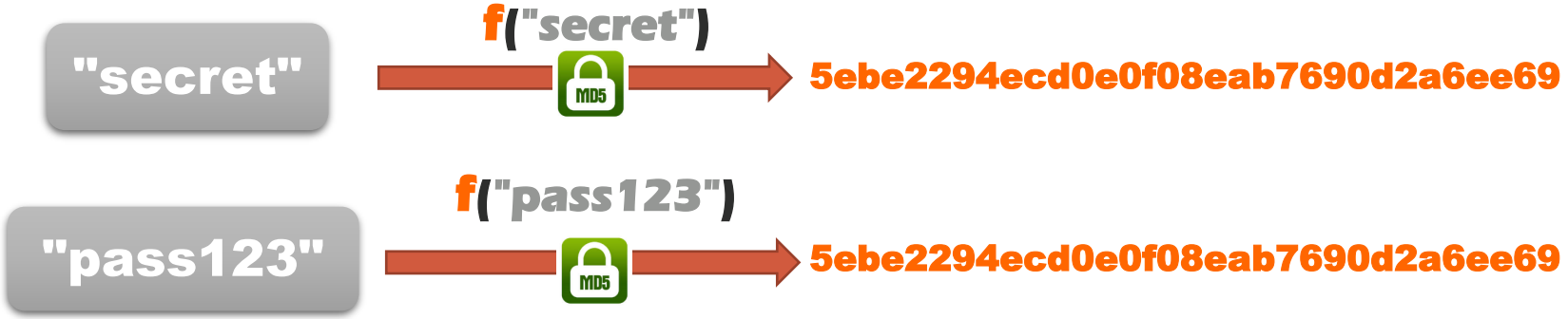
Fonction de hachage

Une fonction de hachage est irréversible

SHA-1 MD5 **Obsolètes pour les mots de passe**

Collision

Dans la théorie, 2 messages différents ne doivent pas avoir le même hash



Collision

Collision

Une bonne fonction de hachage ne doit pas permettre les collisions

Quelques problèmes…

Problème n°1 : le même hash

johndoe@symfony.com
secret ->md5(secret)=
5ebe2294ecd0e0f08eab7690d2a6ee69

janedoe@symfony.com
secret ->md5(secret)=
5ebe2294ecd0e0f08eab7690d2a6ee69

Le hacker sait donc que johndoe et janedoe ont le même mot de passe

Quelques problèmes…

Problème n°2 : mots de passe courants

johndoe@symfony.com
secret ->md5(secret)=
5ebe2294ecd0e0f08eab7690d2a6ee69

janedoe@symfony.com
pass123 ->md5(secret)=
32250170a0dca92d53ec9624f336ca24

Le hacker va utiliser des bases de donnés contenant les hash des mots de passe les plus courants

Sel cryptographique

Utilisation du sel cryptographique (salt) afin de générer un hash qui soit difficile à cracker

Valeur aléatoire

Sel cryptographique

johndoe@symfony.com
password -> secret
sel -> 098Dy

md5(secret + sel)
559426a2960c56a69ceacd4be394049a

janedoe@symfony.com
password -> secret
sel -> 943FB

md5(secret + sel)
1004c5d097f2bf27eefcbd237ce96f10

Casse-tête pour notre ami !

Sel cryptographique

La plupart des algorithmes de hachage modernes génèrent un sel aléatoirement et le stocke dans le hash

Partie pratique

Hacher un mot de passe

Fonction permettant de hacher un mot de passe

password_hash(...)

Plusieurs algo de hachage possible

Hacher un mot de passe

```php
$password = "secret";
$hash = password_hash($password, PASSWORD_DEFAULT);
echo $hash;
```

3X

```
$2y$10$vxni3I.jMgDPBpCVXeJ7POJlOGIlqPXV6pRXQzJc/FPVZ6HpO.EbG
```

```php
$password = "secret";
$hash = password_hash($password, PASSWORD_ARGON2I);
echo $hash;
```

3X

```
$argon2i$v=19$m=65536,t=4,p=1$eGFEUFRMLzI3Qlpma3hLcw$r/jvge
I5hzaSRzUoJLsQlqj3ftgwBZsGu4ebrGk6QIk
```

Vérifier un mot de passe

Fonction permettant de vérifier un mot de passe

password_verify(...)

Vérifier un mot de passe

```php
$password = "secret";
$hash = password_hash($password, PASSWORD_ARGON2I);

if (password_verify($password, $hash)) {
    echo 'Le mot de passe est valide !';
} else {
    echo 'Le mot de passe est invalide !';
}
```

Vérifier un mot de passe

Principe

```
password_verify($password, $hash)
```

1

Extrait le sel de $hash

2

Hachage de $password avec le même sel

3

Compare les 2 hash