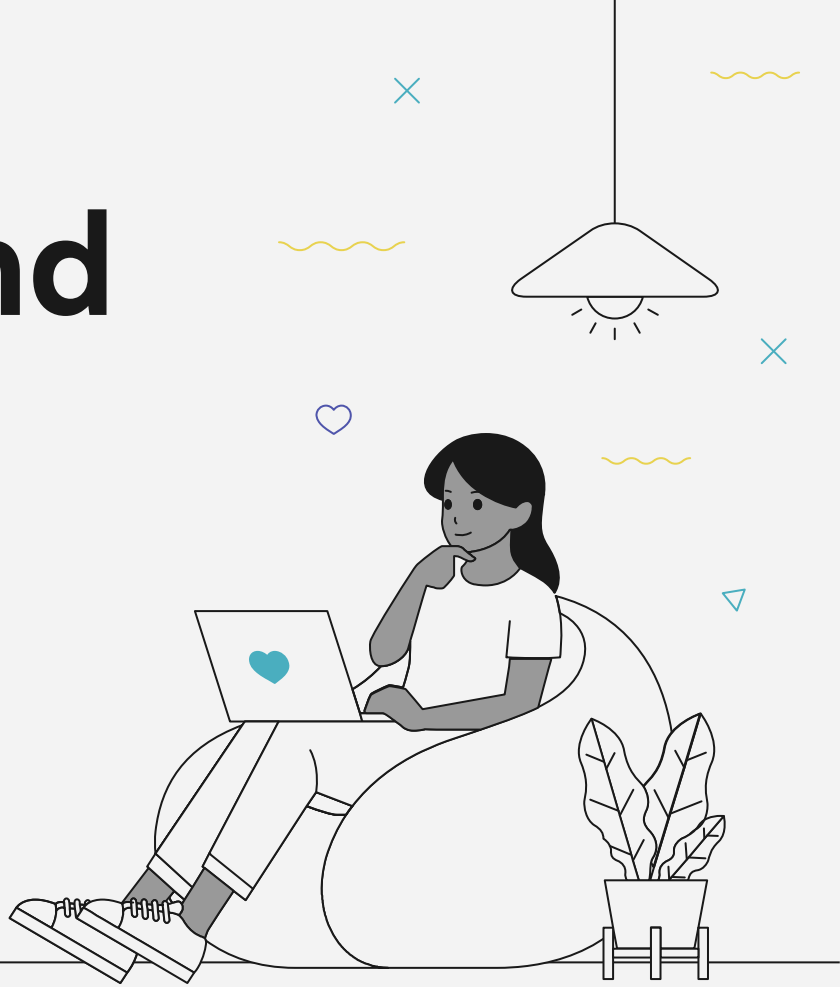
Encryption and secure your database



Léo FERRETTI, Toni DA RODDA, Tibo PENDINO, Alexandre GRARE

Table of contents

01

Secure password in your database

02

Hands-on

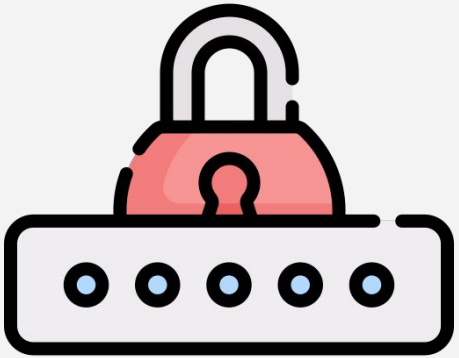
01

Introduction

Example user database

Username	Password
Alice	hello_world
Bob	123456
Leo	hello_world

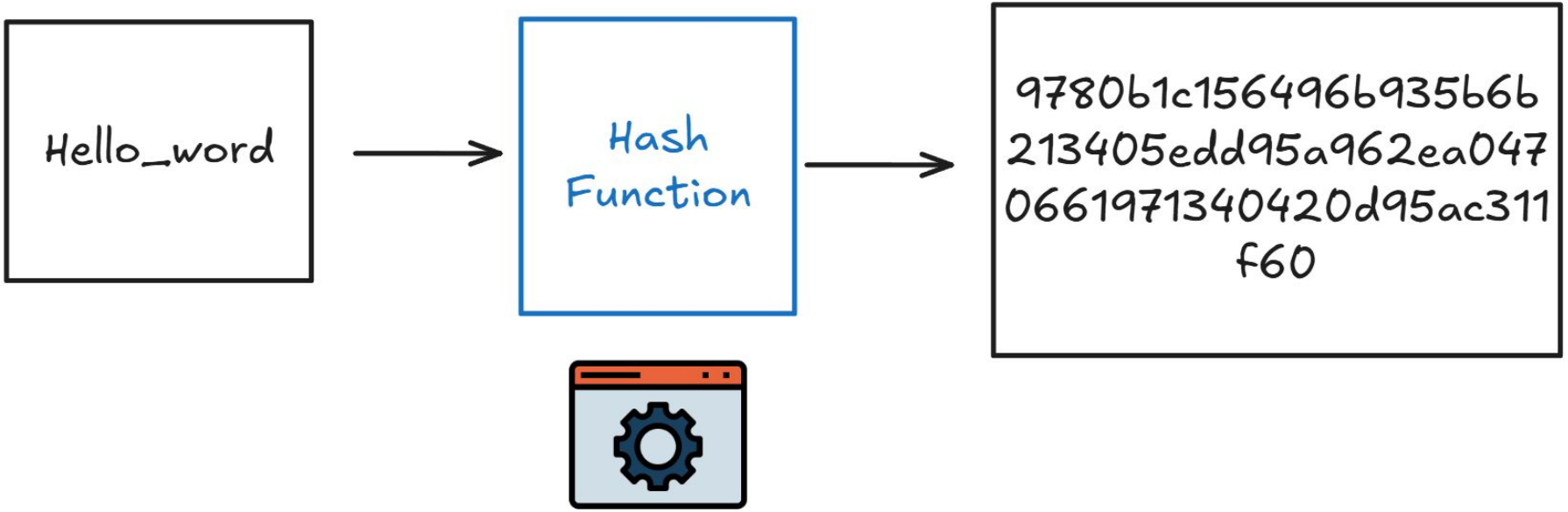
1) Hash your password



Hashing is an Algorithm that converts input data.

- **Deterministic:** Same input always produces the same hash
- **Irreversible:** Impossible to retrieve original data from the hash.

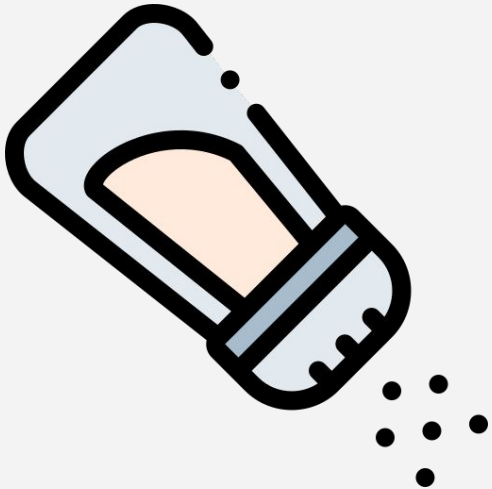
1) Hash your password



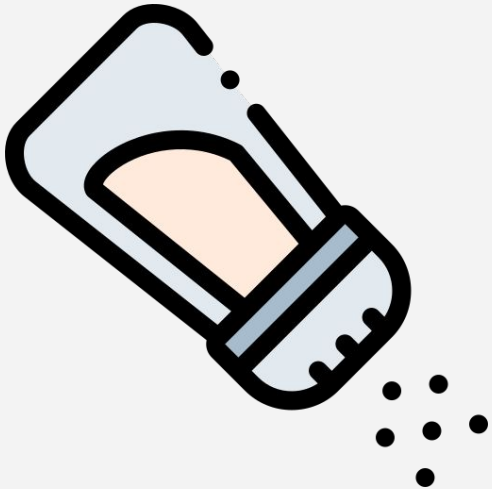
1) Hash your password

Username	Password	Hash_password
Alice	hello_world	9780b1c156496b935b6
Bob	password	4355e15a63214d46b5d
Leo	hello_word	9780b1c156496b935b6

2) Add Salt and Pepper



2) Add Salt and Pepper



A random value added to a password before hashing.

- **Purpose:** Prevents identical passwords from having the same hash.
- **Uniqueness:** Each password gets a unique salt, making attacks on multiple accounts harder.

2) Add Salt and Pepper

Username	Input_Password	**Salt**	**Hash_with_salt_password**
Alice	hello_world	9780b1f5z	161e5c1fa7425e73043362938b9824
Bob	password	2cf24dba5f	4355e15a63214d46b5d
Leo	hello_word	c156496b9	b0a30e26e83b2ac5b9e29e1b

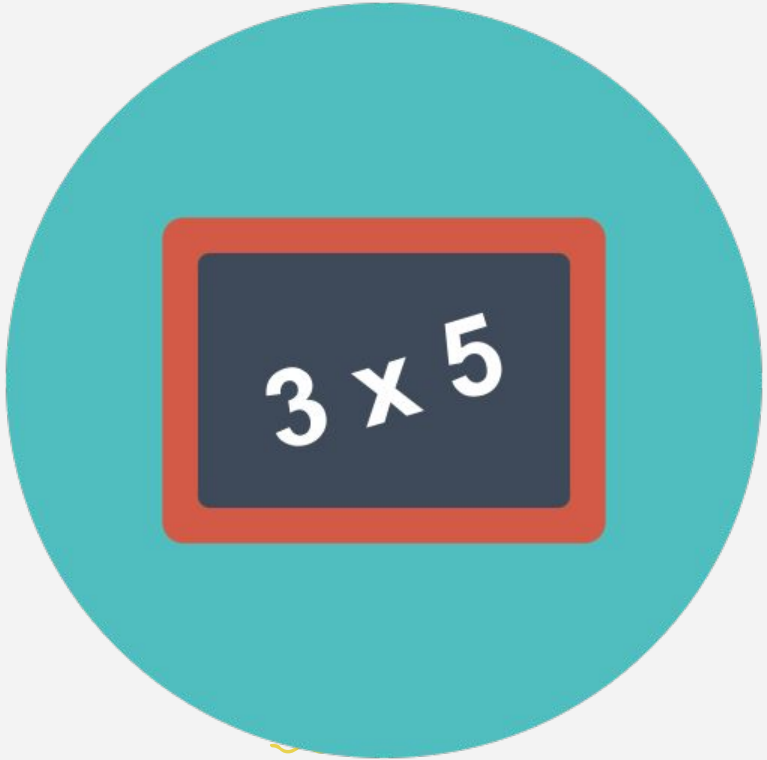
2) Add Salt and Pepper

A secret, fixed value added to passwords before or after hashing

- **Purpose:** Adds an additional layer of security beyond salt
- **Difference from Salt:**
  - Salt is unique for each password and stored alongside it.
  - Pepper is a shared secret, not stored in the database.
- **Enhances Security**: Even if the database is compromised, the pepper adds complexity to cracking hashes

3) Multiply your hash

- **Slows down** brute-force attacks.
- **Adds complexity** to reverse-engineering hashes.



03

Hands-On

https://github.com/etna-alternance/c2wk-2024

Thanks!