ADR-5 Bcrypt

Context

We had to find an encryption framework which was easy to use and the best one we could find was Bcrypt. It is important that a user’s password is encrypted before it is stored on the database. For security purposes, in case the database is hacked only hashed passwords will be obtained which are not what the user knows as their password.

Decision

Bcrypt was used in this project to provide advanced password security. Bycrypt uses Hashing algorithms that are one-way functions.In case the database is compromised only irreversible passwords will be obtained, as a result of hashing. They also have the property that if the input changes by even a tiny bit, the resulting hash is completely different. Bcrypt allows us to choose the value of *saltRounds*, which let us regulate data processing.An added advantage to using Bcrypt is the ability to generate random bytes (the salt) and combining it with the password before hashing creates unique hashes for each users password.Hashing Bcrypt is desirable as it does not require an API key for setup. It is to be noted that bcrypt prevents rainbow table attack and brute force search attacks.

Status

Accepted

Consequences

Bcrypt does not require an API key for setup and it is convenient to generate salts and complete password hashing. Implementation was not time consuming as one did not have to learn what the hashing functions do.