Handling Passwords, Secret values/keys, Credit Card info, CPR etc.

The exercises in this document, will be a mix of theoretical and practical questions and you should take them as guidelines for what we would like to discuss, if you draw a question like this at the exam

a) Consider and write down the Pros & Cons in the two overall strategies for how to handle User passwords.

- **You** save your users passwords, and guarantee to do it in a safe way
- **Someone else** saves your users passwords.

b) Consider and write down why it is considered a BIG security flaw to store user passwords in plain text, even if you are just implementing a simple site for "your son's football club"

c) Is one way hash functions like md5 and sha1 considered safe, if not what is the problem with these algorithms?

d) Consider and write down your interpretation of the terms: Hashing, Dictionary Attacks, Rainbow Tables, Salting and BCrypt.

Timing Hash Algorithms:

Create a new plain Java Maven Project, and include this dependency in your pom-file:

```
<dependency>
    <groupId>org.mindrot</groupId>
    <artifactId>jbcrypt</artifactId>
    <version>0.4</version>
</dependency>
```

Use the password: `My_dog_fido_2800` as the original value for all the calculations requested below:

a) Use one of the strategies found here, to calculate the SHA-256 hash (as a hex string) for the password `My_dog_fido_2800`.

a1) Use this page https://crackstation.net/ for a quick check as to whether this password, was a "wisely" chosen password. If yes, what was good, if no, what is the problem?

b) Measure in ms, using Java's System.nanoTime(), the time it takes to calculate the hash in step-a (1ms = 1.000.000 ns).

c) Use JBcrypt (include in the pom-file above) to create a bcrypt hashed/salted value of the password:

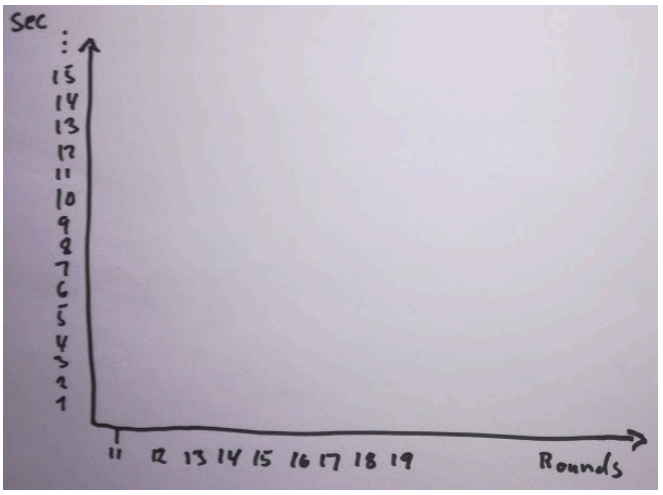
```
String hashed = BCrypt.hashpw(password, BCrypt.gensalt());
```

d) Measure the time in ms, it takes to calculate the hash value requested in step c

e) Compare the calculated times, for the two algorithms and reflect on your observations (which is fastest, and what is best, fast or slow?)

f) The BCrypt.genSalt(x) method takes an argument (default value is 10). What is the purpose of this argument?

g) This part requires two things, which many of you seems to have forgot, *pen* and *paper* ;-)



Write an x,y diagram as sketched in this figure (the "precision" sketched here will be more than enough for this exercise).
Recalculate the BCrypt value for the password, for increasing *log_rounds* values given to the genSalt(..) method. Plot in the time (in seconds).

When you are done (your will hopefully know when quickly), take a photo of your observations, and save it for further reference (the exam).

Explain, the result, and why the observed feature of the algorithm is important for a modern future-proof hash/salting algorithm.

h) Consider, and write down, why it is possible for us to store only one value in the database, and not having to consider separate handling of hash and salt values (explain the colored parts below, to backup your answer).

$2a$10$bTimb9RjOI4H5TOXuIKr4uWF6dc0MSLmzrMY3SNP.lGDKPB6HBrHa

A password safe Login-system:

Design a simple proof-of-concept login system, using BCrypt in whatever language/platform you prefer (*continuing with the jwt-example from the authentication lecture would be fine*)

How to handle our own secrets (db-credentials, API-keys, session secrets, keys for jwt-signing etc.)

a) Consider and write down different ways (and their pros & cons) to handle our own secrets, often needed in our code, when code is deployed to the deployment server.

b) Provide a "safe" example to handle, as a minimum.

- Credentials to the database(s) (*does size/complexity of passwords matter*)
- Secret(s) used to sign JWT's (*does size/complexity of secrest matter)*
- (Secret used to sign (session)cookies *(does size/complexity of passwords matter))*

Supplement the example with a short deployment instruction, explaining, how and what to do, to deploy your example.

Handling Credit Card Information

Consider and write down the "steps" required to handle users Credit Card information. What are the implications of "not following" the rules for this business scenario?