

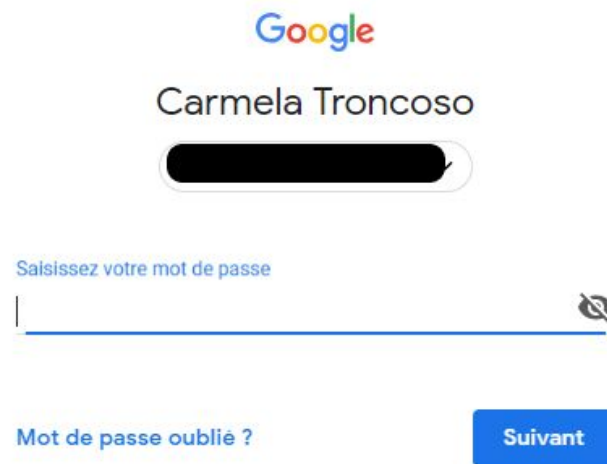
Exercises - Week 5

Authentication

1. Why are both (user, password) required for authentication? Why can't we use just a password for authentication?

Authentication is to verify the validity of a claimed identity. If there is no username, there is no claimed identity. One could have a valid password but, what permissions would be assigned?

[Note that sometimes, e.g., re-entering a password for your email client, you do not explicitly write the username, but the client already has it. Example:



The screenshot shows a Google login page. At the top is the Google logo. Below it, the name 'Carmela Troncoso' is displayed next to a circular profile picture placeholder. Underneath is a password input field with the placeholder text 'Saisissez votre mot de passe'. To the right of the field is an eye icon for toggling visibility. Below the field are two links: 'Mot de passe oublié ?' and a blue button labeled 'Suivant'.

]

2. Servers should store hashed passwords instead of storing them as plaintexts. Would hashing the password in the client, before sending it to the server through the encrypted channel bring any security advantage? How would the server side of the authentication process change if we have hashing at the client? How would the changes affect security?

It does not really add any advantage, and in fact introduces a problem with respect to only encrypting. How would we check at the server side? Two options:

- *Store the password in the clear so that when the hash comes one does:*
$$\text{received_hash} = \text{hash}(\text{stored_password})$$

We are again storing cleartext passwords.

- *Store the hash at the server. Then the hash becomes the “cleartext” password. If the adversary gets hold of the password file, then she can just use those to log in as the users.*

3. In class, we talked a lot about how to counter offline attacks against password cracking. Please discuss online attacks (i.e., attacks where the adversary tries to guess a password by interacting with the system). Are they easier to thwart than offline attacks? What are the typical defenses against them? What is the rationale behind these defenses?

Online attacks are easier to thwart than offline attacks because one can try to reduce the capabilities of the adversary.

The most typical defenses are:

- *Limiting the number of trials that can be done*
- *Using captchas to ensure that a human is introducing the password*

Both methods, as mentioned above, have a goal of restricting the adversary: restrict the number of attempts, and avoid attacks which can be automatically performed by a machine.

Stealing a hashed password storage is an example of changing an online attack to an offline attack. If the attacker needs to connect to a remote server to check password acceptance then the server can limit the interaction, but if the attacker can check with the stolen storage locally then the server cannot limit attacker’s capabilities.

4. Why is having a slow hash function that hinders offline attacks not a problem for the authentication process?

It is not a problem because during authentication the slow hash function is run only once. Therefore it does not affect the usability of the system.

In an offline attack, the adversary has to repeat the operation a large number of times. Therefore, the increase in time from a normal hash function to a slow hash function, which in one execution is not really important, becomes significant.

5. Why do we need to include challenges in authentication protocols? What can go wrong if they are not added?

We need to include challenges to ensure that the authentication information is sent on the moment, i.e., that there is not a replay attack.

If we do not include them, an adversary may copy the authentication information from a previous transaction and use it to gain access to the system later on.

6. If you steal the biometric template from a database with server-side processing of biometrics, can you use it directly to fool authentication in another system with server-side processing?

No, you cannot. When this server receives the template as input, it will process it as if it was a raw biometric, resulting in a new value which would make the comparison with the stored template fail.

7. If you were to implement a biometric authentication system for

(a) Entering a military base

(b) Using your loyalty card at the supermarket

How would you select the parameters for the biometric algorithm? Justify.

(a) The parameters should be set such that the rate of false positives (unauthorized users access the base) is very low. It is not crucial that the rate of false negatives (authorized users are denied access) is somewhat high. Given the security requirements of military resources, having to repeat the authentication is not a big deal.

(b) The parameters should be set such that the rate of false negatives (authorized users not being able to access the card) is very low, and false positives can be high. A supermarket needs fast checkout, and it is not the end of the world if sometimes the loyalty card is misused.