

### Exercise 1

Design a client-server protocol for a file storage provider (like Dropbox). For simplicity, there is only one directory. The client should be able to execute the CRUD operations (create, read, update, delete). The server cannot be trusted, so the client should store a hash for every file in order to verify file contents fetched from the server.

The code for the server is given below. Implement the CRUD operations on the client, which will call server operations.

Which security property of the hash function does the client rely on in order not to have to trust the server?

### Implementation Server

```
code/server.py

"""
Implementation Server
Server stores dictionary stored_content
key: filename
value: content
"""

stored_content = {}

def create(name, content):
    stored_content[name] = content

def read(name):
    return stored_content[name]

def update(name, content):
    stored_content[name] = content

def delete(name):
    del stored_content[name]
```

### Exercise 2

Write functions that produce and verify proof-of-work for SHA-256. You can use any language you like (I used Python).

There should be two functions:

1. `solve(p,d)`, which, given a puzzle string `p` and a difficulty `d` returns a puzzle solution `nonce`

2. `verify(p,nonce)`, which, given a puzzle string `p` and a puzzle solution `nonce` returns the difficulty `d` of the work that has been done to find the solution.