Advanced Topics in Online Privacy and Cybersecurity (67515), Fall 2022

Assignment 1: ORAM

Due date: December 20$^{\text{th}}$, 2022

Introduction

Your task in this assignment is to implement a storage application that hides client access patterns from a remote server - otherwise known as an ORAM. The specific variant of ORAM that you are required to implement is Path ORAM first mentioned in [**?**].

Programming task (functionality)

You are required to implement a client and server.
Definitions:

- **id** - an integer
- **data** - a string with 4 characters

Requirements:

- Server is initialized to support N datablocks
- The client can store data associated with an ID on the server by calling **self.store_data(server, id, data)**
- Given a name, a client can retrieve associated data by calling by calling **self.retrieve_data(server, id, data),** the client should return None if the data does not exist.
- Client can delete data associated with an ID from server by calling **self.delete_data(server, id, data)**
- The client has access to constant memory (can only store data on the server)

Note that the server is data storage, and that the client is supposed to interact directly with said data storage in an oblivious manner.

Programming task (security)

The server should be oblivious to data content and the client's access patterns. This should be accomplished by use of end to end encryption and PATH ORAM

The server should be unable to trick the client into accepting corrupt or outdated data (data integrity). This should be accomplished by use of authentication.

You may use thislibrary for encryption and verification.

Submission Guidelines

Submit your code in **Python** along with a pdf design document. In the design document describe the architecture you built and how it satisfies the above requirements. Additionally, include the following performance benchmarks:

- Throughput (number of requests/sec) vs. N (DB size)
- Latency (time to complete a request) vs. throughput

Make sure to write a brief discussion of your benchmarks, mention how many runs of your ORAM you averaged to receive your results, and state the units you are using clearly.

Does your implementation benefit from multicore?

Good Luck!

References