

Foundations of Cybersecurity

Project documentation

Irene Cantini

Federica Perrone

Content

[Introduction 3](#_Toc108544014)

[Design choices 4](#_Toc108544015)

[Operations’ implementation 5](#_Toc108544016)

[Upload 5](#_Toc108544017)

[Download 5](#_Toc108544018)

[Delete 5](#_Toc108544019)

[List 5](#_Toc108544020)

[Rename 5](#_Toc108544021)

[Logout 5](#_Toc108544022)

[Meeting the security requirements 6](#_Toc108544023)

[User manual 7](#_Toc108544024)

# Introduction

This project is about the implementation of a Client-Server application that resembles Cloud Storage.

Each user has a dedicated storage on the server, and each of them can only access their own dedicated storage.

After the authentication phase, in which the client must authenticate with the server, the client can upload, download, rename or delete data to/from the Cloud Storage in a safe manner.

Users are pre-registered on the server, specifically there are four users. Each user has a long-term RSA key-pair, and the long-term private key is password-protected. The server knows the username of every registered user and the RSA public key of every user. For each registered user, the server has already allocated the user dedicated storage. In the following table (Table 1) are show the four pre-registered users with their username and password.

|  |  |  |
| --- | --- | --- |
| User | Username | Password |
| Alice | alice | alice |
| Bob | bob | bobb |
| Carol | carol | carol |
| Dave | dave | dave |

Table 1 - Pre-registered users

# Design choices

When a user logs in the application, (s)he must authenticate itself with the server.

# Operations’ implementation

## Upload

## Download

## Delete

## List

## Rename

## Logout

# Meeting the security requirements

The security requirements where **Perfect Forward Secrecy**, **encryption and authentication** of the entire session and **resistance to reply attacks**.

**Perfect Forward Secrecy** is guaranteed by the *Ephemeral Diffie-Hellman key exchange*.

**Encryption and authentication** is achieved using *authenticated encryption*. In particular, all the session messages are been encrypted using *AES* with a key length of 128 bits as a block cipher and *Galois-counter mode* as the encryption mode. For each symmetric encryption a random generated IV is used to avoid predictability.

**Resistance to reply attacks** in the authentication phase is given by the use of *Ephemeral Diffie-Hellman*, while during data exchange is given by an incremental counter that is sent in the AAD field of every session message.

# User manual