RDA

Remote Document Access

Group 16

58122, Diogo de Carvalho e Pereira
65937, Ricardo Filipe Fonseca Silva
70584, João Daniel Jorge Machado

Taguspark

Overview

- **Problem**
- Requirements
- Solution
- Demonstration

Cloud Services

- Cloud Storage and File-Sharing are used by millions worldwide
  - As common as sending/receiving emails.
  - Powerful collaborative tool.
  - Online document backup.
- Problems?

Problems

- Most cloud services require the user to trust the service in regards to **confidentiality:**
  - Server has full access to the user's data
  - Server may share the data with unauthorized parties.
  - Data is not safe in case of a server breach
  - **No Privacy**

Example

- Alice wants to backup some files online in case she loses her laptop.
  - Alice finds a cloud storage service that offers no data confidentiality
  - Cloud service provider scans Alice's files and finds some interesting information
  - Provider sells the information found on Alice's files to a 3rd party.

Solution?

- Keep the data contents safe from **all** unauthorised sources.
  - Including the service provider.
- How?
  - Locally encrypt the files before uploading them
  - Don’t allow the encryption information to leave your machine.

Overview

- Problem
- **Requirements**
- Solution
- Demonstration

Communication

- Authentication
  - Mechanisms to counter man-in-the-middle and spoofing attacks
- Authorization
  - Verify that a user has the required permissions to perform a particular task
  - Modify (add/revoke) user permissions

Data Transfer

- Confidentiality
  - Secure transfer between the user and the server
  - Store documents so that only the owner and other authorized users have access to the data within them
  - The server should not be able to access the content of the documents

Document Contents

- Integrity
  - Detect and warn the user about possible access violations and data corruption
- Non-repudiation
  - Users must be able to verify document authorship

Overview

- Problem
- Requirements
- **Solution**
- Demonstration

Solution

- Java RMI
- Client's Box
  - Contains all document keys
- Document encryption is done locally
  - Secret Keys
- Sharing a document key
  - Through Client's bin

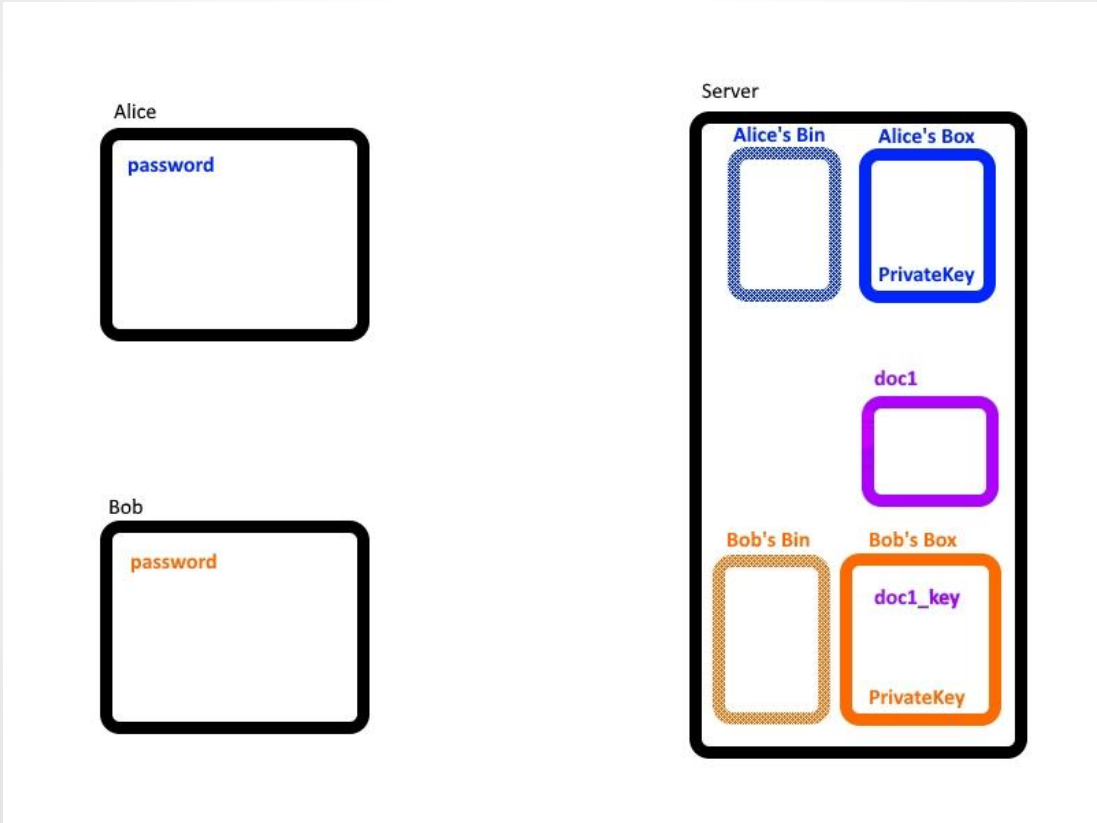
Custom Concepts Used

- Client’s box
  - Encrypted with password-based secret key
  - Only owner can change it
- Client’s bin
  - Encrypted with the client’s public key
  - Everyone can add files
    - But only the owner can get them

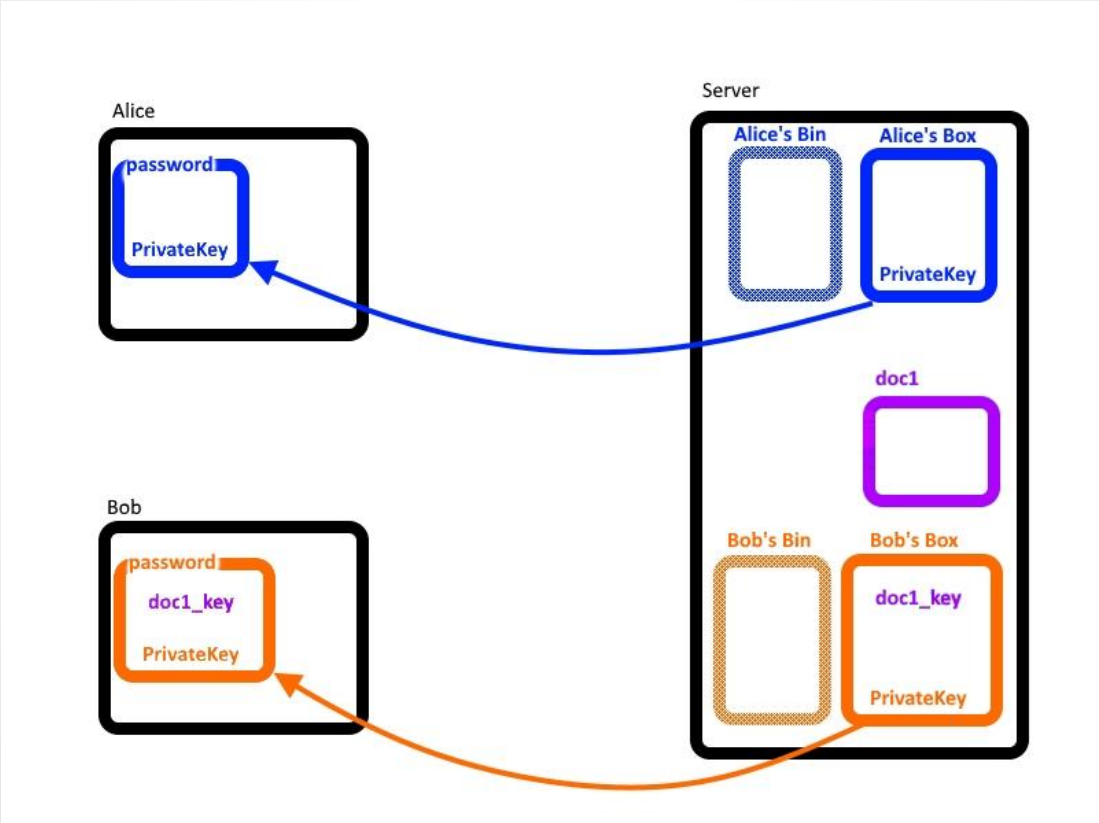
Example’s Context

- **Alice** is registered
- **Bob** is registered
- **Bob** creates a document named **doc1**
- **Bob** wants to share **doc1** with **Alice**

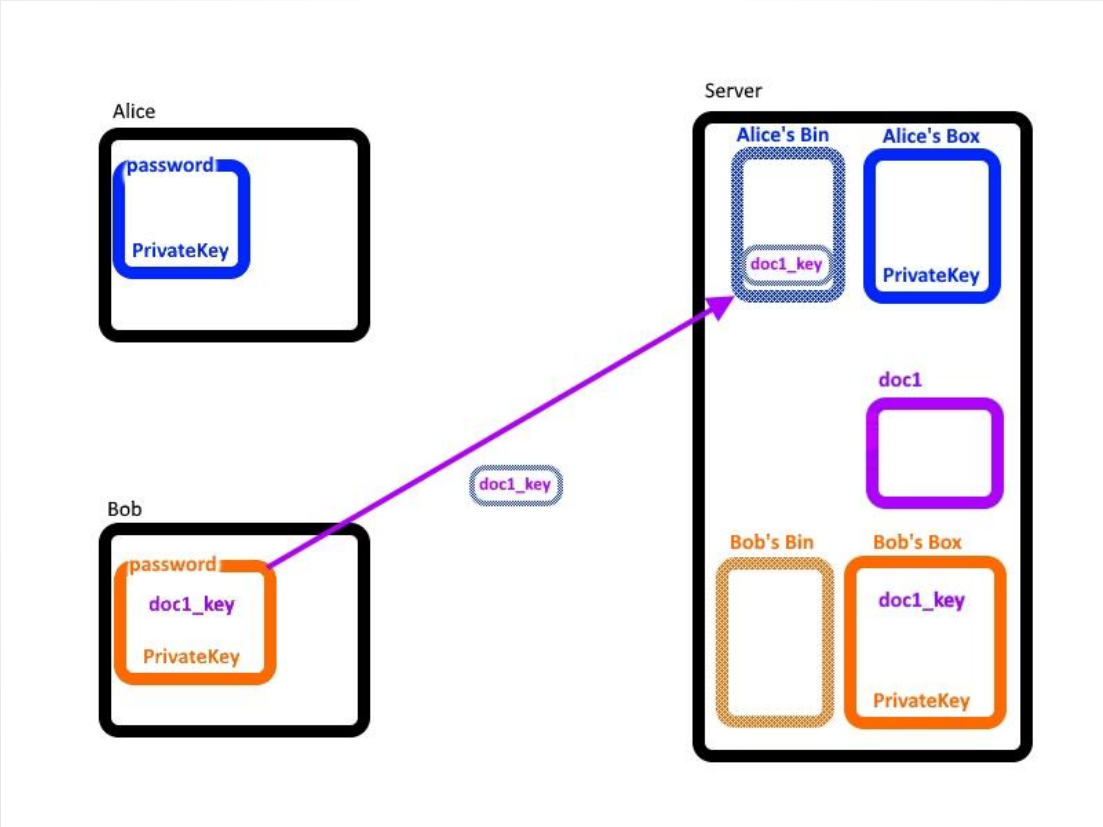
Layout


Alice
password
Bob
password
Server
Alice's Bin
Alice's Box
PrivateKey
doc1
Bob's Bin
Bob's Box
doc1_key
PrivateKey

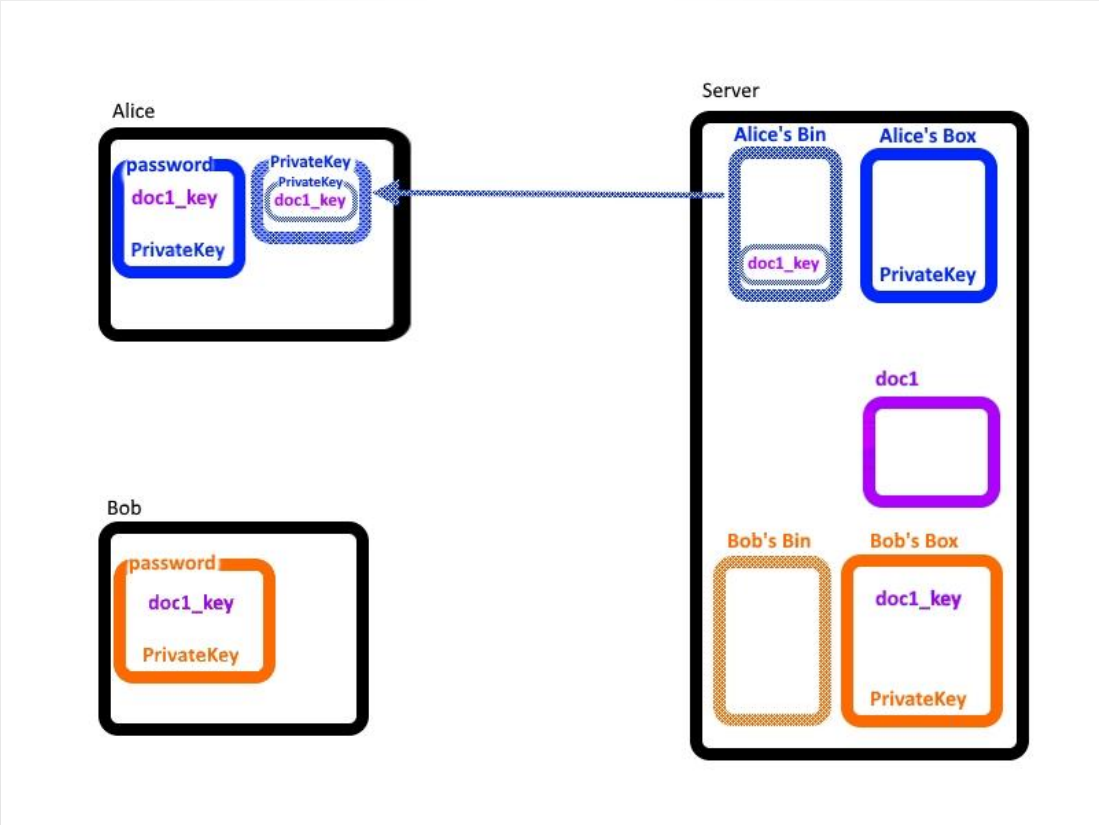
Fetching the Client Box



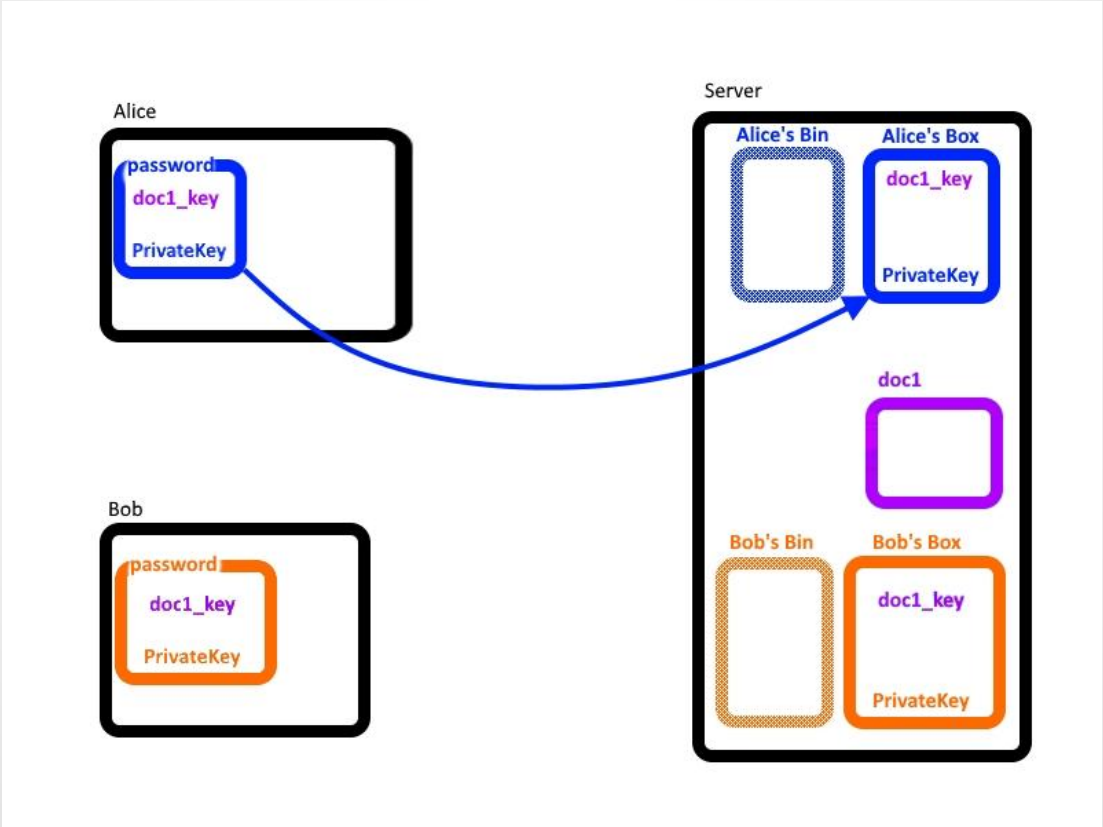
Key Sharing


Alice
password
PrivateKey
Bob
password
doc1_key
PrivateKey
doc1_key
Server
Alice's Bin
Alice's Box
doc1_key
PrivateKey
doc1
Bob's Bin
Bob's Box
doc1_key
PrivateKey

Accessing the Client Bin



Synchronizing the Client Box



Overview

- Problem
- Requirements
- Solution
- **Demonstration**

Demonstration

- [Youtube]

Q & A