EncryptFS: Encrypted Integrity-Preserving Filesystem

Submission

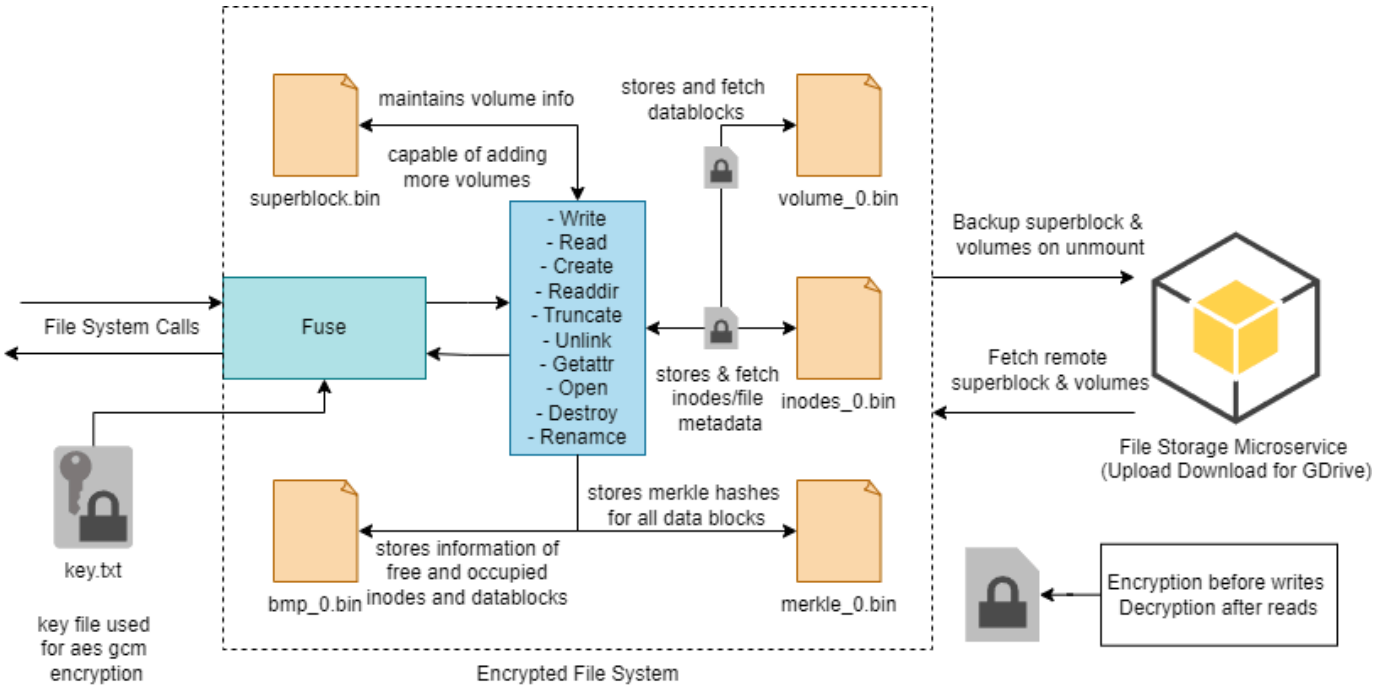
- Run instructions and README.md
- Report
- Demo Documentation

Authors

- Kaushal Patil (653639939)
- Vasu Garg (676073104)

Key Features

- FUSE-based Implementation: Operates in user-space, allowing for enhanced security without kernel modifications.
- AES GCM Encryption: Provides strong encryption and authentication for data.
- Dynamically Expanding Volumes: Facilitates on-the-fly storage expansion to accommodate growing data needs.
- Cloud Backup and Restoration: Supports secure cloud storage solutions for disaster recovery.
- Merkle Tree Integrity Verification: Ensures data remains unaltered and secure against unauthorized changes.



System Requirements

- Linux OS with FUSE support
- Development tools like GCC
- Libraries: FUSE, OpenSSL, libsodium, libcurl

Installation

Install Required Libraries for EncryptFS

```
sudo apt-get install libfuse-dev libssl-dev libsodium-dev libcurl4-openssl-dev # all the required libraries
```

Install Required Libraries for Cloud Storage (Optional)

```
pip insall -r requirements.txt # all the required libraries for cloud storage microservice
```

Build Instructions for EncryptFS

```
make # build the encryptFS
```

Key Generation

```
./encryptFS.out keygen ./key.txt # generate a new key 'path is customizable'
```

Mounting EncryptFS (Local)

Example:

```
./encryptFS.out -f -d ~/hello ./superblock.bin ./key.txt # mount the encryptFS to ~/hello using superblock.bin
```

Mounting EncryptFS (Cloud)

> Requires the Cloud Storage service to be running and authenticated (Have the tokens stored in the tokens.txt file)

Example:

```
./encryptFS.out -f -d ~/hello remote:encryptfs/superblock.bin ./key.txt  # mount the encryptFS to ~/hello using
```

> Your google drive should contain a folder named `encryptfs` and the superblock.bin file should be present in that folder, if its not present then the program will create a new superblock.bin file in the folder.

Unmounting EncryptFS

```
fusermount -u ~/hello # unmount  encryptFS
```

Clean Up

```
make clean # clean the build and volume files
```

> clean the build files and volumes created by encryptFS, store the superblock.bin and associated files in the cloud storage or somewhere else before calling this command

Running Cloud Storage Service

Setup for Google Drive API

1. Go to the Google Cloud Console.
2. Create a new project.
3. Enable the Google Drive API.
4. Create OAuth 2.0 credentials.
5. Download the credentials as a JSON file and save it as `credentials.json` in the same directory as the `cloud_storage.py` file.

Running the Service

```
python cloud_storage.py
```

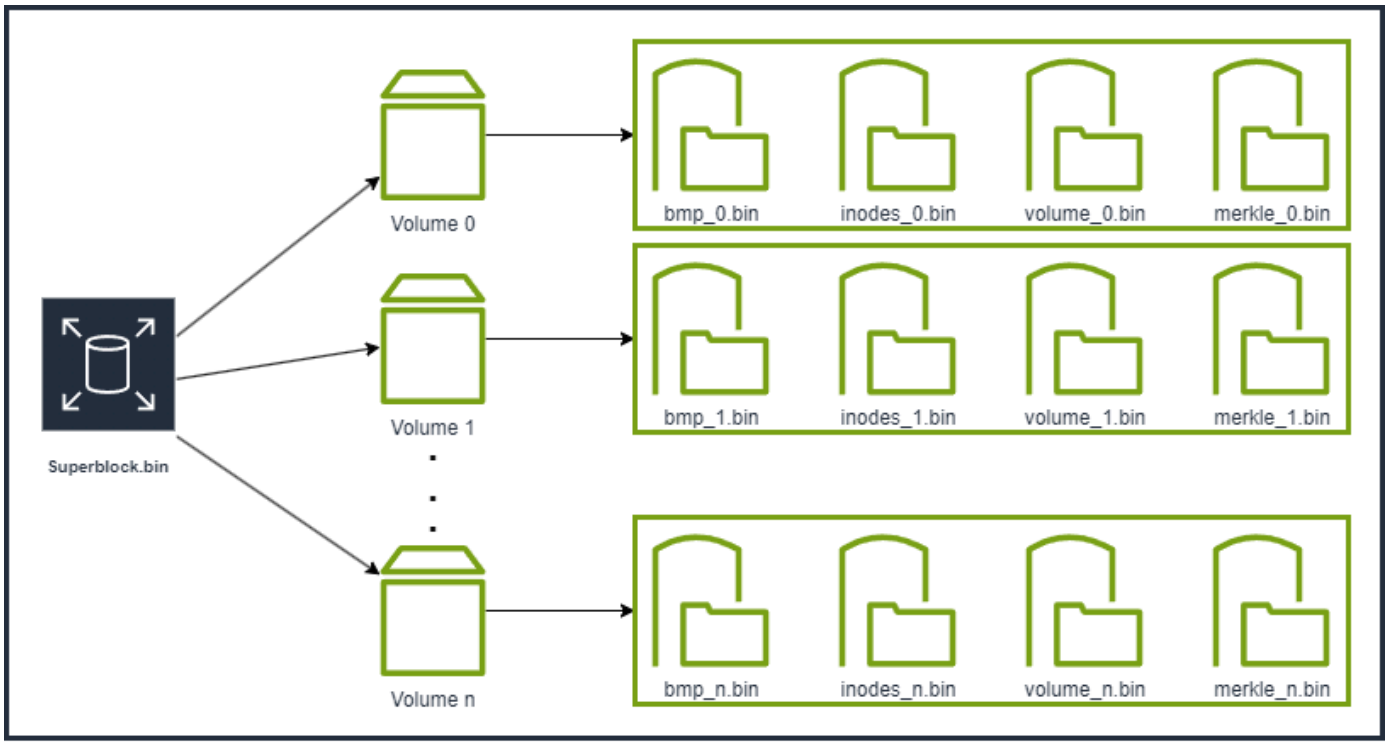
> for creating the tokens.txt to authenticate the cloud storage service, visit service url and authenticate the service, make sure the tokens.txt is present.

Testing Instructions

```
cd ~/hello # Mount point
touch cat.txt # Create a file
echo "hello my name is cat. And I am very cute hehehe hello " > cat.txt # Write to the file, updated merkle tre
cat cat.txt # Read the file, integrity check will be performed
ls -l # List the files, in the root directory
```

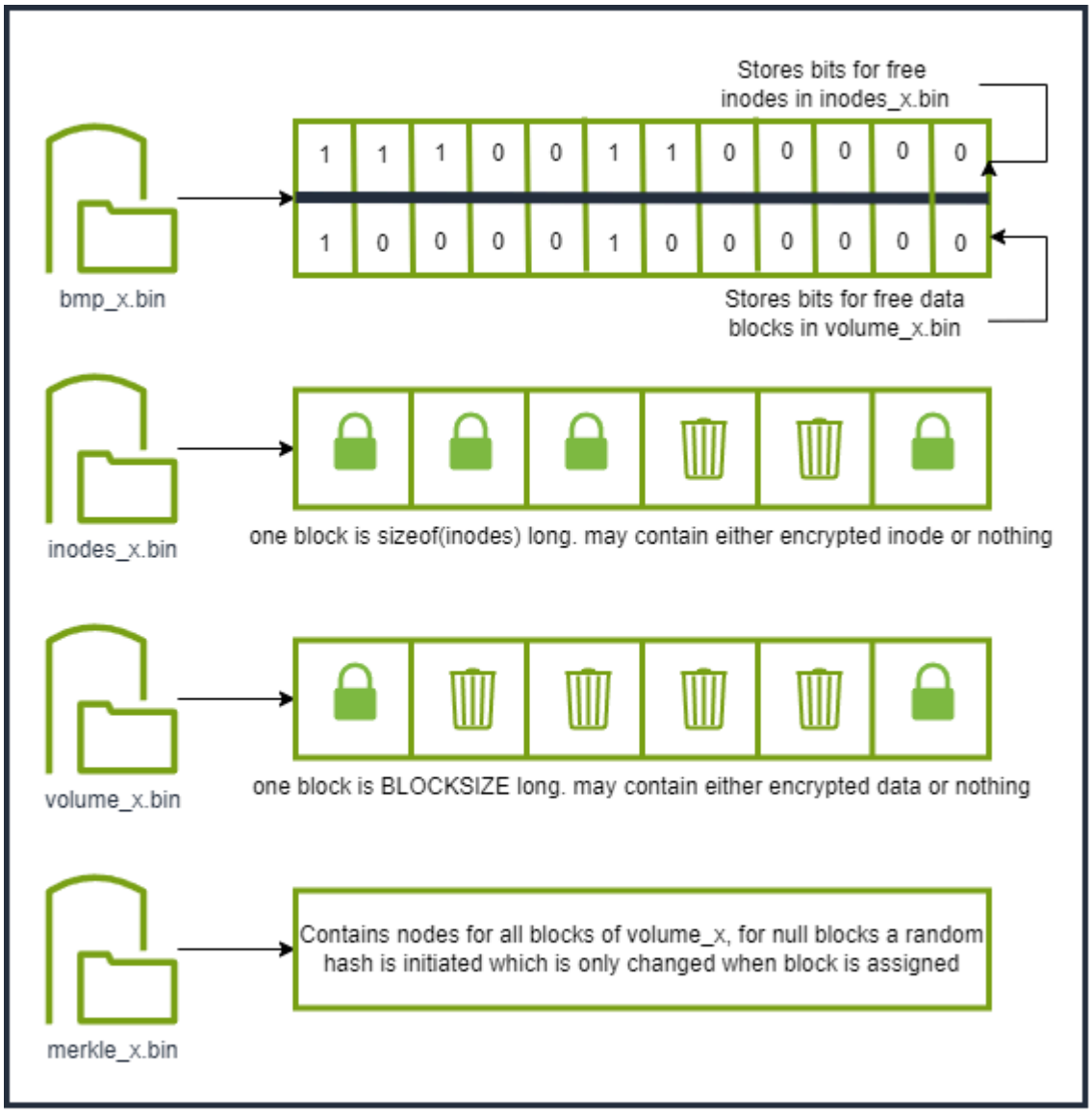
```
nano dog.txt # Create a new file you can paste content more then block size to check multi block allocation and
cat dog.txt # Read the file, integrity check will be performed
# other supported operations can be tested as well like cp, mv, rm -rf, etc.
# test image files
cp ./assets/cat.jpg ~/hello/
cp ./assets/cat.mp3 ~/hello/
feh ~/hello/cat.jpg # view the image
vlc ~/hello/cat.mp3 # play the audio
```

File System Structure



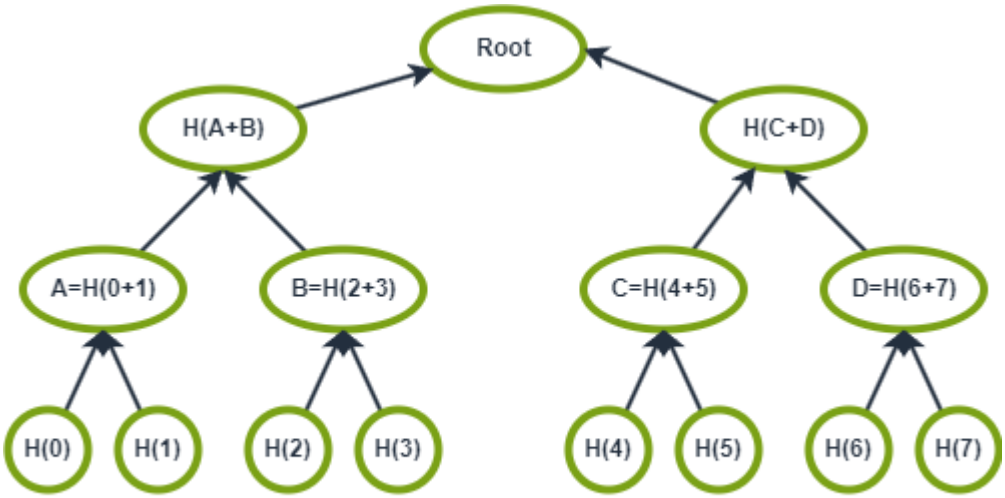
EncryptFS File System Data Structure

Internal Structure



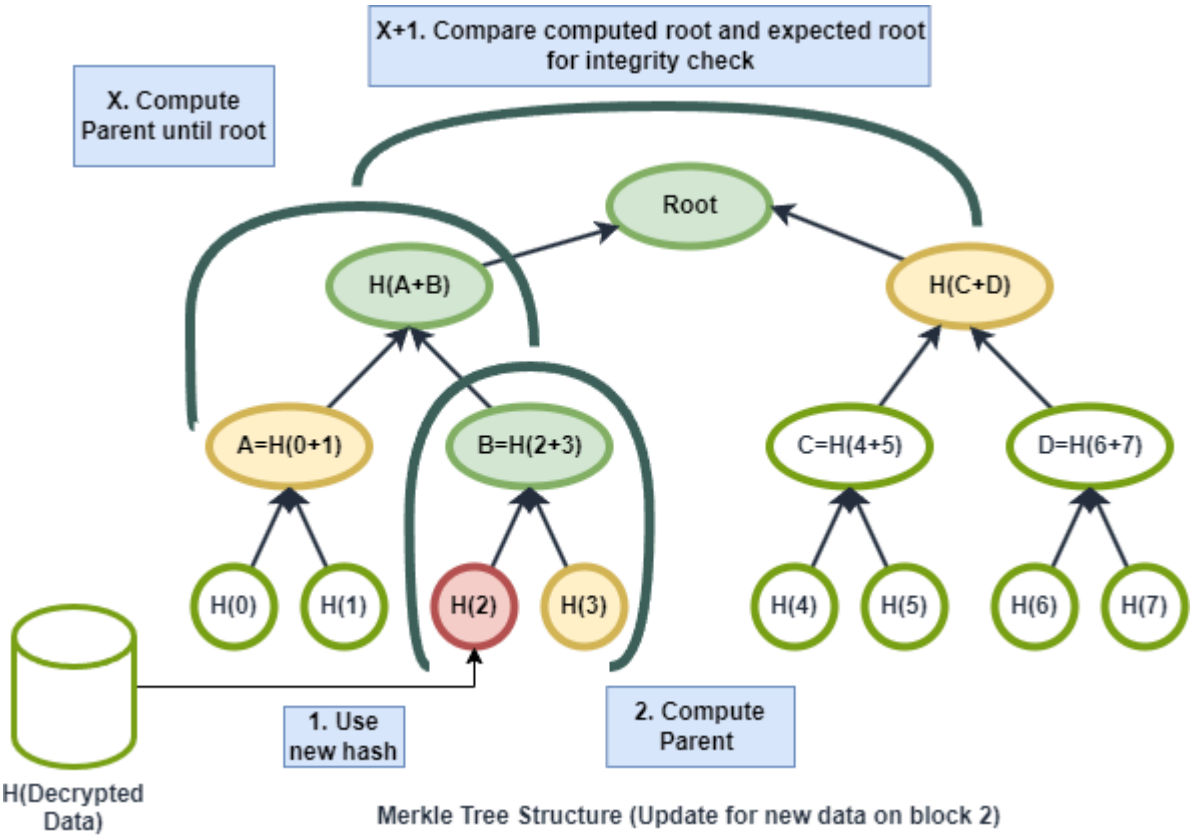
File Structure for encryptfs files

Merkle Tree Structure



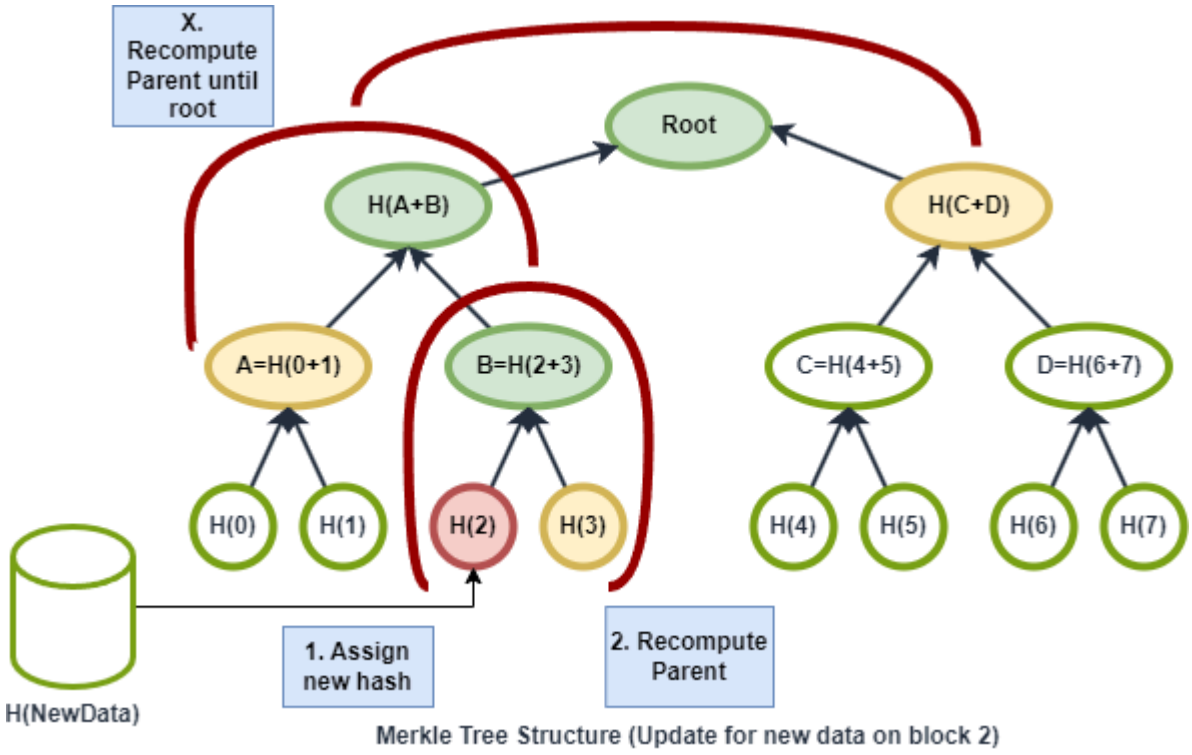
Merkle Tree Structure Built from data blocks

Merkle Tree Verification



Merkle Tree Structure (Update for new data on block 2)

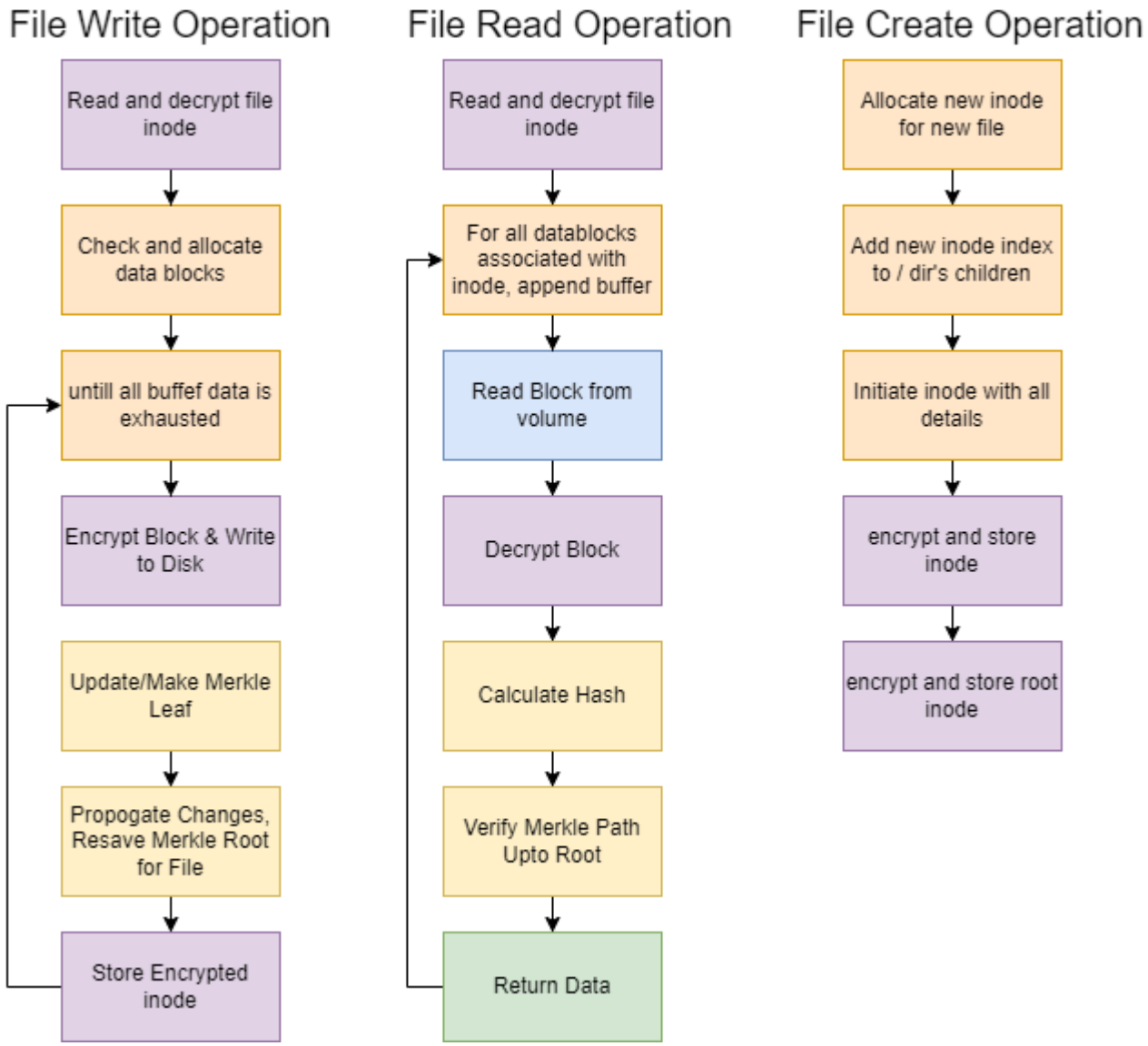
Merkle Tree Update



Merkle Tree Structure (Update for new data on block 2)

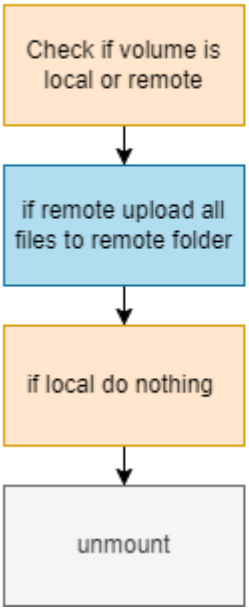
Algorithms

Create, Read, Write

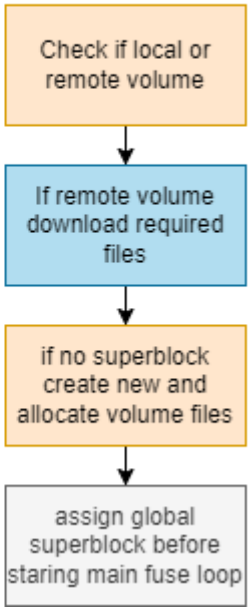


Mount, Unmount

Fuse Unounting

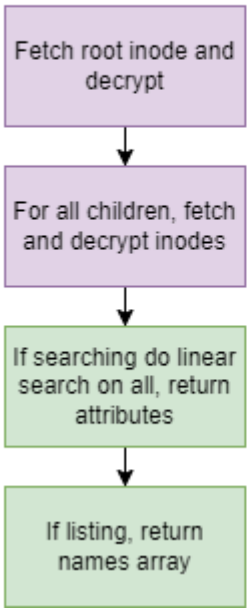


Fuse Mounting



Misc

Search Inode by path(getattr) /List all files



Dynamic Datablock/Inode allocation