Lab #3 - Merkle tree

Léo Gabaix

Francesco Mosanghini

1. Introduction

This lab can be split into 4 parts:

1. Build the tree
2. Update the tree
3. Generate a proof of membership for a given document at a specific position.
4. Verify the proof given the tree information, the document and the proof

🚫 Unfortunately we were unable to implement the proof verification in time.

1.1 Architecture

Example of architecture for a 4-leaves tree

```
.
├── LICENSE             # license
├── README.md           # instructions
├── builder.sh          # script to run
├── doc.pre             # prefix for documents
├── docs                # folder for documents
│   ├── doc0.txt
│   ├── doc1.txt
│   ├── doc2.txt
│   └── doc3.txt
├── node.pre            # prefix for nodes
├── nodes               # folder for nodes
│   ├── node0.0
│   ├── node0.1
│   ├── node0.2
│   ├── node0.3
│   ├── node1.0
│   ├── node1.1
│   └── node2.0
├── proof_x.txt        # proof of membership for docx.txt at position x
└── tree.txt           # tree information
```

We developed a bash script with the following functions:

1.2 Main functions

- `generate_tree()`

Generates all the nodes of all layers.

- `update_tree()`

Creates a documents at the end of the list, generates the corresponding leaf, then updates the necessary nodes.

- `generate_proof()`

Iterates through the tree starting from the given document to the root node, adding all the siblings nodes to the proof file.

- `verify_proof()`

Verify a given proof file with a given document and its position using the public information of the tree.

1.3 Secondary functions

- `main()`

Coordinates the main and secondary functions according to the user choice.

- `initialization()`

Checks for necessary files and directories.

- `clean_dirs()`

When called cleans the `docs/` and `nodes/` directories.

- `generate_documents()`

Allows the generation of x documents inside the `docs/` directory.

- `generate_leaves()`

Computes the documents hashes and store them inside the `nodes/` directory

- `generate_synthesis()`

Once the tree is generated, its information is stored into a synthesis file called `tree.txt`

2. How to run?

The script is named `builder.sh`.

2.1 Prerequisites

For some mathematical operations the `bc` package is used and so mandatory.

Install it with:

```
sudo apt install bc
```

Also the script will probably not have the execution rights

2.2 Modes

Because the script embeds different functions there are different way of running it.

2.2.1 Generate a tree

```
# build tree with x leaves
$ ./builder.sh build 4
```

2.2.2 Update tree

```
# update tree
$ ./builder.sh update
```

2.2.3 Generate proof

```
# generate proof for doc2.txt
$ ./builder.sh gproof 2
```

2.2.4 Verify proof

```
# verify proof for doc2.txt using proof_2.txt
$ ./builder.sh vproof 2 proof_2.txt
```

2.2.5 Clean workspace

Clean the workspace using:

```
# clean 'docs/' and 'nodes/' directories
$ ./builder.sh clean
```