


# Merkle tree

Gianluca Dini  
Dept. of Ingegneria dell'Informazione  
University of Pisa  
Email: [gianluca.dini@unipi.it](mailto:gianluca.dini@unipi.it)  
Version: 2022-04-12

1

## Brief history



- Ralph Merkle patented Merkle Trees in 1979
- Merkle published the paper in 1987
  - R.Merkle. A digital signature based on a conventional encryption function. CRYPTO 1987.
- Patent expired in 2002

apr. '22

Merkle Tree

2

2

# Merkle Tree (1979)

**Merkle Root**

$h_{ij} = H(\text{left son} \parallel \text{right son})$

$h_i = H(B_i)$

**Data blocks**

apr. '22      Merkle Tree      3

3

# Verification

- Verify whether B3 belongs to the data set
  - List of hashes (proof):  $\langle h_4, h_{12}, h_{58}, h_{18} \rangle$
  - Verification algorithm
    - Check whether  $H(H(h_{12}, H(H(B3), h_4)), h_{58}) == h_{18}$
    - Verify authenticity of the root  $h_{18}$
- Complexity  $O(\log n)$ , with B # of blocks

apr. '22      Merkle Tree      4

4

## Properties



UNIVERSITÀ DI PISA

- MT (or hash tree) allows efficient and secure verification of the contents of large data structures
- The root must be trusted
  - Digitally signed
  - Maintained on a trusted source/storage
- Verifying whether a leaf node is part of the MT requires computing a #hashes proportional to the logarithm of the #leaves
  - $O(\log B)$ , with B the number of leaves (blocks)

apr. '22

Merkle Tree

5

5

## Merkle Tree - applications



UNIVERSITÀ DI PISA

- |  |  |
|--|--|
| • File systems <ul style="list-style-type: none"><li>– IPFS, Btrfs, ZFS</li></ul>                      | • Backup Systems <ul style="list-style-type: none"><li>– Zeronet</li></ul>                       |
| • Content distribution protocols <ul style="list-style-type: none"><li>– Dat, Apache Wave</li></ul>    | • P2P networks <ul style="list-style-type: none"><li>– Torrent</li></ul>                         |
| • Distributed revision control system <ul style="list-style-type: none"><li>– Git, Mercurial</li></ul> | • NoSQL systems <ul style="list-style-type: none"><li>– Apache Cassandra, Riak, Dynamo</li></ul> |
| • Blockchain <ul style="list-style-type: none"><li>– Bitcoin, Ethereum</li></ul>                       | • Certificate Transparency framework   |

apr. '22

Merkle Tree

6

6

# Distributed scenario [→]

Content provider

1  
2  
...  
B

H

$h_f$

Untrusted peers

$p_1$   $p_2$  ...  $p_n$

$blk_i$

u

Trusted server

TS

$h_f$

apr. '22

Merkle Tree

7

7

# Distributed scenario


- How does the user know that the information that (s)he is getting from some peer is genuine and hasn't been tampered with (or corrupted)?

apr. '22

Merkle Tree

8

8


  
 UNIVERSITÀ DI PISA

## Distributed scenario

- Solution no. 1 (shown in the slide)
  - Trusted Server stores  $h_f$
- Verification
  - Upon receiving all blocks  $\{blk_i, 1 \leq i \leq B\}$ , compute  $h_f' = H(blk_1 \parallel blk_2 \parallel \dots \parallel blk_n)$ .
  - Return  $(h_f' == h_f)$
- Drawback
  - Check upon completion (possibly long delay)
  - Not possible to determine corrupted/compromised blocks

apr. '22
Merkle Tree
9

9

  
 UNIVERSITÀ DI PISA


## Distributed scenario

- Solution n.2
  - Trusted Server stores  $\langle h_f, h_1, h_2, \dots, h_B \rangle$  with  $h_i = H(blk_i)$ ,  $1 \leq i \leq B$
  - Number of hashes  $B = \text{sizeof}(\text{file})/\text{sizeof}(\text{block})$ 
    - Torrent: block size is 16 kbytes
- User Verification
  - The user can verify each block
- Drawback
  - Increase storage/bandwidth overhead

apr. '22
Merkle Tree
10

10

# Distributed scenario



UNIVERSITÀ DI PISA

- Solution n.3: Merkle Tree
  - Trusted Server stores the root of the Merkle Tree
  - Each peer stores
    - A subset of the blocks  $\{blk_i\}$ ;
    - For each block  $blk_i$ ,  $\langle blk_i, proof_i \rangle$
  - User Verification
    - Upon downloading a block  $blk_i$ , the user verifies it using  $proof_i$  and the tree root


apr. '22

Merkle Tree

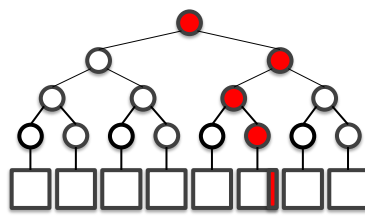
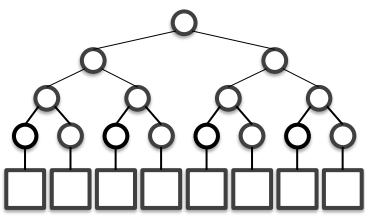
11

11

# File comparison



UNIVERSITÀ DI PISA



- File F gets modified in a block  $blk_i$
- Comparing files takes is  $O(B)$
- Comparing MTs is  $O(\log B)$

apr. '22

Merkle Tree

12

12

Foundations of Cybersecurity

6

# Replication

The diagram illustrates a replication system. At the top center is a circle labeled 'PC' (Primary Component). Below it are two circles labeled 'SC' (Secondary Component). To the right of the PC is a Merkle Tree with 16 leaf nodes (squares) and 4 internal nodes (circles). The root node is red. To the right of the bottom SC is another Merkle Tree with 16 leaf nodes and 4 internal nodes, all of which are white. The top SC also has a Merkle Tree, but it is partially obscured by the PC's tree. The text 'UNIVERSITÀ DI PISA' is in the top right corner. The footer contains 'apr. '22', 'Merkle Tree', and '13'.

# Replication

The diagram illustrates a replication system. At the top center is a circle labeled 'PC' (Primary Component). Below it are two circles labeled 'SC' (Secondary Component). To the right of the PC is a Merkle Tree with 16 leaf nodes (squares) and 4 internal nodes (circles). The root node is red. To the right of the bottom SC is another Merkle Tree with 16 leaf nodes and 4 internal nodes, all of which are white. The top SC also has a Merkle Tree, but it is partially obscured by the PC's tree. The text 'UNIVERSITÀ DI PISA' is in the top right corner. The footer contains 'apr. '22', 'Merkle Tree', and '14'.

- How can the primary replica determine whether a disconnected secondary replica has to be updated?
- Upon reconnection, the primary replica compares its MT with the secondary replica's MT in order to determine the modified blocks