LE MOULLEC
Thomas

–

Ms CSIT

Infnote - Collaboration

Task 2 (11/21/2018)

Problem: Light nodes have limited power and storage

Solution: Merkle Tree

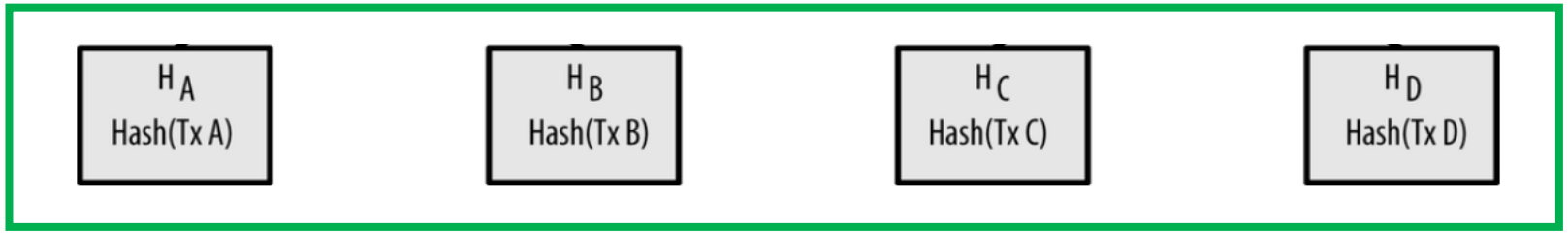
- Allows Merkle Proof: verify that a given input has been included in a particular data set
- Removing all superfluous branches while keeping only the ones we need to establish our proof
- Overall performance and scalability is really adapted to Infnote project

Secure verification of large data structures:

Merkel Tree

Merkle Tree concept

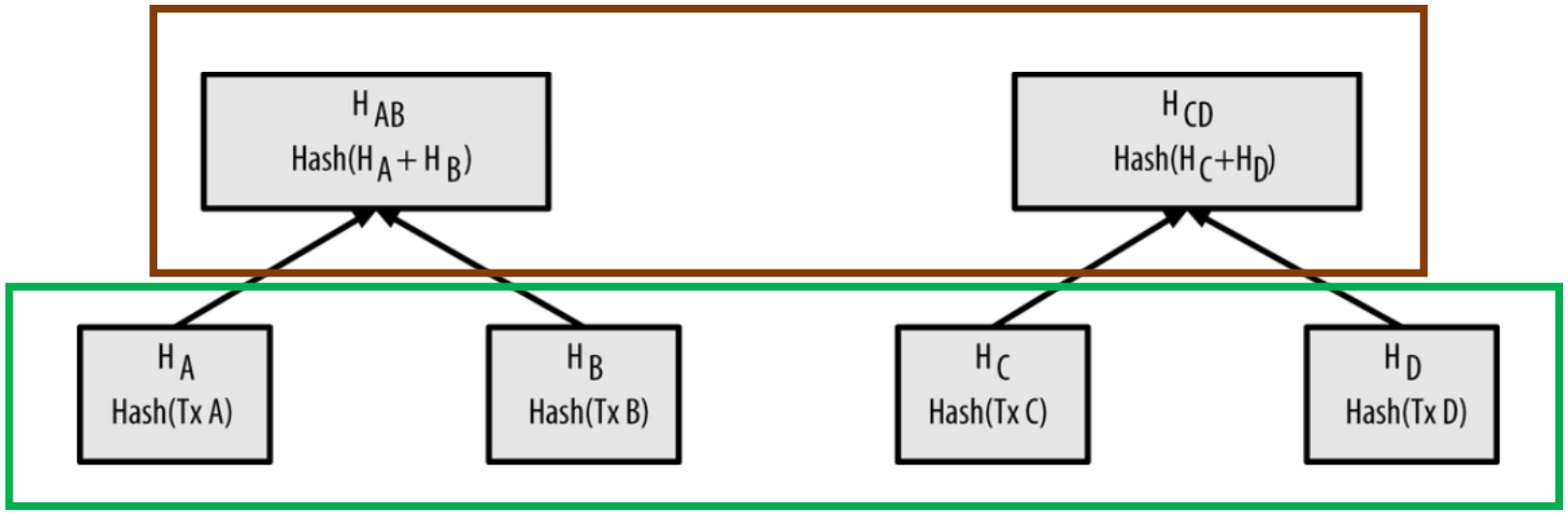
Leaves



Merkle Tree concept

Branches

Leaves

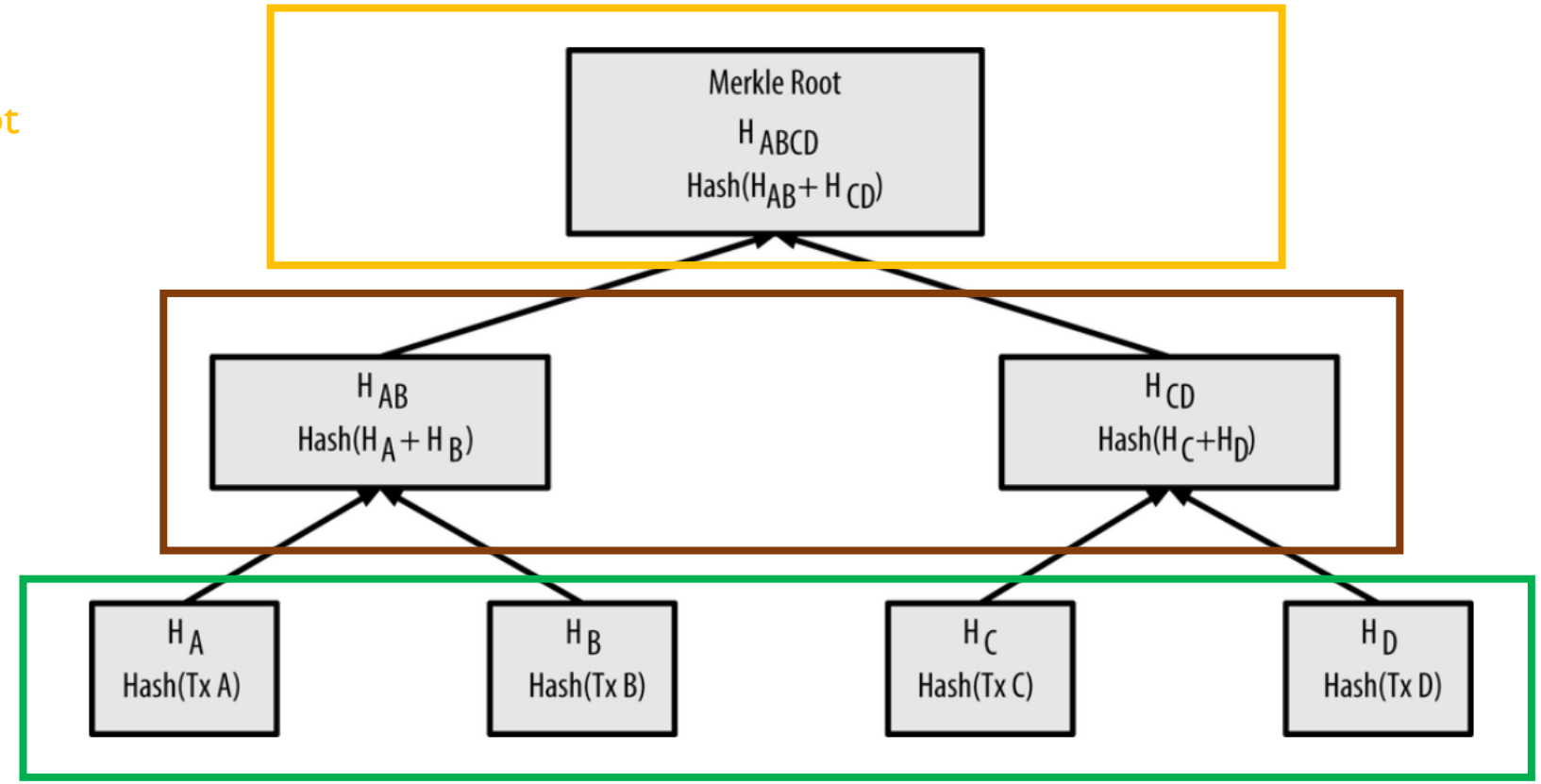


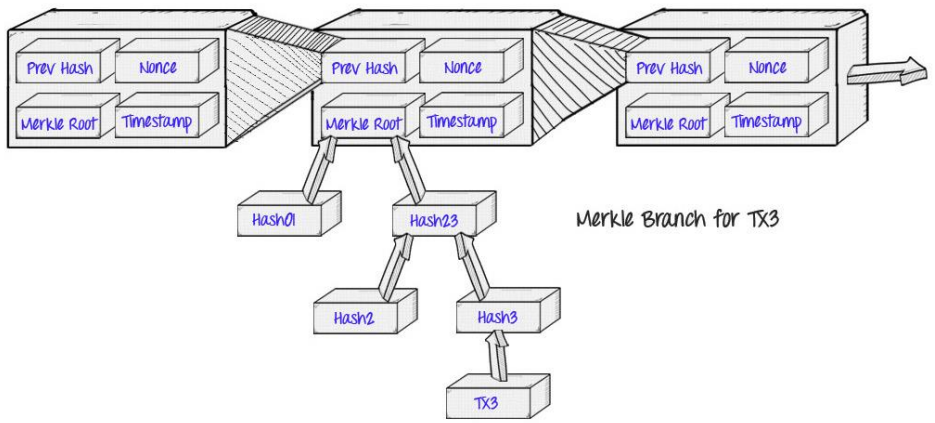
Merkle Tree concept

Merkle Root

Branches

Leaves





How ? – The process

- Merkle Tree by Ralph Merkle
- Merkling in Bitcoin
- Merkling in Ethereum
- Merkle Tree implementation
- Merkle Tree traversal algorithms

Sources:

https://people.eecs.berkeley.edu/~raluca/cs261-f15/readings/merkle.pdf

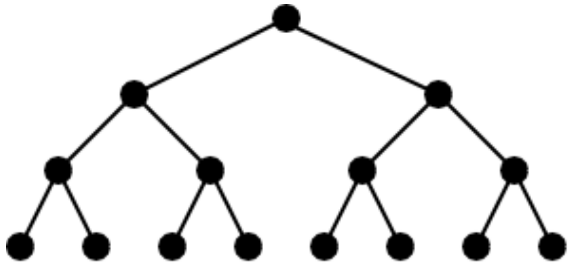
https://blog.ethereum.org/2015/11/15/merkling-in-ethereum/

https://hackernoon.com/merkle-tree-introduction-4c44250e2da7

http://www.righto.com/2014/02/bitcoin-mining-hard-way-algorithms.html

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.84.9700&rep=rep1&type=pdf

Infnote design : Binary Merkle Tree



Number of Levels = log2(leaves)
Number of nodes = 1+2+4+8+⋯+2^k
= (2^(k+1)−1) / (2−1)
= 2^(k+1)−1

Level	Nodes on Level	Nodes on levels up to and including this one
0	$1 = 2^{0}$	$1 = 2^{0+1}-1$
1	$2 = 2^{1}$	$3 = 2^{1+1}-1$
2	$4 = 2^{2}$	$7 = 2^{2+1}-1$
3	$8 = 2^{3}$	$15 = 2^{3+1}-1$
4	$16 = 2^{4}$	$31 = 2^{4+1}-1$
h	2^{h}	$2^{h+1}-1$

Type of Trees: http://cs.lmu.edu/~ray/notes/orderedtrees/

Code Demonstration

- Github: https://github.com/thomas-le-moullec/bin_merkle_tree/

Next step - Implementation of within Infnote

Delivery Last week of the semester

- Add the update_tree method
  - Keeping a balanced tree
- Implement the logic within the P2P network
  - E.g Node A requests branch n to Node B …
  - Distribution of the Tree
- Handle wrong data
  - What to do ?
- Check efficiency and performance (Low priority)