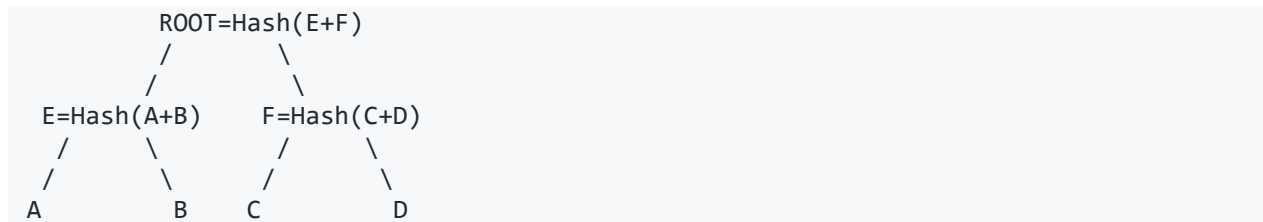


Implementation of Merkle Trees

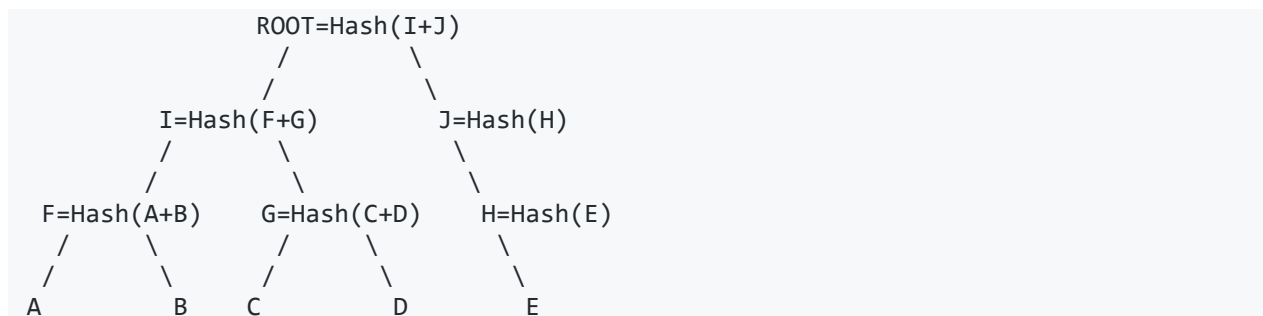
Merkle hash tree is a simple binary tree consisting of hashed leaves and nodes. Leaves are the hashes of individual nodes that have been appended to the next node. Nodes are the hashes of paired child leaves or paired child nodes. The root hash, from which all nodes and leaves stem, is known as the merkle tree hash [1].

*Tree Generation



***Note that A, B, C, D are the hashes of the transaction or list of names in our case.*

*Tree Generation



Validation check of Transaction (Inclusion check)

Merkle tree transaction validation is the missing node hashes required to compute all the nodes between the leaf and the tree root. If the root hash you compute from the audit path matches the currently advertised merkle tree root hash, then the leaf exists in the tree [1]. **checkinclusion.py** file checks for inclusion of a transaction only for 4 leaf merkle tree.

Consistency of Merkle Trees

Merkle tree consistency proof lets you verify that any two versions of a tree are consistent: that is, the later version includes everything in the earlier version, in the same order, and all new entries come after the entries in the older version [1]. **checkconsistency.py** file checks for consistency between two merkle trees i.e if the new merkle tree contains all the transactions and in the same order or not.

Commands & Screenshots on Ubuntu

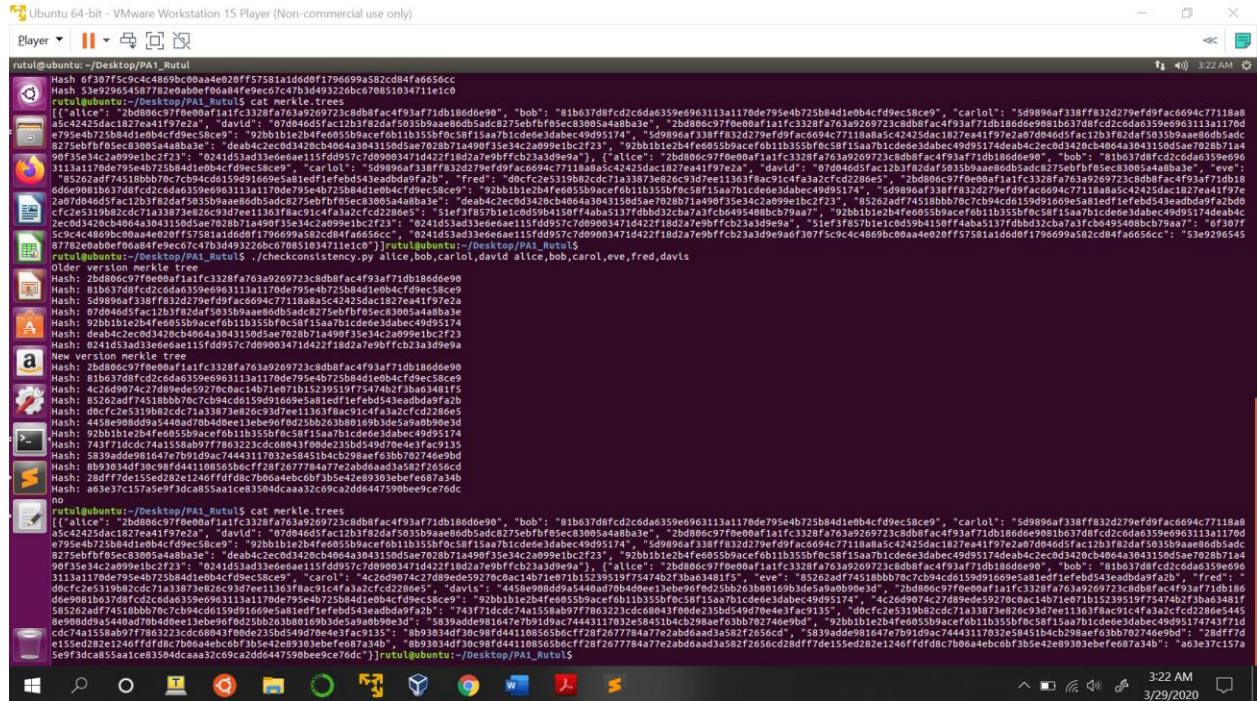
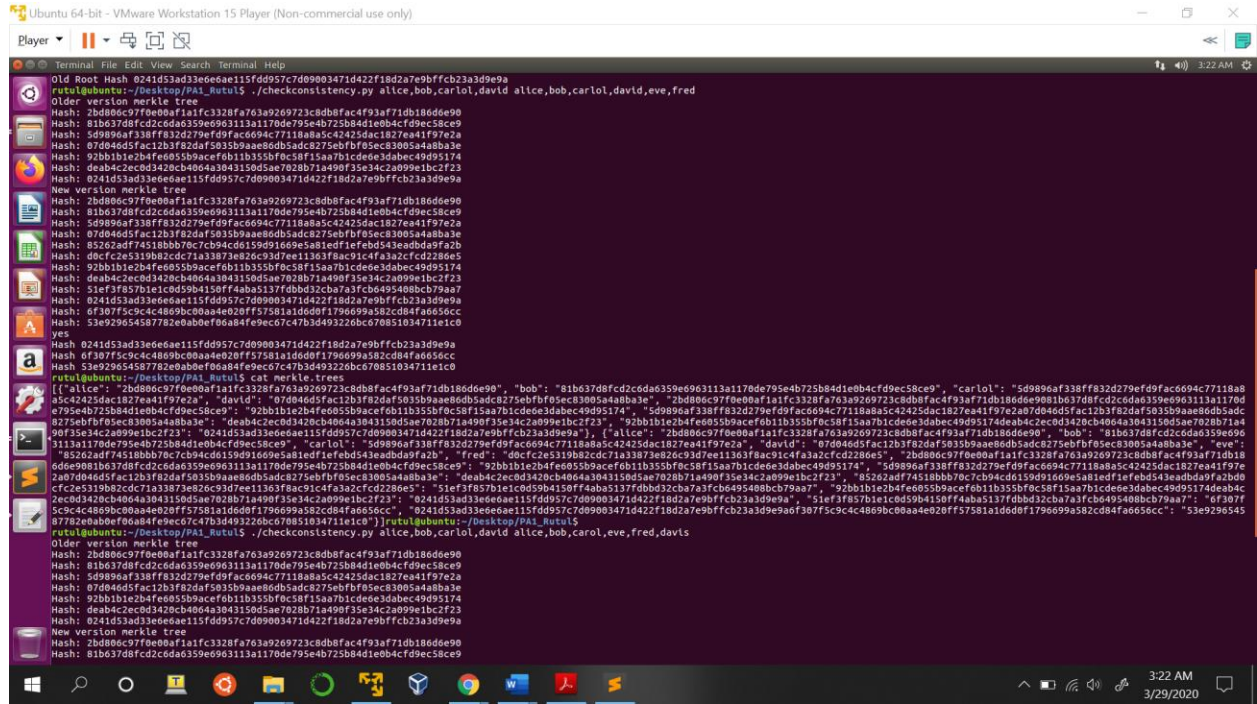
```
./builddmtree.py alice,bob,carlol,david
cat merkle.tree
./checkinclusion.py richard
./checkinclusion.py david
./checkconsistency.py alice,bob,carlol,david alice,bob,carlol,david,eve,fred
cat merkle.trees
./checkconsistency.py alice,bob,carlol,david alice,bob,carol,eve,fred,davis
cat merkle.trees
```

```

rutil@ubuntu:~/Desktop/PA1_Rutul$ ./builddmtree.py alice,bob,carlol,david
Hashes are printed from bottom to top in sequence.
The last one is the root hash.
Hash: 2bd806c97f0e00af1a1fc3328fa763a9269723c8db8fac4f93af71db186d6e90
Hash: 81b637d8fcd2c6da6359e6963113a1170de795e4b725b84d1e0b4cf9ec58ce9
Hash: 5d9896af338ff832d279efd9fac6694c77118a8a5c42425dac1827ea41f97e2a
Hash: 07d046d5fac12b3f82daf5035b9aae86db5adc8275ebfbf05ec83005a4a8ba3e
Hash: 92bb1b1e2b4fe605b9acef6b1b355bf0c58f15aa7b1cde6e3dabec49d95174
Hash: dea4c2ec8d342cb4064a3043150d5ae7028b71a490f35e34c2a099e1bc2f23
Hash: 0241d53ad33e6eae115fd957c7d09003471d422f18d2a7e9bffc23a3d9e9a
rutil@ubuntu:~/Desktop/PA1_Rutul$ ls
builddmtree.py  checkinclusion.py  merkle.tree  __pycache__
rutil@ubuntu:~/Desktop/PA1_Rutul$ cat merkle.tree
{"alice": "2bd806c97f0e00af1a1fc3328fa763a9269723c8db8fac4f93af71db186d6e90", "bob": "81b637d8fcd2c6da6359e6963113a1170de795e4b725b84d1e0b4cf9ec58ce9", "carlol": "5d9896af338ff832d279efd9fac6694c77118a8a5c42425dac1827ea41f97e2a", "david": "07d046d5fac12b3f82daf5035b9aae86db5adc8275ebfbf05ec83005a4a8ba3e", "2bd806c97f0e00af1a1fc3328fa763a9269723c8db8fac4f93af71db186d6e90b637d8fcd2c6da6359e6963113a1170de795e4b725b84d1e0b4cf9ec58ce9": "92bb1b1e2b4fe605b9acef6b1b355bf0c58f15aa7b1cde6e3dabec49d95174", "5d9896af338ff832d279efd9fac6694c77118a8a5c42425dac1827ea41f97e2a07d046d5fac12b3f82daf5035b9aae86db5adc8275ebfbf05ec83005a4a8ba3e": "dea4c2ec8d342cb4064a3043150d5ae7028b71a490f35e34c2a099e1bc2f23", "92bb1b1e2b4fe605b9acef6b1b355bf0c58f15aa7b1cde6e3dabec49d95174dea4c2ec8d342cb4064a3043150d5ae7028b71a490f35e34c2a099e1bc2f23": "0241d53ad33e6eae115fd957c7d09003471d422f18d2a7e9bffc23a3d9e9a"}
rutil@ubuntu:~/Desktop/PA1_Rutul$ ./checkinclusion.py richard
Test String Hash: 61bffe9215f65104ad18b45aff1436c0c165d0d5dd2087ef01b4232ba6dd2c1a
Checking inclusion.....
no
rutil@ubuntu:~/Desktop/PA1_Rutul$ ./checkinclusion.py david
Test String Hash: 07d046d5fac12b3f82daf5035b9aae86db5adc8275ebfbf05ec83005a4a8ba3e
Checking inclusion.....
yes
Hash: 5d9896af338ff832d279efd9fac6694c77118a8a5c42425dac1827ea41f97e2a
Hash: 92bb1b1e2b4fe605b9acef6b1b355bf0c58f15aa7b1cde6e3dabec49d95174
Old Root Hash: 0241d53ad33e6eae115fd957c7d09003471d422f18d2a7e9bffc23a3d9e9a
rutil@ubuntu:~/Desktop/PA1_Rutul$ ./checkconsistency.py alice,bob,carlol,david,eve,fred
Older version merkle tree
Hash: 2bd806c97f0e00af1a1fc3328fa763a9269723c8db8fac4f93af71db186d6e90
Hash: 81b637d8fcd2c6da6359e6963113a1170de795e4b725b84d1e0b4cf9ec58ce9
Hash: 5d9896af338ff832d279efd9fac6694c77118a8a5c42425dac1827ea41f97e2a
Hash: 07d046d5fac12b3f82daf5035b9aae86db5adc8275ebfbf05ec83005a4a8ba3e
Hash: 92bb1b1e2b4fe605b9acef6b1b355bf0c58f15aa7b1cde6e3dabec49d95174
Hash: dea4c2ec8d342cb4064a3043150d5ae7028b71a490f35e34c2a099e1bc2f23
Hash: 0241d53ad33e6eae115fd957c7d09003471d422f18d2a7e9bffc23a3d9e9a
New version merkle tree
Hash: 2bd806c97f0e00af1a1fc3328fa763a9269723c8db8fac4f93af71db186d6e90
Hash: 81b637d8fcd2c6da6359e6963113a1170de795e4b725b84d1e0b4cf9ec58ce9
Hash: 5d9896af338ff832d279efd9fac6694c77118a8a5c42425dac1827ea41f97e2a
Hash: 07d046d5fac12b3f82daf5035b9aae86db5adc8275ebfbf05ec83005a4a8ba3e
Hash: 85262ad7f4518bb70c7cb94cd159d91609e5a81ed1efebd543eadb0a9fa2b
Hash: d0cfe2e319b2cdcd1f33872e0b6c93f7ee113d3f8ac21c4fa3a2c1cd22865
Hash: 92bb1b1e2b4fe605b9acef6b1b355bf0c58f15aa7b1cde6e3dabec49d95174
Hash: dea4c2ec8d342cb4064a3043150d5ae7028b71a490f35e34c2a099e1bc2f23
Hash: 51ef3f87b1e1c0dc59b4150ff4aba5137fddbd32c3ba7a3fcb6495408bcb79aa7

```

CSCI531_Programming Assignment 1



References:

[1] <https://www.certificate-transparency.org/log-proofs-work>

Rutul Bakulkumar Pandya [USC ID: 2154579267]