Experiment 7

Date of Performance : 20-03-23

Date of Submission : 30-03-23

SAP Id: 60004200107 **Name :** Kartik Jolapara

Div: B **Batch :** B1

AIM

Implement Merkle Root creation with the help of SHA-1. Your program will have input as paragraph. Paragraph can be converted to suitable blocks for which hash values can be computed. Finally generate Merkle root based on these computed hash values.

THEORY

A hash tree is also known as Merkle Tree. It is a tree in which each leaf node is labeled with the hash value of a data block and each non-leaf node is labeled with the hash value of its child nodes labels.

In a Merkle tree, transactions are grouped into pairs. The hash is computed for each pair and this is stored in the parent node. Now the parent nodes are grouped into pairs and their hash is stored one level up in the tree. This continues till the root of the tree. The different types of nodes in a Merkle tree are:

- Root node: The root of the Merkle tree is known as the Merkle root and this Merkle root is stored in the header of the block.
- Leaf node: The leaf nodes contain the hash values of transaction data. Each transaction in the block has its data hashed and then this hash value (also known as transaction ID) is stored in leaf nodes.
- Non-leaf node: The non-leaf nodes contain the hash value of their respective children. These are also called intermediate nodes because they contain the intermediate hash values and the hash process continues till the root of the tree.

PROGRAM

```
from hashlib import sha256
```

```
def hash(x):      ans = sha256(x.encode("utf-
8")).hexdigest()   return ans


def hash_value(h):   h1 = []   if
len(h) % 2 == 0:     for i in
range(0, len(h), 2):       text =
h[i] + h[i + 1]
h1.append(hash(text))   else:
for i in range(0, len(h) - 1, 2):
     text = h[i] + h[i + 1]
h1.append(hash(text))
h1.append(h[len(h) - 1])


  return h1


para = input("Enter para (use '.' to seperate lines): ")


l = para.split('.') count
= len(l)


if count % 8 != 0 :
temp = int(count / 8)
for i in range(0, (temp
+ 1) * 8 - count):
    l.append(l[count - 1])
```

```
h = list(map(hash, l))
length = len(h)

while length > 1:  h
=      hash_value(h)
length = len(h)

print("\n\nMerkle root - ", h[0])
```

INPUT AND OUTPUT

```
Enter para (use '.' to seperate lines): hello.goodbye.blue.crystal.puppy.sandalwood.lamp.fineprint.myrtle

Merkle root -  653198460235112299a813791127838c8e0de563
```

CONCLUSION

Thus, we have successfully implemented Merkle root creation using SHA-1.