**Implementation of Bitcoin model using Blockchain​**

A PROJECT REPORT

*Submitted by*

**TEAM - PERFECT DEALERS**

**K. VISHNU SAINADH BL.EN.U4AIE19028**

**K. SATWIK BL.EN.U4AIE19034**

**V. ASHRITH BL.EN.U4AIE19066**

***for the course***

***19AIE203- Data Structures and Algorithms – 2***

***Guided and Evaluated by***

***D. RADHA***

***Asst. Prof (SG),***

***Dept. of CSE,***



**AMRITA SCHOOL OF ENGINEERING, BANGALORE**

**AMRITA VISHWA VIDHYAPEETHAM**

**BANGALORE-560 035**

**December-2020**

**IMPLEMENTATION OF BITCOIN MODEL USING BLOCKCHAIN**

**ABSTRACT OF THE PROJECT**

There have been many attempts to create digital money in the past, but they have always failed. The prevailing issue is trust. If someone creates a new currency called the X dollar, how can we trust that they won't give themselves a million X dollars, or steal your X dollars for themselves. The solution to this problem is given by bitcoin with the help of a specific type of database called a blockchain. In our project a cryptographic protocol using the Merkle tree and SHA-256 hashing technique is performed which makes it possible to confirm the transactions are integrity protected and are not hackable.

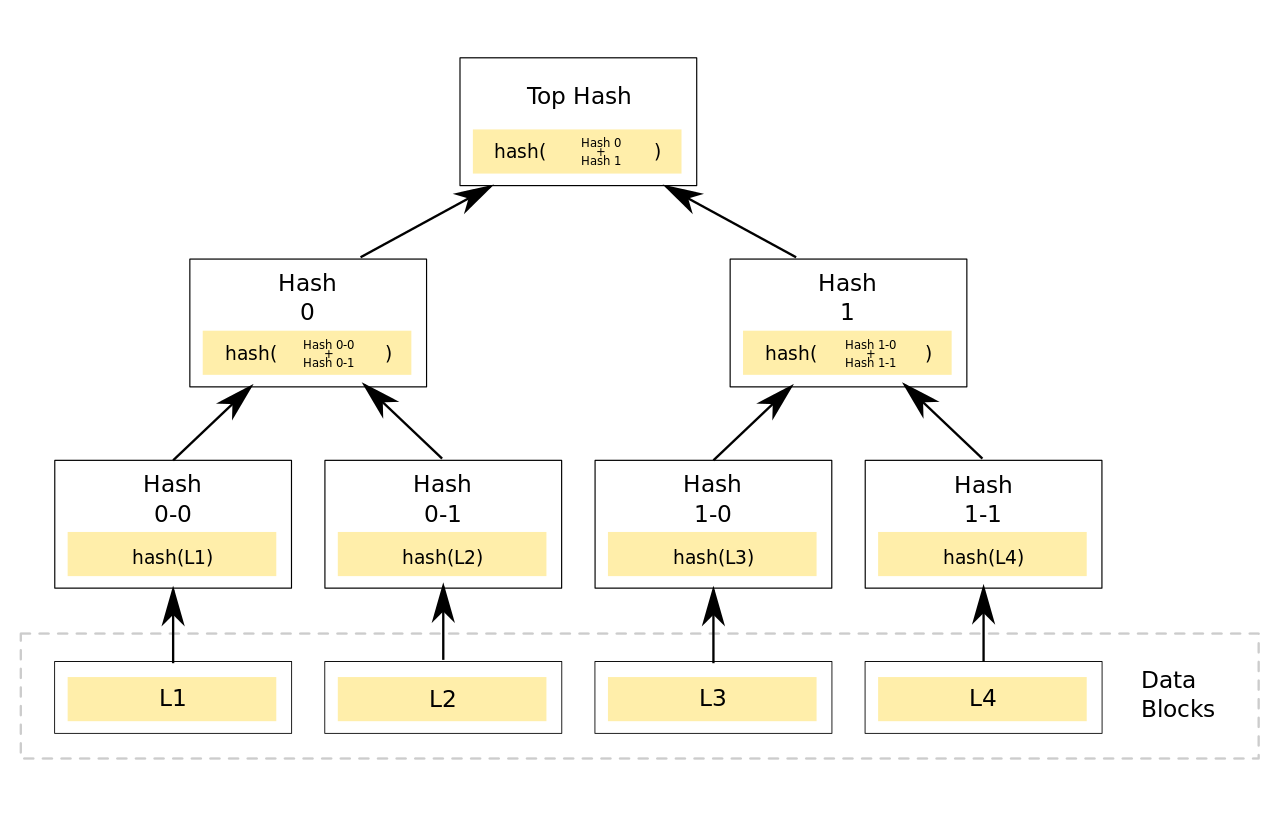
[**INTRODUCTION**](about:blank)

In today’s world digital currency has become more aware. Bitcoin is one of the digital currencies which enables the user with a modern way of payment. These bitcoins work with the help of blockchain technology. A blockchain is a database that is shared across a network of computers. Once a record has been added to the chain it is very difficult to change. To ensure all the copies of the database are the same, the network makes constant checks. Blockchains have been used to underpin cyber-currencies like bitcoin, but many other possible uses are emerging.

**SYSTEM MODEL – DESCRIPTION OF THE DIAGRAM**

Our project basically works on the principle of Merkle trees. Merkle Trees are really important because they allow for “Merkle Proofs”. These enable us to quickly verify that a given input has been included in a particular data set and in what order.

Merkle Trees are also really efficient. They allow us to compress large data sets by removing all superfluous branches while keeping only the ones we need to establish our proof.

****

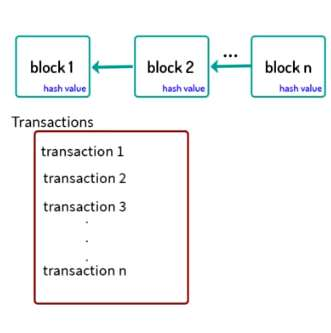
This is how a Merkle tree works:

A Merkle tree is a tree of hashes in which the leaves are hashes of data blocks in, for instance, a file or set of files. Nodes further up in the tree are the hashes of their respective children. For example, in the picture hash 0 is the result of hashing the concatenation of hash 0-0 and hash 0-1. That is, hash 0 = hash (hash (0-0) + hash (0-1)) where + denotes concatenation.

Most hash tree implementations are binary (two child nodes under each node) but they can just as well use many more child nodes under each node.

Usually, a cryptographic hash function such as SHA-2 is used for the hashing.

A Merkle root is the hash of all the hashes of all the transactions that are part of a block in a blockchain network.



A Blockchain database can be divided into 3 parts:

* The Record or A Transaction: Can be any information, a deal for example
* The Block: A bundle of records
* The Chain: All blocks linked together

let us consider a problem:

1. A trade is recorded. For example, let’s say Mr. Pink is selling two of his coins to Mr. Green for $100. The record lists the details, including a digital signature from each party.
2. The record is checked by the network. The computers in the network, called 'nodes', check the details of the trade to make sure it is valid.
3. The records that the network accepted are added to a block. Each block contains a unique code called a hash. It also contains the hash of the previous block in the chain.
4. The block is added to the blockchain. The hash codes connect the blocks together in a specific order.

**Implementation Details**

* class SHA256 : a class which converts strings into hash values
* class Tuple : used to create a tuple
* class Block : it has Merkle\_Root, Previous\_Block\_Hash, next

creates a header block and also assigns the Merkle roots to the blocks

* class Block\_Chain : it has head

creates a head block, adds a block, verifies the block and prints the Previous\_Block\_Hash and Merkle\_Root for the respective blocks

* class Merkle\_Node : it has parent, is\_Left\_of\_Parent, is\_Right\_of\_Parent, left,

right, leaf, H. creates Merkle nodes

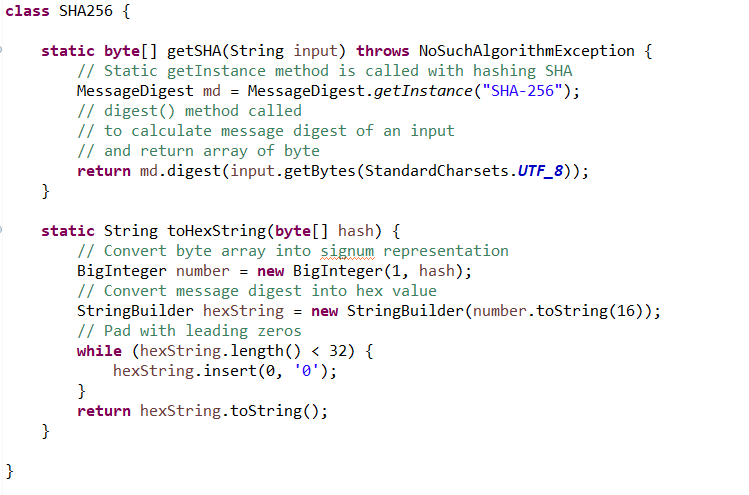
* class Merkle\_Tree : creates a Merkle tree using Merkle nodes, creates and

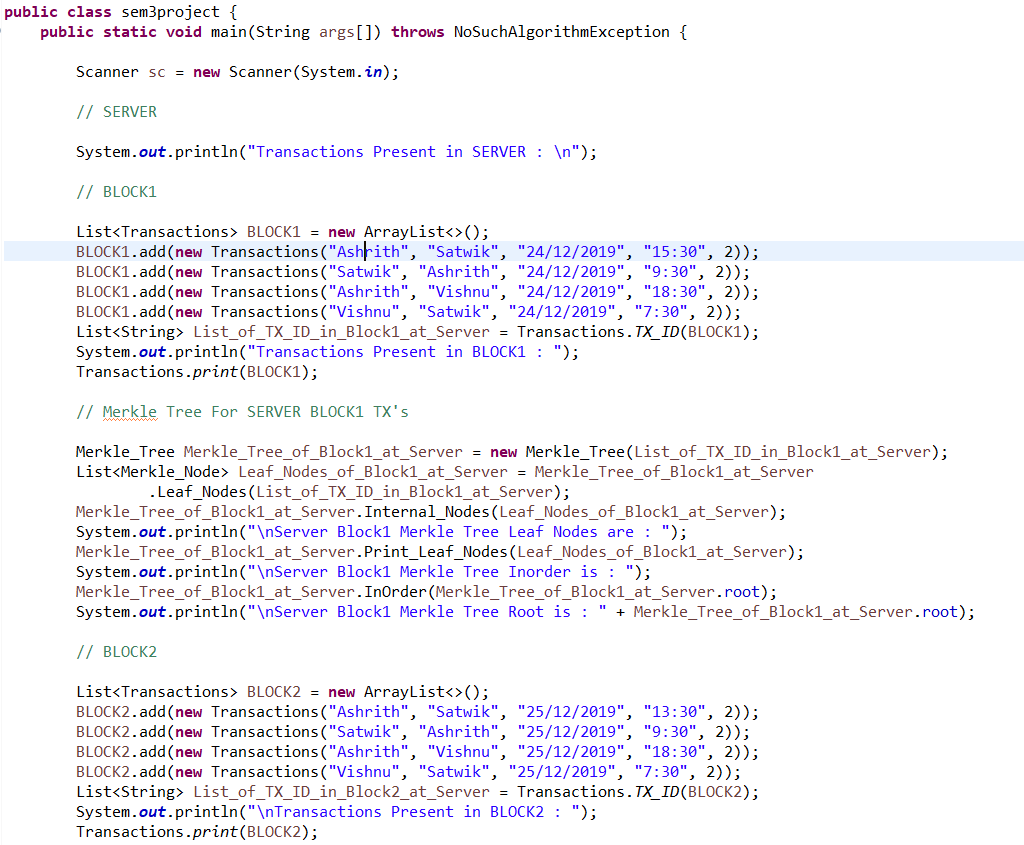
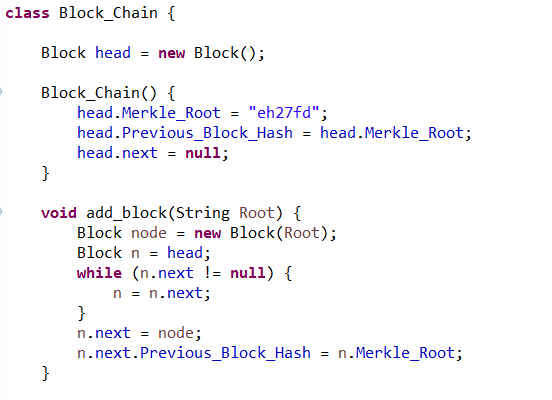
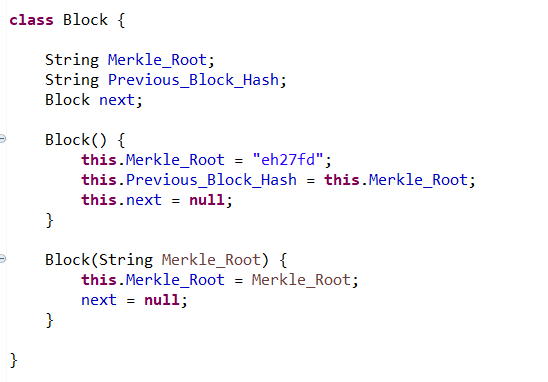
prints list of leaf\_nodes and also creates internal nodes

* class Transactions : it has Sender\_Name, Reciever\_Name, Date, Time,

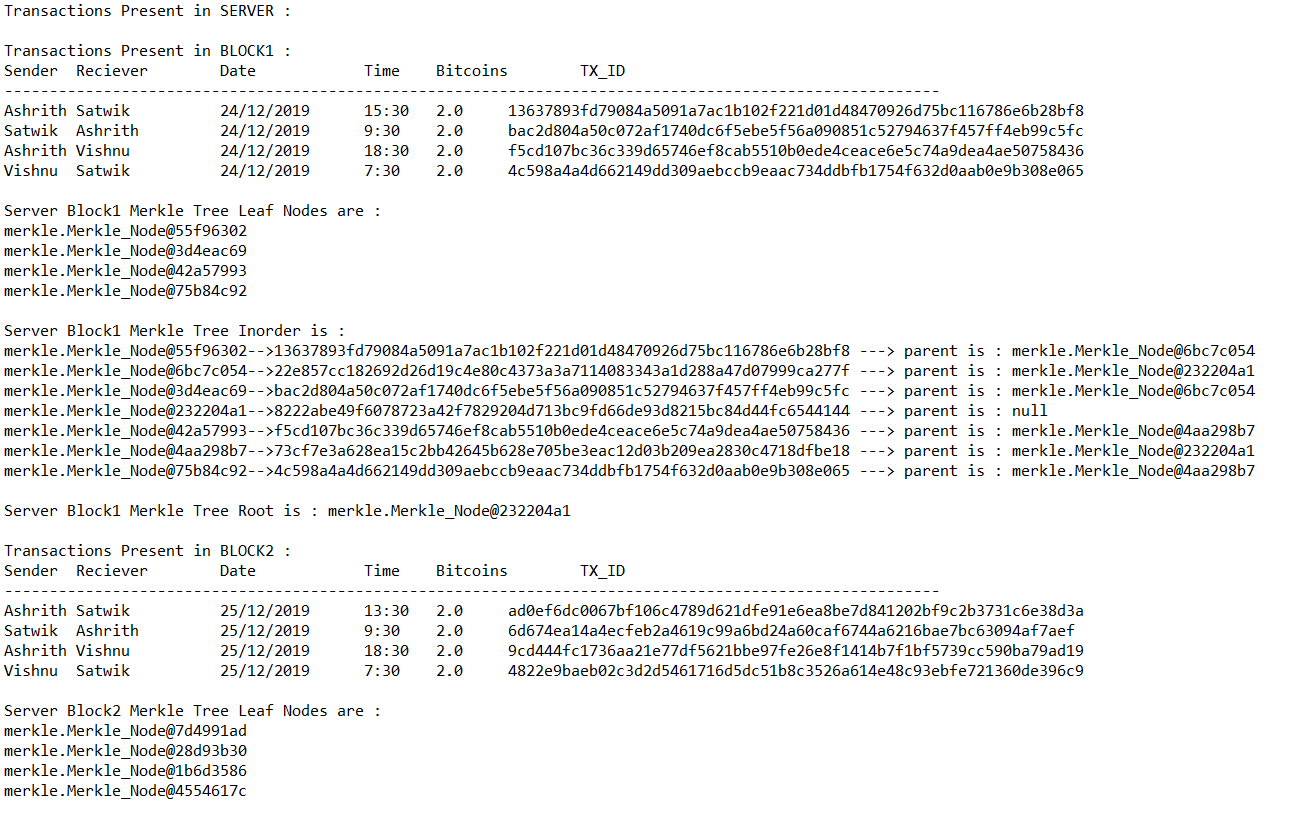
BitCoins, Transaction\_ID. creates a list of all Transactions and also prints the transactions with respect to its block.

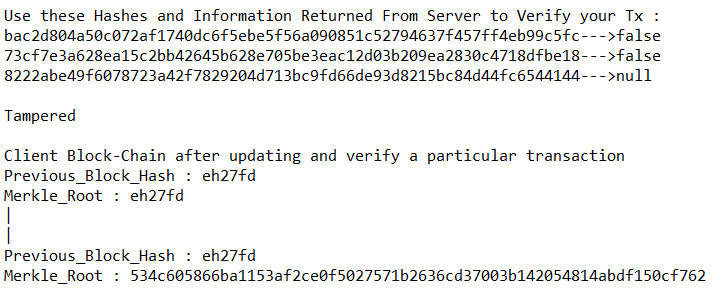
**SAMPLE CODE**

****

****

**SAMPLE OUTPUT**

****

****

**CONCLUSION**

From our project we now understand the role of Merkle Trees in blockchain. The Merkle tree, encodes the blockchain data in an efficient and secure manner. It enables the quick verification of blockchain data, as well as quick movement of large amounts of data in the blockchain network.

**REFERENCES**

**Conference reference paper**

[1]. S. Dhumwad, M. Sukhadeve, C. Naik, M. K.N. and S. Prabhu, "A Peer-to-Peer Money Transfer Using SHA256 and Merkle Tree," 2017 23RD Annual International Conference in Advanced Computing and Communications (ADCOM), Bangalore, India, 2017, pp. 40-43, doi: 10.1109/ADCOM.2017.00013.