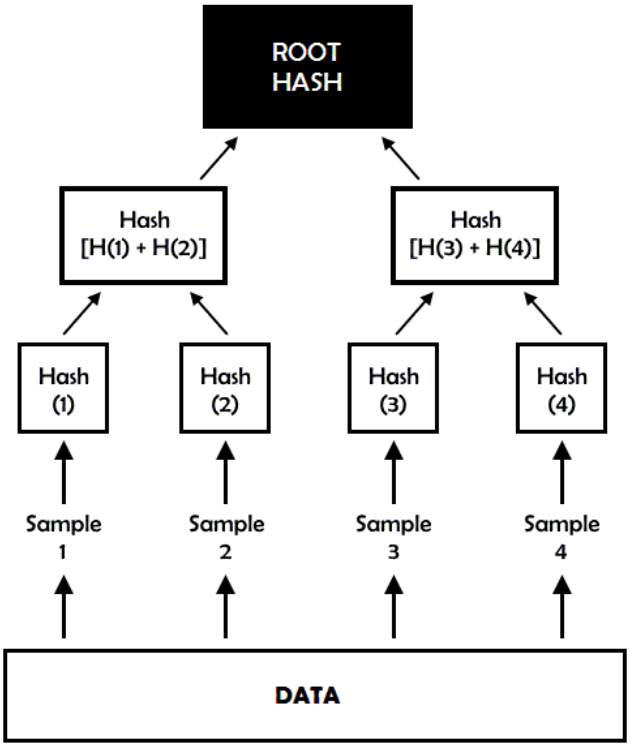

Fig. 1. Generating Merkle tree from the data

After getting the proof, the customer recalculates the final proof by applying the following steps:

1) Calculate the hash of Sample 2: H(2)

2) Calculate the hash of H(1) together with the output of first step : H(H(1),H(2))

3) Calculate the hash of H(H(3),H(4)) together with the output of second step : H(H(H(1),H(2)),H(H(3),H(4)))

After calculating the final root hash, the customer compare it with data's hash value in the ledger to check that the proof is generated using the original data. Also, the sample data can be examined by the customer whether the sample is useful. After the accuracy and usefulness of the data is confirmed; if the customer wants to buy the data, customer contacts with the seller. Then, the smart contract is triggered between seller and customer.

It is necessary to use smart contracts for ensuring a secure money and data exchange. In this way; in the event of any problem, the seller and the customer's victimization is prevented.

Figure 2 shows the basic operations in the smart contract for trading data and money, between seller and customer. In smart contract, the hash value of seller's data is calculated and compared with its hash value in the blockchain. Also, customer's money is checked whether it is enough or not. In case of a problem in one of two situations, the smart contract is terminated unsuccessfully and trade operation is cancelled. Otherwise, the smart contract is terminated successfully and trade operation is completed.

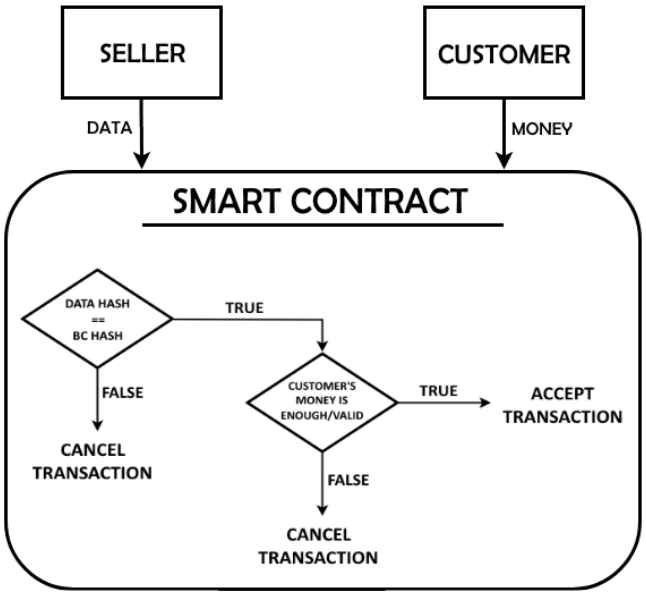

Fig. 2. Smart contract for trade operation.

After the trade is completed, the new transaction is written to the block. When the transactions are verified and the block is completed, it is written on the main chain. As a requirement of blockchain technology, a fee should be paid to the node that works on the block. To do this, a public blockchain that the smart contract can deploy can be used.

IV. SECURITY ANALYSIS

This section provides a discussion on security and performance of our blockchain system and its structures.

When the customer requests sample from data, there is a possibility that the seller may send the wrong or useless sample. On the other hand, the sample may belong to a useful data, but the data to be sent at the end may be an unusable data that is not relevant to the sample. This problem can be solved by Merkle tree with calculating the correlation between sample data and the main data. Also; during the smart contract, the hash value of the data to be sent is compared with the hash value in the blockchain.

A third party node can try to impersonate the seller or customer. In order to do this, it is necessary to produce the signature of the target node. But this type of attack can be avoided easily with digital signature's key generation algorithm. So, it is impossible to forge to someone's digital signature.

Besides these issues, another malevolent third party node can try to break or alter the transaction and message sequence in the chain. But due to the avalanche effect of the cryptographic hash function, block sequences and integrities can be checked fast and correctly. Even the slightest change in the block causes the hash value to change completely. Therefore, the consensus algorithm will not accept the manipulated block.