A Comprehensive Study of Blockchain Technology & Its Applications

Choudhry Bilal Mazhar
Dhanani School of Science & Engineering
Habib University, Karachi, Pakistan
cm00326@st.habib.edu.pk

***Abstract* — In this intensive survey, blockchain technology and its effect on the conventional networking transactional technology has been studied. Its algorithm and operating mechanism have been taken into account to comprehend its firm benefits and associated impacts in technological world. As the world is evolving rapidly and now with the emergence of cryptocurrency, the blockchain technology has gained significant attention globally for various transactional purposes in oil industry, retail industry etc. as well. The blockchain technology is different from any other technology used in online transactions in a manner that it creates node/sub-blocks which keeps getting created subsequently upon getting the analytical validation from the preceding block(s). This makes it a paradoxical network of de-centralized and centralized networking model simultaneously.**

Keywords—blockchain; transaction; sub-blocks; retail transaction; (*key words****)***

I. INTRODUCTION

Blockchain technology has already been getting its footprints not only on cryptocurrencies, but also in various online applications such as banking transactions, online shopping transactions, data management and commodities handling [1]. Blockchain technology operates such that in any given transaction, various nodes validates the transaction upon successful communication by checking the data being sent, hash value, time stamp and identification means of previous block. Hence, the whole communication is integrated in all these blocks. In the present world, with various hacking threats out there in the cyber world, the blockchain technology is generally considered transparent and safe as far as safe data transaction is concerned. Despite the fact that this communication is done in the form of various blocks, it is still a peer-to-peer connection making it a de-centralized and centralized network simultaneously.

For a transaction based on blockchain technology, when the first block is build which is also regarded as genesis block, the data transmitted in it cannot be altered in any case. However at the receiving end of the peer-to-peer connection, certain cases have been reported where the encrypted data being sent in blocks is corrupted and is nullified by changing the decisions of more than half of the blocks involved in that transaction.

Below is the peer-to-peer (P2P) connection system of blockchain technology [2]:

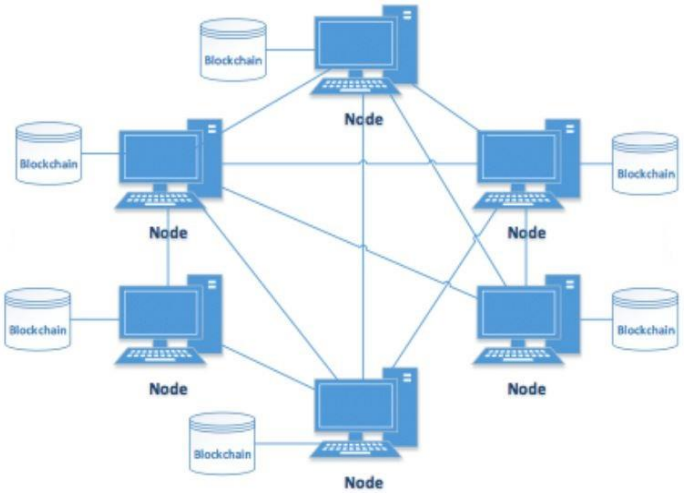

Fig. 1 P2P Connection in blockchain technology [2]

II. BLOCKCHAIN TECHNOLOGY OPERATION

This technology is a landmark addition in the online transaction domain as it gives more leverage to the both the end-to-end users as it provides ample autonomy to both end users to control and access the data as per their need. In blockchain technology, the data is kept secured by means of hashing in which the state of data in any given block is marked at a specific time. So while authenticating, it also checks the timestamp of each block.

During a blockchain transaction, the local copy of blockchain is located at each node that is already created.

In transaction between two nodes, if the time stamps, header file values the information regarding the data being provided to the succeeding block is matched, only then the transaction between these two blocks occurs successfully. This gives the user or customer to transmit the data from one end user to the other safely.

III. Data Encryption in Blockchain Technology

In peer-to-peer (P2P) connection via blockchain technology, the data sent from one end-user is first encrypted to ensure it safety during the transmission phase. This encryption method is regarded as hashing. A hashing algorithm is set up in every communication which converts the transmitted data in a form un-readable and hackable by any means. A simple illustration of the hashing process is provided below:

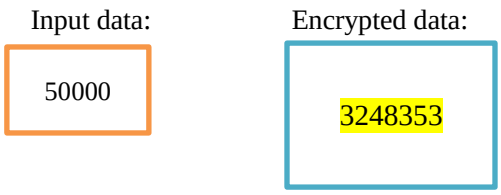

Fig. 2 Input/encrypted values before and after hashing

The above mentioned image is a general illustration of hashing outcome. As it can be clearly observed that the input value cannot be determined by simply looking at the encrypted value. This is what distinguishes blockchain technology, i.e. the communication security, which however can be compromised as discussed later in this article. The hashing function is essential in order to preserve various digital assets such as cryptocurrency, digital signatures, finger print records and other similar kind of information where there is a risk of mismanagement/ re-shuffling with a slight change in original value.

According to Jscrambler's article [3], the important parameters to consider while designing a hashing algorithm for any given communication involves a standard fixed kind of encryption value for any given fixed or arbitrary input data. For instance, regardless of the nature of data, either numerical, alphabetical or combination of both, the encrypted value should be of same nature to avoid inconvenience at the receiving end of the succeeding blocks in a P2P communication. The other important factors while designing hashing algorithms are:

- The computation of encrypted/hash values should be quick regardless of the size and nature of message/data.
- The original message/data should not be regenerated by any means when it is encrypted.
- Each hash value in a given P2P communication must be unique to avoid mismanagement and or loss of message.
- An avalanche effect must be present inherently in every hashing algorithm that can distinguish between two identical messages with a very tiny difference, so that two separate hash values could be generated.

The hashing algorithm is presented in the form of flow chart below:

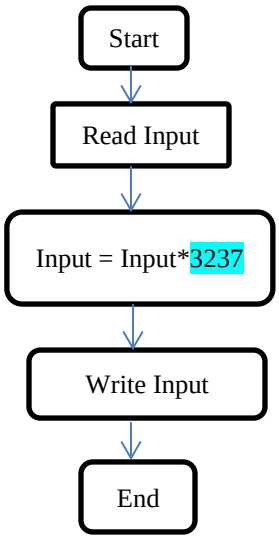

Chart 1. Flowchart of hashing algorithm

In chart 1 above, it is to be noticed the *input* value has been overwritten in the hashing process, and there is no way the original input value could be restored. This is the entire objective of hashing in communication, i.e. to make it highly secure. The number "3237" is used randomly and it could be any number/value to encrypt the original value. The validation parameters are pre-defined in the blockchain system and incase if the hashes are deemed similar for two

identical but slightly different input messages, then the data could be lost, as the system would no more be able to locate original data sent initially for the two same hash values.

In hashing process, it is crucial to set the hashing values in order for each subsequent block that is being created. The succeeding block has to be in continuation/synchronization with the previous block. In case, if at any block, the hash value is attempted to be changed by means of hacking, it would not be successful because the previous hash will no more remain synchronous with the under-attack block, hence the hash value will not change due to the violation of pre-defined protocols already set in blockchain system. However, there is always a risk of 51 percent attack with which the data/message can be corrupted if not retrieved. For instance, in a P2P communication, 20 blocks have been built however if more than half of them are under-attack, then there are high chances of the data being corrupted because in blockchain technology, the *majority rule* is followed, i.e. if more than half of the hashes are altered somehow, then the network is no more safe. At the same time, it is important to highlight that such incidents cannot happen in public blockchain network such as of cryptocurrency due to its worldwide real time usage and un-controlled applications. This threat exists in private blockchain network which in any case has centralized control and the identifiable users/nodes hence not very threatening in this case either. However, multiple cases are reported in literature where the data has been corrupted by means of 51 percent attack.

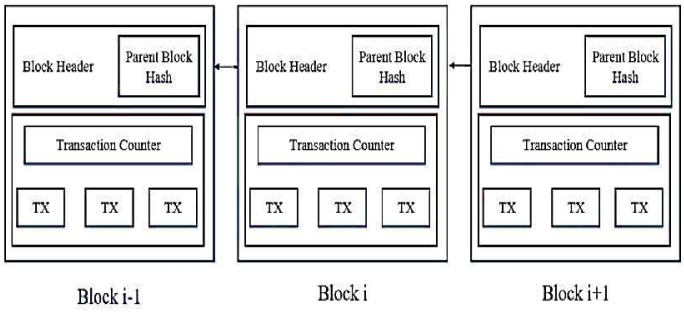

Fig. 3 Blockchain network architecture

The hash values of the succeeding blocks refer to the hash values of previous blocks in some manner (identical few initial digits/ particular order etc.) as presented in figure 3 above. So the original data/message can only be altered if more than half of the blocks are successfully corrupted. Apart from hash values, the time-stamp and the index is also the affirming factor in blockchain which however makes this technology extremely safe with very minute threats for cyber-attacks.

To be more precise, the hash values is not exactly the encrypted data, instead it is just the representation of the encrypted data, so incase if hackers attack the data and try to alter it, the hash value will also get altered automatically and since hash values correlates among the preceding and subsequent blocks, therefore upon automatic change in the hash value it would not correlate with its previous and next block hence would most likely decline and nullify the hacker's attack.

In [3], Jscrambler have presented the clearer representation of hash values as shown below:

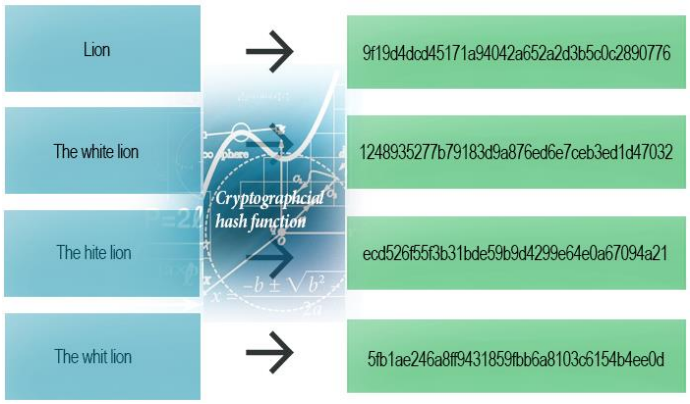

Fig. 4 Original value to hash value [3]

Each message has its own separate hash value as presented in figure 3, even for the identical values with a slight change of single character. This is to ensure that there is no overlapping of the data at any stage of the block wise communication in any given P2P communication.

IV. HASH TABLE

The concept of hashing dates back to the early age of internet, i.e. even way longer before the invention of blockchain technology, and is being used since then in various applications. Since bitcoin is an advanced complex integrated networking system, it also uses hashing method for P2P communication. In order to retrieve the original value at the end-user, a hash table is set up in a blockchain network which maps the corresponding hash value of each

original input message. A hash table is kind of a pointer array which helps in locating the messages. The following example explains the concept of hash table [4]:

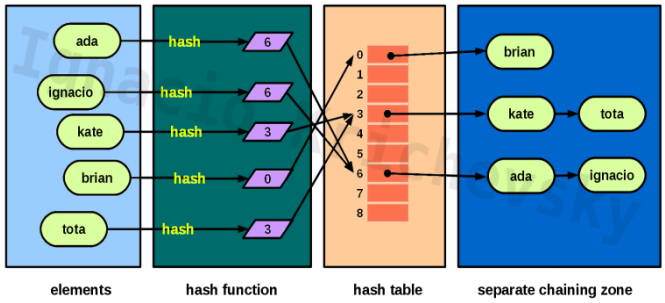

Fig. 5 Correlation of hashing algorithm and hash table [4]

In figure 5, all the original values, such as 'ada', 'ignacio' etc. have been encrypted and assigned hash value via hash functions and their values are mapped at the respective indexes of the hash table. One important thing that is taken into account while performing hashing tasks is to do a *separate chaining* process, which ensures that no hash value has a collision with any other hash value within the transaction and it is done via storing various original data in the same index of hash table as the index of hash function, as seen in figure 5. The separate chaining task is essential because if exactly the same message is send from one end-user twice, there is no guarantee that the hash value will not collide, so separate chaining becomes important in order to ensure that collision does not occur

While extracting the original message from the hash table, the format of the table matters. As it is mentioned above that hash table is a kind of data structure, i.e. an array which stores the original values at its respective indexes identical with the indexes of hash functions, so in order to decrypt the message at the receiving end, each message gets searched through the whole array incorporating certain time complexity whose exact measure depends on the implementation of the array and the searching algorithm being used. Using time complexity measurement makes it feasible to understand and conceptualize the running time required for a single transaction to process.

There searching algorithm used in the hash table is very similar to the linear searching algorithms widely used in computer science domain to search for the desired element in any given array. Such searching is also called linear probing. However this is a standard time complexity of unordered associative hash table which is provided below [5]:

Algorithm	Average Case	Worst Case
Space	O(n)	O(n)
Search	O(1)	O(n)
Insert	O(1)	O(n)
Delete	O(1)	O(n)

Table 1. Time complexity of hash table

The complexity is a measure of how long in terms of space and time does the algorithm take to search the desired element in a given data structure. There could be some iteration that is continuous as it is performed the most. Such elements can be left unattended and regarded as dominant.

The fundamental illustration of the searching algorithm followed in hash table to map the original values is presented in figure 6 below.

```
File  Edit  Format  Run  Options  Window  Help
def search(n, A):
    for i in range(len(A)):
        if A[i] == n:
            return True
        else:
            return False
```

Fig. 6 Searching algorithm for hash table

In the algorithm above, the 'n' is the desired original values which are to be mapped and 'A' is the whole array in which all the values are located. Upon running the loop from starting value to the last value located in the given array 'A', the required original value can be extracted.

A. Open Addressing

According to Kriche in [4], a sub-alternate method is provided to map the original message onto the hash table. As mentioned previously, that the hash function has its own

index value which matches with the index values of hash table to feed the original value into it, however there is a possibility that the required index is not available in the hash table. In that case, again a searching algorithm is used to search for the free location to map the original value which also corresponds to the same time complexity as the total worst case time complexity of hash table.

In order to reduce the time complexity, which however is a trial and error based method, the hash function could also be re-performed to change the index value of the hash function so as to make sure the availability of the location at the respective index of hash table, which still does not guarantee the success but is desirable. This method is called the *open addressing* and ensures better search results.

In such searching methods, one more important parameter to be considered is that each preceding and succeeding value provided in the hash tables has to be mapped in such a way that if any previous value is deleted, then it might get troublesome to find the value which was located at the next of that deleted value. For example, if 'a' is deleted, then it might affect the search of 'b'. In order to avoid this situation, the value 'a' is deleted logically however that value is presented physically to locate its previous and next value as long as the preceding and succeeding blocks are not free. The value at index which is logically deleted but physically present cannot be used to store any new value in it.

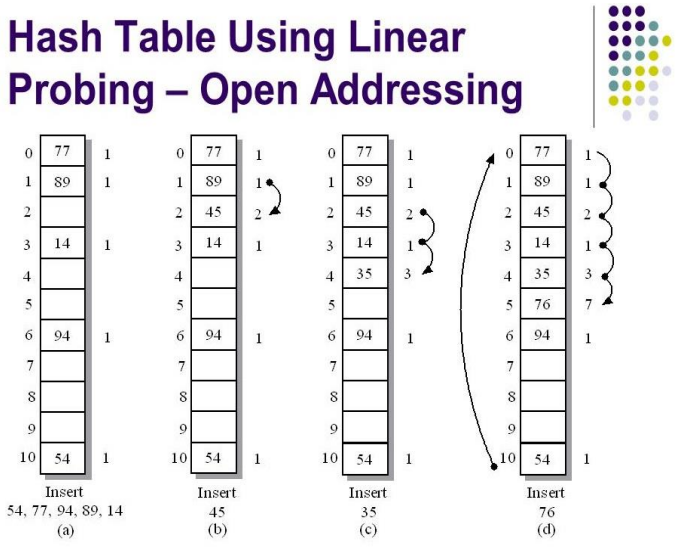

Fig. 7 Linear searching – open addressing

The figure 7 presents the working mechanism of value searching via open addressing. When the location at hash table is already filled, it goes for the next index. In order to make sure the successful operation of open addressing, it is crucial to leave some slots of the table as free slots. Otherwise, in case of collision, the hash table would not be able to locate all the stored original values/message.

V. CONCLUSION

In this paper, the study on blockchain technology is done and their operating algorithms and working principle has been studied. It is determined that the hashing is the essential phase in any given blockchain network. The model of individual blocks in P2P communication has also been studied. In blockchain technology, the encryption and decryption is a tedious task as once the message is encrypted, it can only get decrypted at the receiving end by means of mapping which requires the effective usage of hash table which itself has its own pros and cons. However, it is understandable that the blockchain technology is the considerably safe and secure for P2P networking.

REFERENCES

[1] Z. Zheng, S. Xie, H. Dai, H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus and Future Trends". 6th IEEE International Congress on Big Data, 2017.

[2] B. Koteska, A. Mishev, "Blockchain Implementation Quality Challenges: A Literature Review", Proceedings of the SQAMIA 2017: 6thWorkshop of Software Quality, Analysis, Monitoring, Improvement, and Applications, Belgrade, Serbia, 11-13.9.2017

[3] Jscrambler's articles, "Hashing Algorithms", Web Link: https://blog.jscrambler.com/hashing-algorithms/, retreieved date: December 3, 2018.

[4] Ignacio Kirchevsky, "Hash Table" Web Link: http://www.kriche.com.ar/root/programming/data%20structures%20&%20algorithms/hashTable.html, retreived date, Decemeber 5, 2018.

[5] Codility, chapter 3, "Time Complexity" Web Link: https://codility.com/media/train/1-TimeComplexity.pdf, retreived date: December 5, 2018.

[6] C. Holotescu, "Understanding Blockchain Technology and How To Get Involved". 14th International Scientific Conference E-learning and Software for Education, Apr. 19-20, 2018.

[7] What is Blockchain Technology? A Step-by-Step Guide For Beginners", Blockgeeks, 2018. [Online]. Available: https://blockgeeks.com/guides/what-is-blockchain-technology. Date retreived: 7 December, 2018.