International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

Discovering the Bitcoin Double Spend using Lost Agreement Amount

A. Murugan, J. Vijayalakshmi

Abstract: ***In the modernized world like digital world, traditional way of payments through banks and other third parties are out of sphere. To meet the digital competency digital token like bitcoin based crypto currency payment is required. Lots of business persons are moving towards the digital way of secure payment. Intruders like hackers hamper the digital token and make immortality in the transaction which in turn create the double spend. Double spend is a serious threat in the Bitcoin network. Our research work focuses on double spend detection of transaction before it gets confirmed and added to the block by the miners. The proposed new architecture for detecting double spend using Dual Payout based on Lost Agreement Amount (DPL2A) will identify one of the ways that double spend attack occur before it is added to the blockchain. This architecture gives the clear identification of double spend attack and their full details of transaction occurrence so that when it is broadcasted into the peer-to-peer network, the network nodes will use this architecture to detect double spend, its occurrence is fully prevented and only the genuine transaction will be added to the blockchain.***

Keywords: Bitcoin, Double spend attack, Blockchain, UTXO, Mining

I. INTRODUCTION

Blockchain is a data structure which holds the collection of linked blocks where each block contains financial transactions which is replicated across number of peer-to-peer systems in almost real time. Blockchain uses cryptography and digital signatures techniques in order to prove identity, authenticity and enforce access rights for read or write operations. The data inside blockchain was written by certain participants and can be read by a wider audience around the world. This uses the mechanism which makes harder to change historical records and can be easily detected when someone is trying to do mischief [1]. A blockchain is a shared, immutable public ledger of all transactions or digital events which are executed and shared among participating transactors. Each transaction in the blockchain is verified by consensus mechanism by considerable participants.

Manuscript published on 30 September 2019
* Correspondence Author
Dr. A. Murugan*, Associate Professor & Head, PG & ResearchDepartment of Computer Science,Dr. Ambedkar Government Arts College (Autonomous),Affiliated to University of Madras, Chennai, India.amurugan1972@gmail.com
J. Vijayalakshmi, Research Scholar, PG &Research Department of Computer Science, Dr. Ambedkar Government Arts College (Autonomous),Affiliated toUniversity of Madras, Chennai, India. jeyamaha2002@gmail.com

The blockchain contains a verifiable record of every single transaction that had done previously. This allows for trust minimized transactions between pseudonymous parties without requiring a trusted intermediary [2]. Blockchain builds trust among peers based on the following attributes like distributed and viable, secure, private and indestructible, transparent and adaptable, consensus based flexible and valid transactional support. The transaction data are stored in blocks which are linked together to form a chain. Blocks maintain the logging time of transaction sequence and record the confirmation time which are governed by rules agreed by the participants of network [3]. Bitcoin blockchain is a complicated system which aims to support reading and writing data by open access technique and lack of centralized power or control.

In Bitcoin systems data is stored in blockchain and it is distributed through peer-to-peer network. Consensus is implemented based on longest chain rule and the rules upgradation is carried out based on Bitcoin Improvement Proposals (BIPS). The transaction submission, validating transactions, reading data and adding blocks to blockchain are done based on anonymous name. Misbehavior in network is handled through proof-of-work [1]. The bitcoin system makes use of blockchain architecture and principles for making a database of both secured and widely distributed. Distributed ledgers like blockchain eliminate the need for central authorities to certify the possession of property and clearing transactions.

The bitcoin blockchain environment is depicted in Figure 1. The Bitcoin's blockchain environment holds four layers [4]. The transaction part holds the state transition function and unspent transaction outputs (UTXO) which takes roughly 60 minutes or 6 blocks time for confirming the transaction. The block part includes one coinbase transaction and the order of transactions. It also additionally includes timestamp, hash of the previous block, a merkle root, a difficulty target, target adjustment and nonce value. The miner node attempt to reach the difficulty target value based on the nonce value. Whenever it found nonce it writes one coinbase transaction to earn bitcoin rewards and can start to use that rewards after passing 100 blocks from mining succeed.

Bitcoin Economy
Miners
Blocks
Transactions

Figure 1. Bitcoin Blockchain Environment

People and exchanges use bitcoin economy to exchange their fiat currency to bitcoin and vice versa. Since bitcoin is a digital money which is easier to duplicate than actual money which means some fraud persons can manipulate their way to paying more than once with the same bitcoin which is known as "double spending". Whenever a transaction is double spent, the network latency causes the synchronization issue which can be solved by verifying the transaction existence across the whole network [4]. In section 2, a study is carried out on handling of double spending problem. Section 3 discuss about the blockchain implementation mechanism and its principle of working under bitcoin network. In Section 4, how blockchain detects double spend attack and how it handles the situation was depicted. In section 5 the proposed Dual Payout based on Lost Agreement Amount (DPL2A) architecture which was elaborately discussed and how its implementations were carried out. Finally, the paper concludes the reason behind the prevention of double spend attack.

II. II. LITERATURE SURVEY

Double spending problem is hidden flaw in a crypto currency or other digital cash scheme in which the same single digital token can be used up more than once and this is possible because the digital token which consists of a digital file that can be duplicated or falsified. The originator of Bitcoin, Satoshi Nakamoto gave the solution to the double spending problem in his whitepaper of Bitcoin by the way of introducing distributed consensus protocol using blockchain which was applied to bitcoin for solving double spend attack and he also specified the rate of probability that the attacker would use to create double spend. Pinar et al. [5] show the derivation of mathematics relied upon by Nakamoto for created model of double spend attack and also validate the model with Monte Carlo simulation for determine the imperfect component model

Joseph et al. [6] made an analysis on three main technical components of Bitcoin which are transactions (including scripts), the consensus protocol and the communication network. They said that the bitcoin solved double spending by publishing all transactions in a global permanent transaction log and any individual transaction output may only be redeemed in one subsequent transaction. Ivan et al. [7] propose a new a new peer-to-peer system architecture to prevent double spending without requiring an online trusted party, hardware or any tamper resistant software. They provide a proof of concept implementation for internet vendors along with security proofs. They adapt the witness approach to ensure real time double spending prevention. Diego et al. [8] compares the traditional model, double spent prevention handled by trusted parties with the current bitcoin system double spend prevention handling. They said every participant in the bitcoin network keeps a copy of ledger recording transfers related to digital coins or assets over time. They concluded that chains of standard transactions can protect against double spending problem.

John et al. [9] discuss about the double spending happened in day to day business activities like fast food restaurants and vending machines that deliver their products quickly upon payment. They introduce a shadow framework for solving this problem of double spending and to determine the effectiveness of these defenses and the impact of this effect on individual client and network performance. Carlos et al. [10] deliberate about the attack models that can assign possible time advantage to attacker agents in the bitcoin network. They present two double spend attack models for bitcoin generalized and time-based models. Both are hash rate-based models which supports partial advancement towards block production is influenced by time. They concluded that partial block production is not negligible for analyzing and detecting double spend attacks in bitcoin.

Mauro et al. [12] specified the possibility rate of double spend and its variants attack like Finney attack, Bruteforce attack and vector 76 attack is high when the miner or the mining pool is mining the blocks at faster rate than the rest of the bitcoin network. Sarah et al. [13] examined the limitation of bitcoin anonymity and able to cluster the anonymity based on some heuristics which determine the shared ownership of bitcoins using public and private key pairs. Ozisik et al. [14] proposed a new architecture for increasing transparency and detecting eclipse attacks in both bitcoin and blockchain based network protocols.

III. BLOCKCHAIN MANAGEMENT IN BITCOIN SYSTEM

Blockchain technology runs bitcoin network which was conceived in 2008 and first implemented in 2009 where it serves as an open access distributed ledger of all transactions [3]. This technology was first conceived by Satoshi Nakamoto. According to Wikipedia's definition, "A blockchain is a database that is distributed and continuously growing records called blocks which are secured from tampering and revision and support peer-to-peer exchanges in a secured, public and non-reputable way". The blockchain holds the proof of ownership for every bitcoin creation in the network. This holds the consensus of replicated, shared and synchronized digital data geographically distributed across multiple sites, countries and institutions. Every node holds the copy of blockchain in decentralized system. Blockchain follows certain principles in which the network consists of set of independent nodes together communicated by means of message broadcasting. Every node has its own copy of the blockchain. A node can at least connect too few nodes on the peer-to-peer network except all node connections. A blockchain is a universal ledger that uses cryptography and incentives to record transactions in meddle evident way which allows trust minimized transactions without requiring trusted intermediary [11].The primitive construction of blockchain was based on cryptographic hash functions. In the following Figure 2 the architecture was depicted with the use of cryptographic hash functions in both transaction and block construction which was depicted clearly. Blocks implements the hash function as transitive hash function and transaction applied hash function for Merkle tree construction.

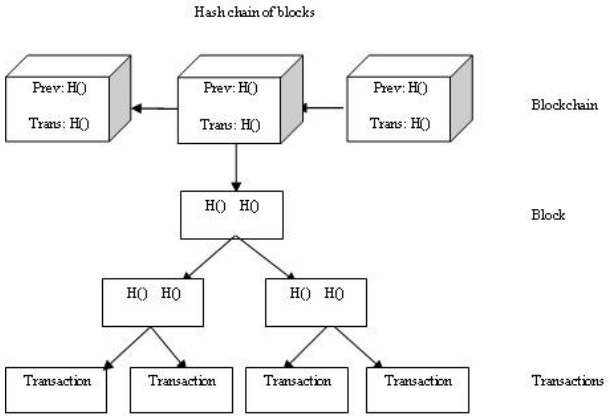

Figure 2. Blockchain Transaction and Block Construction

Cryptography techniques provide secure communication in the presence of third parties. The cryptographic hash function requires three additional properties like collision resistance, deterministic hiding and puzzle friendliness. This cryptographic hash function is used in bitcoin to generate public key of our wallet from the private key. The Blockchain implements the transitive hash functions which mean changing anything in one part would result changes in the subsequent part also hence the blockchain is called it as chain of hashes. Hash pointers are a core component of trust minimized property of blockchain which is used to point out previous block data structure. Hash pointers ensure the integrity of ledger. To pass through each data, Merkle tree is used. Merkle tree is the hash digest representation of all transactions inside the block.

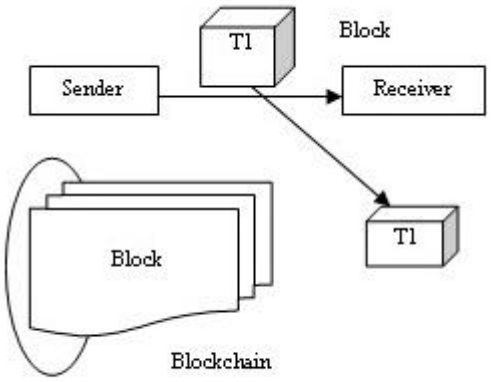

Figure 3. Blockchain Operation

In Figure 3, the operation of blockchain was discussed based on transaction inclusion into the block. The sender sends a transaction which is put inside a new block. The new block is broadcasted to the network for validation. The miners in the blockchain network works together as a pool approve the incoming new transaction and validate it using cryptographic techniques. The block is then added to the chain which provides public, permanent, unrepeatable record of the transaction. The receiver then received the money from sender and now he can send goods based on order by sender.

BLOCKCHAIN DIAGNOSE OF DOUBLE SPEND ATTACK AND ITS PREVENTION METHOD

The following progression discusses the conventional way of transaction inclusion in block; double spend attack routine, and how original transaction becomes invalid transaction due to double spend attack. The Figure 4 denotes valid transaction that is added to the blockchain in a honest manner [1]. The Figure 5 denotes part of the attacker which secretly creates longer chain of blocks which excludes the payment to the retailer and includes the payment to him by publishing the longer chain. The nodes on the network follows longest chain rule adoption hence they ignore the honest block with receiver payment and make that block as orphaned block.

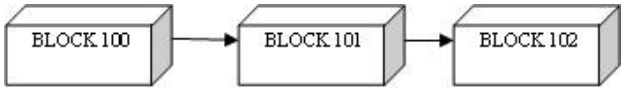

Figure 4. Pay receiver transaction is included in the blockchain (honest blockchain)

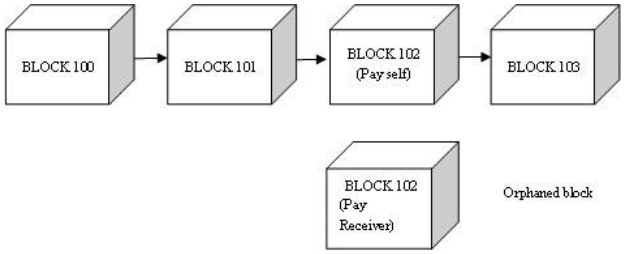

Figure 5. The attacker publishes the long chain which includes double spend attack

In Figure 6, the original block was rejected because the original payment to the receiver will be deemed invalid by the honest nodes because those bitcoins were already spent in the longer chain. This attack is called it as double spend attack because the same bitcoin was spent twice and the dishonest block was added to the blockchain and the honest block was rejected. In order to avoid this attempt, the bitcoin provided certain restrictions to add blocks on the blockchain by implementing computationally expensive proof of work for adding blocks. The implementation of proof of work process was depicted in Figure 7. The computation of proof of work is like guessing a game where block makers needed to guess a number which when added to the block data contents results in a hash that is smaller than certain number. The number is related to the difficulty value of total network processing power. The following picture depicts how proof of work is done on the mining pool.

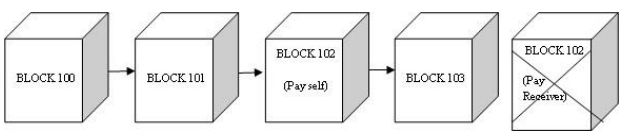

Figure 6. Original transaction is no longer valid now

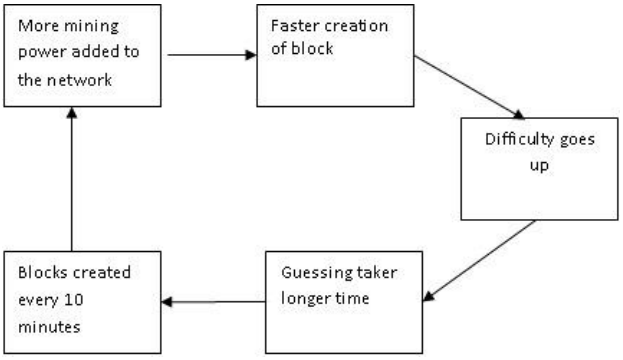

Figure 7. Proof of work implementation cycle

IV. PROPOSED WORK

To build an effective environment which supports the detection of double spended transaction before it gets added to the block of the blockchain. In Figure 8, the current scenario of blockchain working principle is described. Generally, the blockchain adds all incoming transactions, their input and output to the *unspent transaction output* (UTXO) pool. The miners validate the incoming transactions by checking whether input is available in UTXO pool then they consider that the transaction is unspent elsewhere and they add the incoming transaction to the upcoming block without validating whether they spent that output already or not. To prevent mischief attack like this double spend attempt, this paper had proposed a new architecture of finding double spending using *Dual Payout based on Lost Agreement Amount* (DPL2A) which was described in Figure 9, which includes various additional pools for verifying the double spend transaction and storing the original transaction to the *confirmed inputpool*.

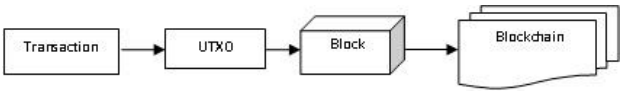

Figure 8. Blockchain Architecture

The transaction from the *unspent transaction output* (UTXO) pool was taken and it was moved to *unconfirmed inputpool* along with certain additional attributes for finding double spend attempt which was described in Figure 9. From the *unconfirmed inputpool* the original transaction was identified based on comparing the values of *unconfirmed input pool* to the *sender utxopool* and sender *stxopool*. If there was a match found in sendername, *transactionhash*, *outputindex*, transfer *value* and *time* then those records were taken and update the *usedflag* and *usedcounter* part of both *unconfirmed inputpool* as well as *sender utxopool*. Similarly, the values of *unconfirmed input pool* were compared with *sender stxopool* and their *usedflag* and *usedcounter* part of *sender stxopool* was also updated. Then based on the status of *usedflag* and *usedcounter* part of inputpool the double spended data were identified and stored into the double spend table. The above process was described in Figure 10.

The advantage of this method is this use additional verification of transaction comparing to the standard bitcoin transaction verification along with various pools like *sender utxopool*, *sender stxopool* and *unconfirmed input pool* for identifying double spended data. In some cases, like if buyer spent his output multiple times or if intruder tries to replicate data for creating double spend or in some case of network delay the double spend might happen without changing the UTXO content. The miner may consider that the data present in UTXO is valid then they may take the faulty transaction as valid. To address this issue, our architecture would find how many times that input was reused based on counter and usage flag which cannot be easily modified by the sender or others.

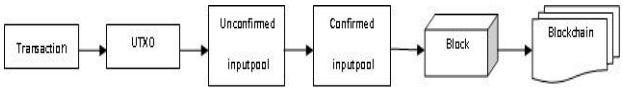

Figure 9. Dual Payout based on Lost Agreement Amount (DPL2A) Architecture

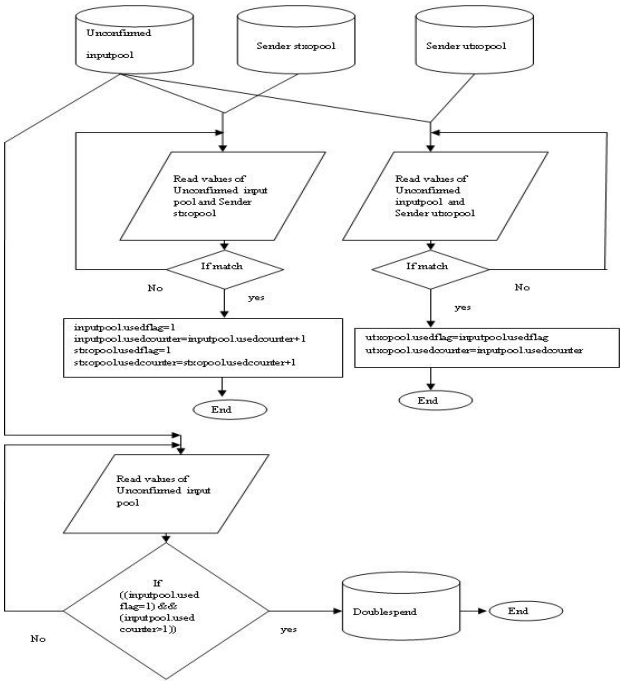

Figure 10. Finding Double spend data from Unconfirmed Inputpool

This model was implemented and tested based on four transactors (Alpha, Bravo, Charlie, and Delta) who played the main role in the transaction process whose blockchain structure was discussed in Table 1. Generally, all pool structure followed the sequence of sender name, transaction hash, output index, transfer value, time, usage flag, usage counter and recipient name. Transaction structure followed the sequence of sender name, recipient name, set of inputs and outputs, transfer value to the recipient, timestamp of transaction occurrence, size of each transaction, and transaction hash which includes all fields like timestamp, transfer value, size, sender, recipient, inputs and outputs. Block structure contained the fields of block height, transactions that were included in the blockchain, transaction counter, and previous block hash/digital signature, block timestamp, block hash, proof of work values like nonce and transaction hash digest value like merkle.

The Blockchain structure includes global chain of blocks that were constructed earlier and set of pools like unspent transaction pool, input pool and spent transaction pool along with public and private keys used to verify, validate and to enforce data integrity across peer to peer network transfer. For each transactors blockchain they maintained their own utxopool, stxopool, and pool.

Table 1: Table structure followed in DPL2A architecture

transaction

sender	recipient	inputs	outputs	hash	value	time	size

block

block height	Transactions	transaction counter	previous hash	Time stamp	blockhash	nonce	merkle

blockchain

blockchain	utxopool	pool	stxopool

inputpool,outputpool,utxopool

sender name	transactionhash	output index	value	time	used flag	used counter	recipient name

alphastxopool,bravostxopool,charliestxopool,deltastxopool

sender name	transactionhash	output index	value	time	used flag	used counter	recipient name

alphautxopool, bravoutxopool, charlieutxopool, deltautxopool

sender name	transactionhash	output index	value	time	used flag	used counter	recipient name

alphapool, bravopool, charliepool, deltapool

sender name	transactionhash	output index	value	time	used flag	used counter	recipient name

doublespend

sender name	transactionhash	output index	value	time	used flag	used counter	recipient name

confirmedinput

sender name	transactionhash	output index	value	time	used flag	used counter	recipient name

The *utxopool* part stored the unspent transaction of each transactors involved in the transaction. Similarly *pool (transpool)* part was implemented as same as *utxopool* architecture without digital signature. The *transpool* include sender name of the transaction, transaction hash value, output index of the transaction, transfer value and time of transaction carried out. The *transpool* was a common pool for every transactor in which they hold all transactions including incoming (previous transaction output) and outgoing (sending value to recipient) transactions. The *stxopool* follows the same structure as *transpool* which holds only the spent transaction details. Alternately, the *transpool* structure holds involvement of both *utxopool* and *stxopool* transactions of each transactors. For every transactor these pools were individually maintained by blockchain. Whenever a transaction arises the blockchain will collect information of sender and recipient transactions based on their transaction pool management. The double spend attempt is recognized based on comparing each and every values of input in the unconfirmed input pool to the corresponding transactors spent transaction pool. If matches found then they determine the number of occurrences of transaction based on the usage flag and usage counter. If the counter value exceeds normal limit then it is treated as double spend.

Algorithm getdoublespenddata()

```
// Getting the double spend data from unconfirmed input pool based on comparing sender stxopool and sender utxopool
{
    for every row in inputpool
    {
        for every row in senderstxopool
        {
            if match exists
                read row
                if row = None
                    break
                updatedata()
        }
    }
    for every row in inputpool
    {
        for every row in senderutxopool
        {
            if match exists
                read row
                if row = None
                    break
                updatedata()
        }
    }

for every row in inputpool
    {


 if ((usedflag=1) && (usedcounter>1)
            read row
            if row = None
                break
            write each row in doublespend table
        }
    }

}
```

Algorithm updatedata()

```
//update the inputpool and senderstxopool table values based on condition
{
     for every row in inputpool
    {
        for every row in senderstxopool
        {
            if match exists
                read row
```

```
                    if row = None
                        break
                    set inputpool.usedflag=1
                    set inputpool.usedcounter = inputpool.usedcounter+1
                    set senderstxopool.usedflag=1
                    set senderstxopool.usedcounter=senderstxopool.usedcounter +1
            }
        }
         for every row in inputpool
         {
            for every row in senderutxopool
            {
                if match exists
                    read row
                    if row = None
                        break
                    set senderutxopool.usedflag=inputpool.usedflag
                    set sendersutxopool.usedcounter=inputpool.usedcounter
            }
        }
}
```

V.RESULTS AND DISCUSSION

The projected architecture of Dual Payout based on Lost Agreement Amount (DPL2A) was to find the double spending attack which was implemented on Intel i3 Core processor with 2GB RAM at Ubuntu 32-bit operating system with an application of Python language with backend as Postgresql database. The implementation work was carried out based on 4 transactors and their 4 corresponding blockchain nodes. Each node of the blockchain separately maintained *transpool*, *utxopool* and *stxopool* databases. In addition to it globally maintained two input pools one for unconfirmation and another for confirmation. Whenever a new transaction arrives it is added to the *transpool* of transactor as well as unconfirmed input pool. For every transaction the corresponding pools of transactors are updated. We have tested unconfirmed *inputpool* with total of 52 ,114 and 156 records based on the inclusion of transactions with the minimum of 5 to maximum of 24 transactions. The projected architecture determines the number of occurrences of double spends and it was verified based on manual checking. The outcome of this experiment is shown in the below Table 2. For 52 records transactor 1 was double spender for 8 times, transactor 2 was double spender for 3 times, transactor 3 for 1 time and transactor 4 for 0 times. For 114 records transactor 1 was double spender for 11 times, transactor 2 for 10 times,transactor 3 for 6 times and transactor 4 for 3 times. For 156 records transactor 1 was double spender for 33 times, transactor 2 for 42 times, transactor 3 for 45 times and transactor 4 for 36 times Hence, we concluded that if number of transactions increase there might be a greater number of double spend attack. The proposed architecture will determine the number of occurrences of double spend attack accurately. We store all these double spender data list in a separate table which can be broadcasted to the other nodes in the network to avoid double spend further.

Table 2: Determining double spend count based on number of transactors and input records.

Total Number of Transactors	Total Number of Records which portrays double spend occurrence		
	52	114	156
Transactor 1	08	11	33
Transactor 2	03	10	42
Transactor 3	01	06	45
Transactor 4	0	03	36

VI. CONCLUSION

This paper had carried out the analysis of double spend attack and it reviewed the solution of double spending handled by the Bitcoin network system. This also discuss about the how the blockchain will diagnose this double spend attack and how it provides solution were discussed. It also discussed about the root cause of double spend attack that happened inside the network both in transaction wise inclusion as well as block wise inclusion. The proposed DPL2A architecture detected the double spend data from the transaction list by comparing with the sender unspent output and spent output list, which is earlier than the block inclusion in a reliable and efficient manner. Hence, the paper concludes that modest adjustment in the execution of transaction handling will definitely prevent the double spend attack in future.

REFERENCES

1. "A gentle introduction to blockchain technology." Brave new coin,2015, "https://bravenewcoin.com/insights/a-gentle-introduction-to-blockchain-technology", last access:03/04/2019.
2. Crosby, Michael, Pradan Pattanayak, Sanjeev Verma, and Vignesh Kalyanaraman. "Blockchain technology:
3. Beyond bitcoin." *Applied Innovation* 2, no. 6-10 (2016): 71.
4. Jerome kehrli, "Blockchain explained part 1", 2017, "https://blog.netguardians.ch/blockchain-explained-part-1", last access: 18/12/18.
5. Rakuten, "Blockchain momentum part 2 of 2", 2018," https://techblog.rakuten.co.jp/2018/08/16/blockchain-momentum2 ", last access: 18/12/18.
6. Ozisik, A. Pinar, and Brian Neil Levine. "An explanation of Nakamoto's analysis of double-spend attacks." *arXiv preprint arXiv: 1701.03977* (2017).
7. Bonneau, Joseph, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A. Kroll, and Edward W. Felten. "Sok: Research perspectives and challenges for bitcoin and cryptocurrencies." In *2015 IEEE Symposium on Security and Privacy*, pp. 104-121. IEEE, 2015.
8. Osipkov, Ivan, Eugene Y. Vasserman, Nicholas Hopper, and Yongdae Kim. "Combating double-spending using cooperative P2P systems." In *27th International Conference on Distributed Computing Systems (ICDCS'07)*, pp. 41-41. IEEE, 2007.
9. Romano, Diego, and Giovanni Schmid. "Beyond Bitcoin: a critical look at Blockchain-Based systems." *Cryptography* 1, no. 2 (2017): 15.
10. Podolanko, J., Jiang Ming, and Matthew Wright. "Countering Double-Spend Attacks on Bitcoin Fast-Pay Transactions." In *Workshop on Technology and Consumer Protection*, pp. 1-3. 2017.
11. Pinzón, Carlos, and Camilo Rocha. "Double-spend attack models with time advantange for Bitcoin." *Electronic Notes in Theoretical Computer Science* 329 (2016): 79-103.
12. Taylor pearson, "How does blockchain technology work", 2018,"

https://taylorpearson.me/blockchain-explained", last access: 20/12/18.
13. Conti, Mauro, E. Sandeep Kumar, Chhagan Lal, and Sushmita Ruj. "A survey on security and privacy issues of bitcoin." *IEEE Communications Surveys & Tutorials* 20, no. 4 (2018): 3416-3452.
14. Meiklejohn, Sarah, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage. "A fistful of bitcoins: characterizing payments among men with no names." In *Proceedings of the 2013 conference on Internet measurement conference*, pp. 127-140. ACM, 2013.
15. Ozisik, A. Pinar, Gavin Andresen, G. D. Bissias, Amir Houmansadr, and Brian Neil Levine. "A Secure, Efficient, and Transparent Network Architecture for Bitcoin." *UMass Amherst, Tech. Rep. UM-CS-2016–006* (2016).

AUTHORS PROFILE

Dr. A. Murugan is an Associate Professor and Head of the Department of PG and Research in Computer Science. His research interests focus on Molecular Computation, Graph Theory, Data Structure, Analysis of Algorithms, Theoretical Computer Science and Cryptocurrency Security. He completed Ph.D in Computer Science, Master of Science(M.Sc) in computer science. He published 3 books in computer science subjects. His publication focuses on DNA Computing, Cloud Computing, Mobile Communication and Security aspects of all networking. He published more than 90 papers in both International and National Journals and springer proceedings. He has produced 8 Ph. D research scholars. He is an Editorial Board Member, Technical Advisory and Reviewer in various conferences and journals. He acted as a Board of Studies member in various reputed colleges of Chennai. He plays a variety role in both colleges and university.

J. Vijayalakshmi is a full-time research scholar of PG & Research Department of Computer Science in Dr. Ambedkar Govt. Arts College. She obtained her Master of Philosophy(M.Phil) degree in Computer Science at Bharathidasan college, Tiruchirappalli in the year 2006 and obtained her Master of Computer Application(M.C.A) under University of Madras in the year 2004 and she also completed her Bachelor of Science(B.Sc) in Computer science under University of Madras in the year 2001. She had working experience of 9 years in private colleges. She has published her papers in international journal and also in springer proceedings. Her interest is focus on providing security in virtual currency management.