BITCOIN AND DOUBLE-SPENDING: HOW PAVING THE WAY FOR BETTERMENT LEADS TO EXPLOITATION

Nujud A. Alabdali, Mohammed A. AlZain, Mehedi Masud, Jehad Al-Amri, Mohammed Baz
College of Computers and Information Technology,
Taif University, Saudi Arabia

Abstract - Bitcoin is a cryptocurrency that has gained popularity over the last decade. The scheme of its implementation does not call for a trusted third party to validate the coins spent or the currency made. Therefore, Bitcoin has been developed to grant users the privilege of purchasing anonymously and performing other transactions while not going through a financial institution. In this paper, we review the challenge of keeping the identities of Bitcoin users anonymous while maintaining secure payments. More specifically, we present double-spending, a security attack, in fast payments and examine some of the counterattack initiatives taken to protect payments in Bitcoin.

***Keywords*:** double-spending, security attack, fast payments, Bitcoin**.**

I. Introduction

Bitcoin has made it a reality for people to digitally perform transactions without consulting a third party for validating their payments. It has been integrated in many businesses such as fast food chains [1] and other places are accepting Bitcoin as a valid currency. The number of Bitcoin users is growing as well as its economic value [2] which makes it an attractive hub for trading coins of value.

Double-spending attack is a security issue in Bitcoin that allows users to spend the same coin on different items. A user can spend his coin on purchasing pizza and uses the same coin to purchase glasses at a different store. This attack made the developers of Bitcoin implicitly acknowledge the vulnerability in the implementation of the cryptocurrency and advise users to perform fast payments for low-cost purchases [1].

In this paper, we argue that maintaining completely secure fast payments in Bitcoin is not always reached. We believe that, with the countermeasures proposed, there is a way for the attacker to successfully double-spend a coin, especially in fast payments. We show how maintaining security while abiding to the implementation standards and expectations of Bitcoin can be challenging, mostly in keeping the identity of users private at all cost.

II. Proposed Methodology

We structure the paper to give an extensive introduction to Bitcoin. We explain how transactions are issued and communicated through peers. We also delve into the steps taken to process a payment using Bitcoin, detailing the different types of payments and how they are received and validated. We then present the conditions needed for a successful double-spending attack to take place, mainly in fast payments. We finally introduce known proposed countermeasures and show how they can still not provide a completely secure environment in protecting fast payments against double-spending attacks. The methodology is shown in Figure 1.

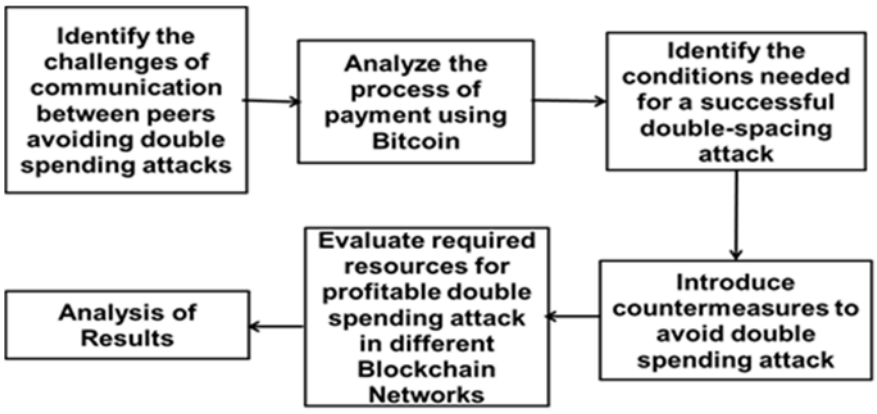

Fig 1 : Methodology of research

III. Bitcoin in a Nutshell

Bitcoin is a decentralized payment system that relies on Peer-to-Peer infrastructure. Transactions are made in the form of blocks. Each transaction is issued by a user and once verified, it is included in a block. The block is appended to a chain of blocks unless it is the first block to be generated for the BTC. The longest blockchain in the network is trusted amongst nodes/users to be valid and secure. To purchase items using BTCs, a number of computational steps have to be performed beforehand.

A. Transacting in Bitcoin

The main objective of issuing transactions in Bitcoin is to allow for communication between peers to grow the network on which users process payments. The communication is therefore entitled to transfer BTCs between peers and their ownership as well. The payment system used in Bitcoin is based on Proof-of-Work scheme that helps in securing transactions.

Users transfer their coins by using their digital signatures. Each user uses his/her own private key along with other related information to sign a hash of the previous transaction and the public key of the intended new owner.

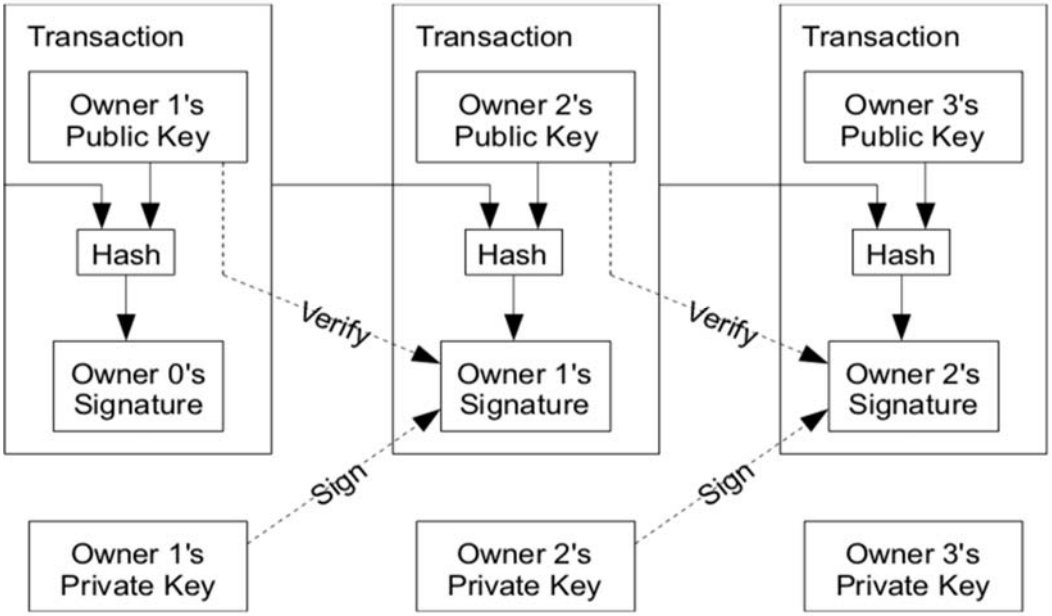

Fig 2 : How digitally signing a transaction works in Bitcoin [7]

B. Validating Transactions in Bitcoin

Bitcoin depends on the timestamp server to validate transactions. Timestamp server is implemented on the P2P network by taking a hash of a block containing items and publicly publishing the hash [7]. The public publication of the hash discards the need for a trusted authority to validate transactions, making peers agree on receiving the to-be-validated transaction at the specified time [7].

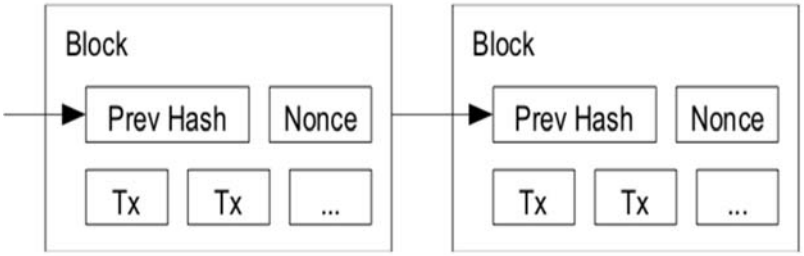

Fig 3: Publicly publishing a hash of a block of items [7]

PoW evaluates transactions based on the result the nonce value gives. The nonce value is hashed along with the previous block's hash and produces a value. The value should lead to a result that begins with a special number of zeros [7]. If the value is not found, computation is repeated, incrementing the nonce value until the exact number of leading zeros is achieved [7].

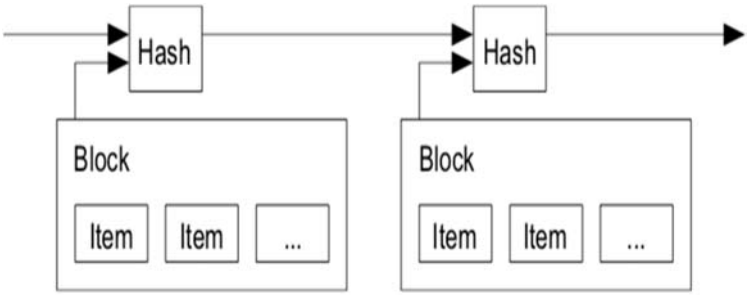

Fig 4: PoW illustration [7]

Table 1 shows a Bitcoin block having two transactions. The special leading zeros are shown in the hash attribute, meaning that the transaction was confirmed. In addition, there is an incentive reward given to a node/user that generates a new block to the chain. The incentive is 50 BTCs [1]. The incentive is shown in the Source & Amount column in the table below.

Table 1. A Bitcoin block containing two transactions with the special number of zeros shown in the hash attribute [1]

Hash: 000000000043a8c0fd1d6f726790caa2a406010d19efd2780db27bdbbd93baf6
Previous block: 00000000001937917bd2caba204bb1aa530ec1de9d0f6736e5d85d96da9c8bba
Next block: 00000000000036312a44ab7711afa46f475913fbd9727cf508ed4af3bc933d16
Time: 2010-09-16 05:03:47
Difficulty: 712.884864
Transactions: 2
Total BTC: 100
Size: 373 bytes
Merkle root: 8fb300e3fdb6f30a4c67233b997f99fdd518b968b9a3fd65857bfe78b2600719
Nonce: 1462756097

Input/Previous Output	Source & Amount	Recipient & Amount
N/A	Generation: 50 + 0 total fees	Generation: 50 + 0 total fees
f5d8ee39a430....:0	1JBSCVF6VM6QjFZyTnbpLjoCJ....: 50	16ro3Jptwo4asSevZnsRX6vf...: 50

IV. Payments Types

If a user wants to buy a meal at a fast food restaurant such as Burger King, he will not be happy with standing in front of the cashier waiting for an avenge time of 10 minutes to get his transaction validated. Frustration would also be sensed in the cashier as he waits for the transaction to be confirmed, resulting in a negative user experience on both ends. Therefore, fast payments cannot rely on transaction confirmation to processing payments. Payments are achieved through another verification approach. This demand in fast payments opens up an opportunity to improve the experience of purchasing via Bitcoin. However, this betterment leads to exploitation. Double-Spending happens more frequently in fast payments due to the fact that transactions are not confirmed at the instant of purchase.

Payments in Bitcoin are divided into two types based on the operation performed on the transaction received: *Transaction Confirmation* and *Transaction Reception* [1]. The former is applied or perceived as slow payment. Slow payments do not necessarily deliver goods at the instant of purchase and therefore there is a decent amount of time to check the validity of the transaction and confirm it. Transactions will be checked and blocks will be generated and appended to the chain eventually once verified.

Therefore, the chances of having a double-spending attack is not high compared to fast payments unless the user is dishonest: the payee has the will to tell the vendor that the transaction will take time to be confirmed and it is the vendor's choice to either accept the offer or reject it. If the vendor accepts to receive a transaction without having it confirmed and it happens to have a dishonest user, then in this case a successful double-spending takes place.

Fast payments, on the other hand, have greater chances for double-spending attacks since they do not require verification of transactions at the moment of purchase. Fast payments rely on *Transaction Reception* where the vendor only cares about receiving the transaction instantly rather than the validity of the transaction. Places such as vending machines and fast food chains are likely to accept this form of payment.

Fast payments rely on Zero-Confirmation mechanism where there is no real work of confirmation of the transaction. The vendor will accept the transaction of his interest, the one holding the requested number of BTCs, and he will search his wallet to see if there is a matching transaction or address.

Bitcoin users are highly encouraged to perform fast payments on purchases of low value. In the next section, we examine the scenarios in which a successful double-spending attempt takes place, mainly in having the attacker send his transaction before letting the vendor redeem it subsequently.

V. How Payments Are Processed in Bitcoin?

There is a number of steps for payments to occur in Bitcoin. A great focus is directed towards issuing and validating transactions. The steps are adopted from [1]

- transactions are broadcast to the entire public network.
- transactions are received by peers.
- verification process undergoes the use of a nonce value.
- nonce value is hashed into a computational function along with other important information. If the result of the computation leads to an expected number of leading zeros, the transaction is verified and a block can be generated and appended to the user's digital wallet.

In more details, the nonce is hashed along with the remaining fields of information of the transaction such as "the Merkle hash of all valid and received transactions, the hash of the previous block, and a timestamp" [1]. If the result is below a specified value, the block is generated based on the proof of work submitted by the user.

The ownership of BTCs can be transferred as well. If user A wishes to send a number of BTCs to user B, then upon the process of sending and receiving, user B can claim ownership of the BTCs.

The process encompasses more details. For example, each user is referenced upon owning the bitcoin. Therefore, when user A sends his coin to user B, the coin is already stamped with the address of user A which is included in the hash value. The final result is a chain of blocks with each block representing the previous owner/user [1].

To authenticate BTCs, the user has to check the correctness of each block in the chain. Hence, the process of double-spending is not easy since the attacker has to re-compute all the calculations made for each block in order to insert his new double-spent coin without having the user/vendor know that it was spent before.

An important note should be given about the steps mentioned above: transactions are broadcast to the public network: each user on the network has a local storage known as a "memory pool" [1]. This pool is responsible for storing all the broadcast transactions which are not confirmed yet. Once a transaction is confirmed, it is taken out of the pool and appended to a chain of blocks. If the transaction happens to be confirmed by another user, it will be taken out from the current pool and sent to that user to be appended to his chain of blocks. Moreover, having a pool of unconfirmed public transactions does not obligate the owner of the pool to verify/confirm them since they were broadcast on the network with and without his will.

VI. Requirements for a Successful Double-Spending Attempt in Fast Payments

Two requirements are to be satisfied to possibly double-spend a coin. The two conditions work in favor of a scenario where the attacker is attempting to double-spend a coin at a different store rather than the same store. Also, it is assumed that the attacker knows the IP address of the vendor.

The trick is as follows: the attacker sends two transactions that have the same inputs but different outputs. The inputs construct the details of the transactions and the outputs are the IP addresses to which transactions are sent. The address of one coin is the vendor's and the other of a helper's. A helper performs under the control of the attacker [1]. The two transactions are sent at the same time and their probability to be received by peers on the network is similar.

For the first requirement to be satisfied, the time of receiving the original/first transaction (T_V) by the vendor must be less than the time of receiving the second one (T_A). If vendor receives (T_V) first, he is most likely going to ignore another transaction holding the same inputs. The second requirement to be satisfied is to have (T_A) confirmed first on the network by another node.

These two conditions are required to double-spend a coin in fast payments where the vendor relies on *Transaction Reception* rather than *Transaction Confirmation* to process payments.

VII. How to Maintain Security?

Security in cryptocurrency is a hot topic that keeps evolving. There is no definite way of completely securing a chain of blocks from any attempted attacks. There are, however, countermeasures that help reduce the effect of the attack and its chance to happen.

In this section we present some of the proposed countermeasures that have gained attention as means to reasonably prevent double-spending attacks. These countermeasures could work alone or may be combined to slightly have a better security [17-21]. We also introduce design standards to be followed in attempting to design double-spending attack countermeasures.

The cryptocurrency aspects detailed below are better to be considered when designing security measures in Bitcoin. These aspects are crucial in providing the promising results/outcomes upon using Bitcoin in purchasing.

A. Keep These Standards in Mind

- Performance
- Anonymity
- Decentralization

High computational performance is one major characteristic sought in fast payments using cryptocurrency. The transactions are expected to be accepted/confirmed fast and goods to be delivered at the instant of purchase. Performance in processing fast payments in Bitcoin is expected to have the following characteristics: short time response for processing tasks, high rate of processing work (throughput), low utilization of computer resources, high bandwidth, and short data transmission time. Nonetheless, major means of security may not work in favor of these characteristics.

Anonymity is a long-lasting challenge in cryptocurrency. The primary concept of developing cryptocurrencies, Bitcoin in specific, is to enable the user/customer to spend money without being known to the vendors or other customers. This aspect of Bitcoin presents a never-ending question of how to maintain complete anonymity and complete security at the same time. Nonetheless, some of the countermeasures work its way to give the best protection that could be made without affecting the anonymity of the users.

Decentralization was also a primary concern that led to the development of cryptocurrency and should be kept consistent in the design and improvement of any countermeasure in Bitcoin.

In the next section, we examine some of the countermeasures taken in stopping double-spending attacks.

B. Countermeasures Overview

Here, we take the two requirements/conditions needed for a successful double-spending attack and run scenarios on which the countermeasures explained below could be used. In other words, one requirement or the other or both may not be satisfied in order to attempt a successful double-spending attack. We also show how with each countermeasure there exists a possibility at which the attacker could still perform the attack.

Before we examine the countermeasures, we briefly review the steps taken to achieve a successful double-spending attack. The two conditions/requirements along with the process of performing the attack are explained below:

1. The attacker sends two transactions T_A and T_V where they both have the same input (amount of money included) and different outputs (the addresses to which they are sent)
2. which they are sent)

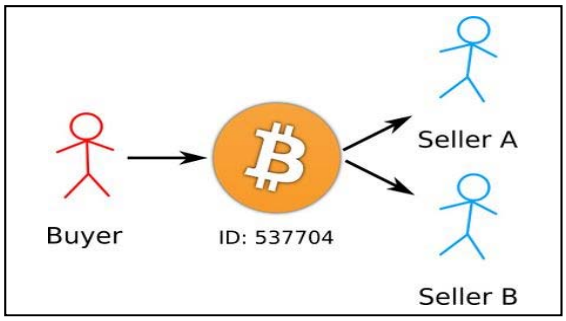

Fig 5: Illustration of step 1 (mentioned above) in forming a double-spending attack [15]

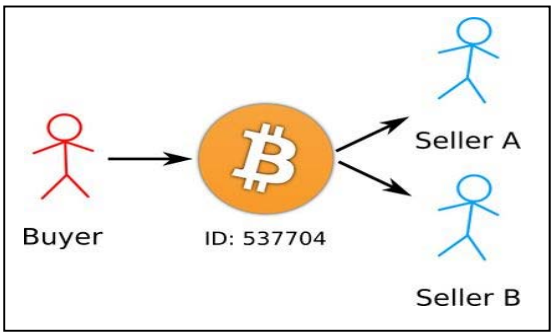

Fig 6: Illustration of step 4 (mentioned above) in forming a double-spending attack [16]

3. T_V is concerned with the vendor (only sent to him) and T_A is working under the supervision of the attacker (a helper to him)
4. The first requirement is for the vendor to receive T_V long enough before receiving T_A. This will enable the attacker to have his transaction accepted immediately by the vendor since there is no transaction confirmation in fast payments.
5. The attacker by now achieves his goal: getting the goods he desires and manipulating the vendor into accepting T_V while having the other transaction T_A received by another user/node on the network.
6. The attacker keeps his money by having his transaction T_V dropped later from the network since there is a similar transaction that was received first, which is T_A. — The helper's transaction was received and confirmed first in another node, by another user, leading to discarding the vendor's for having the same input. This leads to satisfying the second condition which is having T_A confirmed first.

The countermeasures mentioned in this section were proposed by Bamert et al. [13] and Karame et al. [14] and were designed to prevent double-spending.

The first countermeasure is inspecting propagation depth of transactions, proposed by Bamert et al. [13]. This proposal aims at reaching a higher depth in terms of nodes. A threshold is specified and has to be met if the vendor seeks secure purchases. The idea of this proposal is to let the vendor wait for the incoming transaction to be sent and received by a number of nodes before accepting it. The vendor is advised to not rush into accepting the transaction upon arrival and to evaluate its validity based on the number of nodes receiving it. The assumption here is that if a transaction was propagated to a number of nodes, then it is more likely to be a valid one and not a double-spent coin. The greater the propagation depth reached, the better. Nonetheless, an attacker could possibly create malicious nodes around the vendor and have his double-spent coin received by these nodes and baffle the vendor to believe it is a valid transaction.

The second countermeasure is blocking incoming connection requests, proposed by Bamert et al. [13]. The proposal aims at blocking any incoming connections directed to the vendor and this will not satisfy the first condition/requirement for a successful double-spending attack since the attacker will not be able to connect directly to the IP address of the vendor and send T_V. However, the attacker could target those newly joining users who are eager to get the latest information updates on the network. These users could possibly be hungry for any connection to start building up their wallets and eventually their chain. This countermeasure may not work in this case.

The third countermeasure is implementing a listening period, proposed by Karame et al. [14]. This countermeasure relies on having honest nodes where they check their chains for any occurrence of a similar transaction. These honest nodes help the vendor in validating a transaction. Once a transaction received by the vendor, honest nodes act immediately and examine their stored transactions to see if it had existed before. This countermeasure works well with slow payments since they require an extended period of time to confirm transactions. However, fast payments will not favor this secure mechanism as the time needed to accept a customer's transaction is estimated to be within seconds. In addition, the attacker can still attempt double-spending by delaying the release of T_A to other nodes to avoid being detected by the vendor.

The fourth countermeasure is using observers, proposed by Karame et al. [14]. The concept of this mechanism is to use nodes called observers, randomly distributed on the network, to protect the vendor. The observers forward all transactions coming to them to the vendor. These nodes do not necessarily have to be near or around the vendor's node. They might be placed far away and yet with the use of this mechanism they can immediately forward received transactions to the vendor within seconds. Nonetheless, observers can be easily detected by the attacker by performing a network pattern analysis. Once detected, the attacker could perform a Denial of Service Attack (DoS) and prevents these observers from receiving any connections including the one contains T_A, enabling double-spending.

The fifth countermeasure is alerting peers of double-spending, proposed by Karame et al. [14]. This countermeasure relies on peers in conducting a deep investigation on their blocks and check if there is a conflict resulted from finding a similar transaction. Once found, peers will immediately send a double-spending alert to the entire network. Peers will be notified directly if an attack occurs. However, this countermeasure does not prevent attacks in the first place. They help in decreasing the chance of getting another attack on the network but they do not prevent it from happening in the first time.

The countermeasures mentioned above could be combined to result in a more effective security measure. The listening period along with the use of observers could be combined. The vendor will wait until all transactions received by near and far observers get broadcast to him after they undergo investigation of double-spent coins.

C. Results

In this section we discuss the resources required for profitable double spending attack in two main blockchain networks mainly Syscoin and Bitcoincash as evaluated in [22]. The parameters include computing power, cut time, BitCoin(BTC) value, expected operating expense (OPEX), and expected AS time.

Table 2: Required Resources for Profitable double spending attack in different Blockchain Networks

Network	Cut Time	Required BTC	Expected OPEX	Expected AS time
Syscoin	1153	13.13	1.81	546
Bitcoin	11530	20.64	2.84	5466

The Table 2 shows the results of evaluation of required resources using the computing power 0.35. The result shows that Syscoin requires less operating expense for profitable double spending attacks than Bitcoin.

VII.Conclusion

Bitcoin is tremendously used as a valid currency to purchase goods. It is used for both slow and fast payments where the time taken to deliver goods makes a major difference. Slow payments rely on transaction confirmation in which an average of 10 minutes is needed to confirm a transaction. Fast payments rely on transaction reception where the vendor only needs to validate a transaction through its arrival. Double-spending is easier to perform on fast payments since there is no investigation of whether the transaction was received before on the network or not.

Five countermeasures were explained above and they are proposals that help reduce the chance of getting a double-spending attack. The countermeasures are the following: inspecting propagation depth of transactions, blocking incoming connection requests, implementing a listening period, using observers, and alerting peers of double-spending.

No single mechanism can provide a full and complete security measure. They can however be combined to increase efficiency. The challenge with Bitcoin is meeting standards that were established through its development such as performance, anonymity, and decentralization.

Users are always advised to take security measures from their side such as changing passwords every now and then and not putting all their assets in one digital wallet and most importantly not sharing a wallet on the public network.

References

[1] Androulaki, Elli, Ghassan O Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun. "Evaluating User Privacy in Bitcoin." Paper presented at the International Conference on Financial Cryptography and Data Security, 2013.
[2] Chohan, Usman W. "The Double Spending Problem and Cryptocurrencies." Available at SSRN 3090174 (2017).
[3] Decker, Christian, and Roger Wattenhofer. "Bitcoin Transaction Malleability and Mtgox." Paper presented at the European Symposium on Research in Computer Security, 2014.
[4] Gervais, Arthur, Ghassan O Karame, Vedran Capkun, and Srdjan Capkun. "Is Bitcoin a Decentralized Currency?". IEEE security & privacy 12, no. 3 (2014).
[5] Karame, Ghassan O, Elli Androulaki, and Srdjan Capkun. "Double-Spending Fast Payments in Bitcoin." Paper presented at the Proceedings of the 2012 ACM conference on Computer and communications security, 2012.
[6] Karame, Ghassan O, Elli Androulaki, Marc Roeschlin, Arthur Gervais, and Srdjan Čapkun. "Misbehavior in Bitcoin: A Study of Double-Spending and Accountability." ACM Transactions on Information and System Security (TISSEC) 18, no. 1 (2015): 2.
[7] Nakamoto, Satoshi. "Bitcoin: A Peer-to-Peer Electronic Cash System." (2008).
[8] Ober, Micha, Stefan Katzenbeisser, and Kay Hamacher. "Structure and Anonymity of the Bitcoin Transaction Graph." Future internet 5, no. 2 (2013): 237-50.
[9] Rosenfeld, Meni. "Analysis of Hashrate-Based Double Spending." arXiv preprint arXiv:1402.2009 (2014).
[10] Ruffing, Tim, Pedro Moreno-Sanchez, and Aniket Kate. "Coinshuffle: Practical Decentralized Coin Mixing for Bitcoin." Paper presented at the European Symposium on Research in Computer Security, 2014.
[11] Bitcoin – Wikipedia, Available from https://en.bitcoin.it/wiki/Introduction.
[12] Vukolić, Marko. "The Quest for Scalable Blockchain Fabric: Proof-of-Work Vs. Bft Replication." Paper presented at the International workshop on open problems in network security, 2015.
[13] T. Bamert, C. Decker, L. Elsen, R. Wattenhofer and S. Welten, "Have a snack, pay with Bitcoins," IEEE P2P 2013 Proceedings, Trento, 2013, pp. 1-5.
[14] G. O. Karame, E. Androulaki, M. Roeschlin, A. Gervais, and S. Cap- kun, “Misbehavior in bitcoin: A study of double-spending and ac-countability,” ACM Transactions on Information and System Security
[15] AndrewMarshall. (2018, December 10). Double-spending problem. All about cryptocurrency. Retrieved from https://en.bitcoinwiki.org/wiki/Double-spending.
[16] Harsh AgrawalAn award-winning blogger with a track record of 10 years. (2019, November 6). What is Double Spending & How Does Bitcoin Handle It? Retrieved from https://coinsutra.com/bitcoin-double-spending/.
[17] M. A. AlZain, Utilization of Double Random Phase Encoding for Securing Color Images, International Journal of Computer Applications, 975 (2018), pp. 8887.
[18] M. A. AlZain and J. F. Al-Amri, Application of Data Steganographic Method in Video Sequences Using Histogram Shifting in the Discrete Wavelet Transform, International Journal of Applied Engineering Research, 13 (2018), pp. 6380-6387.
[19] M. A. AlZain, A. S. Li, B. Soh and M. Masud, Byzantine Fault-Tolerant Architecture in Cloud Data Management, International Journal of Knowledge Society Research (IJKSR), 7 (2016), pp. 86-98.
[20] M. A. AlZain, A. S. Li, B. Soh and M. Masud, Managing Multi- Cloud Data Dependability Faults, Knowledge-Intensive Economies and Opportunities for Social, Organizational, and Technological Growth, IGI Global, 2019, pp. 207-221.
[21] M. A. AlZain, E. Pardede, B. Soh and J. A. Thom, Cloud computing security: from single to multi-clouds, 2012 45th Hawaii International Conference on System Sciences, IEEE, 2012, pp. 5490-5499.
[22] J. Jang and H-N. Lee. Profitable double-spending attacks, arXiv:1903.01711, 2019