2018 IEEE Confs on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Blockchain, Computer and Information Technology, Congress on Cybermatics

A Strong Adaptive Strategic Double-spending Attack on Blockchains

Gholamreza Ramezan, Cyril Leung, and Z. Jane Wang

The University of British Columbia, Vancouver, Canada.
Email:{gramezan, cleung, zjanew}@ece.ubc.ca

***Abstract*—In this paper, we first propose an adaptive strategy for double-spending attack on blockchains. The attacker in our strategy observes the length of the honest branch when a submitted transaction becomes available in the blockchain, and then updates the attack strategy accordingly. This provides a stronger strategy than conventional double-spending attack. We then derive closed-form expressions for the probability of a successful attack and the expected reward of attacker miners. Our analysis shows that the probability of a successful attack by convincing the network nodes to follow the counterfeit branch under the proposed attack strategy is 60% higher than what is expected from the conventional attack strategy when the attackers acquire 40% of the total network processing power. To counter this increase in the probability of attack, the network nodes are required to use a bigger number of confirmation blocks for validating any transaction in the blockchain. We computed the expected reward of an attacker for mining a counterfeit branch on a blockchain and observed that the expected reward drops to zero after a few number of block confirmations.**

***Index Terms*—blockchain, double-spending, security, attack**

I. Introduction

In recent years, numerous blockchain-based applications have been proposed to solve problems in different areas. These include Namecoin [1] to decentralize Domain Name System (DNS) service, Filecoin [2] to decentralize file storage, Litecoin [3], Bitcoin [4], and Zcash [5] for decentralized payment systems, Ethereum for developing autonomous systems [6], and the proposed method in [7] for decentralized energy trading. Also, different blockchain development platforms have been introduced, such as Hyperledger Fabric by IBM [8], and Nexledger by Samsung [9].

Different attacks on blockchain technology are analyzed in [4], [10]–[13]. In the original Bitcoin paper [4], a simplified attack model was considered. In this model, the number of blocks in a valid chain in a blockchain is assumed to follow a Poisson distribution with mean $(r * q/p)$, where r is the number of blocks formed by the honest miners, and q and p are the probabilities of generating the next block by the attacker miners and the honest ones, respectively. The average time needed by the attacker miners to make s blocks (the attack is successful if $s > r$) is then rT/p, where T is the average time to create a block. This is not a precise model because only the average number, and not the actual number, of blocks created by an attacker is considered. The author in [14] considered a more accurate model for the attack scenario by assuming a negative binomial distribution on the number, s, of blocks generated by the attacker miners when, in the same time period, the honest miners generate r blocks.

In [10], the authors introduce the *balance attack* in which the attacker disrupts communications between subgroups of similar mining power. This study was done on Greedy Heaviest Observed Subtree (Ghost) protocol in Ethereum [6]. They show that the Ghost protocol is vulnerable to double-spending attacks with high probability. In [11] the authors review a rushing attack on a blockchain protocol, when the network has a fixed number of miners and the network is synchronous which means it has a maximum delay when transmitting the transactions. This paper shows that the blockchain protocol is $T-$consistent with some assumptions on mining hardness and attacker nodes hash power. By $T-$consistent, we mean that when the protocol operates with a maximum delay, the honest miners agree on all the blocks in a blockchain except the last T blocks at the end of the chain. The authors in [12] analyze the consistency of the blockchain in an asynchronous network with a bounded adversarial delay and show that the blockchain has a strong consistency.

Several papers focus on the security of blockchain consensus algorithms [15]–[19]. In a selfish mining attack [15], the attacker drops any blocks discovered by the victim miner that compete with the blocks discovered by the attacker. In other words, the attacker feeds only its own view of the blockchain to the victim miner. This misuses the victim's computing power to mine on the attacker's blockchain. In a 0-confirmation transaction, an attacker pays a transaction to a merchant who releases goods to the customer before seeing the transaction in the blockchain. Then, the attacker blocks the communication lines of the merchant's node and sends another double-spending transaction to the rest of the network. The merchant releases the goods to the attacker, but since the attacker controls all the merchant's connections, the merchant cannot tell the rest of the network about the original transaction.

In this paper, we propose an adaptive strategy for a double-spending attack on blockchains. The results show that caution is needed in calculating the number of confirmation blocks which is required when validating a transaction. The main contributions of this paper are as follows:

- *Developing Adaptive Attack Strategy*: We propose a strategy for a double-spending attack on blockchains that substantially increases the probability of a successful attack.

978-1-5386-7975-3/18/$31.00 ©2018 IEEE
DOI 10.1109/Cybermatics_2018.2018.00216

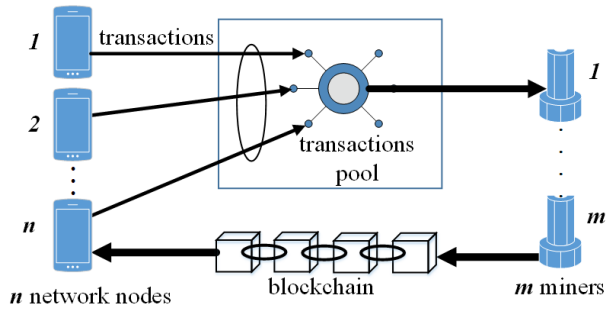

Figure 1: The blockchain traffic flow in a decentralized communication network. m and n are the number of miners and network nodes, respectively.

In the proposed attack strategy, the attacker proactively observes the counterfeit branch, chooses its strategy based on the length of the honest chain at the exact time when a submitted transaction appears on a block in the blockchain.

- *Attack Analysis*: We derive closed-form expressions for the probability of a successful attack. We take into account the number of new blocks added to the valid chain until a submitted transaction appears on a block, and the number of confirmation blocks required for a transaction to be considered valid when calculating the attack success probability. We also derive a closed-form expression for the expected reward of attacker miners in a blockchain in terms of system parameters including the block rewards and creating new blocks probability.
- *Analysis Results*: We study the successful attack probability and the expected reward when an attacker acquires a fraction, f, equal to 0.4, 0.3, and 0.2 of the processing power of the miners network. Our results show that the success attack probability is increased 60% compared to the Bitcoin double-spending attack [4], when $f = 0.4$. To make the expected rewards negligibly small for the attacker in case of receiving the transaction in a block after 5 waiting blocks, the honest nodes should wait for 30 confirmation blocks before validating any transaction.

The remainder of this paper is structured as follows. Section II presents the system model and the motivation for our research. In Section III, we analyze the attack scenario, derive analytical expressions for the probability of a successful attack and the expected reward, and discuss numerical results. Concluding remarks are provided in Section IV.

II. System Model

In Fig. 1, we illustrate the traffic flow in a typical network between the network nodes and the miners. Our system model includes the following:

Network Nodes: We have n network nodes that can communicate with each other to provide or request services. Such nodes submit their transactions via the network to miners.

Miners: Miners receive transactions and process them to form new blocks containing the submitted transactions. After forming the blocks, the miners compete to add the newly formed blocks to the blockchain. The incentive for the miners to provide the mining service is a reward in the form of blockchain tokens.

Transaction Pool: We assume that the submitted transactions go to a transaction pool in a network. Then, they can reach miners for mining process.

Blockchain Tokens: Network nodes and miners agree on the value of a blockchain token. Nodes pay tokens to other nodes for services. Tokens are also used for the mining process. Each miner receives some tokens as a reward whenever it creates a new block for the blockchain. The reward amount is determined by the blockchain rules.

We assume that any node can join miners or network nodes; i.e. the network is *permissionless*. Also, each node is capable of securely generating and storing a public/private key. If it is unable to generate one, it is at least required to securely store a public/private key.

In a double-spending attack, an attacker issues a transaction in favour of another network node that is an honest node. The attacker must first convince the honest network node that the transaction has been confirmed through the blockchain mechanism. Therefore, the attacker waits until the honest network node receives the transaction in a block of the blockchain. Then, the attacker creates a block that contains another transaction that conflicts with the first transaction with the honest node. For example, in the first transaction, the attacker declares that she/he has sent R tokens from her/his account to the honest node's account, but in the second transaction, the attacker declares that the same R tokens, are to be transferred to the account of the attacker's friend. Thus the attacker is trying to spend the same tokens twice. If other network nodes accept the second transaction in the blockchain then, in effect, the attacker has convinced those network nodes that the second transaction is valid, and the first transaction is not. Therefore, the first transaction is valid only for a short period. The honest node may irreversibly provide a service to the attacker in that period.

III. Attack Analysis

We now analyze the probability of attacking a blockchain when an attacker makes a counterfeit branch that contains chained blocks in the blockchain, longer than a valid branch which contains chained blocks made by honest miners. The design of any security scheme has two major components; an attack model and a security goal. The attack model describes what abilities an attacker has, but it does not place any constraints on how the attacker uses these abilities. The security goal demonstrates how the attacker's scheme can be defeated [20]. The attack model and the security goals for our system are presented next.

A. Attack Model

In this study, we consider a more comprehensive attack model by defining a new attack scenario. As in conventional

attack models, the attacker in our model starts making a counterfeit branch immediately after submitting a transaction for an honest node to the miner pool. However, unlike in conventional attack models, the attacker in our model verifies whether or not its counterfeit branch is longer than the valid branch, when the submitted transaction appears on a block in the blockchain. If the counterfeit branch is longer, the attacker continues to generate more blocks until it can successfully attack the system by making a long enough counterfeit chain. If the valid branch is longer, the attacker replaces its counterfeit branch with a copy of the honest chain and adds new blocks to the duplicated valid one until it can successfully attack the system. Such a proactive approach allows the attacker in our model to achieve a higher probability of success in taking control of the blockchain.

The attackers are miners in the miner network, or network nodes that are in collaboration with a few miners to take control of the network by attacking the blockchain. Attacking a blockchain means creating a new chain of blocks in order to make a new branch in the blockchain and invalidate previously added blocks to the blockchain and their transactions.

The attacker has access to the control channel of the network and can acquire a copy of the blockchain to gain knowledge concerning the transactions within a blockchain. Also, an attacker, which does not have any tokens, may initially provide some services to other nodes to acquire sufficient tokens. Then, using the tokens, the attacker is able to submit false transactions.

We assume that the nodes are not compromised; that is, the attacker does not have access to the private keys of the legitimate network nodes or the miners. A legitimate network node can process and properly follow the blockchain protocol. For example, if a miner node receives an invalid transaction such as a transaction for a smart contract with a zero-token bond, it will drop it as an invalid request.

An attacker cannot generate faulty transactions such as adding an invalid digital signature on a transaction. This is due to the nature of the blockchain. After inserting a faulty transaction into a block, that block cannot receive sufficient confirmation from other miners to be considered a new block in a blockchain.

B. Security Goals

By definition, a scheme is said to be (t, ϵ)-secure if every attacker running at the maximum time t succeeds in attacking the security scheme with the highest probability ϵ [20]. This means that in order to prove the level of security of a scheme, we need to show the effort required to attack the scheme and its related probability. With this definition in mind, in the following section we study the probability of an attacker's ability to successfully attack a blockchain scheme.

It is possible that different miners and network nodes follow different branches of equal length in the blockchain that create a fork in the system as illustrated in Fig. 2. A fork has two branches of equal length that results in a tie because both branches can be considered valid. However, as shown in Fig. 2(b), as soon as a new block is mined in one of branches of the fork, all the network nodes will accept the longer branch as the valid one. Therefore, the blocks in the other chain of the fork will be ignored. A transaction taking place within a given block in the longer chain is said to have z confirmations if the block is z blocks away from the last block in the chain. For example, in Fig. 2(b), there is one confirmation for all the transactions in Block 4b of the chain, and two confirmations for all the transactions in Block 3.

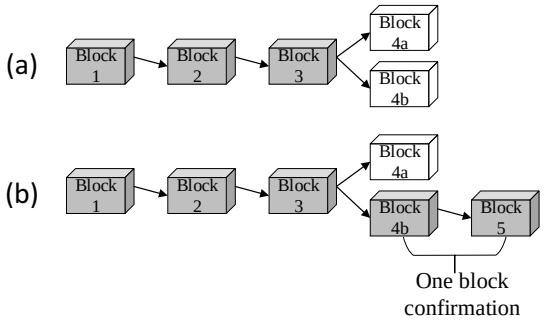

Figure 2: A fork in the blockchain: (a) There is a tie in the blockchain as the nodes attempt to find the longest branch. Blocks 4a and 4b have created two branches in the blockchain. (b) The tie will be resolved as soon as the nodes in one of the branches are able to generate the next block.

The next assumption is about the source of tokens. We assume that tokens are created only by the defined mechanism in the blockchain, such as a Proof-of-Work (PoW) mechanism. Therefore, the source of tokens is based on the effort spent by the miners in the mining process; the miners are either honest or attackers. If they are attackers they cannot create new tokens in the blockchain without any effort.

1) Conventional double-spending attack: Let us now review the incidence of a double-spending attack taking place in a blockchain that we will map on our system model. An attacker generates a transaction to request a service from an honest network node and agrees to pay some tokens in return. For example, Node A transfers R tokens to Node B, if Node B provides a specific network service to Node A. We call this a good transaction. This transaction is then broadcast to the network (Step A in Fig. 3).

After broadcasting the good transaction, the attacker creates a second transaction that conflicts with the good one. For example, Node A transfers R tokens, which refers to the exact tokens in the good transaction, to Node C. Then, without broadcasting the second transaction to the good miners, the attacker miner secretly starts generating new blocks and then, adds these blocks to a different chain of blocks in parallel to the honest branch. We call this a counterfeit branch. One of the blocks in this chain contains the second transaction.

After a certain amount of time, the good transaction will appear in Block k of the valid chain (Step B in Fig. 3); i.e., the valid branch will have successfully created k successive blocks until the submitted transaction appears in one of them. We

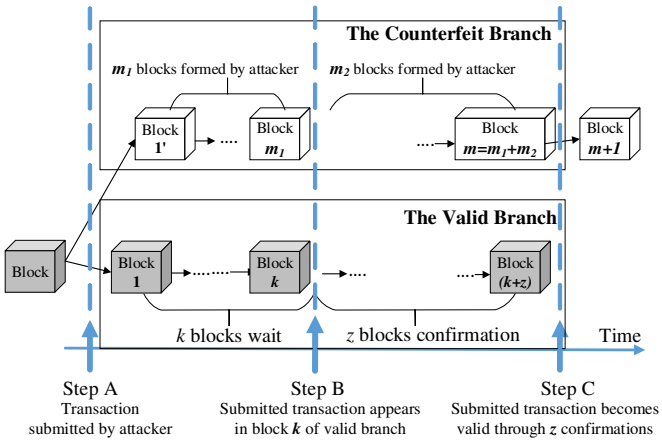

Figure 3: A double-spending attack in a blockchain: (Step A) An attacker creates a good transaction to pay some tokens to an honest node to receive a service. The attacker broadcasts this transaction to the network. Also, the attacker begins in parallel to create m_1 blocks in a hidden counterfeit chain. (Step B) The good transaction appears in the block k of a valid chain. (Step C) The honest network node believes there are z blocks as confirmation for block k and provides service to the attacker. Meanwhile, the attacker has created m_2 more blocks. Now, the attacker can release the counterfeit chain if that chain size, $m = m_1 + m_2$, is longer than the valid chain. In one of the blocks in the counterfeit chain there can be a false transaction that sends the attacker's tokens elsewhere (for example, to itself), instead of to the honest node's account.

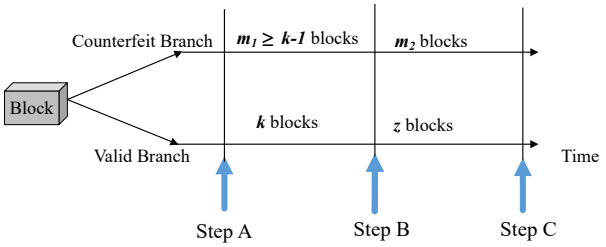

Figure 4: State 1: At Step B of creating blocks for a blockchain, the attacker uses its own chain (m_1 blocks) to continue the attack and create m_2 more blocks if the created counterfeit branch is longer than the current valid branch in the blockchain $(m_1 \geq k-1)$.

assume that the attacker has secretly been creating m_1 blocks until that moment. The attacker will continue to secretly add new blocks to its counterfeit chain until the network node sees z blocks following block k of the valid chain. It is assumed that the attacker has created m_2 more blocks between Steps B and C. Therefore, the length of the counterfeit chain is $m = m_1 + m_2$.

At Step C, upon receiving z confirmation blocks, the network node, which is the receiver of the R tokens mentioned

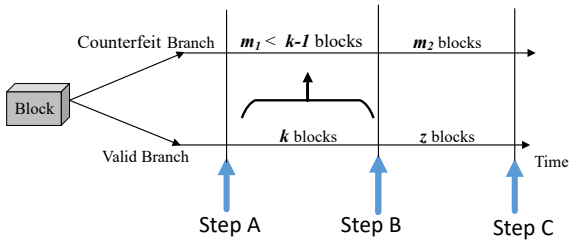

Figure 5: State 2: At Step B of creating blocks for a blockchain, the attacker uses the already created valid chain (the first $k-1$ blocks of the valid chain) to continue the attack and create m_2 more blocks if the created counterfeit branch is smaller than the current valid branch in the blockchain $(m_1 < k-1)$.

in the good transaction, assumes the payment is finalized and provides service to the attacker (Step C in Fig. 3). As soon as the attacker receives service from the honest network node, it reveals the counterfeit chain to the network. This makes a fork in the blockchain with two branches; the valid branch and the counterfeit branch. Now, if the counterfeit chain is longer than the valid one $(m_1+m_2 > z+k)$, the network nodes will follow the counterfeit branch and ignore the valid one. Thus, the good transaction is replaced by the second transaction, i.e., the double-spending attack is successful.

2) Adaptive strategic double-spending attack: As in the conventional attack strategy, the attacker begins to create a counterfeit chain following Step A in Fig. 3. At Step B, the transaction appears in a block in the valid chain. At this moment, if the attacker observes that the length of the counterfeit chain m_1 is longer than the valid one k, it continues with the conventional attack strategy of adding new blocks to the m_1 blocks in the counterfeit chain (State 1 in Fig. 4). Even if the attacker calculates a longer chain during the period between Steps A and B, it cannot reveal this to the network because this prevents the receiver node in the good transaction from providing service at Step C. However, if the valid chain is longer at Step B, i.e., if m_1 is less than k, the attacker realizes that it is already lagging and that it has to incur a higher computational cost if it continues to use the counterfeit chain (State 2 in Fig. 5). Therefore, unlike in the conventional attack strategy, the attacker does not continue with the counterfeit chain. Instead, it proactively makes a copy of the $k-1$ blocks from the valid chain and adds new blocks to this copy, to improve its probability of attack. The valid chain contains the good transaction in Block k. Therefore, the attacker does not copy the Block k and consider only the first $k-1$ blocks of the valid chain. Both scenarios are illustrated in Figs. 4 and 5.

Let the probabilities with which the attacker and honest miners can generate the next block be q and $p = (1-q)$, respectively. The attacker can take control of the blockchain as soon as it creates a counterfeit chain that is longer than the valid one. The probability of successfully attacking a

blockchain $P_V(k,z)$ is then:

$$\begin{aligned} P_V(k,z) &= P(m_1 \geq k-1, m_1+m_2 > z+k) \\ &\quad + P(m_1 < k-1, m_2 > z), \end{aligned} \tag{1}$$

where $P(m_1 < k, m_2 > z)$ is the joint probability that $m_1 < k$ and $m_2 > z$, and $P(m_1 \geq k, m_1+m_2 > z+k)$ is the joint probability that $m_1 \geq k$ and $m_1+m_2 > z+k$. Let $p(g)$ denote the probability that an attacker creates a counterfeit chain which is longer than the valid chain at any time, when it is g blocks behind the valid chain at any time after Step A, be

$$p(g) = \begin{cases} 1 & if \quad q \geq p \\ a_g & if \quad q < p, \quad g = 0,1,2,\ldots \end{cases}$$

That is, if $q \geq p$, the attacker will eventually take control of the chain with probability 1. Otherwise, the attacker will take control of the chain with probability a_g. **Proposition 1** below gives an expression for a_g in terms of q and p.

Proposition 1. *From* [4]*, we model the creation of new blocks for a blockchain as a Markov process (continuous-time, discrete-state-space) as shown in Fig. 6. Let the probabilities with which the attacker and honest miners generate the next block be q and $p = (1-q)$ (with $q < p$) respectively. It is assumed the current counterfeit chain made by the attacker is g blocks shorter than the valid chain that is made by the honest miners. Then the probability, a_g, that the attacker can take control of the blockchain by making a counterfeit branch that is one block longer than the valid branch is given by*

$$a_g = (q/p)^{(g+1)}, \quad g = 0,1,2,\ldots \tag{2}$$

Proof. From Fig. 6, we have

$$a_g = a_{(g+1)}p + a_{(g-1)}q, \quad g = 0,1,2,\ldots \tag{3}$$

where $a_{(g-1)}$ is the conditional probability that the attacker can successfully mine $g-1$ blocks, given that the last block has also been mined by the attacker, and $a_{(g+1)}$ is the probability that an attacker can successfully mine $g+1$ blocks given that the last block was mined by honest miners.

Now, as Eq. (3) shows, we can rewrite the probability of a successful attack, a_g, as the probability that first the honest miner makes one block, p, and then the attacker miner attacks successfully when it is $g+1$ blocks behind, $a_{(g+1)}$, or that first the attacker miner creates one block, q, and then the attacker miner attacks successfully when it is $g-1$ blocks behind, $a_{(g-1)}$.

Eq. (3) takes the form of a recurrence relation that considers the initial conditions $a_{-1} = 1$ and $a_0 = q/p$ proves Eq. 2. □

Let us now examine the probability that the attacker's counterfeit chain is g blocks behind the valid one. If the attacker and honest chains have f and h blocks, respectively, then we know $g = h - f$ blocks. The probability $p(f,h)$, that there are f and h blocks in the counterfeit and valid branches, when the attacker and honest miners generate the next block with probabilities q and p respectively, is given by the below assumption.

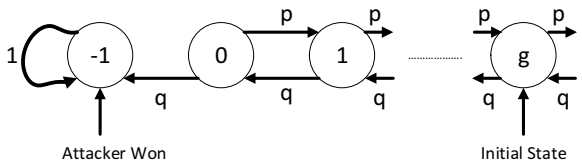

Figure 6: A Markov chain by which attackers can make a counterfeit chain that is one block longer than the valid one, when the counterfeit chain is g blocks behind the valid chain. q and p are the success and failure probabilities of the attacker, respectively, in creating the next block. Each state in the Markov chain refers to the difference between the number of blocks in the valid and counterfeit chains.

Assumption 1. *We assume that the time unit is sufficiently small that no two blocks can be generated by the attackers and honest miners simultaneously. Therefore, the probability that the attacker's counterfeit chain is $g = h - f$ blocks behind the valid chain can then be assumed equivalent to the probability that the attacker successfully generates f blocks and fails to create h blocks (because h blocks are generated by the honest miners), and also, $f < h$.*

We then model the number of blocks f generated by the attacker as a negative binomial random variable, i.e. $f \sim NB(h,q)$ because f is the number of successes before h failures, with a success probability of q. The probability $p(f,h)$ of the attacker having f successes and h failures is then [21]:

$$\begin{aligned} p(f,h) &= \binom{h+f-1}{h} q^f p^h, \\ & f < h, \\ & f = 0,1,2,\ldots, \\ & h = f,\ldots,\infty \end{aligned} \tag{4}$$

Now, using Eqs. 2 and 4, we derive an expression for the probability of attack on the system when the attacker observes that the valid chain contains k blocks between Steps A and B in Figs. 4 and 5.

Proposition 2. *When the attacker observes that the valid chain has k blocks between Steps A and B in Fig. 2, the probability of a successful attack under $z-$confirmation block validation, which is shown in Fig. 2, is given by*

$$\begin{aligned} P_V(k,z) = &\sum_{m_1=k-1}^{\infty} \sum_{m_2=0}^{\infty} p(m_1,k)p(m_2,z)a_{((z+k)-(m_1+m_2))} \\ &+ \sum_{m_1=0}^{k-2} \sum_{m_2=0}^{\infty} p(m_1,k)p(m_2,z)a_{(z-m_2)}, \end{aligned} \tag{5}$$

where $p(m_1,k)$ is the joint probability that the number of blocks between Steps A and B in the counterfeit and valid chains are m_1 and k respectively, $p(m_2,z)$ is the joint probability that the number of blocks between Steps B and C

in the counterfeit and valid chains are m_2 and z respectively, and a_g is the probability that the attack is successful when the counterfeit chain is g blocks behind the valid chain.

Proof. Using (2) and (4), the probability that the attacker is successful in attacking the blockchain, when the valid chain has h blocks and $q < p$, is:

$$P(h) = \sum_{f=0}^{\infty} p(f,h) a_{(h-f)}, \tag{6}$$

Then, using (6), the probability that the attacker is successful in attacking the blockchain, assuming State 1 in Fig. 4, is:

$$P(m_1 \geq k-1, m_1 + m_2 > z + k) = \sum_{m_1=k-1}^{\infty} \sum_{m_2=0}^{\infty} p(m_1,k) p(m_2,z) a_{((z+k)-(m_1+m_2))} \tag{7}$$

Similarly, the probability that the attacker is successful in attacking the blockchain, assuming State 2 in Fig. 5, is:

$$P(m_1 < k-1, m_2 > z) = \sum_{m_1=0}^{k-2} \sum_{m_2=0}^{\infty} p(m_1,k) p(m_2,z) a_{(z-m_2)}. \tag{8}$$

Finally, substituting (8) and (7) in (1) gives us the probability of a successful attack as

$$\begin{aligned} P_V(k,z) = & \sum_{m_1=k-1}^{\infty} \sum_{m_2=0}^{\infty} p(m_1,k) p(m_2,z) a_{((z+k)-(m_1+m_2))} \\ & + \sum_{m_1=0}^{k-2} \sum_{m_2=0}^{\infty} p(m_1,k) p(m_2,z) a_{(z-m_2)}, \\ & k, z = 1, 2, \ldots, \\ & m_1, m_2 = 0, 1, 2, \ldots \end{aligned} \tag{9}$$

□

Fig. 7 plots $P_V(k,z)$ in (9) for different z values when the attacker's success probability in creating the next block is $q = 0.2$, 0.3, and 0.4 respectively. As a baseline attack scheme, we consider the attack model commonly studied in the blockchain literature, where the attacker does not observe (and make use of) the length of the honest chain at Step B of Fig. 2. For a wide range of z values in (9), we observe that the attacker in our study has a higher chance of successfully attacking the blockchain than the baseline attack scheme. The higher chance of successful attack is mainly because our attacker can use the valid chain (the first k blocks) at Step B, if it fails to create a longer chain until Step B (as described in Fig. 5). The successful attack probability decreases when q is decreased because a smaller q implies that the attacker has a lower chance of generating the next block. It can also be observed that the probability of a successful attack decreases rapidly when the number of block confirmations, z, is increased. This shows that, even if the attacker opts for the smart strategy, the blockchain system is resistant to double-spending attacks when a sufficiently large number of confirmation blocks are used to validate each transaction.

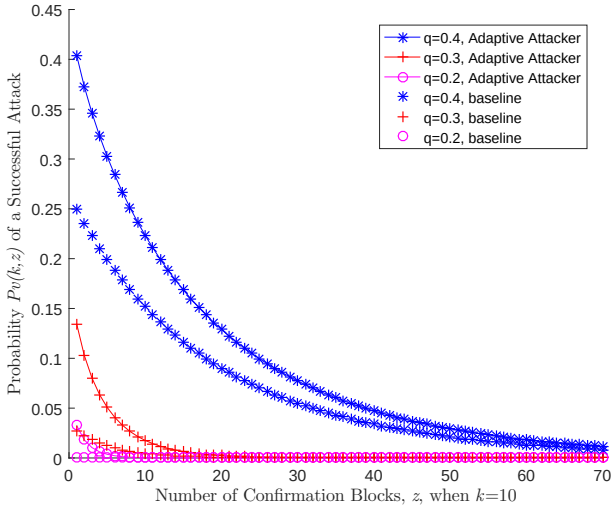

Figure 7: The probability $P_V(k,z)$ of attacking a blockchain by an attacker for a different number of confirmation blocks, z, when $k = 10$ and $q = 0.2, 0.3, 0.4$. "Adaptive Attacker" refers to the attack model in our study, while "baseline" refers to the attack model considered in conventional blockchain literature

Discussion on the chosen q values: We have chosen q values up to 0.4 in our analysis. We now explain why the chosen values for q is high enough.

Conventionally, in blockchain literature, it is assumed that the probability q of the attacker making a new block is always less than the corresponding probability p for the honest miners, i.e., we assume $(q < p)$. In other words, attackers are always assumed to have less than half of the total network processing power when making new blocks in the blockchain. Let us now analyze the amount of investment that an attacker has to make, in order to achieve $q = 0.4$, 0.3, or 0.2. Assume that the blockchain operates with a PoW mechanism which requires that the miners find hash values with d leading zeros, in order to form a valid block. The average number of hash operations required to find a valid hash value is often referred to as the *difficulty* of the blockchain. For example, in Bitcoin, as of September 2017, the *difficulty*, calculated as $2^{(d-32)}$, is $1,103,400,932,964$ hash/second (H/s). This corresponds to $d = 72.0051$ bits; i.e., any valid block hash value in Bitcoin should contain about 72 leading zeros, as of September 2017.

The PoW mechanism is designed to control the time duration T taken by the miner network to generate a new block. Example values of T in existing systems are $T = 600$ (s) for Bitcoin blockchain and $T = 10$ (s) for Ethereum blockchain. Now, the network hash power requirement (in hash/s) can be calculated as $(2^d)/T$. Substituting $d = 72$ bits, we find that the average hash power requirement for miners in the Bitcoin

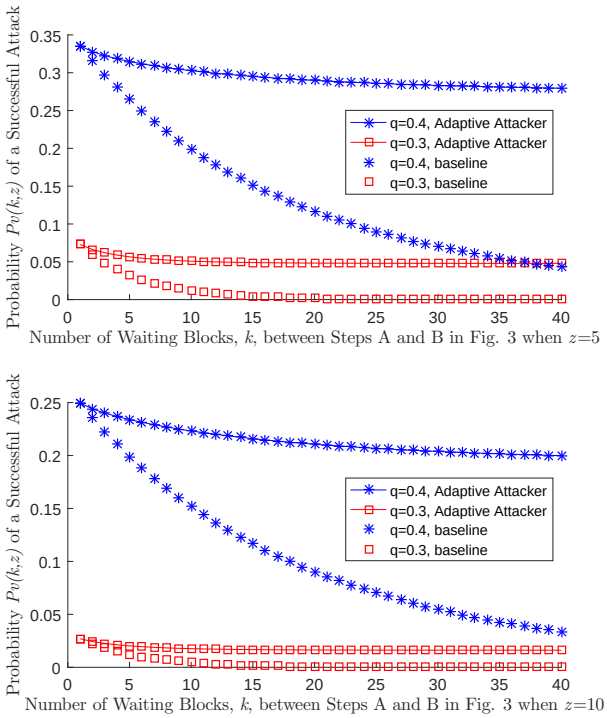

Figure 8: The probability $P_V(k,z)$ of attacking a blockchain for different block waits, k, when $q = 0.4$ and 0.3. "Adaptive Attacker" refers to the attack model in our study, while "baseline" refers to the attack model considered in conventional blockchain literature.

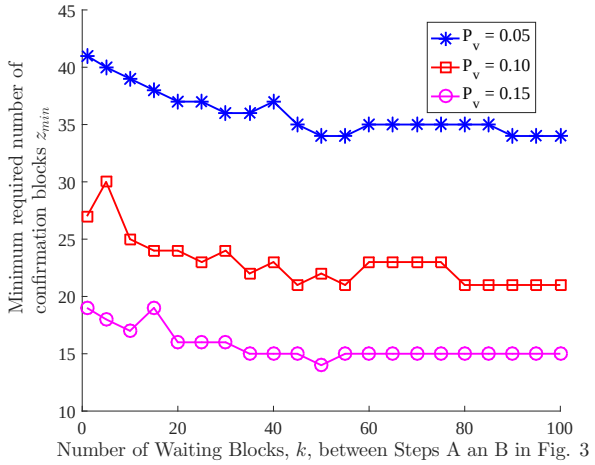

Figure 9: Plots of the lowest number of confirmation blocks z_{min} required to keep the attack probability $P_V(k,z)$ below 0.05, 0.10, and 0.15 respectively, versus the number of blocks k in the honest chain between Step A and Step B in Fig. 4-5.

blockchain is 7.87×10^{18} (H/s) as of September 2017. If we assume that each hardware module used by miners has a maximum computing power of 20 (GH/s), the miners should spend about 500 USD per hardware module, as per the current market value. This shows that the total investment cost from the miners in Bitcoin blockchain is about 196 billion USD. To achieve a q value of 0.4, 0.3, and 0.2, respectively, the attacker should own 40%, 30%, and 20% of the total hardware modules in the miner network respectively. This corresponds to a total investment of 78.7, 59.0, and 39.3 billion USD respectively. Such large investment costs show that the q values chosen for our analysis are high enough for practical purposes.

Fig. 8 illustrates the blockchain successful attack probability for different k values when the attacker's success probability in creating the next block is $q = 0.4$ and 0.3, and the number of confirmation blocks for valid transactions is $z = 5, 10$. For both $z = 5$ and $z = 10$, the successful attack probability of the smart attacker decreases very gradually with k. This shows that, when the attacker is smart, the network cannot significantly decrease the probability of a successful attack by slowing down the transaction receipts. In contrast, when the attacker opts for a conventional attack strategy, the successful attack probability decreases quite rapidly with increasing k. The above difference in behaviour shows that the proactive attack strategy considered in our analysis is more powerful.

In Fig. 9, we plot z_{min} against k, where z_{min} is the smallest number of confirmation blocks required to keep the successful attack probability, $P_V(k,z)$, below 0.05, 0.10, and 0.15 respectively. First, we observe that a smaller number of confirmation blocks is generally required as k is increased, until a threshold level (around $k = 40$ in Fig. 9). Beyond this threshold level, increasing k does not impact z_{min}; that is, slowing down the network to increase k does not necessarily mean that the network can start validating transactions with a smaller number of confirmation blocks. For this analysis, we used the Brent optimization method which is a real value optimization algorithm [22]. Therefore, in Fig. 9, z_{min} may not decrease monotonically with k due to quantization error.

In Figs. 7, we observe that the successful attack probability can be very small when a sufficiently large number of confirmation blocks is used to validate a transaction. Nevertheless, the block reward that miners receive is an incentive that encourages attackers. Next, we study the expected reward for an attacker.

C. Expected reward for attackers

As Fig. 1 shows each node sends transactions out to the miner pool and each miner selects a set of transactions from the pool to create a new block. In our blockchain framework, miners can receive rewards based on their PoW, that is, the miners prove that they have spent a certain amount of time in solving a puzzle by finding a valid hash value. For example, miners calculate the hash value of a certain block using the hash function of the Secure Hash Algorithm 256 (SHA256)

[4]. With this algorithm, there are 2^{256} possible hash values. If we assume that a valid hash value has d leading zeros, the number of hash operations required to find the first hash value is geometrically distributed with an expected value of 2^d. Hence, the expected cost of generating a valid block, C_b, is given by

$$C_b(d) = C_h 2^d, \tag{10}$$

where C_h is the cost rate of each hash operation performed by the miner. Let R_b be the reward that each miner receives if it can form a block for the blockchain. Otherwise, it does not receive any reward and merely needs to pay the block calculation cost $C_b(d)$. The expected net reward per block R_{net} for the attacker is therefore

$$R_{\text{net}}(R_b, C_h, d) = \begin{cases} R_b - C_b(d) & \text{if attack successful} \\ -C_b(d) & \text{otherwise} \end{cases} \tag{11}$$

Now, assuming that there are k and z blocks in the valid chain between Steps A and B and between Steps B and C, respectively and that there are m_1 and m_2 blocks in the counterfeit chain between Steps A and B, and between Steps B and C in Fig., 3 respectively, we can use the net reward per block $R_{\text{net}}(R_b, C_h, d)$ from (11) and the probability $P_V(k, z)$ of a successful attack from (9) to obtain the expected reward $\mathbb{E}_{(m_1,m_2)}(R|z, k, R_b, C_h, d)$ for an attacker in mining a counterfeit chain as

$$\begin{aligned} &\mathbb{E}_{(m_1,m_2)}(R_{\text{net}}|z, k, R_b, C_h, d) = \\ &\sum_{m_1=k-1}^{\infty} \sum_{m_2=0}^{\infty} p(m_1, k)p(m_2, z)a_{((z+k)-(m_1+m_2))} \\ &((R_b - C_b(d))(m_1 + m_2)) \\ &+ \sum_{m_1=0}^{k-2} \sum_{m_2=0}^{\infty} p(m_1, k)p(m_2, z)a_{(z-m_2)} \\ &(R_b m_2 - C_b(d)(m_1 + m_2)) \\ &+ \sum_{m_1=k-1}^{\infty} \sum_{m_2=0}^{\infty} p(m_1, k)p(m_2, z)(1 - a_{((z+k)-(m_1+m_2))}) \\ &(-C_b(d)(m_1 + m_2)) \\ &+ \sum_{m_1=0}^{k-2} \sum_{m_2=0}^{\infty} p(m_1, k)p(m_2, z)(1 - a_{(z-m_2)}) \\ &(-C_b(d)(m_1 + m_2)), \end{aligned} \tag{12}$$

where, on the right-hand side, the first and second summation terms refer to the net reward from mining a counterfeit chain that is at least one block longer than the valid chain, i.e., $m_1 + m_2 > k + z$, when $m_1 \geq k$ and $m_1 < k$ (State 1 and State 2 in Figs. 4 and 5) respectively. The third and fourth summation terms refer to the net reward from mining a counterfeit chain that is not longer than the valid chain, i.e., $m_1 + m_2 \leq k + z$.

Fig. 10 plots the expected reward for an attacker when the number, d, of leading zeros required for a hash value to be valid is $d = 40$, the cost rate, C_h, of each hash operation is $(1/18) * 10^{-9}$ unitCost/Hash (C/H) (corresponding to a hardware with a computing power of 20×10^6 Hash/s (H/s), operating at a cost of \$4 per hour), and the per-block reward R_b is 250 unitCost (C). It can be seen that the attacker's expected reward decreases to zero following 30 blocks even if it has a high success probability $q = 0.4$, of creating the next block.

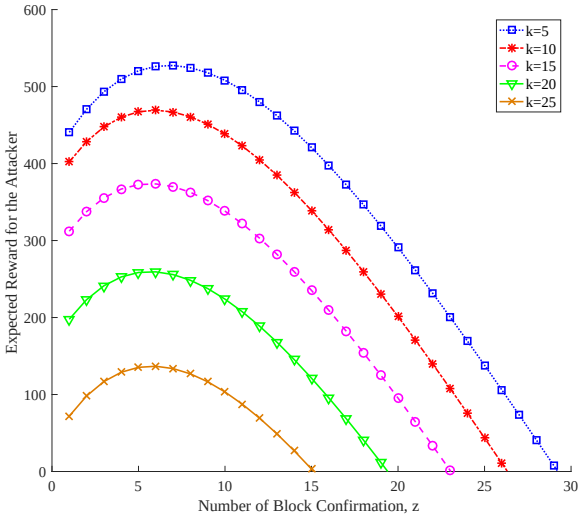

Figure 10: The expected reward $\mathbb{E}_{(m_1,m_2)}(R_{\text{net}}|z, k, R_b, d)$ of the attacker in mining a counterfeit chain, as a function of, z, for different values of waiting blocks, k, with $R_b = 250$ unitCost, $C_h = (1/18) * 10^{-9}$ (C/H), $d = 40$, and $q = 0.4$.

Fig. 10 shows that the expected reward is affected by the number, k, of waiting blocks which is related to the delay in creating new blocks by the miners.

IV. Conclusion

In this paper, we consider a decentralized network using blockchain technology, and propose an adaptive attack strategy that increases the probability of a successful double-spending attack. The attacker injects a false transaction into the blockchain by making a counterfeit chain that is at least one block longer than the valid chain made by honest miners. In our adaptive attack model, the attacker observes the length of the honest chain when the submitted transaction becomes available in a block of the blockchain. If the valid chain is longer than the counterfeit one, the attacker adds new blocks to a copy of the valid chain. Otherwise, the attacker follows the conventional strategy of adding new blocks to the existing counterfeit chain.

Our analysis shows that the probability of a successful attack with the proposed adaptive attack model can be much higher than that with the conventional double-spending attack. For example, for a successful attack probability of 0.01, with

the conventional attack model, the network nodes need to wait the number of confirmation blocks, z, to be at least 52 to validate a transaction. In contrast, in the proposed adaptive attack model we need z to be at least 60. Thus, to mitigate impact of the proposed attack, we need to wait for a larger number of confirmation blocks. Nevertheless, the successful attack probability can be made arbitrarily small if a sufficiently large number of confirmation blocks is used by the honest nodes to validate a transaction.

We also analyzed the expected reward for the attacker when mining a counterfeit chain. Our results show that the expected reward is very small when a large number of block confirmations are used to validate a transaction. For example, in the case of receiving the transaction in 5 waiting blocks, the honest nodes should wait 30 confirmation blocks to make the expected received reward negligibly small for the attacker.

ACKNOWLEDGEMENT

This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada under Grant RGPIN 1731-2013 and by the UBC PMC-Sierra Professorship in Networking and Communications.

REFERENCES

[1] H. A. Kalodner, M. Carlsten, P. Ellenbogen, J. Bonneau, and A. Narayanan, "An Empirical Study of Namecoin and Lessons for Decentralized Namespace Design." in *WEIS*, 2015.

[2] "Filecoin project," http://filecoin.io/filecoin.pdf, accessed: Apr. 3, 2018.

[3] C. Lee, "Litecoin," https://litecoin.org/, 2011, accessed: Feb. 20, 2018.

[4] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," https://bitcoin.org/bitcoin.pdf, 2008, accessed: Feb. 20, 2018.

[5] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous Distributed E-Cash from Bitcoin," in *IEEE Symposium on Security and Privacy*, May 2013, pp. 397–411.

[6] V. Buterin, "Ethereum: a next generation smart contract and decentralized application platform (2013)," *http://ethereum.org/ethereum.html*, 2017.

[7] N. Z. Aitzhan and D. Svetinovic, "Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams," *IEEE Transactions on Dependable and Secure Computing*, 2016.

[8] C. Cachin, "Architecture of the hyperledger blockchain fabric," in *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*, 2016.

[9] "Samsung Nexledger," https://www.samsungsds.com/global/en/solutions/off/nexledger/Nexledger.html, accessed: Feb. 20, 2018.

[10] C. Natoli and V. Gramoli, "The Balance Attack or Why Forkable Blockchains are Ill-Suited for Consortium," in *47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, June 2017, pp. 579–590.

[11] J. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol: Analysis and applications," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2015, pp. 281–310.

[12] R. Pass, L. Seeman, and A. Shelat, "Analysis of the blockchain protocol in asynchronous networks," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2017, pp. 643–673.

[13] T. Neudecker, P. Andelfinger, and H. Hartenstein, "A simulation model for analysis of attacks on the Bitcoin peer-to-peer network," in *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, May 2015, pp. 1327–1332.

[14] M. Rosenfeld, "Analysis of hashrate-based double spending," *arXiv preprint arXiv:1402.2009*, 2014.

[15] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *International conference on financial cryptography and data security*. Springer, 2014, pp. 436–454.

[16] B. Johnson, A. Laszka, J. Grossklags, M. Vasek, and T. Moore, "Game-theoretic analysis of DDoS attacks against Bitcoin mining pools," in *International Conference on Financial Cryptography and Data Security*. Springer, 2014, pp. 72–86.

[17] J. A. Kroll, I. C. Davey, and E. W. Felten, "The economics of bitcoin mining, or bitcoin in the presence of adversaries," in *Proceedings of WEIS*, 2013, p. 11.

[18] A. Laszka, B. Johnson, and J. Grossklags, "When bitcoin mining pools run dry," in *International Conference on Financial Cryptography and Data Security*. Springer, 2015, pp. 63–77.

[19] A. Shomer, "On the phase space of block-hiding strategies." *IACR Cryptology ePrint Archive*, p. 139, 2014.

[20] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. CRC press, 2014.

[21] S. M. Ross, *Introduction to Probability Models*. Academic press, 2014.

[22] R. P. Brent, *Algorithms for minimization without derivatives*. Courier Corporation, 2013.