Bitcoin Double-Spending Profitability Analysis

Ehab Zaghloul, Tongtong Li, Jian Ren

***Abstract*—Blockchain is a technology invented to enable the decentralized digital currency, Bitcoin, for secure and private asset transfer and storage. As a cryptocurrency, Bitcoin should be difficult to double-spend. This paper analyzes the profitability of double-spending Bitcoin over a blockchain. We first introduce the major attacks that can be performed to double-spend Bitcoin. Next, we derive the profitability for attackers to perform such attacks. We provide a quantitative characterization between the risk of double-spending and the number of blocks to be added to the blockchain before a transaction is accepted. Our findings are useful to both Bitcoin users and miners. Miners can obtain more insight into the mining process and potential methods to maximize their profits.**

***Index Terms*—Blockchain, Bitcoin, cryptocurrency, double-spending.**

I. Introduction

Bitcoin BTC [1] is a decentralized online cryptocurrency designed to eliminate the need for a trust third party to facilitate online payments between two parties. It allows users to securely process their transactions faster and avoid the modern-day financial institution costly service fees. The system is designed to run over an online Peer-to-Peer (P2P) network that stores and maintains user transactions in a public ledger, blockchain [2]–[5].

As a technology designed to enable Bitcoin, blockchain allows the entire network to store Bitcoin transactions in a distributed manner while maintaining its persistence, liveness, and consistency. It also validates new transactions as they come in. Users generate transactions that utilize cryptographic protocols and release them into the P2P network to trigger BTC transfers from one user account to another. As the transactions propagate through the network, miners compete to validate these transactions by solving a hard cryptopuzzle, referred to as the *Proof-of-Work* (PoW). The first miner that solves the PoW earns a specified amount of newly released BTC and all the transaction fees associated with the transactions included in their block. At that point, the competing miners accept the solution and append the winning block to the blockchain to finalize the payments included.

Unfortunately, validating and storing Bitcoin transactions over a blockchain could be susceptible to double-spending attacks [6]. In this attack, the attacker tries to use the same inputs for two Bitcoin transactions, one paying the receiving user and another paying the same inputs back to herself. Once the miners validate and store the transaction paying the receiving user (the transaction is accepted by the receiver), the attacker attempts to reverse the transaction by competing in the mining process and forking the blockchain. However, this attack requires significant computational resources to reverse all transactions stored into blocks over the blockchain.

In this paper, we will analyze the profitability of double-spending attacks based on the computational resources of the attacker versus the rest of the honest network. This analysis shows a break-point in time when the attacker will not be profitable on the attack beyond this point (i.e. the time when the cost is greater than the revenue). This presents a trade-off between the waiting time before accepting a transaction versus the profits/losses of the attacker. It also proves that an attacker that controls 51% or more computational power will continue to profit indefinitely. From the perspective of a miner, the analysis also reveals how profitable mining Bitcoin could be.

The rest of the paper is organized as follows. In Section II, we present the common approaches of performing double-spending attacks. Next, in Section III, we present our profitability analysis. Finally, we conclude our findings in Section IV.

II. Double-Spending Attacks

Double-spending attacks come in various forms.

1) Race Attack: In a race attack, the receiving user of a transaction accepts an unverified Bitcoin transaction that has not been mined or stored in the blockchain. The attacker, acting as the transaction sender, generates two transactions; a transaction that pays the receiving user and a fraudulent transaction that uses the same inputs to pay herself, hence, conflicts with the other transaction. Both transactions are then released into the Bitcoin P2P network simultaneously by the attacker. Once received by the miners, they begin to validate and append both transactions. Only one transaction will be validated and accepted while the others will be rejected by the network since its inputs have already been used. The attacker hopes that the fraudulent transaction becomes validated first as the receiving user has already accepted the payment and it is not possible to reverse this action. To prevent a race attack, the receiving user must first wait for the transaction to be permanently stored before accepting it to ensure that any other fraudulent transactions do not exist within the network.

2) Finney Attack: A Finney attack [7] is similar to the race attack where the receiving user accepts a transaction that has not been stored in the blockchain. The attacker also generates two transactions as those in a race attack, however, she does not release them into the network. Next, the attacker secretly begins to mine a block that contains the fraudulent transaction. The attacker strives to successfully win the PoW competition while generating the block and then immediately releases both transactions into the network. Given that the attacker has already been able to mine a block containing the fraudulent

The authors are with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824-1226, Email: {ebz, tongli, renjian}@msu.edu

transaction, this block will be accepted by the P2P network and appended to the block while the transaction paying the receiving user will be rejected. A Finney attack relies solely on the computational resources of the attacker.

3) Vector76 Attack: In a Vector76 attack, the receiving user waits for a single block to be mined and appended to the blockchain before accepting the transaction. In this case, the attacker must possess significant computational power to create a fork in the blockchain. Similar to the Finney attack, the attacker generates a transaction paying the receiving user and does not release it into the network. Following that, the attacker attempts to mine the transaction into a block. Once successful, the attacker only releases the block into the P2P if the honest miners are able to mine another block. This causes the blockchain to fork since both blocks are accepted. Before the fork is resolved, the attacker aims to generate a fraudulent transaction that uses the same inputs as the previous transaction and releases it to the honest miners unaware of the forked blockchain that carries the transaction paying the receiving user. Since those miners do not recognize the fraudulent transaction as a double-spending attempt, they validate it and append it to their blockchain. Finally, in order for the attack to be complete, this fraudulent fork must grow longer in comparison with the other branch and become the dominant one.

4) 51% Attack: A 51% attack (also referred to as the majority attack) is the main concern to the Bitcoin system. In this attack, the attacker possesses more than half the computational resources of the P2P network. In most cases, the attacker is in the form of a pool of miners where computational power is accumulated. This attack allows the attacker to always win the PoW competition given the advantage in computational resources, thus can reverse any block of transactions.

For explanation purposes, consider an attacker that generates two transactions as previously discussed and releases both of them into the P2P network. The receiving user usually waits for six blocks to be appended to the blockchain as confirmation before accepting the payment. Given that the attacker possesses 51% of the total computational power of the network, she can always win the PoW competition and mine blocks in a shorter time. Therefore, the attacker generates a mined block containing the fraudulent transaction in parallel with other miners. Once the receiving user accepts the payment, the attacker releases the secretly mined blocks to creating a fork in the blockchain. At that point, the transaction paying the receiving user is dropped and no longer considered valid. It is worth mentioning that even an attacker with slightly less than 50% of the total computational power stands a good chance to severely control the network since her chances of winning the PoW competition may be higher than all honest miners combined.

The probability of success P_s of a double-spending attack, as presented in [8], shows that as the number of blocks appended to the blockchain increases, an attacker with computational power larger than that of the entire honest network combined can always succeed, i.e. $P_s = 1$. However, with computational power that is less than 50% of the entire network, P_s declines exponentially.

III. Double-Spending Attack Profitability

A double-spending attack is only profitable when the attack returns are greater than the cost of performing the attack. We consider an attacker that tries to double-spend v BTC. The attacker generates a transaction that pays the receiving user v BTC and releases it into the P2P network. Next, the attacker immediately begins to secretly mine blocks of transactions that include a fraudulent transaction that pays the same v BTC back to herself. The receiving user only accepts the transaction after observing that n blocks have been appended to the blockchain. The attacker is successful in performing the attack and can fork the blockchain if she can secretly mine $m = n + 1$ blocks and replace the n blocks generated by the honest miners. The return of the attacker includes the v BTC, a product/service obtained from the receiving user equivalent to the v BTC payment, the mining reward for each block successfully mined, and the transaction fees included in each transaction. Therefore, the revenue gained by the attacker can be formulated based on her corresponding P_s as follows

$$\text{Revenue} \approx v + P_s(v + Rm) \text{ BTC}, \tag{1}$$

where R is the block reward and the transaction fee per block.

The costs of a double-spending attack are determined using multiple factors such as the price and depreciation value of machinery used, the cost of electricity, and the amount of BTC being spent in the transaction. Accounting for all the possible factors is infeasible. Therefore, we simplify our analysis to account only for the factors that significantly change while the attack is being performed. Our analysis includes the v BTC the attacker spends, the cost of mining m blocks, and the depreciation cost $d(t)$ of the computing device used in BTC at time t. We derive the cost as follows

$$\text{Cost} \approx v + me_q(t) + d(t) \text{ BTC} \tag{2}$$

where $e_q(t)$ is the estimated mining electrical cost in BTC per block of a miner with a share q of the total computational power of the system. We assume $e_q(t)$ remains constant during the total time T the attack is performed. We also assume that the average lifespan of the mining equipment is approximately two years. Using straight-line depreciation, $d(t)$ is a negligible value for an attack over a short period of time. Therefore, we can reduce the cost equation as follows

$$\text{Cost} \approx v + me_q(t) \text{ BTC}. \tag{3}$$

In the Bitcoin network, the approximate time to mine a single block is ten minutes. The probability of a successful double-spending attack, revenue, and cost can be derived as

$$P_s \approx 1 - \sum_{m=0}^{\frac{T}{10}-1} \binom{m + \frac{T}{10} - 1}{m} \times (p^{\frac{T}{10}} q^m - p^m q^{\frac{T}{10}}), \tag{4}$$

$$\text{Revenue} \approx v + P_s \left(v + \frac{RT}{10} \right) \text{ BTC}, \tag{5}$$

$$\text{Cost} \approx v + \left(\frac{T}{10} \right) e_q(t) \text{ BTC}. \tag{6}$$

Therefore, the profit/loss can be formulated as

$$
\begin{aligned}
\mathsf{Profit/Loss} &= \mathsf{Revenue} - \mathsf{Cost} \\
&\approx P_s\left(v + \frac{RT}{10}\right) - \left(\frac{T}{10}\right) e_q(t) \text{ BTC.} \qquad (7)
\end{aligned}
$$

Today, to stand a chance in mining Bitcoin, miners accumulate their computational power into mining pools where each individual miner uses an Application-Specific Integrated Circuits (ASIC), Field-Programmable Gate Array (FPGA), Graphics Processing Unit (GPU), or Central Processing Unit (CPU) as their mining machine. Each computing machine consumes electricity differently based on its specifications. Therefore, formulating the cost of electricity spent by a miner in the mining process becomes challenging. It is well known that ASICs are the dominant machines nowadays given their powerful computational power.

Our goal now is to formulate the estimated electrical cost $e_q(t)$ of a mining pool. We begin by estimating the total number of miners $N(t)$ based on the total hashrate $H(t)$ of the system at a certain time t as

$$N(t) \approx \frac{H(t)}{h(t)}, \qquad (8)$$

where $h(t)$ is the average hashrate of a single mining machine involved in mining at time t.

The cost of electricity is measured in cents/kWh and varies based on the end-use sector and time t. We denote the average cost of electricity of all sectors at time t as $e_a(t)$. The average running cost $c(t)$ of a machine at time t is

$$c(t) \approx e_a(t) \times w \text{ cents/hour}, \qquad (9)$$

where w is the computing wattage of the machine.

Using equations (8) and (9), the total cost $E(t)$ for all miners at time t is

$$E(t) \approx N(t) \times c(t) \text{ cents/hour}. \qquad (10)$$

Given the approximated 10 minutes to generate one block, it takes $T = 1$ hour for miners to generate $m = 6$ blocks. As a result, the total cost $C(t)$ for all miners to generate one block at $T = 10$ minutes can be estimated as

$$C(t) \approx \frac{E(t)}{6} \text{ cents/10 minutes (1 block).} \qquad (11)$$

We estimate the average electricity cost $e_q(t)$ of a mining pool based on its computational power q as

$$
\begin{aligned}
e_q(t) &\approx C(t) \times q \\
&\approx \frac{H(t) \times e_a(t) \times w \times q}{6h(t)} \text{ cents/block.} \qquad (12)
\end{aligned}
$$

For our simulations, we assume that the total cost of mining blocks $C(t)$ by all miners and computational power of the mining pool are fixed during the total mining time T. We also assume a mining environment consists of miners using only ASICs such as Antminer S9 with specifications of $h = 14$ TH/s and $w = 1.375$ kWh. We then consider an attacker that attempts to perform a double-spending attack during October 2019. During that period, 1 BTC was equal to approximately \$9,200. The total hashrate was approximately $H = 95000000$ TH/s and the average cost of electricity for all sectors in the U.S. was approximately 10.45¢/kWh, based on the data collected by the U.S Energy Information Administration [9]. Under these circumstances, in Fig 1, we present the expected profits/losses of double-spending attacks for various computational powers q. For this analysis, we consider an attacker trying to double-spend $v = 5$ BTC.

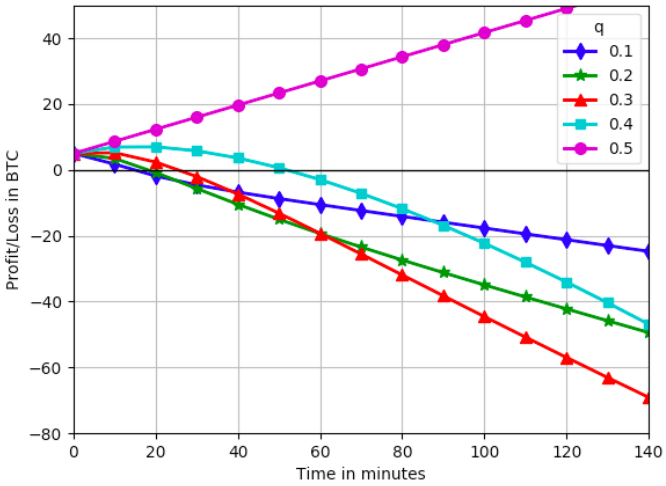

Fig. 1. Profit/loss of attackers with varying computational power q trying to double-spend $v = 5$ BTC.

As depicted by Fig. 1, any point above $y = 0$ represents a profit while below $y = 0$ represents a loss. The point of intersection of a curve with $y = 0$ represents the break-even point. By further analyzing the figure, we derive the following conclusions:

1) For all values q at $t = 0$, the attacker turns a profit of exactly 5 BTC. This may occur when the receiving user accepts an unconfirmed transaction where the attacker has a theoretically perfect chance to succeed.
2) If the receiving user waits for n confirmations before accepting a transaction, the attacker must compete to mine blocks for the blockchain. Based on the previous analysis, we know that the probability of success P_s of mining blocks is based on the computational power q of the attacker. We also know that P_s declines as n (or t) increases for all values $q < 0.5$. Therefore, not only does a profit turn into a loss as the attack time progresses, but an attacker with a smaller q will more likely lose at an earlier point in time. However, with smaller values q, losses are also smaller. This is evident as larger values q impose higher electricity costs $e_q(t)$.
3) Looking closer at the attack with $q = 0.1$, we notice that the attacker breaks-even after approximately 15 minutes, i.e. after mining at most one block. Beyond that time, if the attacker has not been able to fork the block, she will begin losing if she continues to perform the attack. This means, the attacker would most likely surrender at that time to avoid losses.
4) For $q = 0.2$ to $q = 0.4$, the potential profits first grow as t increases and rewards are accumulated until they reach a turning point where they begin to decline and eventually turn into losses. This turning point occurs when P_s starts to decline with t.

5) For $q \geq 0.5$, the attacker will always turn a profit representing the majority attack. This is evident as represented by the straight line with a continuous positive slope.

By analyzing Eq. 7, we note that the attack profitability is mainly based on variables such as P_s and $e_q(t)$ that differ from one attacker to the other. While changing the value v of an attack would shift profitability, it would not affect the overall trend shown in Fig. 1.

It is also worth mentioning that this analysis does not incorporate the *luck* factor. To better comprehend this, we consider two miners with computational powers q_1 and q_2 respectively, where $q_1 > q_2$. Given that the miner with q_1 has more computational power to compete when solving the PoW, she can perform the attack faster. However, the miner with computational power q_2 may still get lucky and win the competition since PoW is based on an exhaustive method to solve the crypto-puzzle. However, from a probabilistic standpoint, the chances are low.

IV. Conclusion

In this paper, we analyzed the double-spending attacks of Bitcoin. This analysis aims to educate both Bitcoin users and miners. We presented our double-spending attack profitability analysis showing the potential profits/losses of performing the attack. Our results show that an attacker with a computational power $q < 0.5$ will eventually lose at some point in time as t increases whereas one with computational power $q \geq 0.5$ will always succeed with a profit.

References

[1] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system.” https://bitcoin.org/bitcoin.pdf, 2008.

[2] S. Haber and W. S. Stornetta, “How to time-stamp a digital document,” in *Conference on the Theory and Application of Cryptography*, pp. 437–455, Springer, 1990.

[3] D. Bayer, S. Haber, and W. S. Stornetta, “Improving the efficiency and reliability of digital time-stamping,” in *Sequences II*, pp. 329–334, Springer, 1993.

[4] S. Haber and W. S. Stornetta, “Secure names for bit-strings,” in *Proceedings of the 4th ACM Conference on Computer and Communications Security*, pp. 28–35, ACM, 1997.

[5] H. Massias, X. S. Avila, and J.-J. Quisquater, “Design of a secure timestamping service with minimal trust requirement,” in *the 20th Symposium on Information Theory in the Benelux*, Citeseer, 1999.

[6] G. Karame, E. Androulaki, and S. Capkun, “Two bitcoins at the price of one? double-spending attacks on fast payments in bitcoin.,” *IACR Cryptology ePrint Archive*, vol. 2012, no. 248, 2012.

[7] H. Finney, “Best practice for fast transaction acceptance - how high is the risk?’.” https://bitcointalk.org/index.php?topic=3441.msg48384#msg48384, Feb. 2011.

[8] E. Zaghloul, T. Li, M. Mutka, and J. Ren, “Bitcoin and blockchain: Security and privacy,” *arXiv preprint arXiv:1904.11435*, 2019.

[9] “U.S. energy information administration.” https://www.eia.gov/electricity/monthly/epm_table_grapher.php?t=epmt_5_6_a. [Online; accessed 9-October-2017].

Ehab Zaghloul received his B.S. and M.Sc degrees in Computer Engineering from Arab Academy for Science and Technology (AAST), Alexandria, Egypt in 2012 and 2015 respectively.

He is currently pursuing his Ph.D. degree in Electrical and Computer Engineering at Michigan State University (MSU), East Lansing, USA. His research interests include applied cryptography, secure and private cloud data sharing, electronic voting, cryptocurrencies, and blockchain.

Tongtong Li (SM'09) received her Ph.D. degree in Electrical Engineering in 2000 from Auburn University. From 2000 to 2002, she was with Bell Labs, and had been working on the design and implementation of 3G and 4G systems. Since 2002, she has been with Michigan State University, where she is now an Associate Professor. Prof. Li's research interests fall into the areas of wireless and wired communications, wireless security, information theory and statistical signal processing, with applications in neuroscience. She is a recipient of a National Science Foundation (NSF) CAREER Award (2008) for her research on efficient and reliable wireless communications. Prof. Li served as an Associate Editor for IEEE Signal Processing Letters from 2007-2009, and an Editorial Board Member for EURASIP Journal Wireless Communications and Networking from 2004-2011. She served as an Associate Editor for IEEE Transactions on Signal Processing from 2012-2016.

Jian Ren (SM'09) received the BS and MS degrees both in mathematics from Shaanxi Normal University, and the Ph.D. degree in EE from Xidian University, China. He is an Associate Professor in the Department of ECE at Michigan State University. His current research interests include network security, cloud computing security, privacy-preserving communications, distributed network storage, and Internet of Things. He is a recipient of the US National Science Foundation Faculty Early Career Development (CAREER) award in 2009. Dr. Ren served as the TPC Chair of IEEE ICNC'17, General Chair of ICNC'18 and Executive Chair of ICNC'19 and ICNC'20. Currently Dr. Ren serves as an Associate Editor for IEEE Transactions on Mobile Computing, ACM Transactions on Sensor Networks (TOSN) and a Senior Associate Editor for IET Communications. Dr. Ren is a senior member of the IEEE.