POSTER: Deterring DDoS Attacks on Blockchain-based Cryptocurrencies through Mempool Optimization

Muhammad Saad
University of Central Florida
saad.ucf@knights.ucf.edu

My T. Thai
University of Florida
mythai@cise.ufl.edu

Aziz Mohaisen
University of Central Florida
mohaisen@ucf.edu

ABSTRACT

In this paper, we highlight a new form of distributed denial of service (DDoS) attack that impacts the memory pools of cryptocurrency systems causing massive transaction backlog and higher mining fees. Towards that, we study such an attack on Bitcoin mempools and explore its effects on the mempool size and transaction fees paid by the legitimate users. We also propose countermeasures to contain such an attack. Our countermeasures include fee-based and age-based designs, which optimize the mempool size and help to counter the effects of DDoS attacks. We evaluate our designs using simulations in diverse attack conditions.

KEYWORDS

Blockchains, DDoS, Cryptocurrency, Mempool

ACM Reference Format:
Muhammad Saad, My T. Thai, and Aziz Mohaisen. 2018. POSTER: Deterring DDoS Attacks on Blockchain-based Cryptocurrencies through Mempool Optimization. In *Proceedings of ACM Asia Conference on Computer and Communications Security (ASIACCS'18).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/XXXXXX.XXXXXX

1 INTRODUCTION

Cryptocurrencies have witnessed enormous growth in the last two years and the number of attacks on them have also increased. Some of the well-known attacks on Blockchain-based cryptocurrencies include the 51% attack, selfish mining, double-spending, Blockchain forks and distributed denial-of-service (DDoS) attacks [6, 9]. In Bitcoin, block size is limited to 1MB and the average block mining time is 10 minutes. The size of individual transaction varies from 200 Bytes to 1K Bytes. Under these constraints, Bitcoin can verify only 3-7 transactions per second [5]. Low transaction throughput makes Bitcoin vulnerable to flood attacks [1], where malicious users flood the network with low-valued dust transactions. A digital coin can be divided into smaller factions of coin (ie., 1 bitcoin can be divided into 10^8 satoshis). Leveraging that, a user can generate a series of dust transactions from a single bitcoin and overwhelm the network and its resources.

In Bitcoin, the memory pool (mempool) acts as a repository of all the unconfirmed transactions. Once a user generates a transaction, it is broadcast to the entire network and stored into the mempool where it waits for confirmation. If the rate of incoming transactions is higher than the throughput of the network (3-7 transactions/sec), a transaction backlog starts. Transactions that remain unconfirmed for a long time eventually get rejected. On November 11, 2017, the mempool size exceeded 115k unconfirmed transactions, resulting in USD 700 million worth of transaction stall [7]. Mempool flooding creates uncertainty among users so they pay higher mining fees to prevent their transactions from being rejected.

In Bitcoin, a user generates a transaction from a spendable balance in his wallet. Spendable balance consists of "Unspent Transaction Outputs" (UTXO's) that the user previously received from other transactions. The relay fee in Bitcoin is the minimum fee paid for a transaction to be relayed among the peers and the mining fee is the fee paid to a miner as an incentive to include the transaction into a block. Confirmation of a transaction means that a transaction has been successfully mined into a block. A confirmation score, also known as the age of a transaction, is the difference between the block number in which it was mined and the most current block computed by the network. If a transaction is mined in any block, its age grows as the Blockchain size grows. A confirmation score of 0 means that the transaction has been broadcast to the network but not yet mined. High confirmation scores protect the transaction from reversibility and double-spending.

2 DDOS ATTACK ON BITCOIN

Threat Model. In this work, we assume an attacker who owns a full node in the Bitcoin network with a complete Blockchain and a memory pool at his machine. The attacker's wallet has spendable bitcoins that have been previously mined into the Blockchain. We assume that the balance in the attacker's wallet is large enough that it can be split into a large number of small transactions, where each of them is capable of paying the mining fee. The attacker also controls a group of Sybil accounts, each with multiple public addresses. The attacker and the Sybil accounts have an apriori knowledge of each others' public addresses. Also, the attacker and Sybils have client side software and scripts [2], which enable them to initiate a flood of "raw transactions" [10] in a short time.

Attack Objective. When launching a mempool flood, the objective of the attacker is to maximize the size of the mempool and minimize the cost of the attack. The cost of the attack is the fee paid to miners—including relaying and the mining fees—if the attacker's transactions are mined. A higher fee increases the transaction's priority, which determine the transaction's mining chances. To avoid such fees, the attacker's goal is to produce transactions that are less likely to be prioritized. At the same time, the attacker wants his transactions to stay in the mempools for as long as possible.

Attack Procedure. To achieve this objective, the attacker estimates the minimum relay fee of the network, divides his spendable bitcoins ("UTXO's") into various transactions, and sends those transactions to a group of Sybil accounts. All transactions to the Sybil accounts will have input "UTXO's", which are previously mined in

ASIACCS'18, June 4–8, 2018, Incheon, Republic of Korea

ACM ISBN 978-1-4503-5576-6/18/06…$15.00
https://doi.org/10.1145/XXXXXX.XXXXXX

the Blockchain. As such, these transactions will have greater-than-zero age, and will be capable of paying the minimum mining fee. Then, all Sybils generate "raw transactions" [10] of minimal value and exchange them with one another. The rate of transactions will be much higher than the throughput of the network, causing transaction backlog, whereby the size of the mempools will increase. The transactions made among the Sybil accounts will have unconfirmed parent transactions, causing their age score to be zero.

Attack Validation. In Figure 1, we show the effect of the attack by plotting the size of the mempool, which is determined by the number of transactions, against the average transaction fee paid to the miners. The data used in generating this figure was obtained by crawling the mempool size and the average fee paid to the miners from May 2016 to November 2017. We use the min-max normalization, defined as $z = \frac{x_i - \min(x)}{\max(x) - \min(x)}$, to scale the data in the range $[0, 1]$, and plot the normalized values of the mempool size and mining fee. It can be observed in the figure that there is a high correlation between the mempool size and the transaction fee. In May, August, and November 2017, it was reported [4, 7] that Bitcoin mempool was under spam attacks with unconfirmed dust transactions. From Figure 1, it can be seen that during the attack, the size of the mempool was much larger than the average mempool size. As a result, the mining fee pattern also followed similar trend as the mempool size, with a high Pearson correlation coefficient of $\rho = 0.69$–$\rho(X, Y)$ is calculated as $\frac{\text{Cov}(X,Y)}{\sqrt{\text{Var}(X)\text{Var}(Y)}}$.

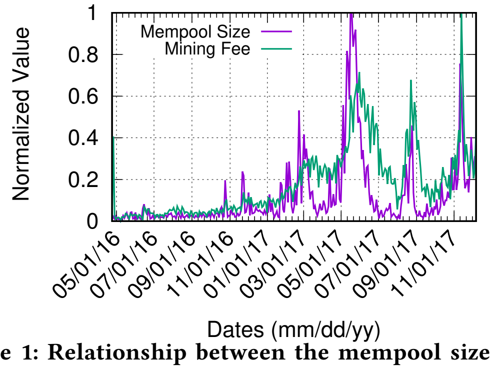


Figure 1: Relationship between the mempool size and the fee paid to the miners. Notice a high correlation between the two curves indicating the possibility of DDoS attack.

In November 2017, when the mempool was flooded, $700 million USD worth of bitcoins remained stuck in the pool for two days [7]. Delay in transaction verification can create multiple problems, including possibilities of equivocation and double-spending [8]. As a result, we conclude that overwhelming the mempool size can lead to other problems in the Blockchain.

Attack Implications and Constraints. A legitimate user who wants his transactions mined will pay relay fees and a high mining fee as a result of the attack. On the other hand, an attacker who aims to get his transactions only into the mempool will only have to pay the relay fee. In such settings, the maximum loss an attacker can incur would happen if all his transactions get mined. As such, the cost will be equal to the product of the total number of transactions by the relay fee. However, given that the attacker has a fixed budget, per our threat model, the attacker can relaunch the attack as long as the total cost of the attack remains below the budget.

```
Inputs : incoming transactions, minimum relay fee;
         minimum mining fee, Threshold Size;
Output: Mempool Size
State: Mempool Size Exceeds Threshold Size
while Mempool Size > Threshold Size do
    while transaction relay fee > minimum relay fee do
        if (transaction mining fee > minimum mining fee)
         then
            Mempool ← transaction; UPDATE(mempool); if
             (transaction mining fee < minimum mining fee)
             then
                transaction rejected ; /* transaction only
                 pays relay fee */
            else
return Mempool Size;
Result: Spam Transactions Rejected
```

Figure 2: Fee-based Mempool Design

```
Inputs : incoming transactions, minimum relay fee;
         minimum mining fee, minimum age limit;
         age of each input of transaction, Threshold Size;
Output: Mempool Size
State: Mempool Size Exceeds Threshold Size
foreach transaction ∈ incoming transactions do
    initialize;
    average age = 0;
    N ← number of parent transactions of current transaction;
    while (transaction relay fee > minimum relay fee) do
        while (transaction mining fee > minimum mining fee)
         do
            average age = $\frac{(\sum_{i=1}^{N} parent_i)}{N}$ if ( average age >
             minimum age limit) then
                ;
            Mempool ← transaction;
            UPDATE(mempool) if (average age < minimum
             age limit) then
                transaction rejected; ;        /* Reject
                 transaction age factor is low */
            else
    return Mempool Size
Result: Spam Transactions Rejected
```

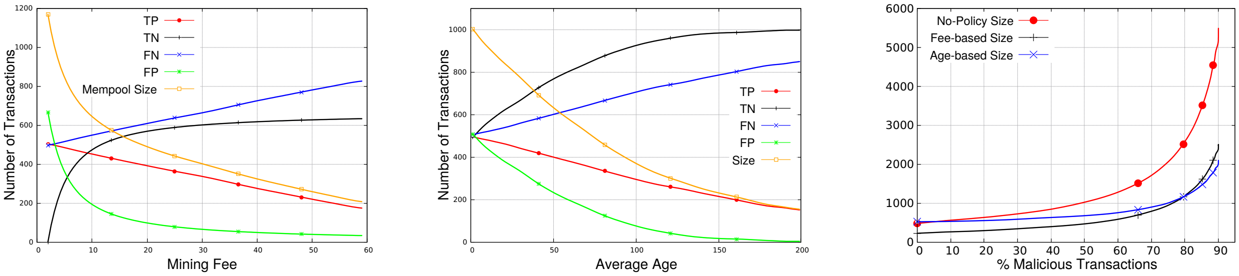
Figure 3: Age-based Mempool Design

3 COUNTERING MEMPOOL DDOS ATTACK

To counter DDoS attacks on the Bitcoin mempool, we propose fee-based and age-based designs. Both designs leverage the intrinsic nature of transactions and prevent transmission of spam.

Fee-based Design. In the fee-based design, an incoming transaction is accepted by the mempool if it pays both the minimum relay fee and the minimum mining fee. The key idea behind this scheme is to counter the strategy of the attacker, by allowing only those transactions to be accepted, which eventually aim to get mined into the Blockchain. As a result, this technique puts a cap on the incoming transactions and filters spam transactions, thereby reducing the mempool size, as shown in Figure 2.

Age-based Design. In the age-based design, shown in Figure 3, for each incoming transaction, we count the number of its inputs or parent transactions. We initialize a variable "average age" and set its value to 0. Next, we calculate the average age of the transaction by adding the age of each parent transaction and dividing by the total number of parent transactions. This gives an estimate of mean confirmation score of the incoming transaction. Then, we apply a



(a) Performance of fee-based design design

(b) Performance of age-based design

(c) Performance under spam attack

Figure 4: Analysis of proposed countermeasures for DDoS attack on mempool

"minimum age limit" filter on the mempool. The "minimum age limit" can take any arbitrary value greater than 0. According to Bitcoin Developers Guide [3], a confirmation score of 6 is considered good for any transaction. If the transaction's mean age value fulfills the age criteria, only then the mempool accepts the transaction. In this way, all the unconfirmed transactions generated by the attacker and the Sybil nodes will be rejected by the mempool, while the transactions of legitimate users will be accepted. If the attacker would still want to spam the network, he will have to get all his transactions mined and wait for them to acquire significant age. As such, this increases the cost of the attack and reduces the time window in which the attack can be launched.

4 EXPERIMENT AND RESULTS

To evaluate the performance of our proposed designs, we carry out two experiments. For the fee-based design, we select a suitable budget for the attacker that results into 1000 transactions with a minimum mining fee. We generate 1000 legitimate transactions, each with a mining fee normally distributed over the range of the minimum and maximum mining fee. Using discrete-event time simulation, we increase the mining fee and monitor its effects on transactions of attacker and legitimate users. For the age-based design, we set a minimum age limit and a maximum age limit as thresholds for the incoming transactions. Transactions from the attacker to the Sybils were assigned an age value greater than 1 due to confirmed parent transactions, while the transactions among the Sybils were assigned 0 age value due to unconfirmed parent transactions. To capture that, we normally distribute the average age value of all malicious transactions from 0 to the minimum age limit. The average age value of all legitimate transactions was set from 0 to the maximum age limit. A total of 2000 transactions were generated with 1000 malicious transactions and 1000 legitimate transactions. Then we applied the age-based design on all the incoming transactions at the mempool. We increased the age requirement for the incoming transactions and evaluated the mempool state.

We plot the results of our experiments in 4(a) and 4(b), and use the confusion matrix (for actual and mempool transactions each with legitimate and malicious, we define the true positive, true negative, false positive, and false negative, as in the literature) to evaluate the effect of fee-based and age-based designs. From the two experiments, we derive an optimum cut-off for mining fee and average age. Finally, we fix the average number of legitimate transactions and increase the percentage of malicious transactions. We observe the change in mempool size with optimum cut-off values. We report our results in 4(c). Using this knowledge and the results obtained from our experiments, we derive Equation 1 that effectively reduces the spam transactions under varying attack conditions.

$$\underset{f,a}{\text{minimize}} \quad R_{spam}(f,a) = \alpha \frac{\Omega(f)}{N} + (1-\alpha)\frac{\Phi(a)}{N}, \tag{1}$$

In Equation 1, f and a are the mining fee and average age cutoffs, used to minimize the accepted spam ratio R_{spam}. $\Omega(f)$ and $\Phi(a)$ are two functions of mining fee f and the average age a learned from the simulations to show the numbers of accepted spam under the two designs. $0 \leq \alpha \leq 1$ is a hyperparameter for balancing the weights of the designs, and N is the total number of transactions.

5 CONCLUSION

In this paper, we identify a DDoS attack on Bitcoin mempools that traps users into paying higher mining fees. Attacks on Bitcoin mempools have not been addressed previously, so we propose two countermeasures to the problem: fee-based and age-based designs. From our simulations, we conclude that when the attack is not severe, the fee-based design is more effective in mempool size optimization. However, it affects both the attacker and the legitimate users. In contrast, when the attack is severe, the age-based design is more useful in helping legitimate users while discarding a maximum of spam transactions.

REFERENCES

[1] Khaled Baqer, Danny Yuxing Huang, Damon McCoy, and Nicholas Weaver. 2016. Stressing out: Bitcoin "Stress Testing". In *International Conference on Financial Cryptography and Data Security*. Springer, 3–18.

[2] BitcoinJS. 2017. The clean, readable, proven library for Bitcoin JavaScript development. (2017). Retrieved November 28, 2017 from https://goo.gl/tNMGvj

[3] Bitcoin Community. 2009. Bitcoin Developer Guide. (2009). Retrieved March 5, 2018 from https://bitcoin.org/en/developer-guidepeer-discovery

[4] Bitcoin Community. 2017. Someone is spamming the mempool with extremely low-fee transactions. (2017). Retrieved March 4, 2018 from https://goo.gl/ggSULm

[5] Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin Gün Sirer, et al. 2016. On scaling decentralized blockchains. In *International Conference on Financial Cryptography and Data Security*. Springer, 106–125.

[6] T. N. K. De Zoysa Iresha Dilhani Rubasinghe. 2017. Transaction Verification Model over Double Spending for Peer-to-Peer Digital Currency Transactions based on Blockchain Architecture. Vol. 163. Foundation of Computer Science, 24–31. http://www.ijcaonline.org/archives/volume163/number5/rubasinghe-2017-ijca-913531.pdf

[7] Francisco Memoria. 2017. 700 Million Stuck in 115,000 Unconfirmed Bitcoin Transactions. (2017). Retrieved March 6, 2017 from https://goo.gl/cvSTCD

[8] Muhammad Saad, Aziz Mohaisen, Charles Kamhoua, Kevin Kwait, and Laurent Njilla. 2018. Countering Double Spending in Next-Generation Blockchains. In *IEEE International Conference on Communications.*

[9] Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. 2016. Optimal selfish mining strategies in Bitcoin. In *Financial Cryptography and Data Security*. Springer, 515–532.

[10] Bitcoin Wiki. 2017. Raw Transactions. (2017). Retrieved August 28, 2017 from https://en.bitcoin.it/wiki/Raw_Transactions