UDC 621.391:519.2:519.7

Estimation of the Probability of Success of a Suppression Attack

Anton Vykhlo[1], Lyudmila Kovalchuk[1]

[1]*National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Institute of Physics and Technology*

Abstract

This work presents the results of research on suppression attacks, which are a specific case of frontrunning attacks. We provide a formal step-by-step algorithm for executing the attack, along with a mathematical model and explicit analytical formulas for calculating an upper bound on the success probability of such an attack with numerical examples.

This study continues the research presented in [1], which investigated insertion and displacement attacks.

Keywords: suppression attack, frontrunning attack, blockchain, mempool, smart contract.

Introduction

Suppression attacks, which represent a particular case of frontrunning attacks, pose a significant threat to blockchain networks that are based on public mempools. The core idea of such attacks is to manipulate the order in which transactions are included in a block, with the goal of delaying the execution of the original transaction. These attacks do not exploit vulnerabilities in smart contracts and their execution uses only publicly available information. This greatly simplifies the attack process, resulting in a high frequency of their occurrence.

In this work, a step-by-step algorithm for performing a suppression attack is formalized. Based on this algorithm, a mathematical model of the attack is proposed, which enables the derivation and rigorous proof of explicit analytical formulas for the upper bound of its success probability.

1. Review of Related Work

One of the first groups of researchers to distinguish between different types of frontrunning attacks were S. Eskandari, S. Musavi, and J. Clark. In their work [2], they identified and defined the suppression attack, and explained why it cannot be reduced to a large number of repeated displacement attacks. The key difference lies in wider variety of techniques utilized by suppression attacks. Suppression can be achieved not only through displacement (i.e., delaying a transaction due to block size limits), but also by other means, for example, by exhausting the gas limit allocated to a single block. Also worth mentioning a similarity between suppression attacks and censorship attacks, which were studied by by Z. Wang [3] and J. Kim [4]. Both of attacks aim to leverage delaying the processing of a transaction, but they rely on different mechanisms. A suppression attack aims to manipulate the order in which transactions were processed, and this can be achieved through multiple completely different strategies. The main ones include:

1) Manipulating the gas limit – this strategy is based on exploiting the block gas limit, i.e., the maximum total amount of gas that can be used by all transactions included in a block. This mechanism is described in more detail in the work by M. Varun [5].
2) Influencing validators – this approach involves censoring transactions through validators. It is thoroughly examined in the works of A. Wahrstätter [6, 7], which investigate methods of exerting influence on network validators.
3) Influencing miners – this strategy relies on altering miners' behavior with additional profit beyond the standard block reward

(Mining Extractable Value) to manipulate the transaction order within the blocks they produce [8, 9, 10]. Studying the incentives and mechanisms that influence miners provides further insight into how suppression attacks can be performed.

As we can see, the objective of a suppression attack can be achieved even in the absence of network or smart contract vulnerabilities, which makes protection against such attacks an important subject of research.

Various scientific studies propose different approaches to defending against suppression attacks and attacks that involve similar manipulations for gaining profit. For example, the use of private mempools complicates the attacker's ability to gather information about vulnerable transactions [11]. Implementing fair transaction ordering mechanisms weakens the levers of influence required for a successful attack [12, 13, 14]. The development of models for detecting and preventing such attacks [15] provides additional insights into the emerging strategies and techniques available to adversaries.

This article is a continuation of the research presented in [1], which enables the estimation of success probabilities for other types of frontrunning attacks.

2. Formalization of the Suppression Attack

A suppression attack is a specific case of a frontrunning attack. The primary goal of this attack is to delay the execution of an original transaction. The targets of such an attack are typically transactions that have a noticeable impact on market conditions. By knowing the exact effect of the original transaction, the attacker seeks to postpone its inclusion in a block in order to gain time for manipulating the asset before the market conditions change due to the original transaction. One of the most common implementations of this attack is delaying the original transaction by one block, which gives the attacker time for manupulations before the next block is produced.

The process of the attack can be described in more detail as follows.

To obtain the necessary information for the attack, the attacker begins monitoring transactions in the public mempool that are waiting to be included in a block. Upon identifying a transaction whose delay could potentially yield a profit, the attacker selects one of the available methods to influence the transaction ordering within the block. The implementation of strategies involving influence over miners or validators is difficult to formalize mathematically, as it depends on network-specific mechanisms and reward structures. Therefore, in this work, we focus on the manipulation strategy based on the gas limit within the Ethereum blockchain network.

Given that the attacker's objective is to delay the inclusion of the original transaction until the next block, the attacker must create a series of transactions that will be included before the original one, consume the entire block gas limit, and thereby prevent it from being included into the current block.

One of the key parameters that defines the characteristics of this transaction series is the block gas limit. The block gas limit is a network parameter that specifies the maximum amount of gas that can be used by all transactions included in a block. Although this limit is not fixed, its value can be accurately predicted by analyzing recent blocks. Since each transaction consumes a certain amount of gas, the attacker's next step is to estimate the block gas limit and determine the optimal gas usage per transaction so that including this series (or a sufficient subset of it) uses the entire block gas capacity before the original transaction can be included.

Another important characteristic of the attacker's transactions is the size of their fees. To increase the likelihood that these transactions will be included before the original one, their priority among miners must be raised. This is typically achieved by significantly increasing the fee of the attacker's transactions compared to the original. Thus, the attacker's next step is to determine an appropriate fee level, depending on the desired probability of success.

Once these parameters are determined, the attacker generates the transaction series and submits it to the network.

This results in the original transaction being delayed by at least one block, giving the attacker time to exploit market conditions for profit.

The formal algorithm for suppression attacks can be written as follows:

Algorithm 1. Execution of Suppression Attack

Input:

- Publicly available information about transactions (including their fees) that are currently in the mempool and awaiting processing (i.e., inclusion in a block);
- Expected gas limit of the next block.

Steps:

1) Among the pending transactions in mempool, identify which one can provide profit if successfully attacked.
2) Estimate the amount of gas for each attacker's transaction to consume and the number of such transactions needed to consume all the gas of the upcoming block.
3) Estimate the fee for each attacker's transaction to ensure it will be included in the block before the original transaction with sufficient probability.
4) Create a series of transactions according to the calculated parameters.
5) Send the series to the blockchain network.

Output: A series of the attacker's transactions

The attack is considered successful if the generated series of transactions (or its sufficient subset) is included in the block before the original transaction. The original transaction may be evicted either because there is no remaining space in the block, or because the attacker's transactions included in the block have consumed all available gas.

3. Probability of Success of a Suppression Attack

Let us introduce the following notations. Let the value τ denote the fee size of a transaction, and let the random variable T_τ represent the processing time t of a transaction with fee τ. This random variable is typically modeled as having an exponential distribution [16] with a parameter λ, where the expected waiting time is equal to $1/\lambda$. Thus, the higher the parameter λ, the shorter the expected processing time of the transaction.

In this work, we adopt the same model as in [1], assuming that the parameter λ is an increasing function of the transaction fee: $\lambda = \lambda(\tau)$.

Next, let $\{\Delta_i\}_{i=1}^{n+k}$ denote the time intervals required by the attacker to create a series of alternative transactions $\{\mathrm{Tx}_i\}_{i=1}^{n+k}$, and define $\Delta = \sum_{i=1}^{n+k} \Delta_i$.

Denote τ_{orig} as the fee set in the original transaction.

Also, let $\Lambda_i = \frac{\lambda(\tau_i)}{\lambda(\tau_{\text{orig}})+\lambda(\tau_i)}$ and $N_{n+k} = \{1, \dots, n+k\}$.

Theorem 1. *Let the attacker generate a series of $n+k$ transactions, such that any subset of n of them is sufficient to fully consume the gas limit of the upcoming block. Denote by $\tau_1, \dots, \tau_{n+k}$ the fees assigned by the attacker for processing these transactions.*

Then the probability of a successful suppression attack, denoted by $P_{\text{sup}}(\tau_{\text{orig}}, \tau_1, \tau_2, \tau_3, \dots, \tau_{n+k}, \Delta)$, can be upper-bounded by the following expression:

$$P_{\text{sup}}(\tau_{\text{orig}}, \tau_1, \dots, \tau_{n+k}, \Delta) \leq$$
$$\leq e^{-\lambda(\tau_{\text{orig}})\cdot\Delta} \sum_{\{i_1,\dots,i_n\}\subset N_{n+k}} \prod_{u=1}^{n} \frac{\lambda(\tau_{i_u})}{\lambda(\tau_{\text{orig}})+\lambda(\tau_{i_u})}$$

Proof. According to the suppression attack algorithm, the attack is successful if the time required to identify a suitable transaction, generate a series of alternative transactions, and process at least any n of them is less than the processing time of the original transaction. Let A_{i_t} be an event $T_{\tau_{i_t}} + \Delta_{i_t} < T_{\tau_{\text{orig}}}$. Following the defined notation, this probability can be calculated as:

$$P_{\text{sup}}(\tau_{\text{orig}}, \tau_1, \tau_2, \tau_3, \dots, \tau_{n+k}, \Delta) =$$
$$= \sum_{\{i_1,\dots,i_n\}\subset N_{n+k}} P(A_{i_1}, A_{i_2}, \dots, A_{i_n})$$

As long as suppression attack does not need a specific order of attacker's transactions, the events of transaction inclusion are mutually independent. Therefore, the probability of the intersection of these events:

$$\sum_{\{i_1,\dots,i_n\}\subset N_{n+k}} P(A_{i_1}, A_{i_2}, \dots, A_{i_n})$$

Can be calculated as the product of the probabilities of independent events:

$$\sum_{\{i_1,\dots,i_n\}\subset N_{n+k}} P(A_{i_1})\cdot P(A_{i_2})\cdot\ldots\cdot P(A_{i_n})$$

Thus, we observe that the expression under the summation corresponds to a series of displacement attacks [1]:

$$\sum_{\{i_1,\dots,i_n\}\subset N_{n+k}} \prod_{u=1}^{n} P_{\text{dis}}(\tau_{\text{orig}}, \tau_{i_u}, \Delta_{i_u})$$

Using the results for the probability of a displacement attack presented in [1], we obtain:

$$P_{\text{sup}}(\tau_{\text{orig}}, \tau_1, \tau_2, \tau_3, \dots, \tau_{n+k}, \Delta) =$$

$$= \sum_{\{i_1,\dots,i_n\}\subset N_{n+k}} \prod_{u=1}^{n} e^{-\lambda(\tau_{\text{orig}})\cdot\Delta_{i_u}} \cdot \Lambda_{i_u} \leq$$

$$\leq \sum_{\{i_1,\dots,i_n\}\subset N_{n+k}} e^{-\lambda(\tau_{\text{orig}})\cdot\Delta} \cdot \prod_{u=1}^{n} \Lambda_{i_u} =$$

$$= e^{-\lambda(\tau_{\text{orig}})\cdot\Delta} \cdot \sum_{\{i_1,\dots,i_n\}\subset N_{n+k}} \prod_{u=1}^{n} \Lambda_{i_u}$$

which concludes the proof ■

Let us now present several corollaries of Theorem 1 that provide upper bounds on the success probability of the attack under certain conditions.

Corollary 1. *Probability of a suppression attack under equal transaction fees.*

Assuming that, under the conditions of Theorem 1, all $n+k$ transactions created by the attacker have identical fees τ. Then, in our notation, we obtain the following inequality:

$$P_{\text{sup}}(\tau_{\text{orig}}, \tau, \Delta) \leq$$

$$\leq e^{-\lambda(\tau_{\text{orig}})\cdot\Delta} \cdot C_{n+k}^{n} \cdot \left(\frac{\lambda(\tau)}{\lambda(\tau_{\text{orig}}) + \lambda(\tau)}\right)^{n}$$

Proof. According to the proof of Theorem 1, the suppression attack reduces to a series of displacement attacks:

$$P_{\text{sup}}(\tau_{\text{orig}}, \tau_1, \tau_2, \dots, \tau_{n+k}, \Delta) = \sum_{\{i_1,\dots,i_n\}\subset N_{n+k}} P_{\text{dis}}(\tau_{\text{orig}}, \tau_{i_1}, \Delta_{i_1})\cdot$$

$$\cdot P_{\text{dis}}(\tau_{\text{orig}}, \tau_{i_2}, \Delta_{i_2}) \cdot \ldots \cdot P_{\text{dis}}(\tau_{\text{orig}}, \tau_{i_n}, \Delta_{i_n})$$

Given the imposed constraint on the transaction fee, we have $\tau_{i_1} = \tau_{i_2} = \ldots = \tau_{i_n} = \tau$. Thus, we consider a selection of n transactions from the full set of $n+k$ transactions created by the attacker, for which the probability of a displacement attack is calculated. Therefore, we obtain the following inequality:

$$P_{\text{sup}}(\tau_{\text{orig}}, \tau, \Delta) = C_{n+k}^{n} \cdot \prod_{i=1}^{n} P_{\text{dis}}(\tau_{\text{orig}}, \tau, \Delta_i) \leq$$

$$\leq e^{-\lambda(\tau_{\text{orig}})\cdot\Delta} \cdot C_{n+k}^{n} \cdot \left(\frac{\lambda(\tau)}{\lambda(\tau_{\text{orig}}) + \lambda(\tau)}\right)^{n}$$

which concludes the proof ■

Corollary 2. *Probability of a suppression attack under the minimum number of transactions*

Assume that under the conditions of Theorem 1, $k = 0$ (i.e., the attacker has created the minimally sufficient number of transactions). Let the fees of the created transactions be denoted by $\tau_1, \dots, \tau_n$, and let the fee of the original transaction be τ_{orig}. Then the probability of a successful suppression attack can be upper-bounded by the following expression:

$$P_{\text{sup}}(\tau_{\text{orig}}, \tau, \Delta) \leq$$

$$\leq e^{-\lambda(\tau_{\text{orig}})\cdot\Delta} \cdot \prod_{i=1}^{n} \frac{\lambda(\tau_i)}{\lambda(\tau_{\text{orig}}) + \lambda(\tau_i)}$$

Proof. According to the assumption, the suppression attack reduces to a series of n frontrunning attacks of the displacement type, and therefore:

$$P_{\text{sup}}(\tau_{\text{orig}}, \tau_1, \tau_2, \dots, \tau_n, \Delta) = P_{\text{dis}}(\tau_{\text{orig}}, \tau_1, \Delta_1)\cdot$$

$$\cdot P_{\text{dis}}(\tau_{\text{orig}}, \tau_2, \Delta_2) \cdot \ldots \cdot P_{\text{dis}}(\tau_{\text{orig}}, \tau_n, \Delta_n).$$

Using the results for the success probability of a frontrunning attack of the displacement type [1], we obtain:

$$P_{\text{sup}}(\tau_{\text{orig}}, \tau_1, \dots, \tau_n, \Delta) =$$

$$= e^{-\lambda(\tau_{\text{orig}})\cdot\Delta_1} \cdot \Lambda_1 \cdot \ldots \cdot e^{-\lambda(\tau_{\text{orig}})\cdot\Delta_n} \cdot \Lambda_n =$$

$$= \prod_{i=1}^{n} e^{-\lambda(\tau_{\text{orig}})\cdot\Delta_i} \cdot \Lambda_i \leq e^{-\lambda(\tau_{\text{orig}})\cdot\Delta} \cdot \prod_{i=1}^{n} \Lambda_i$$

which concludes the proof ■

4. Example calculations

In this section, we provide example calculations to demonstrate the application of Theorem 1 for the calculation of the upper bound for the success probability of suppression attacks. These calculations illustrate how theoretical results can be applied to practical scenarios.

We present two examples: one for a case with an easier attack scenario, where suppression attack is more likely to succeed, and another for a more difficult attack scenario, where additional constraints reduce the probability of success.

Example 1.

Let the attacker be performing a suppression attack targeting a transaction with a fee of $\tau_{\text{orig}} = 4$, creating a series of four transactions with fees $\tau_1 = 8$, $\tau_2 = 9$, $\tau_3 = 12$, and $\tau_4 = 16$, spending a total of $\Delta = 1$ units of time to generate them.

Assume that the attack is successful if at least one of these transactions is included in the block before the original transaction.

In the notation of Theorem 1, this corresponds to $n = 1$ and $k = 3$. Let the parameter λ be defined by the function $\lambda(\tau) = \sqrt{\tau}$.

Then, the success probability of the attack can be upper-bounded by the following expression:

$$P_{\text{sup}}(\tau_{\text{orig}}, \tau_1, \tau_2, \tau_3, \ldots, \tau_{n+k}, \Delta) \leq$$

$$\leq e^{-\lambda(\tau_{\text{orig}})\cdot\Delta} \cdot \sum_{\{i_1,\ldots,i_n\}\subset N_{n+k}} \prod_{u=1}^{n} \Lambda_{i_u} =$$

$$= e^{-\lambda(\tau_{\text{orig}})\cdot\Delta} \cdot \Big(\Lambda_1 + \Lambda_2 + \Lambda_3 + \Lambda_4\Big) \leq$$

$$\leq e^{-2} \cdot 2.487 \leq 0.34$$

This example shows that the upper bound of the probability of a suppression attack under such conditions is significant. Such high value can be explained by the fact that only 1 out of 4 of the attacker's transactions had to be included in the block before the original one. Additionally, it is worth mentioning that the attacker's transactions had significantly higher fees than the original transaction.

Example 2.

Let the attacker be performing a suppression attack targeting a transaction with a fee $\tau_{\text{orig}} = 2$, by creating a series of four transactions with fees $\tau_1 = 4$, $\tau_2 = 6$, $\tau_3 = 8$, and $\tau_4 = 10$, spending a total of $\Delta = 2$ units of time to generate them. Assume that the attack is successful if any three transactions from the generated series are included in the block before the original transaction, which corresponds in the notation of Theorem 1 to $n = 3$ and $k = 1$.

Let the parameter λ be defined by the function $\lambda(\tau) = \tau^2$.

Then the success probability of the attack can be upper-bounded by the following expression:

$$P_{\text{sup}}(\tau_{\text{orig}}, \tau_1, \tau_2, \tau_3, \tau_4, \Delta) \leq$$

$$\leq e^{-\lambda(\tau_{\text{orig}})\cdot\Delta}\Big(\Lambda_1 \cdot \Lambda_2 \cdot \Lambda_3 + \Lambda_1 \cdot \Lambda_2 \cdot \Lambda_4 +$$

$$+ \Lambda_1 \cdot \Lambda_3 \cdot \Lambda_4 + \Lambda_2 \cdot \Lambda_3 \cdot \Lambda_4\Big) \leq$$

$$\leq e^{-8} \cdot (0.6777 + 0.692 + 0.7238 + 0.8143) \leq$$

$$\leq 0.000975.$$

This example shows that the upper bound of the probability of a suppression attack under such conditions is significantly lower than in the scenario of Example 1. In this example we see that the attack is very unlikely to succeed which can be explained by the fact that 75% of the attacker's transactions had to be included in the block before the original. Also, it is worth mentioning that the time required for transaction creation is higher than in the previous example.

5. Acknowledgements

The research was conducted as part of the project 'Development of Distributed Energy in the Context of the Ukrainian Electricity Market Using Digitalization Technologies and Systems,' implemented under the state budget program 'Support for Priority Scientific Research and Scientific-Technical (Experimental) Developments of National Importance' (CPCEL 6541230) at the National Academy of Sciences of Ukraine.

Conclusions

This work presented a detailed description of suppression attacks. The relevance of this research is highlighted by exploring diverse methods of manipulating transaction order for attacks that use similar approaches for gaining profit. A step-by-step algorithm for performing suppression attacks is formulated. Based on this algorithm, formulas for calculating the upper bound of this probability are derived and proven.

The direction of further research involves an in-depth study of methods for influencing transaction ordering and identifying specific characteristics of these methods across different blockchain networks.

This includes analyzing the impact of various ordering mechanisms on the network fairness, exploring potential vulnerabilities of validators and investigating other methods of attack detection.

Another research direction is the analysis of the dependency of transaction processing time on the fee to improve the estimation precision.

References

[1] L. V. Kovalchuk and A. A. Vykhlo, “Estimation of the probability of success of a frontrunning attack on smart contracts,” Cybernetics and Systems Analysis, vol. 60, pp. 881–890, 2024.

[2] S. Eskandari, S. Moosavi, and J. Clark, “SoK: Transparent dishonesty: Front-running attacks on blockchain,” Proceedings of the 2019 Workshop on Trusted Smart Contracts, 2019.

[3] Z. Wang, X. Xiong, and W. Knottenbelt, “Blockchain Transaction Censorship: (In)secure and (In)efficient?,” Mathematical Research for Blockchain Economy. MARBLE 2023, 2023.

[4] J.-S. Kim, J.-M. Shin, S.-H. Choi, and Y.-H. Choi, “A study on prevention and automatic recovery of blockchain networks against persistent censorship attacks,” IEEE Access, vol. 10, pp. 110770–110784, 2022.

[5] M. Varun, B. Palanisamy, and S. Sural, “Mitigating frontrunning attacks in Ethereum,” Proceedings of the Fourth ACM International Symposium on Blockchain and Secure Critical Infrastructure (BSCI ’22), pp. 115–124, 2022.

[6] A. Wahrstätter, L. Zhou, K. Qin, D. Svetinovic, and A. Gervais, “Time to bribe: Measuring block construction market,” arXiv, 2023.

[7] A. Wahrstätter, J. Ernstberger, A. Yaish, L. Zhou, K. Qin, T. Tsuchiya, S. Steinhorst, D. Svetinovic, N. Christin, M. Barczentewicz, and A. Gervais, “Blockchain censorship,” Proceedings of the ACM Web Conference 2024 (WWW ’24), pp. 1632–1643, 2024.

[8] Z. Alipanahloo, A. S. Hafid, and K. Zhang, “Maximum Extractable Value (MEV) Mitigation Approaches in Ethereum and Layer-2 Chains: A Comprehensive Survey,” IEEE Access, vol. 12, pp. 185212–185231, 2024.

[9] V. Gramlich, D. Jelito, and J. Sedlmeir, “Maximal extractable value: Current understanding, categorization, and open research questions,” Electronic Markets, vol. 34, p. 49, 2024.

[10] H. Materwala, S. M. Naik, A. Taha, T. A. Abed, and D. Svetinovic, “Maximal extractable value in decentralized finance: Taxonomy, detection, and mitigation,” arXiv, 2025.

[11] A. Capponi, R. Jia, and Y. Wang, “Do private transaction pools mitigate frontrunning risk?,” Cryptology ePrint Archive, Paper 2023/1461, 2023.

[12] A. Kiayias, N. Leonardos, and Y. Shen, “Ordering transactions with bounded unfairness: Definitions, complexity and constructions,” Advances in Cryptology – EUROCRYPT 2024, vol. 14653, pp. 35–65, 2024.

[13] M. Kelkar, S. Deb, S. Long, A. Juels, and S. Kannan, “Themis: Fast, strong order-fairness in Byzantine consensus,” Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS ’23), pp. 475–489, 2023.

[14] W. Yahyaoui, J. Bruneau-Queyreix, J. Decouchant, and M. Völp, “Mitigating front-running attacks through fair and resilient transaction dissemination,” Proceedings of the 55th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2025.

[15] Y. Zhang, P. Liu, G. Wang, P. Li, W. Gu, H. Chen, X. Liu, and J. Zhu, “FRAD: Front-running attacks detection on Ethereum using ternary classification model,” 2023.

[16] N. T. Thomopoulos, Exponential, pp. 21–29. Cham: Springer International Publishing, 2017.