F3B: A practical per-transaction front-running mitigation solution for blockchain

IC3

Follow

--

Listen

Share

by Haoqian Zhang (EPFL, IC3)

Contributors: Louis-Henri Merino (EPFL, IC3), Ziyan Qu (EPFL), Mahsa Bastankhah (EPFL), Vero Estrada-Galinanes (EPFL, IC3), Bryan Ford (EPFL, IC3)

Introduction to Front-running

The name “front-running” came from when a broker needs to deliver the clients’ orders to the trading desk physically. The term vividly describes how it works: an attacker who knows a large order could run ahead to execute a trade before the client’s order goes through.

What is the incentive for someone to do that? Here is an example that explains why. Suppose a broker receives a large order from a client, say, buy 500,000 shares of a company’s stock. The order is big enough to drive up the share’s price. Knowing this information, an attacker can place his small order, say 10,000 shares of the same stock, before the large order. The attacker can sell his shares at a higher price when the price goes up after the large order went through.

The formal definition of front-running is a practice of benefiting from the advanced knowledge of pending transactions. Although benefiting some entities involved, this practice puts others at a significant financial disadvantage, making this behavior illegal in traditional markets with established securities regulations.

[Stock brokers working at the New York Stock Exchange in 1963. They need to physically go to the trading desk to place an order.]

Front-running in Blockchain

With the ambition of the blockchain community to mimic the traditional Centralized Finance to the Decentralized Finance (DeFi), more and more financial applications appear in the blockchain world. At the same time, this creates enormous opportunities for attackers to front-run clients’ transactions to gain an unfair profit. One estimate suggests that front-running attacks amount to $280 million in losses for DeFi actors each month.[1]

However, due to the open and pseudonymous nature of blockchain transactions, it is virtually impossible to rely on regulations to mitigate this issue; thus, we must rely on technology to protect users.

How front-running works in blockchain is very similar to the traditional exchange example. Instead of the broker, a blockchain miner/validator is the middleman who places users’ transactions into the blockchain. As all the transactions are in plaintext by default, anyone can observe the pending transaction and become a front-runner, including miners/validators.

[An example of how front-running works in the blockchain world. A miner can insert his transactions before and after the customer’s transaction to make a profit.]

Flash Freezing Flash Boys (F3B)

Our solution, F3B, mitigates front-running attacks in a very simple way. Instead of plaintext transactions, users encrypt their transactions so that no other entities can know the content of the transaction. When the encrypted transactions are firmly written into the blockchain, the consensus nodes can decrypt, verify, and execute the transactions.

We can reason why F3B mitigates the attacks from the definition of the front-running. Before the consensus group finalizes the transaction, an attacker can no longer read the content of it, thus preventing the adversary from benefiting from advanced knowledge of pending transactions.

To properly ensure no single point of failure, F3B adopts threshold encryption. A Secret-Management Committee (SMC) can generate a public key for a user to encrypt their transaction. When the consensus group finalizes the transaction, SMC releases the encryption key so that the consensus group can validate and execute it. Threshold encryptions guarantee as long as there are more than half of the nodes following the protocol honestly, F3B can successfully mitigate the front-running

attacks. In practice, the SMC and the consensus group can consist of the same set of servers.

[F3B architecture. Senders publish encrypted transactions to the consensus group. The secret-management committee (SMC) releases the decryption shares once the transactions are no longer pending. Finally, the consensus group reconstruct the key and decrypt and execute the transaction. The secret-management committee and the consensus group can consist of the same set of servers. For clarity in this figure, we logically separate them into two different entities.]

F3B unique features

Architecture level solution: F3B addresses front-running attacks at the architecture level. Deploying F3B requires a change of the execution layer, requiring a hard fork of the existing blockchain. This inevitably brings some difficulty for adoption, but the benefit is substantial: all the smart contracts on the blockchain, by default, have front-running protection with low overhead costs.

Per-transaction protection: Notably, F3B adopts per-transaction basis protection rather than per-block basis, as presented in Fairblock and Shutter. These schemes require clients to choose a future block to derive the encryption key, which raises security concerns. Suppose a transaction failed to be finalized in the client-chosen block due to, for example, a crypto mania that overwhelms the blockchain network or a deliberate denial of service attack. In this case, the transaction is undesirably revealed. F3B, on the other hand, asks clients to generate an encryption key for each transaction. This ensures that a transaction remains confidential until the transaction has received enough confirmations.

Spamming issue: As the consensus group cannot execute encrypted transactions, an adversary could, at a low cost, spam the blockchain with non-executable transactions (e.g., inadequate fees, malformed transactions), thus delaying the finality of honest transactions. To make such an attack costly, we introduce a storage deposit alongside the traditional execution fee (e.g., gas in Ethereum) and adjustable based on the transaction's size. The underlying blockchain can deduct the storage deposit from the sender's balance, much like paying a transaction fee. Then, the blockchain can partially refund the deposit after successful execution by the consensus nodes. This approach imposes a low-cost fee on compliant users and a penalty on those who misbehave.

Practical to deploy: Our prototype on Ethereum found only a negligible (0.026%) increase in transaction latency, specifically due to running threshold decryption with a 128-member secret-management committee after a transaction is finalized; this indicates that F3B is both practical and low-cost to deploy.

Conclusion

We have introduced F3B, a novel blockchain architecture that addresses front-running attacks on a per-transaction basis. Our prototype demonstrates that F3B can practically deploy to Ethereum. Given that the deployment of F3B would require modifications to a blockchain's execution layer, F3B, in return, would also provide a substantial benefit: the F3B-deployed blockchain would now, by default, contain standard front-running protection for all applications in need at once without requiring any modifications to smart contracts themselves.

[1] Source: https://cybernews.com/crypto/flash-boys-2-0-front-runners-draining-280-million-per-month-from-crypto-transactions/

Original paper:

[https://arxiv.org/abs/2205.08529

](https://arxiv.org/abs/2205.08529)

Corresponding author: Haoqian Zhang (EPFL, IC3,

[haoqian.zhang@epfl.ch

](mailto:haoqian.zhang@epfl.ch))

Editor: Bria Han (IC3,

[jh2584@cornell.edu

](mailto:jh2584@cornell.edu))