Impacto do Paradigma de Separação Proponente-Construtor em Ataques Sanduíche na Rede Ethereum

Josué N. Campos[1], Isdael R. Oliveira[2], Alexandre Fontinele[2], Glauber D. Gonçalves[2], Alex B. Vieira[3] e José Augusto M. Nacif[1]

[1] Universidade Federal de Viçosa (UFV) – Florestal, MG – Brasil
[2] Universidade Federal do Piauí (UFPI) – Teresina, PI – Brasil
[3]Universidade Federal de Juiz de Fora (UFJF) – Juiz de Fora, MG – Brasil

{josue.campos, jnacif}@ufv.br, alex.borges@ufjf.edu.br

{isdael, alexandre.fontinele, ggoncalves}@ufpi.edu.br

Resumo. *A Máxima Extração de Valor em Blocos (MEV) surgiu como uma questão de extrema importância, particularmente no ecossistema Ethereum DeFi. As práticas de MEV permitem que os traders maximizem seus lucros ao reordenar, inserir ou bloquear transações dentro de um bloco. Recentemente, a rede Ethereum implementou o Paradigma de Separação Proponente-Construtor (PBS), dividindo a função dos mineradores em nós construtores e validadores. Apesar deste novo design, o fenômeno MEV permanece. Neste artigo, investigamos o ataque sanduíche, uma prática especial de MEV baseada na manipulação de preços por meio de técnicas de front-running - na rede Ethereum sob o paradigma PBS. Nossa análise abrangeu mais de 1 milhão de blocos ao longo de 2023, onde identificamos aproximadamente 1,5 milhão de ataques sanduíche, com um lucro médio de US$ 3,2 mil para os atacantes. Nossos resultados mostram que o PBS contribuiu para encorajar as atividades de sanduíche, uma vez que os atacantes geralmente pagam as taxas mais altas aos construtores de blocos, cerca de 60% dos blocos, de acordo com nossas medições. Além disso, identificamos que poucos construtores se beneficiam dos ataques sanduíche. Neste caso, apenas 4 nós construtores receberam mais de 70% das taxas dos atacantes em 2023.*

1. Introdução

As Finanças Descentralizadas (DeFi) revolucionaram o mercado financeiro global. Em Dezembro de 2024, somente a rede Ethereum representou 55% do Valor Total Bloqueado (TVL) de todo o mercado de blockchains [DeFiLlama 2024]. Por meio da implementação de contratos inteligentes, o mercado DeFi atua como um ecossistema de aplicações descentralizadas (dApps) que visam replicar, aprimorar ou substituir os serviços financeiros tradicionais por meio de negociações e transferências de tokens [Zhou et al. 2021]. Um dos principais projetos DeFi que desempenham um papel importante na identificação de oportunidades lucrativas são as Corretoras Descentralizadas (DEXes). Uma DEX é responsável principalmente pela negociação de ativos através da compra ou venda automática realizadas pelos próprios usuários da rede blockchain.

Entretanto, os contratos inteligentes de DEXes são dependentes da ordem de execução das transações, *i.e.*, o resultado das operações de compra e venda de um ativo

financeiro (lucro ou prejuízo) é determinado pela ordem em que as transações são efetivadas na rede. Nesse sentido, *traders* ajustam a posição de suas transações para potencializar o lucro com base nas outras transações de um bloco [Chi et al. 2024]. Esta prática de incluir, excluir ou reordenar transações ficou conhecida como *Maximal Extractable Value* (MEV) e os *traders* foram denominados de MEV *searchers*. A prática de MEV pode ser executada de diversas maneiras, mas as estratégias mais populares são: Arbitragem, Liquidação e Sanduíche. Enquanto Arbitragem e Liquidação são "benéficas" para manter o ecossistema DeFi consistente em meio a diferentes DEXes, elas ainda assim podem gerar malefícios pelo fato de aumentarem as taxas pagas por transações. Dependendo da estratégia de Sanduíche empregada, o resultado pode ser prejudicial por fazer usuários obterem prejuízo pelas suas transações [Ferreira Torres et al. 2024].

Os trabalhos propostos na literatura abordam análises sobre os ataques sanduíche somente sob as perspectivas de lucro e prejuízo dos atacantes e das vítimas. Além disso, as atividades MEVs observadas por estes trabalhos são referentes a rede Ethereum antes da implementação do Paradigma de Separação Proponente-Construtor (*Proposer-Builder Separation* - PBS), criado para dividir o papel do nó minerador em duas partes. Neste novo paradigma, o nó construtor (*builder*) fica responsável por selecionar e ordenar as transações de um bloco e o nó proponente (*validator*) valida o conteúdo das transações selecionadas pelo *builder* e propõe a criação de um novo bloco. Nesse sentido, as pesquisas atuais abrem lacunas sobre como os ataques sanduíche são executados com o intuito de serem vantajosos para os nós deste novo paradigma, de maneira que um ataque não pode ser recusado no processo de construção de um bloco.

Para identificar essas lacunas, o presente trabalho tem como objetivo conduzir uma análise compreensiva dos ataques sanduíche que ocorrem na rede blockchain Ethereum sob a perspectiva do paradigma PBS. Nós analisamos os ataques sanduíche e transações realizados durante o ano de 2023 com o intuito de agrupar os ataques em relação aos atacantes e os contratos inteligentes utilizados. A partir do agrupamento, é possível extrair os nós construtores que mais se relacionam com os ataques estudados. Além disso, analisamos os ataques sanduíche em detrimento das outras transações que compõem um bloco, com o intuito de ranquear os ataques a partir do preço das taxas de cada transação. Dessa maneira, é possível estabelecer como os atacantes aproveitam da taxa paga aos nós construtores para garantir que o ataque sanduíche seja anexado ao bloco que será construído. Nós resumimos nossas principais contribuições a seguir:

- Nossas análises demonstram como os atacantes utilizam deste novo paradigma PBS para garantir que o ataque sanduíche ocorra com sucesso;
- Constatamos que em cerca de 60% dos ataques a segunda transação do atacante possui a maior taxa dentre as transações dos blocos com ataque. Além disso, em 30% dos ataques o pacote de transações do atacante possui em média a segunda maior tarifa dentre o restante das transações;
- Por meio do agrupamento de ataques, identificamos que apenas alguns nós construtores se beneficiam da maioria dos grupos. Os 4 principais nós construtores de blocos com ataque se beneficiaram das tarifas de pelo menos 70% dos grupos de ataques em 2023;
- O ataque sanduíche como atividade MEV pode aumentar as taxas da rede. Observamos por meio dos grupos de ataques que os atacantes gastam em média mais de 780 mil dólares em taxas de transação durante o sanduíche.

As próximas seções deste artigo estão organizadas da seguinte maneira: Na Seção 2 discutimos os principais conceitos relacionados a este trabalho. Na Seção 3 apresentamos a metodologia criada e o conjunto de dados utilizado para identificação dos resultados, apresentados na Seção 4. Por fim, nas Seções 5 e 6 definimos os trabalhos relacionados e a conclusão, respectivamente.

2. Fundamentos

Para quantificar o impacto dos ataques sanduíche mediante o paradigma PBS, é importante entender como eles funcionam, como funciona a rede Ethereum e as estratégias de MEV para maximização de lucro.

2.1. Ethereum

A Ethereum é uma rede blockchain pública e descentralizada potencializada pelo suporte de contratos inteligentes [Buterin et al. 2013]. A rede blockchain Ethereum é composta por uma sequência de blocos ordenados e interligados. Cada bloco contém um conjunto de transações ordenadas, sendo que cada transação pode ser desde uma transferência de uma criptomoeda nativa da rede, *e.g. Ether* (ETH), até mesmo a execução de um contrato inteligente.

Toda transação possui um custo computacional associado. Para mensurar este custo, a rede Ethereum conta com um mecanismo de cobrança denominado *gas*. O *gas* é a taxa cobrada pela execução de qualquer instrução de uma transação [Wood 2014]. Cada instrução possui uma quantidade fixa de *gas*, que é paga utilizando a moeda nativa da rede, o *Ether* (ETH). Nesse sentido, toda transação possui um preço de *gas* (*gasPrice*) e um limite de *gas* (*gasLimit*) associados, com o objetivo de prevenir gastos extensivos de recursos computacionais. Portanto, um contrato inteligente apesar de ser um programa escrito em uma linguagem Turing-completa, deve possuir operações que não sejam computacionalmente complexas.

Atualmente, de acordo com o mecanismo de consenso Prova por Participação (PoS) e o paradigma PBS, antes de qualquer transação ser efetivada na rede, elas são enviadas para uma fila pública de transações pendentes (*mempool*). Qualquer usuário da rede pode consultar uma transação inserida na *mempool* enquanto ela não for incluída em um bloco. Além da *mempool*, nós responsáveis pela construção do bloco (*builders*) podem oferecer filas de transações privadas (*private pools*) não nativas da rede Ethereum para usuários que desejarem ocultar suas transações durante este processo de efetivação [Heimbach et al. 2023]. Dessa forma, o nó construtor seleciona transações da *mempool* e da fila privada para compor um novo bloco. Os blocos mais lucrativos são encaminhados aos nós proponentes pelos nós que servem como ponte (*relays*). Por sua vez, os nós proponentes são responsáveis por aceitar os novos blocos criados e anexá-los à cadeia. Estas transações selecionadas são priorizadas pelo alto preço de *gas* e oportunidades MEV lucrativas, visto que parte das taxas pagas nas transações é direcionada para os próprios nós construtor e proponente, mas podem haver outros fatores [Weintraub et al. 2022]. Logo, um usuário da rede pode definir a posição e garantir a inserção da sua transação no bloco a partir do preço de *gas* que ele está disposto a pagar.

2.2. Máxima Extração de Valor em Blocos (MEV)

Maximal Extractable Value (MEV) é definido como a estratégia de maximizar o lucro por meio da análise histórica de transações ou por meio de transações pendentes [Lyu et al. 2022]. Normalmente, a maximização de lucro acontece pela ordem em que as transações ocorrem na rede. Nesse sentido, são consideradas atividades MEV qualquer estratégia que seleciona, ordena ou reajusta transações durante o processo de criação de um novo bloco [Daian et al. 2019].

Existem três principais práticas MEV que representam cerca de 90% das transações que ocorrem na rede Ethereum [Yang et al. 2022]. São elas: Arbitragem, Liquidação e Sanduíche. Arbitragem e Liquidação baseiam-se em monitorar a *mempool* ou analisar as transações do último bloco em busca de oportunidades MEV. Por outro lado, o Sanduíche baseia-se em identificar uma vítima em potencial para gerar transações que irão manipular o preço do ativo transacionado, fazendo com que a vítima possa ter prejuízo. Apesar de Arbitragem e Liquidação serem estratégias que regulam os preços de ativos em meio a diferentes corretoras, todas as estratégias MEV resultam no aumento das taxas para todos os usuários e no congestionamento da rede. Diferentes MEV *searchers* competem pelas mesmas posições em um bloco ou identificam a mesma vítima. As filas privadas foram implementadas para resolver estes problemas, porém vítimas que utilizam a *mempool* ainda podem ser atacadas pela estratégia de sanduíche.

2.3. Ataques Sanduíche

Os ataques sanduíche são considerados uma estratégia MEV baseados no conceito tradicional do mundo financeiro de *front-running* [Campos et al. 2024], na qual um usuário com visão privilegiada do sistema identifica um negócio promissor de outro usuário e o executa antecipadamente para benefício próprio. De acordo com a Figura 1, um atacante consegue executar um sanduíche a partir da inclusão de transações que irão anteceder (*front-run*) e suceder (*back-run*) uma transação-alvo, enviadas através de pacotes (*bundles*) em uma fila de transações privadas não nativas da rede Ethereum.

Uma característica dos pacotes é que eles podem incluir transações privadas e transações públicas, *i.e.* transações da *mempool*. O atacante então monitora a *mempool* em busca de uma vítima que transaciona um ativo em uma quantidade que vá gerar lucro. Em seguida, o atacante envia o pacote para a fila privada contendo as transações dele e a transação da vítima, configurando estratégias de manipulação do preço de *gas* para o nó construtor incluir o pacote nas posições do bloco que o satisfaçam. Para realizar este processo, os atacantes fazem uso de contratos inteligentes que monitoram as transações em busca de oportunidades de ataque. Estes contratos são também chamados de MEV *bots*, pois incluem a estratégia MEV para identificar transações em potencial.

3. Metodologia

Nossa metodologia quantifica os ataques sanduíche a partir da perspectiva dos nós construtores e atividades MEV. Nesse sentido, para quantificar o quão lucrativo um ataque sanduíche como atividade MEV pode ser para um nó construtor, nós simulamos a estratégia de identificar transações lucrativas e que compensam ser adicionadas na construção de um bloco através do preço de *gas*. Além disso, nós avaliamos os ataques utilizando técnicas de clusterização e grafos para identificar o relacionamento entre ocorrências de ataques e construtores.

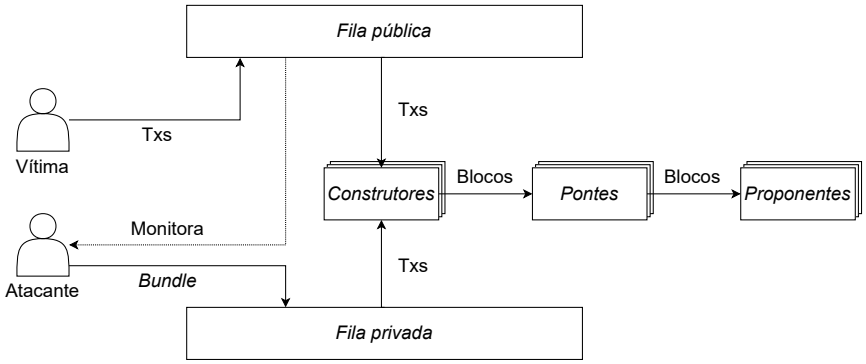


Figura 1. Arquitetura geral de ataques sanduíche sob o paradigma PBS.

3.1. Conjunto de Dados

Para medir o impacto dos ataques sanduíche sob a perspectiva do paradigma PBS, nós utilizamos o conjunto de dados inicialmente proposto pelos próprios autores em [Fontinele et al. 2024]. Utilizamos estes dados por representar os ataques que ocorreram após o primeiro ano do paradigma PBS. Conforme a Tabela 1, o *dataset* conta com 1.553.362 ataques sanduíche ocorridos do período de janeiro a dezembro de 2023. Estes ataques estão distribuídos em 1.008.607 blocos dos 2.599.105 blocos processados. A média de lucro destes ataques é de 3.202,82 dólares.

Tabela 1. Visão geral do conjunto de dados de ataques sanduíche.

Blockchain	Ethereum
Período	1 Jan, 2023 - 31 Dez, 2023
Quantidade de ataques	1.553.362
Blocos com ataques	1.008.607
Blocos processados	2.599.105
Média de lucro	USD 3.202,82
Fonte de coleta	APIs QuickNode/Alchemy

Todo o processo de coleta dos ataques foi realizado utilizando APIs externas provedoras de serviços para Web3 [QuickNode 2025], [Alchemy 2025]. Cada registro de ataque contém as transações do atacante e a transação da vítima, sendo que cada transação possui o índice referente à posição no bloco, o valor transacionado e o preço de *gas* estabelecido. Além das informações dos ataques, realizamos uma coleta de dados adicional para avaliarmos como as transações do atacante são lucrativas em relação às transações de um bloco. Este conjunto de dados adicional consiste em transações não relacionadas aos ataques, mas que foram incluídas nos blocos que possuem ataque.

A base de dados utilizada conta com ataques sanduíche coletados a partir do estabelecimento de heurísticas difundidas na literatura. Portanto, reconhecemos que toda a análise representa um limite inferior da quantidade real de ataques que ocorrem na rede Ethereum. Além disso, o processo de construção do conjunto de dados utiliza bases de

terceiros, que apesar de serem fontes utilizadas em outros trabalhos podem gerar falsos positivos.

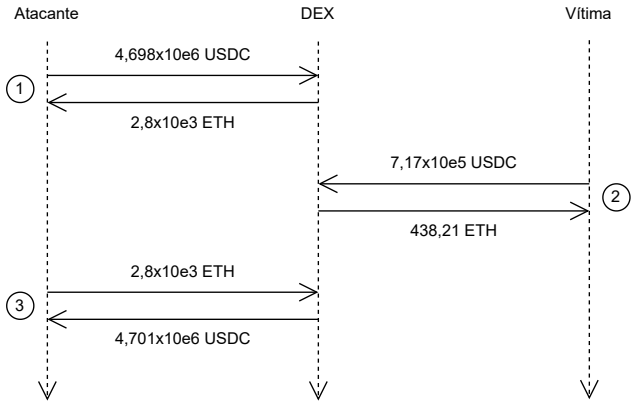


Figura 2. Fluxo de execução de um ataque sanduíche.

A Figura 2 mostra um ataque retirado do conjunto de dados que transaciona o *token* USD Coin (USDC), uma *stablecoin* que acompanha o dólar dos Estados Unidos e utiliza uma corretora decentralizada (DEX). Em (1) o atacante antecipa a compra da vítima adquirindo um grande volume do ativo, elevando o seu preço. Em seguida, em (2) a vítima executa sua compra e acaba encontrando o ativo já valorizado e, ao comprar, faz o preço subir ainda mais. Por fim, em (3) o atacante vende os ativos que comprou antes para obter lucro, enquanto o preço do ativo cai novamente. Dessa forma, a vítima paga mais caro do que deveria e sofre prejuízo, enquanto o atacante se beneficia da manipulação de preços.

3.2. Ranqueamento de transações

Para determinar o quão lucrativas são as transações de um ataque sanduíche para os nós construtores (*builders*), nós analisamos o os blocos da Ethereum contendo ataques detectados. Trabalhos anteriores, como em [Torres et al. 2021] e [Qin et al. 2021], observaram que analisar ataques sanduíche em um mesmo bloco já é suficiente, visto que no caso da rede Ethereum ataques sanduíche espalhados por mais de um bloco podem aumentar as chances do atacante obter prejuízo.

Além disso, nós consideramos nesta análise que o atacante utiliza apenas pacotes de tamanho 3 para realizar os ataques. Logo, cada ataque consiste em um pacote de transações com a seguinte estrutura: T_{A_1} como a primeira transação do atacante, T_V como a transação da vítima e T_{A_2} como a segunda transação do atacante. A partir desta definição, nós simulamos a ordenação das transações dentro do bloco, considerando a política de prioridade por preço do *gas* usada pelos construtores. Dessa forma, medimos a posição das transações do atacante em relação às demais transações do bloco. É importante notar que, diferentemente de [Torres et al. 2021], nós consideramos ataques sanduíche que possuem qualquer configuração no preço de *gas* das transações. O ataque sanduíche da maneira clássica possui a característica de que T_{A_1} possui um preço de *gas* maior do que T_{A_2}. Porém os dados analisados apontam que, com o novo paradigma PBS,

atacantes preferem configurar a última transação (T_{A_2}) como a que possui o preço de *gas* maior. Esta transação acaba funcionando como o identificador para o nó construtor sobre o quanto o atacante pagará por todo o pacote com o ataque. O atacante pode definir a sequência das transações do seu pacote. Dessa forma, os preços de *gas* em T_{A_1} e T_{A_2} são independentes entre si e também independentes da transação da vítima. Nesse sentido, para que o pacote seja aceito basta que o atacante pague um valor de *gas* suficiente somando T_{A_1} e T_{A_2}.

Como o nó construtor precisa incluir todas as transações de um pacote caso o mesmo seja selecionado, usuários podem decidir o preço de *gas* que está disposto a pagar em qualquer transação do pacote. Nesse sentido, o atacante utiliza a estratégia de definir o preço de *gas* apenas em T_{A_2}. Logo, para ranquear o quão lucrativo o ataque pode ser para o nó construtor, consideramos o caso em que T_{A_2} é a primeira transação do bloco após a ordenação. Para casos em que o atacante não utiliza esta estratégia, *i.e.* o preço de *gas* está distribuído entre T_{A_1} e T_{A_2}, nós ranqueamos as transações considerando uma transação fictícia T_{A_f} que possui como preço de *gas* a média do preço de *gas* das duas transações do atacante $gas_{T_{A_f}} = (gas_{T_{A_1}} + gas_{T_{A_2}})/2$.

Neste novo modelo de construção de blocos, o construtor não pode definir a ordem de transações de um pacote, mas ainda pode definir a ordem dos pacotes em relação às demais transações do bloco [Li et al. 2023]. Geralmente, as transações de um ataque sanduíche consomem a mesma quantidade de *gas*, porém apenas as taxas de *gas* de T_{A_1} e T_{A_2} podem ser manipuladas pelo atacante. Isso porque o atacante não pode modificar a transação da vítima, apenas incluí-la em seu pacote. Por tais motivos decidimos realizar uma média entre as taxas pagas em T_{A_1} e T_{A_2}, pois o nó construtor terá que executar duas transações e ser pago apenas por uma delas.

Para o ranqueamento das transações, nós consideramos os pacotes de atacante com tamanho 3. Consideramos este tamanho, pois ele reflete a quantidade mínima de transações que o atacante precisa para realizar o ataque. Porém, em um mesmo pacote, o atacante pode incluir outras transações que conterão o preço de *gas* distribuído. Em trabalhos como [Chi et al. 2024], os autores apontam casos que em um mesmo pacote um atacante pode executar diversos ataques sanduíche.

3.3. Clusterização de ataques

Para agrupar os ataques sanduíche considerando o relacionamento com nós construtores, foram utilizados apenas ataques realizados através de contratos de atacantes (99,83% da base de dados). Isso significa que os ataques em que o atacante utiliza apenas o contrato da corretora não foram utilizados na criação dos *clusters* (0,17% da base de dados). A partir destes ataques, nós agrupamos através de um grafo os endereços de contas de atacantes com os contratos utilizados por eles, conforme a Figura 3a.

Cada vértice do grafo representa os endereços de conta ou contratos de atacantes, e cada aresta conecta um atacante ao contrato que ele utilizou para realizar ataques sanduíche. Dessa maneira, sempre que um novo ataque é lido da base de dados, é criada uma aresta com pesos entre o contrato e/ou endereço de conta. Os pesos das arestas representam métricas como: custo monetário para realização do ataque, lucro obtido, quantidade de ataques realizados por estes endereços, quantidade de contas e contratos envolvidos. Ao final, o grafo construído possuirá diversos componentes desconectados,

como mostra a Figura 3b. Cada componente é considerado como um *cluster*, pois assumimos que os atacantes não compartilham suas contas e seus contratos com outros atacantes.

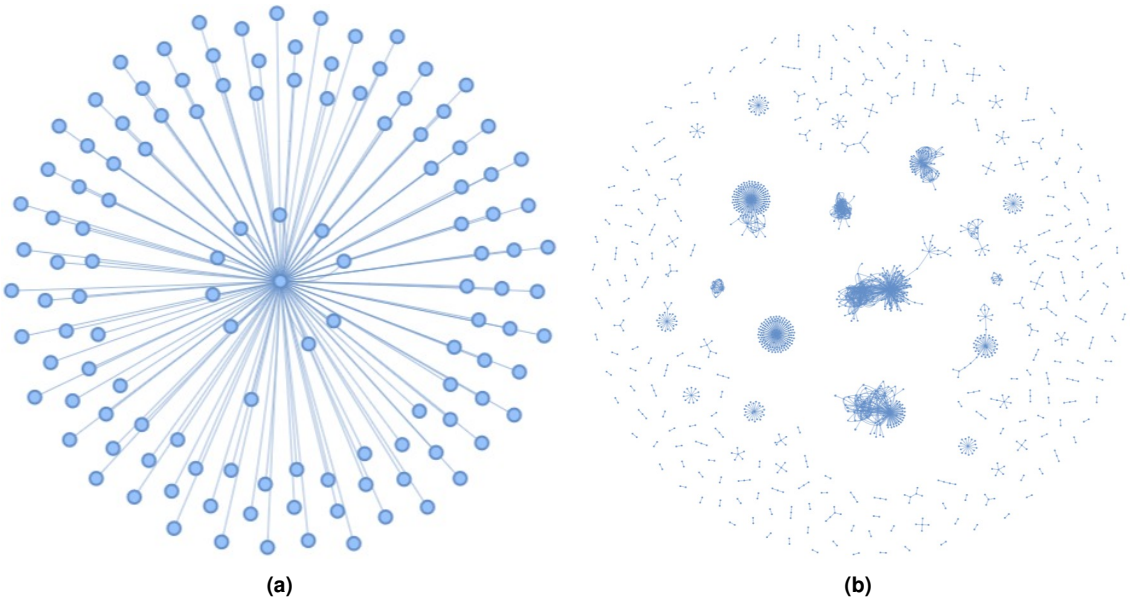


Figura 3. *Clusters* gerados como grafos de contratos e contas.

Para a construção da clusterização, consideramos que os atacantes não compartilham seus contratos com outras contas, sendo que na prática isto pode não ocorrer. A metodologia adotada neste caso é uma adaptação do método utilizado em [Torres et al. 2021]. Os autores relacionam um endereço de conta e um contrato inteligente para representar a atividade do atacante. Porém, a abordagem baseia-se em duas suposições. A primeira é que os atacantes desenvolvem seus próprios contratos. A segunda é que os atacantes não publicam o código-fonte de seus contratos, para não compartilhar os detalhes de implementação com concorrentes. A partir destas suposições o grafo de relacionamento pode ser criado.

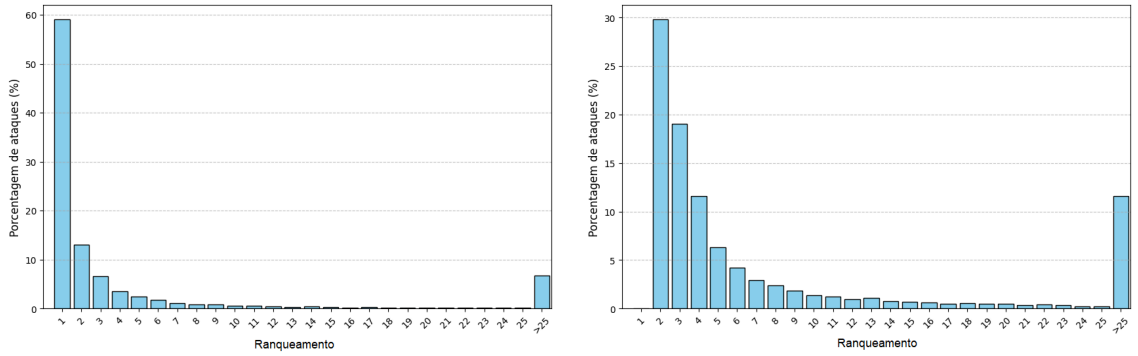
4. Resultados

Nesta seção nós apresentamos nossos resultados. Em primeiro lugar, apresentamos os ranqueamentos de transações dos blocos atacados para quantificar o quão vantajoso o ataque sanduíche acaba sendo para os nós mantenedores da rede. Além disso, apresentamos como os grupos de ataques acabam selecionando nós construtores específicos da rede para realização dos ataques.

4.1. Análise dos ranqueamentos

O ranqueamento das transações leva em consideração as transações do atacante e as demais transações de um bloco, conforme a Seção 3.2. A Figura 4a apresenta o resultado do ranqueamento da transação T_{A_2} em relação ao restante do bloco. É possível notar que em cerca de 60% dos casos a transação do atacante possuirá o índice 1 do bloco, *i.e.* será a primeira transação incluída. Este resultado aponta como T_{A_2} frequentemente aparece entre as transações com maior preço do *gas*, indicando que os atacantes ajustam essa taxa

para garantir que seu ataque seja executado. Como os construtores priorizam transações mais lucrativas, isso sugere que ataques sanduíche são estrategicamente estruturados para maximizar a extração de valor.



(a) (b)

Figura 4. (a): Ranqueamento da transação T_{A_2} em relação ao bloco. (b): Ranqueamento da transação T_{A_f} em relação ao bloco.

Na Figura 4b, o ranqueamento da transação T_{A_f} em relação as outras transações do bloco segue comportamento semelhante da abordagem anterior. O gráfico aponta como a transação T_{A_1} apenas complementa o preço de um pacote para garantir o sucesso do ataque, indicando diferenças do trabalho em [Torres et al. 2021]. A frequência de cerca de 30% em que as transações do atacante atingem a segunda transação do bloco representa o quanto o ataque sanduíche como atividade MEV pode influenciar as taxas da rede. A transação T_{A_2}, de acordo com o ranqueamento, funciona como a última transação do atacante que conterá o pagamento por todo o pacote.

4.2. Comportamento dos atacantes

Nós inicializamos a análise obtendo uma visão geral dos nós construtores responsáveis pelos blocos com ataques sanduíche seguindo a metodologia da Seção 3.3. A Tabela 2 detalha esta relação. Para cada nó construtor é associado um identificador (ID), cada um possui um endereço de conta associado e a quantidade de blocos construídos. É possível observar que apenas um pequeno número de construtores (4) são responsáveis por mais por processar 70% dos blocos com ataques.

Tabela 2. Top 10 maiores construtores do conjunto de dados, identificados (ID) em ordem decrescente por número de blocos atacados.

ID	Endereço	Blocos	Nome
1	0x95222290dd7278aa3ddd389cc1e1d165cc4bafe5	227.265	beaverbuild
201	0x1f9090aae28b8a3dceadf281b0f12828e676c326	202.065	rsync-builder.eth
5	0x690b9a9e9aa1c9db991c7721a92d351db4fac990	152.889	builder0x69
6	0xdafea492d9c6733ae3d56b7ed1adb60692c98bc5	120.857	Flashbots: Builder
1067	0x4838b106fce9647bdf1e7877bf73ce8b0bad5f97	61.686	Titan Builder
11	0x388c818ca8b9251b393131c08a736a67ccb19297	53.912	Lido: Execution Layer Rewards Vault
27	0xbaf6dc2e647aeb6f510f9e318856a1bcd66c5e19	19.792	MEV Builder
10	0x4675c7e5baafbffbca748158becba61ef3b0a263	19.506	Coinbase: MEV Builder
1796	0x5124fcc2b3f99f571ad67d075643c743f38f1c34	10.955	Faith Builder
9	0xbd3afb0bb76683ecb4225f9dbc91f998713c3b01	9.893	BuildAI.net

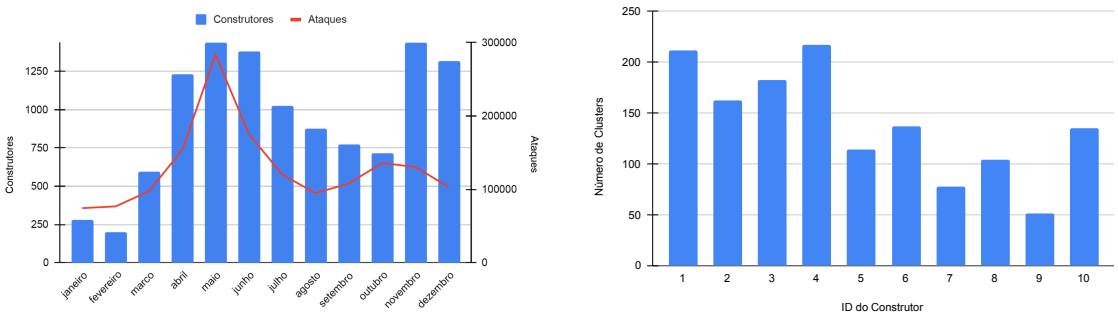
Após o agrupamento de contas e contratos inteligentes, nós conseguimos obter uma visão geral dos *clusters* obtidos, conforme a Tabela 3. Nesse sentido, foram encontrados 252 *clusters* com ataques sanduíche. Os ataques são realizados por poucos grupos organizados (433 contas diferentes). Além disso, em 88 *clusters* existem atacantes que realizaram ataques utilizando mais de um contrato inteligente, variando a estratégia que cada contrato utiliza para executar o ataque sanduíche. Os atacantes não compartilham contratos inteligentes entre si, mas alguns optam por utilizar mais de um contrato inteligente (88 *clusters*). Cada *cluster* está associado a pelo menos um nó construtor.

Tabela 3. Visão geral do método de clusterização realizado.

Informações	Quantidade
Bots e atacantes	1.524
Contas de atacantes	433
Contratos de *bots*	1.091
Clusters	252
Clusters com mais de um *bot*	88

Ao relacionarmos os *clusters* gerados com os nós construtores mapeados, é possível observar como os atacantes acabam tendo suas transações selecionadas pelos principais construtores responsáveis pela criação dos blocos da rede. Na Figura 5a é possível observar a curva de tendência de ataques e nós construtores em cada mês de 2023. Exceto pelo final do ano, nos outros meses a quantidade de ataques (representada pela linha vermelha) acompanha a quantidade de nós construtores responsáveis pela inclusão de transações nos blocos da rede. A Ethereum adotou o paradigma PBS em setembro de 2022, o que justifica a crescente de ataques e construtores no começo do ano de 2023. Já a Figura 5b mostra como os 4 principais nós construtores se relacionam com mais da metade dos *clusters* criados. Além disso, os construtores 1 e 6 se relacionam com praticamente todos os *clusters* de ataques sanduíche. Este resultado aponta como os atacantes utilizam o recurso dos pacotes de transações para definir preços de *gas* irrecusáveis para os nós construtores que seguem o paradigma PBS, *i.e.* o nó construtor constrói o bloco levando em consideração o lucro que pode ser obtido pelas taxas das transações.

A Tabela 4 apresenta a distribuição acumulada para os ataques sanduíche com base nos *clusters*. Podemos observar que em média os *clusters* possuem 6.153,68 ataques executados. Entretanto, um único atacante conseguiu executar 900.059 ataques (*max*). Em relação ao lucro obtido, cerca de 50% dos *clusters* obtém prejuízo, atingindo um valor máximo de perda de mais de 809 milhões de dólares. Ainda assim, o *cluster* mais lucrativo conseguiu atingir um lucro acumulado de mais de 4 bilhões de dólares. Semelhantemente, em relação ao custo gasto para realizar as transações, o maior *cluster* gastou mais de 132 milhões dólares em taxas de transações, mostrando como os ataques sanduíche influenciam as taxas da rede e o paradigma de criação de blocos. Também é possível observar que na maioria dos *clusters* o atacante opta por utilizar apenas uma conta e um contrato inteligente, mas há casos de uso de 42 contas diferentes e 143 bots diferentes para realizar ataques.



(a) (b)

Figura 5. (a): Distribuição de nós construtores e ataques por mês. (b): Quantidade de *clusters* por nó construtor.

Tabela 4. Distribuição dos *clusters* de ataques sanduíche.

Distribuição	Custo (USD)	Lucro (USD)	Ataques	Contas	Contratos
mean	784.204,34	19.742.931,84	6.153,68	1,72	4,33
std.	8.419.310,61	305.501.049,35	58.375,00	3,51	15,63
min.	3,37	-809.509.070,44	1,00	1,00	1,00
25%	114,32	-485,89	5,00	1,00	1,00
50%	812,38	-38,30	30,50	1,00	1,00
75%	9.301,24	0,34	291,00	1,00	2,00
max.	132.318.131,39	4.693.998.256,49	900.059,00	42,00	143,00

5. Trabalhos Relacionados

Nesta seção nós discutimos os trabalhos existentes referentes aos métodos de detecção de ataque sanduíche, as medições e análises de transações da rede Ethereum, e a identificação de atividades MEV no paradigma PBS e em blockchains com filas privadas.

Detecção de ataques sanduíche. Os trabalhos existentes na literatura que abordam a detecção de ataques sanduíche baseiam-se em heurísticas e análise de atividades na rede, como em [Torres et al. 2021], [Varun et al. 2022] e [Fontinele et al. 2024]. Existem duas razões principais pelas quais os ataques sanduíche são possíveis em blockchains públicas como a Ethereum: A falta de confidencialidade das transações e a capacidade dos nós construtores ordenarem as transações arbitrariamente. A partir destas características, os autores em [Torres et al. 2021] apresentam as principais heurísticas para detecção deste tipo de ataque. Já em [Varun et al. 2022], os autores definem um modelo de aprendizado de máquina para aprender padrões a partir do conjunto de dados proposto em [Torres et al. 2021] e predizer a ocorrência de ataques sanduíche. Em [Fontinele et al. 2024], os autores estendem as heurísticas da literatura, apontando a evolução dos ataques sanduíche de acordo com as atualizações da rede Ethereum, sendo a principal referência para este trabalho.

Análise de transações. Identificar a frequência e realizar medições de todo o ciclo de vida de uma transação é uma atividade importante em ambientes descentralizados como redes blockchains. Trabalhos como [Heimbach and Wattenhofer 2022],

[Weintraub et al. 2022] e [Chi et al. 2024] destacam como o ataque sanduíche se comporta como atividade MEV. Em [Heimbach and Wattenhofer 2022] os autores utilizam a teoria de jogos para identificar como o ajuste de preço de ativos ao longo de diferentes transações pode suprimir certos ataques. Já em [Weintraub et al. 2022], os autores analisam o fluxo das transações MEV sob a utilização das filas privadas, identificando como estas filas podem impactar na centralização da rede e beneficiar mineradores. Por fim, os autores em [Chi et al. 2024] medem duas estratégias MEV, arbitragem e ataque sanduíche. Os autores constataram a necessidade de filas privadas para reduzir o lucro das atividades MEV, mas apontam como estas filas permitem outras variações destas estratégias. Nosso trabalho estendem essas análises medindo como os ataques sanduíche podem ser vantajosos para os nós da rede a partir da manipulação do preço de *gas*.

Identificação de MEVs. As atividades MEVs se adaptam de acordo com os recursos da rede blockchain em questão. Alguns trabalhos como [Heimbach et al. 2023] e [Ferreira Torres et al. 2024] abordam como as implementações das redes blockchains atuais impactam as estratégias MEV. Os autores em [Heimbach et al. 2023] estudam a adoção do paradigma PBS na rede Ethereum, apontando questões como a centralização da rede, censura e omissão de transações. Já em [Ferreira Torres et al. 2024] foi conduzido um estudo de MEV em blockchains de camada 2 (chamada de *rollups*) ao longo de um período de quase três anos. O trabalho aponta que neste tipo de rede onde há a ausência da fila pública de transações pendentes os ataques sanduíche não podem ser realizados. Além disso, os autores apontam como as tecnologias de Prova de Conhecimento Zero (*Zero-Knowledge Proof*) auxiliam na privacidade das transações destas redes.

6. Conclusão

Neste trabalho, nós conduzimos uma análise compreensiva dos ataques sanduíche sob a perspectiva do novo Paradigma de Separação Proponente-Construtor (*Proposer-Builder Separation* - PBS) da rede Ethereum. Nossas análises demonstram como os atacantes utilizam deste novo paradigma para garantir que o ataque sanduíche ocorra com sucesso. Por meio da manipulação do preço de *gas* e da criação dos pacotes de transações, os atacantes garantem que o ataque seja incluído na construção de um novo bloco. Os nós construtores selecionam transações levando em consideração as transações que irão maximizar o lucro pela construção do bloco.

Nossos resultados apontam que o ataque sanduíche como atividade MEV pode influenciar negativamente as taxas da rede. Por meio da análise de grupos de atacantes, foi encontrado que os ataques consomem em média mais de 780 mil dólares em taxas de transação. Este valor indica como o paradigma PBS juntamente com o mecanismo de consenso *Proof-of-Stake* podem ser desestabilizados pelos ataques sanduíche. Além disso, pelo ranqueamento das transações dos blocos com ataques, os pacotes com ataques sanduíche majoritariamente contém a transação com maior preço de *gas* do bloco. Nesse sentido, pelo fato do paradigma PBS e as filas privadas permitirem o uso de menos *gas* para criar transações e garantirem que se um pacote não é selecionado as transações não são executadas, os atacantes aproveitam para alocar mais valor no pagamento das taxas com o intuito de favorecem os nós construtores.

Por outro lado, o ataque sanduíche como atividade MEV depende da transação da vítima permanecer na *mempool* da rede Ethereum para ser identificada pelo atacante. Com

o paradigma PBS e as plataformas de MEV, a rede Ethereum conta com diversas filas de transações privadas. Trabalhos como [Heimbach et al. 2023] discutem questões práticas como confiabilidade e centralização da rede pela utilização de filas privadas. Porém, em [Ferreira Torres et al. 2024] os autores apontam como em blockchains que não possuem a *mempool* os ataques sanduíche não acontecem da mesma maneira. Nesse sentido, como direções futuras os ataques sanduíche na rede Ethereum podem ser analisados da perspectiva da vítima, para entender como mesmo após a criação das filas privadas as vítimas optam por correr o risco de terem suas transações atacadas.

7. Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001. Fapemig, CNPq e RNP também apoiaram parcialmente a execução deste trabalho.

Referências

Alchemy (2025). Alchemy documentation: Web3 development made easy. Website. Accessed: 2025-01-06.

Buterin, V. et al. (2013). Ethereum white paper. *GitHub repository*, 1:22–23.

Campos, J. N., Mendonça, R. D., Fontinele, A., de Carvalho, L. H. S., Oliveira, I. R., Cardoso, Í. W. F., Coelho, R., Freitas, A. E. S., Gonçalves, G. D., Nacif, J. A. M., and Vieira, A. B. (2024). Finanças descentralizadas em redes blockchain: Perspectivas sobre pesquisa e inovação em aplicações, interoperabilidade e segurança. In *Jornada de Atualização em Informática 2024*, volume 44 of *Congresso da Sociedade Brasileira de Computação*, pages 7–56. SBC, Porto Alegre, 43nd edition.

Chi, T., He, N., Hu, X., and Wang, H. (2024). Remeasuring the arbitrage and sandwich attacks of maximal extractable value in ethereum. *arXiv preprint arXiv:2405.17944*.

Daian, P., Goldfeder, S., Kell, T., Li, Y., Zhao, X., Bentov, I., Breidenbach, L., and Juels, A. (2019). Flash boys 2.0: Frontrunning, transaction reordering, and consensus instability in decentralized exchanges.

DeFiLlama (2024). Defillama: The leading decentralized finance (defi) analytics platform. Website. Accessed: 2024-12-27.

Ferreira Torres, C., Mamuti, A., Weintraub, B., Nita-Rotaru, C., and Shinde, S. (2024). Rolling in the shadows: Analyzing the extraction of mev across layer-2 rollups. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 2591–2605.

Fontinele, A., Campos, J., Oliveira, I., Gonçalves, G., Nacif, J., Vieira, A., and Soares, A. (2024). Análise de ataques sanduíche sob as transações da blockchain ethereum. In *Anais do XLII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 728–741, Porto Alegre, RS, Brasil. SBC.

Heimbach, L., Kiffer, L., Ferreira Torres, C., and Wattenhofer, R. (2023). Ethereum's proposer-builder separation: Promises and realities. In *Proceedings of the 2023 ACM on Internet Measurement Conference*, pages 406–420.

Heimbach, L. and Wattenhofer, R. (2022). Eliminating sandwich attacks with the help of game theory. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '22. ACM.

Li, Z., Li, J., He, Z., Luo, X., Wang, T., Ni, X., Yang, W., Chen, X., and Chen, T. (2023). Demystifying defi mev activities in flashbots bundle. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 165–179.

Lyu, X., Zhang, M., Zhang, X., Niu, J., Zhang, Y., and Lin, Z. (2022). An empirical study on ethereum private transactions and the security implications. *arXiv preprint arXiv:2208.02858*.

Qin, K., Zhou, L., and Gervais, A. (2021). Quantifying blockchain extractable value: How dark is the forest? *CoRR*, abs/2101.05511.

QuickNode (2025). Quicknode documentation: Blockchain infrastructure for developers. Website. Accessed: 2024-12-27.

Torres, C. F., Camino, R., and State, R. (2021). Frontrunner jones and the raiders of the dark forest: An empirical study of frontrunning on the ethereum blockchain. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 1343–1359. USENIX Association.

Varun, M., Palanisamy, B., and Sural, S. (2022). Mitigating frontrunning attacks in ethereum. In *Proceedings of the Fourth ACM International Symposium on Blockchain and Secure Critical Infrastructure*, BSCI '22, page 115–124, New York, NY, USA. Association for Computing Machinery.

Weintraub, B., Torres, C. F., Nita-Rotaru, C., and State, R. (2022). A flash(bot) in the pan: measuring maximal extractable value in private pools. In *Proceedings of the 22nd ACM Internet Measurement Conference*, IMC '22, page 458–471, New York, NY, USA. Association for Computing Machinery.

Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger eip-150 revision.

Yang, S., Zhang, F., Huang, K., Chen, X., Yang, Y., and Zhu, F. (2022). Sok: Mev countermeasures: Theory and practice. *arXiv preprint arXiv:2212.05111*.

Zhou, L., Qin, K., Torres, C. F., Le, D. V., and Gervais, A. (2021). High-frequency trading on decentralized on-chain exchanges. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 428–445. IEEE.