DLT PERUGIA 2025

Approaches to MEV in DAG-based DLTs

Exploring vulnerabilities and mitigation strategies in next-generation ledgers

MEV · DAG-based Ledgers · Transaction & Block Ordering · Fairness · Fully Homomorphic Encryption

Giulio Jan Valentini — ExaGroup.xyz/Università di Torino
Carla Mascia — DataKrypto
Andrea Bracciali — Università di Torino

MEV: THREAT TO FAIRNESS

Maximal Extractable Value (MEV) is the value extracted by **manipulating the ordering of transactions** in distributed ledgers. Its impact on the ecosystem is wide:

- **Higher transaction costs** for all users.
- Frequent **unfavorable trade executions**, undermining trust and fairness.
- **Network Congestions** as it becomes flooded with MEV-driven attempts.

It is not always easy to identify MEV transactions, and **estimates of the total value extracted vary widely**. Current estimates suggest cumulative MEV is approaching **$2 billion**, with the majority originating from Ethereum.

TOXIC & NEUTRAL MEV

MEV is **not always easy to identify** and it is often unclear when it benefits the ecosystem or when it is harmful, leaving plenty of **gray areas.**

1. **Neutral MEV,** such as **arbitrage** and **liquidations**, can **enhance market efficiency** and contribute to the overall health of the ecosystem.

2. **Toxic MEV,** such as **sandwich** attacks, **frontrunning**, and **backrunning**, exploits users and **undermines trust** in the system.

Being able to accurately identify MEV and distinguish between its neutral and toxic forms is **essential for designing effective mitigations.**

MEV IN ETHEREUM

Ethereum's mempool is where all *submitted-but-not-yet-finalized* transactions are visible to everyone. This transparency exposes pending transactions to MEV strategies and has led to **gas bidding wars** among participants. Several mechanisms have been adopted to reduce the negative externalities:

- **EIP-1559:** introduced a **base fee plus tip**, reducing fee volatility but leaving MEV opportunities intact.
- MEV-Geth and MEV-Boost by Flashbots introduced **private bundles** so that MEV transaction could reach the validator while bypassing the mempool, reducing congestions and bidding wars.
- **Proposer–builder separation (PBS)**, also by Flashbots, creates **off-chain markets** for block building and MEV transactions, further reducing congestions and bidding wars.

However, these also increased **reliance on relays and raised centralization concerns.**

MEV IN SOLANA

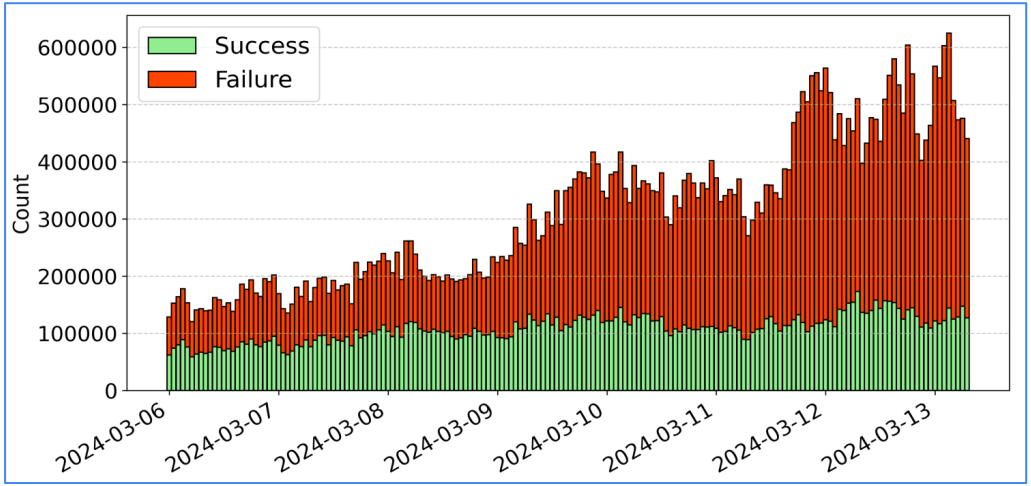
Solana offers a different architecture compared to Ethereum:

- **No public mempool:** pending transactions are not visible.
- **High throughput**: very short reaction time for MEV bots.
- **Low fees**: frontrunning through gas bidding wars is not effective.

Nonetheless, MEV still **emerged** and adapted into a **“spray-and-pray”** strategy, flooding the network with speculative spam.

More than **half of Solana’s traffic consists of failed MEV attempts**.

Jito Labs introduced MEV auctions to reduce spam, but raised new centralization concerns.



During memecoin mania, 75% of the transaction on Solana failed

Image source: Zheng et al., "Why Does My Transaction Fail? A First Look at Failed Transactions on the Solana Blockchain" (2025)

INTERIM TAKEAWAYS

1. MEV seems to be **structural** across decentralized ledgers and **adapts to each architecture**.
2. As long as there is **transaction volume there is an incentive** for MEV to emerge, and it will likely find its way.
3. On Ethereum it leverages **mempool transparency** leading to **gas wars** and **off-chain MEV auctions.**
4. Solana has **no mempool**, but **low fees** encouraged spam and "**spray-and-pray**" strategies.
5. Mitigations have **reduced congestion** but only partially reduced MEV, while introducing new **centralization risks**.

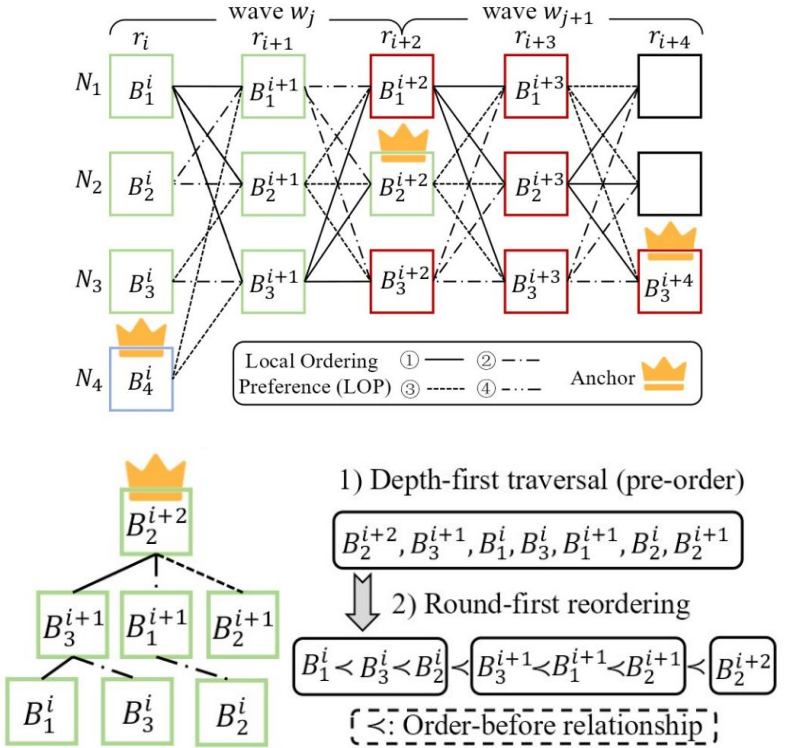
MEV IN DAGs

Directed Acyclic Graph (DAG) systems allow **multiple blocks to be proposed concurrently,** making **single-block transaction reordering extremely difficult** or impossible.

Traditional MEV attacks are blunted, since transaction reordering is not feasible.

However, blocks still need to be ordered, since their transactions cannot execute simultaneously the causal order of transactions has to be maintained.

Manipulating this consensus process makes block ordering itself the new attack surface.



Block ordering process in DAGs

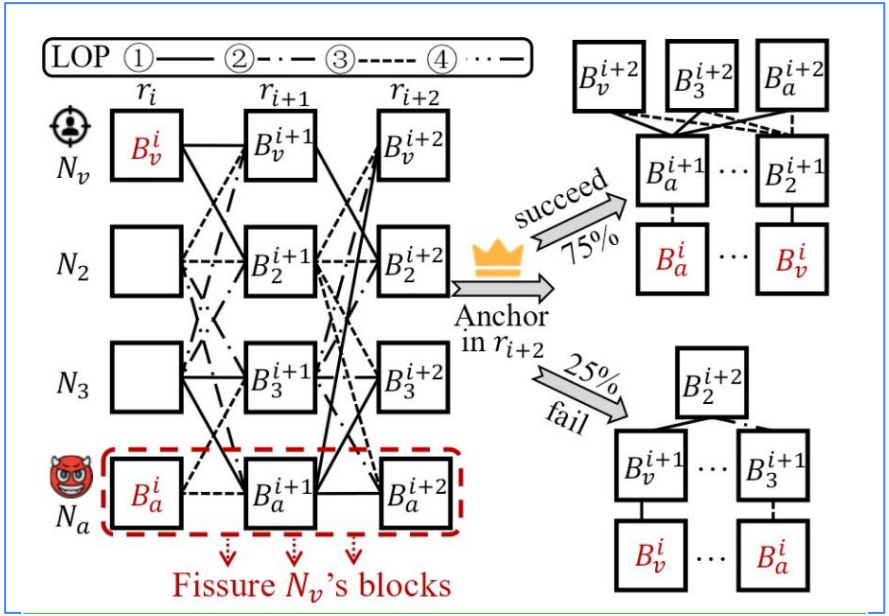
Image source: Zhang, J., & Kate, A. (2024). *No Fish Is Too Big for Flash Boys! Frontrunning on DAG-based Blockchains*.

BLOCK REORDERING ATTACKS

New block reordering attacks are probabilistic: so attackers rely on statistical advantage rather than guaranteed ordering.

- **Fissure attacks**: adversary **avoids linking to a victim's block** to delay its ordering. Less connections effectively is lower ordering priority.
- **Speculative attacks**: **multiple candidate blocks** are created, and **only the most favorable** one is revealed.
- **Sluggish attacks**: attacker **delays block proposals** to defer the victim's transaction to the next round.

Success rates ~80–90% depend on consensus variant (Zhang & Kate, 2024).



A schematic representation of the Fissure Attack

Image source: Zhang, J., & Kate, A. (2024). *No Fish Is Too Big for Flash Boys! Frontrunning on DAG-based Blockchains*.

CONSENSUS IN HEDERA HASHGRAPH

Hedera Hashgraph, a DAG-based distributed ledger, employs a unique consensus mechanism grounded in *gossip-about-gossip* and *virtual voting*.

To mitigate MEV attacks, Hedera assigns each transaction a **timestamp based on the median of the times** reported by a supermajority (i.e., more than two-thirds) of nodes that witnessed the transaction.

This timestamp-based mitigation mechanism relies on several assumptions:

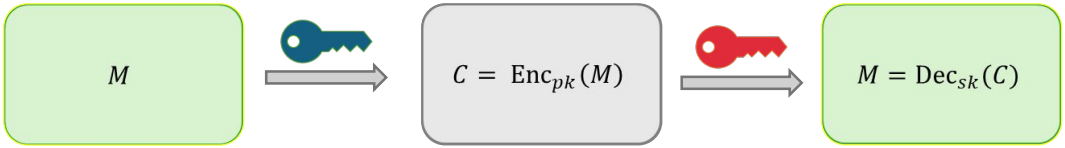
1. participating nodes report honest and accurate timestamps,
2. the dissemination latency of transaction messages is relatively consistent across nodes,
3. timestamps cannot be predicted or influenced by adversaries.

To further reduce the risk of MEV exploitation, we propose a new enhancement:
encrypting the timestamps using Homomorphic Encryption (HE).

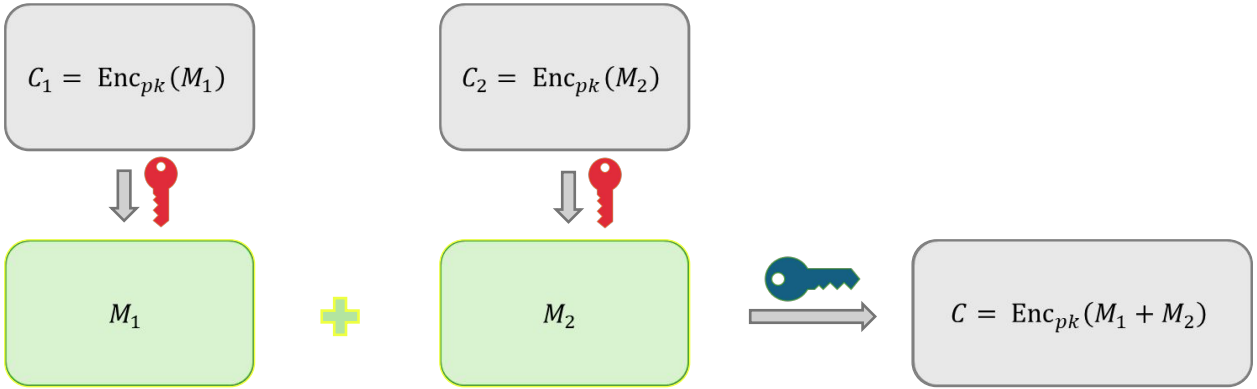
PUBLIC-KEY CRYPTOGRAPHY

pk = public key

sk = secret key



But, to perform the encryption of M_1+ M_2:



HOMOMORPHIC ENCRYPTION (HE)

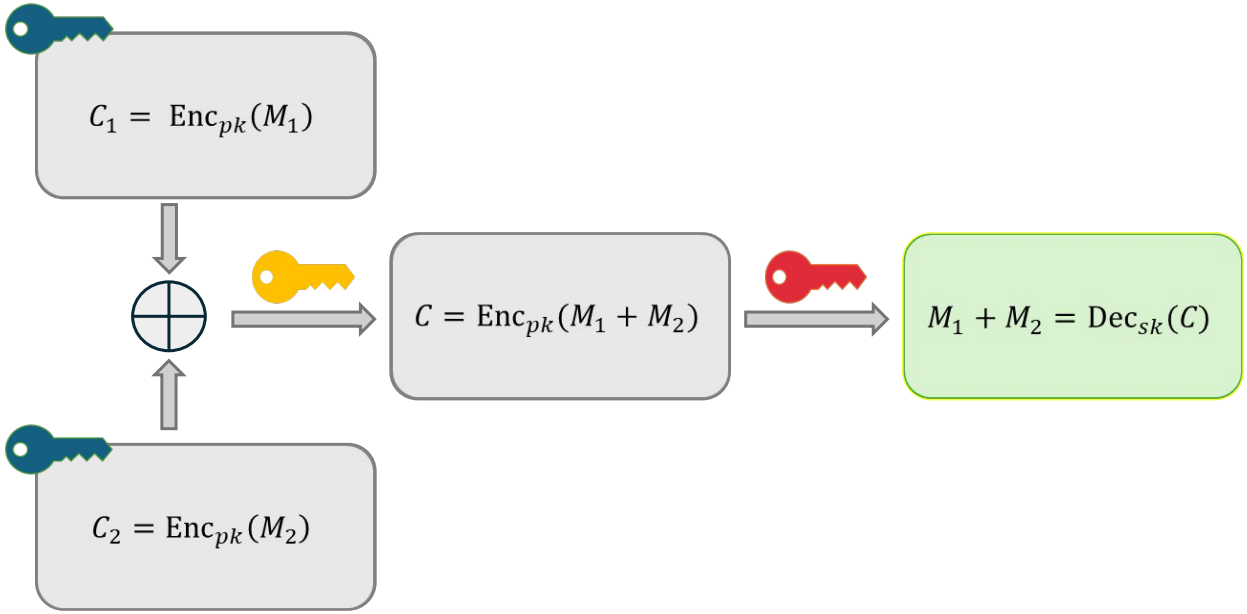
pk = public key

sk = secret key

evk = evaluation key

1. Encryption

3. Decryption



2. Homomorphic Sum

MULTI-PARTY THRESHOLD HE

pk = shared public key

sk_i = secret key of i-th party

evk = evaluation key

1. Encryption

$C_1 = \mathrm{Enc}_{pk}(M_1)$

$C_2 = \mathrm{Enc}_{pk}(M_2)$

2. Homomorphic Sum

$C = \mathrm{Enc}_{pk}(M_1 + M_2)$

3. Partial Decryption

$N_1 = \mathrm{Dec}_{sk_1}(C)$

$N_2 = \mathrm{Dec}_{sk_2}(C)$

$N_3 = \mathrm{Dec}_{sk_3}(C)$

Assuming threshold ≥ 3.

4. Shares Aggregation

Combiner

$M_1 + M_2$

OUR PROPOSED PROTOCOL

1. **Threshold Key Generation**:
   Nodes collaboratively execute a threshold key generation protocol, producing:
   a. A public encryption key *pk* and an associated evaluation key *evk* for the homomorphic median;
   b. Secret key shares ($sk_1, \ldots, sk_N$) distributed among the nodes.

2. **Encrypted Timestamp Submission**:
   Each node *i* encrypts its locally observed timestamp t_i as $c_i = \text{Enc}_{pk}(t_i)$.

3. **Gossip and Collection**:
   The encrypted timestamps $\{c_i\}$ are propagated via the gossip protocol and incorporated into the DAG.

4. **Homomorphic Median Computation**:
   Once an event is deemed famous, it is computed $c_{med} = \text{Eval}_{evk}(\text{Median}, \{c_i\})$.

5. **Threshold Decryption and Timestamp Assignment**:
   Each node computes a partial decryption share d_j of c_{med}. Once *t* valid shares are collected, the final timestamp is recovered: $t_{med} = \text{Combine}(d_1, \ldots, d_t)$ and is assigned as the consensus timestamp.

SECURITY PROPERTIES

- **Data confidentiality**: Raw timestamps are never revealed to any node before the median timestamp has been computed. Indeed:
  - thanks to the semantic security of HE, the encryption of identical timestamps yields different ciphertexts due to randomized encryption;
  - nodes are only given access to the evaluation key necessary to compute the median. They do not possess general-purpose evaluation keys.

- **Collusion resistance**: Decryption is impossible unless a threshold number of parties cooperate, mitigating insider threats.

- **Deterministic ordering**: The consensus timestamp remains unique and reproducible despite encryption.

CONCLUSIONS

- MEV has repeatedly emerged across different ledger architectures, **adapting to the available incentives and mechanisms.**
- In DAG-based systems, MEV does not vanish but might shift from transaction-level manipulation to block-ordering strategies.
- Our contribution proposes a **mitigation mechanism** based on Fully Homomorphic Encryption (FHE) to **enhance timestamp confidentiality**.
- This approach provides confidentiality of **ordering information**, supports **fairness in consensus**, and **strengthens collusion resistance** among participants.

References

- Buterin, V., et al. (2021). *EIP-1559: Fee Market Change for ETH 1.0 Chain*. Ethereum Improvement Proposal 1559. https://eips.ethereum.org/EIPS/eip-1559
- ZeroMEV. (2025). Definition of Terms. https://info.zeromev.org/terms.html (accessed May 28, 2025).
- Flashbots. (2021). *Flashbots: Frontrunning the MEV Crisis*. Ethereum Research. Retrieved from https://ethresear.ch/t/flashbots-frontrunning-the-mev-crisis/8251
- Flashbots. (2025). Flashbots Auction Overview. https://docs.flashbots.net/flashbots-auction/ (accessed May 28, 2025).
- Zheng, X., Wan, Z., Lo, D., Xie, D., & Yang, X. (2025). *Why Does My Transaction Fail? A First Look at Failed Transactions on the Solana Blockchain*. SIGMOD ’23.
- Zhang, J., & Kate, A. (2024). *No Fish Is Too Big for Flash Boys! Frontrunning on DAG-based Blockchains*.