

# Encrypted Mempool

최원재

2024. 12. 3



# Table of Contents

---

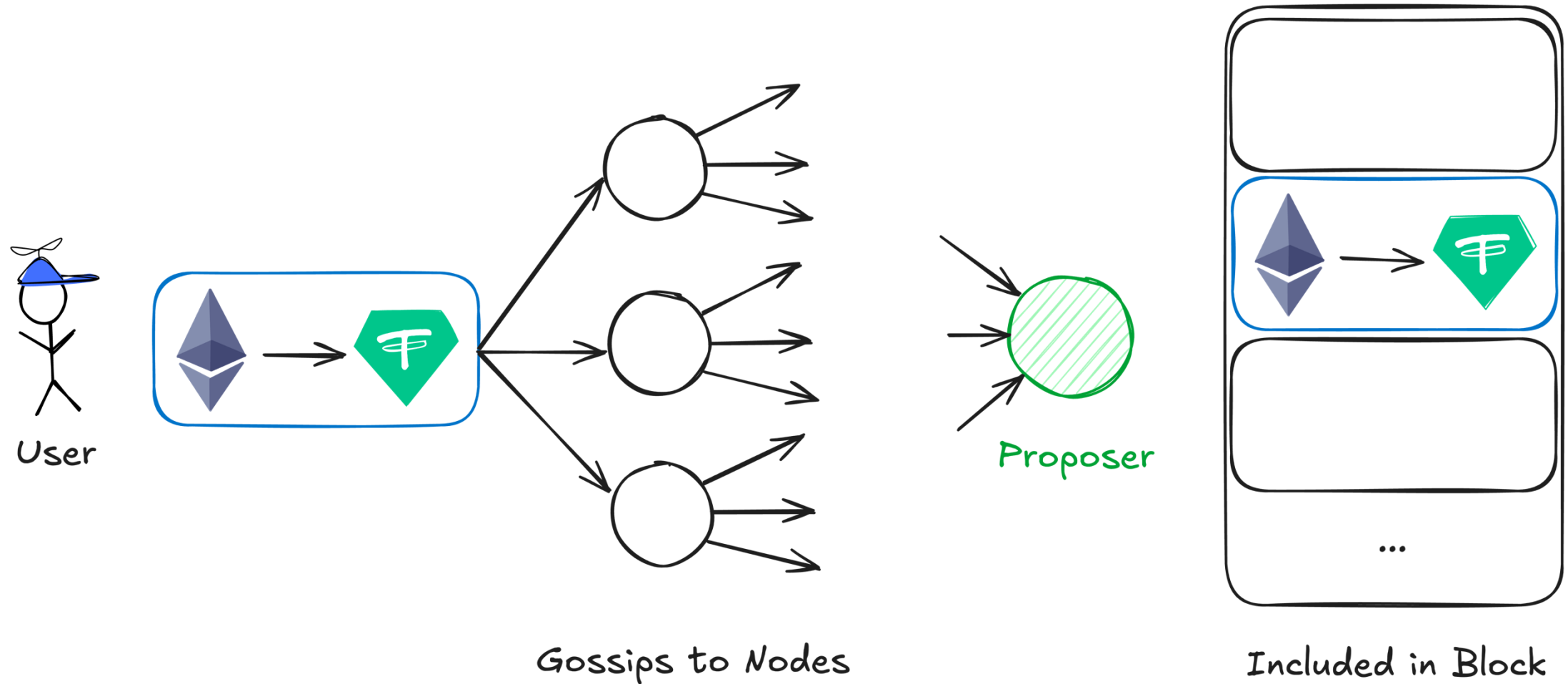
**I. MEV (Max Extractable Value)**

**II. MEV Solutions**

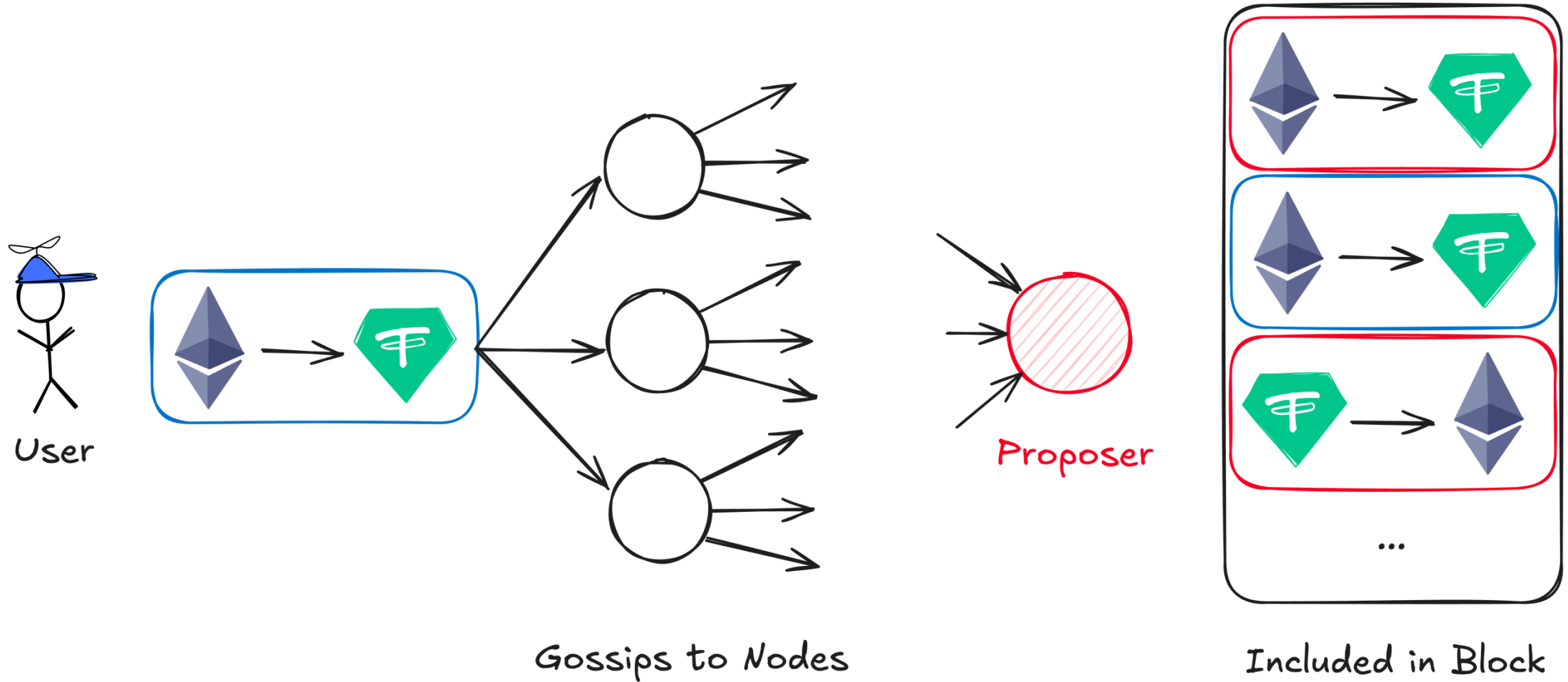
**III. Encrypted Mempool**

**MEV**

# Including a Transaction Onchain











# Problem 1: Frontrunning / Sandwich Attack



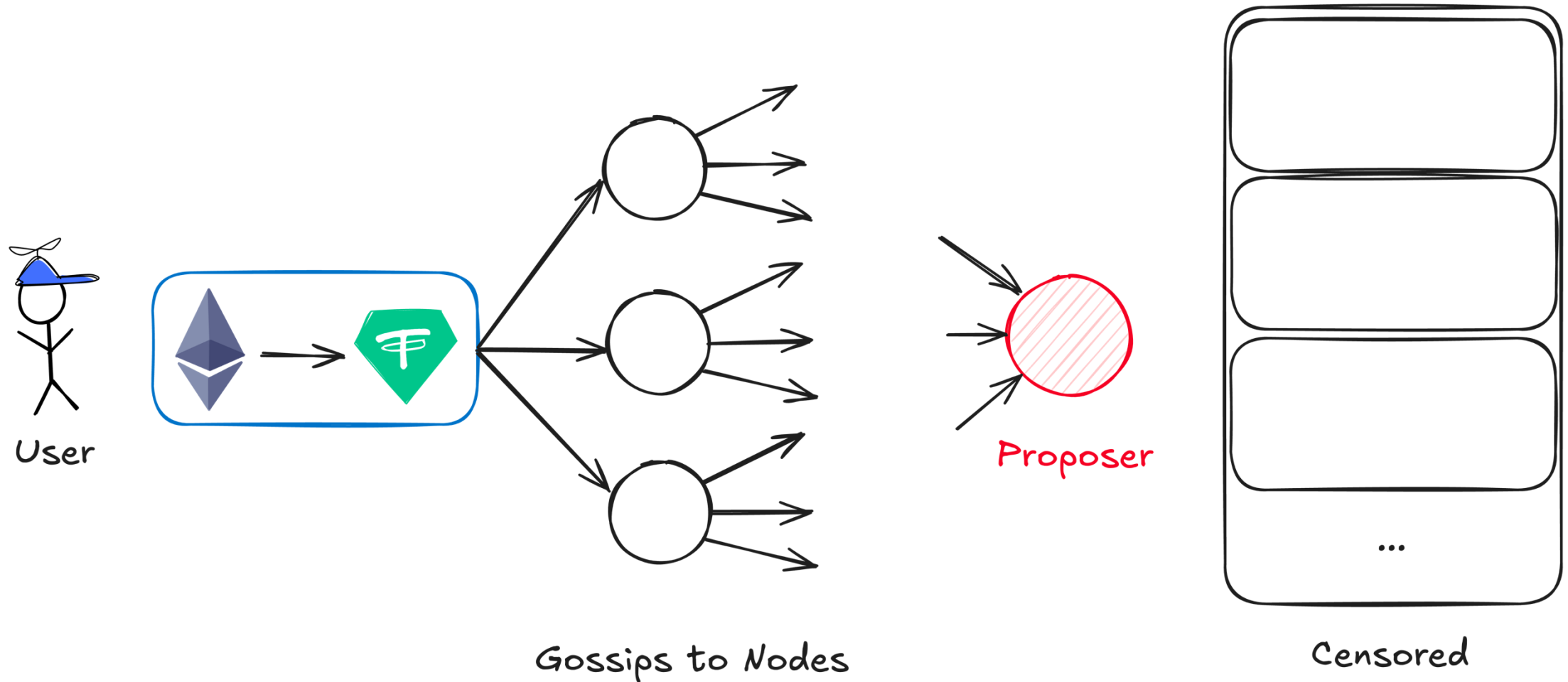
# Example

---

- ▶ From  MEV Bot: 0x000...f56 To  SushiSwap: WBTC For 139.095043641361099086 \$503,307.66  Wrapped Ethe... (WETH)
- ▶ From  SushiSwap: WBTC To  MEV Bot: 0x000...f56 For 5.7648024 \$550,774.99  Wrapped BTC (WBTC)
- ▶ From  MEV Bot: 0x000...f56 To  0xF55D8450...744836007 For 5.76480241 \$550,774.99  Wrapped BTC (WBTC)
- ▶ From  0xF55D8450...744836007 To  MEV Bot: 0x000...f56 For 2,269,314.669822 \$2,269,314.67  Tether USD (USDT)
- ▶ From  MEV Bot: 0x000...f56 To  Uniswap V2: USDT For 2,269,314.669822 \$2,269,314.67  Tether USD (USDT)
- ▶ From  Uniswap V2: USDT To  MEV Bot: 0x000...f56 For 1,352.124212080924112964 \$4,892,586.11  Wrapped Ethe... (WETH)

<https://etherscan.io/tx/0xb72689042f313adbffbe4d192b0febc4c8a8346b75a549d5b4d4795b37180488>

## Problem 2: Censorship



# **MEV Solutions**



# Current Solutions

---

## Dapp Level

- 앱에서 사용자가 최대 감당할 수 있는 슬리피지 설정
  - 예: 5% 이상의 손실 발생 시, 트랜잭션이 실패(Revert) 되도록 설정
- Batch Auction 등 MEV가 발생하기 어려운 구조로 앱을 설계

## Private Mempool

- SUAVE (TEE)

## PBS (Proposer Builder Separation)

- Flashbots에서 개발한 MEV-boost가 PBS의 프로토콜 외부 구현체
  - 현재 이더리움 블록 생성의 90%
- 기존에 블록을 생성하고 제안하던 프로포저의 역할을 빌더와 프로포저로 분리
- 빌더가 수익이 좋은 블록을 프로포저에게 제안, 프로포저는 선택하여 수익의 일부를 받음
- MEV를 막기보단 수익을 공유
- 하지만, 95%의 블록이 두 빌더에 의해 생성
  - BuilderNet 같은 빌더 탈중앙화 해결책 제시

# Proposed Solutions

---

## Fair Ordering

- Priority Fee Ordering
- Random Permutation
- FCFS

## Inclusion Lists

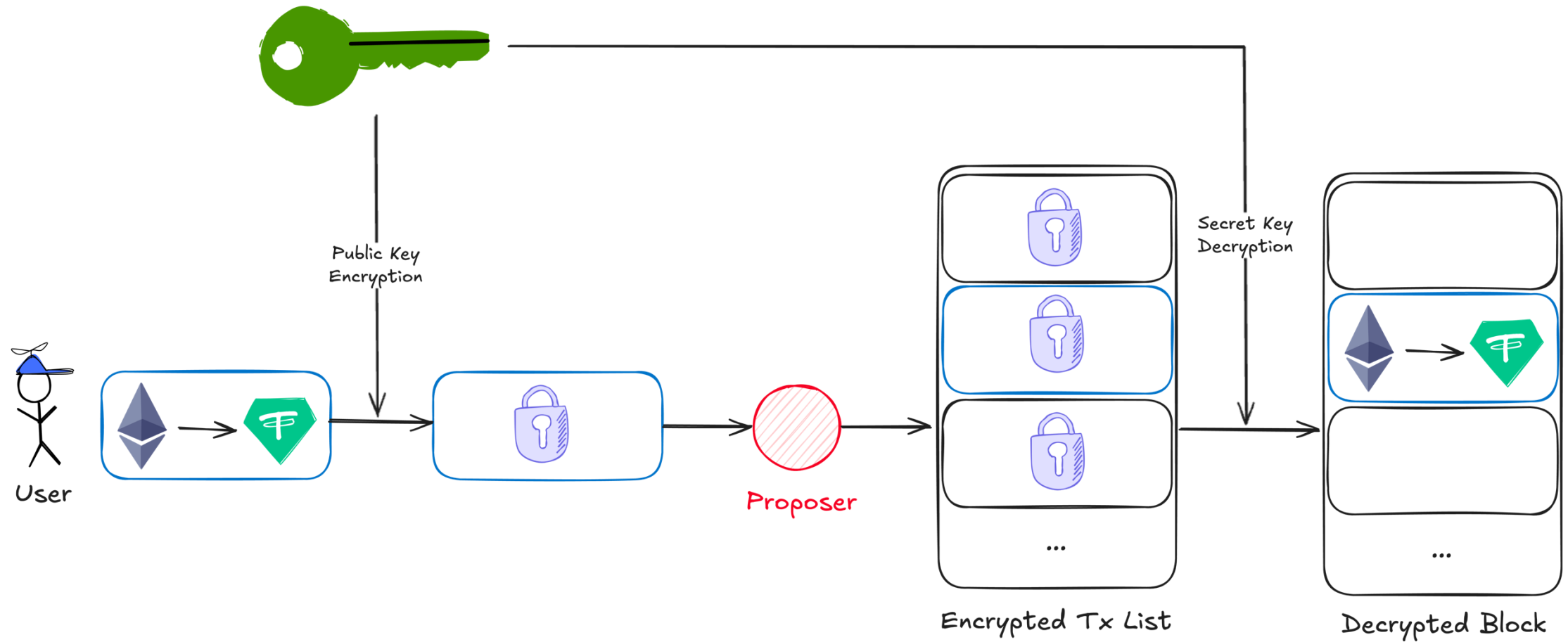
- Proposer가 특정 트랜잭션을 반드시 포함하도록 강제하는 리스트
- Censorship Resistance를 위함
- FOCIL(Fork-Choice Enforced Inclusion Lists)

## Encrypted Mempool

- 사용자가 트랜잭션을 암호화하여 보냄
- 트랜잭션 내용을 알 수 없기 때문에 모든 악의적인 MEV와 검열을 근본적으로 차단

# Encrypted Mempool

# Encrypted Mempool



# Encrypt / Decrypt Designs

---

## Trusted Party

- 공개키로 유저들이 트랜잭션을 암호화하고, 한 주체가 트랜잭션을 복호화
- Flashbots Protect

## Secure Enclave

- TEE 이용
- Flashbots SUAVE

## Threshold Encryption

- 일정 수 이상 모이면 해독할 수 있도록 함.
- 예를 들어, 어떤 Committee를 두고, 2/3 이상이 모이면 해독할 수 있도록

## VDF (Verifiable Delay Function) / Time Lock Puzzle

- 해독하는 데에 일정 시간 이상 걸리도록 함
- 해독이 되기 전에 블록의 트랜잭션들을 확정

# Threshold Encryption

---

## Challenges

- 한 주체는 아니어도, 정해진 Committee가 해독하는 방식은 중앙화와 보안 상의 문제가 발생 가능

## Solution

- Committee를 계속 새롭게 선정하는 방식

## Future Directions

- Optimizations
- 빠르고, 효율적으로 암호학 프로토콜 설계.

# VDF & Timelock Puzzle

---

## Challenges

- 처음 키 생성할 때의 문제
- 이번 블록에 처리되지 않는 트랜잭션들이 모두 공개된다는 문제

# Zero Knowledge Proof

---

## Problem

- 트랜잭션이 암호화된 상태로 있기 때문에, DoS 공격이 가능
- 유효하지 않은 트랜잭션을 암호화 시켜 대량으로 보낼 수 있음

## Solution

- ZKP로 트랜잭션의 세부사항을 공개하지 않으면서 유효한 트랜잭션임을 증명



**Thank You**