# SECURING BLOCKCHAIN-BASED E-VOTING THROUGH SHAMIR'S SECRET SHARING ON ETHEREUM

Esma Beydili
Umut Can Çabuk
Gökhan Dalkılıç
Yusuf Öztürk

April 2025,
Boston

# OUTLINE

Next Slides

Introduction

E-VOTE

Analysis

Conclusion

References

END

# Secret Sharing

- A secret is split into pieces.
- Shamir’ Algorithm

# Smart Contract

- Automatically execute without the need for third parties.
- is immutable
- Can be viewed and verified by anyone.

# Introduction

## -Voting systems

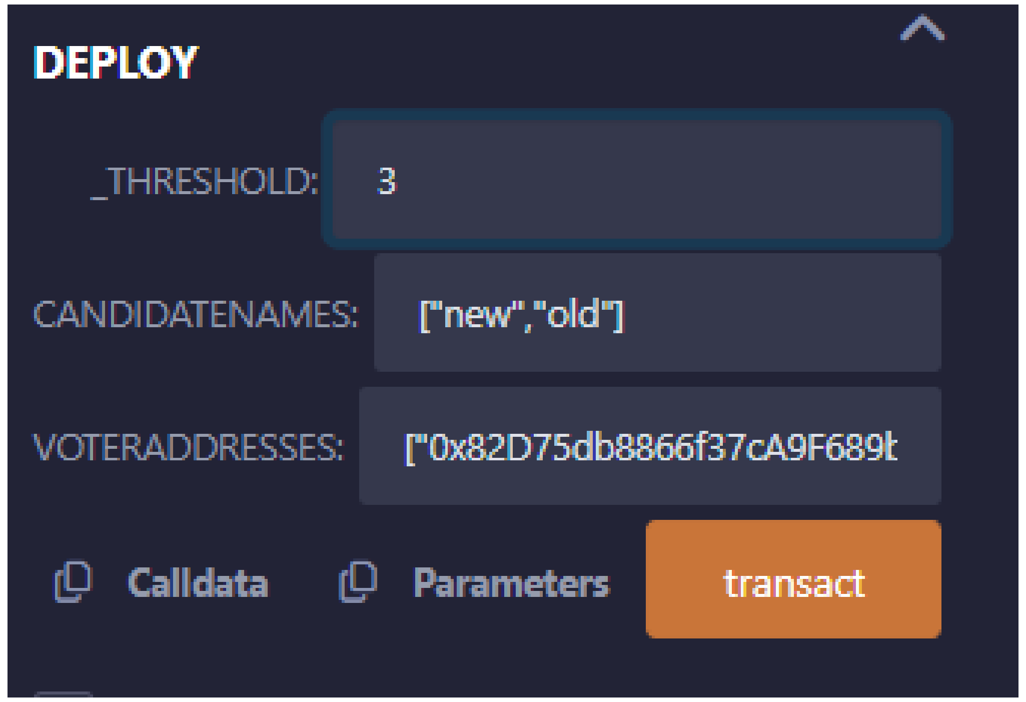
E-voting, or electronic voting, is the use of electronic systems to cast and count votes in an election. It aims to improve efficiency, accessibility, and security in the voting process.

## Analysis Metrics

- Technical Complexity
- Cost
- Privacy
- Accessibility

# E-VOTE

## OWNER



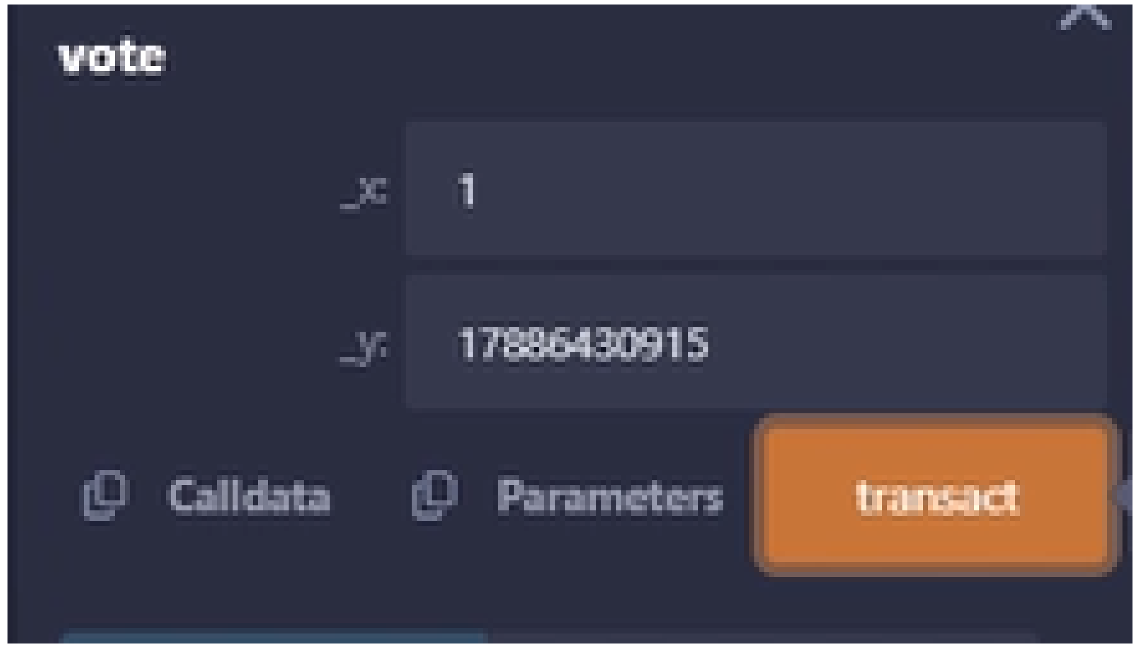
_THRESHOLD: 3

CANDIDATENAMES: ["new","old"]

VOTERADDRESSES: ["0x82D75db8866f37cA9F689t

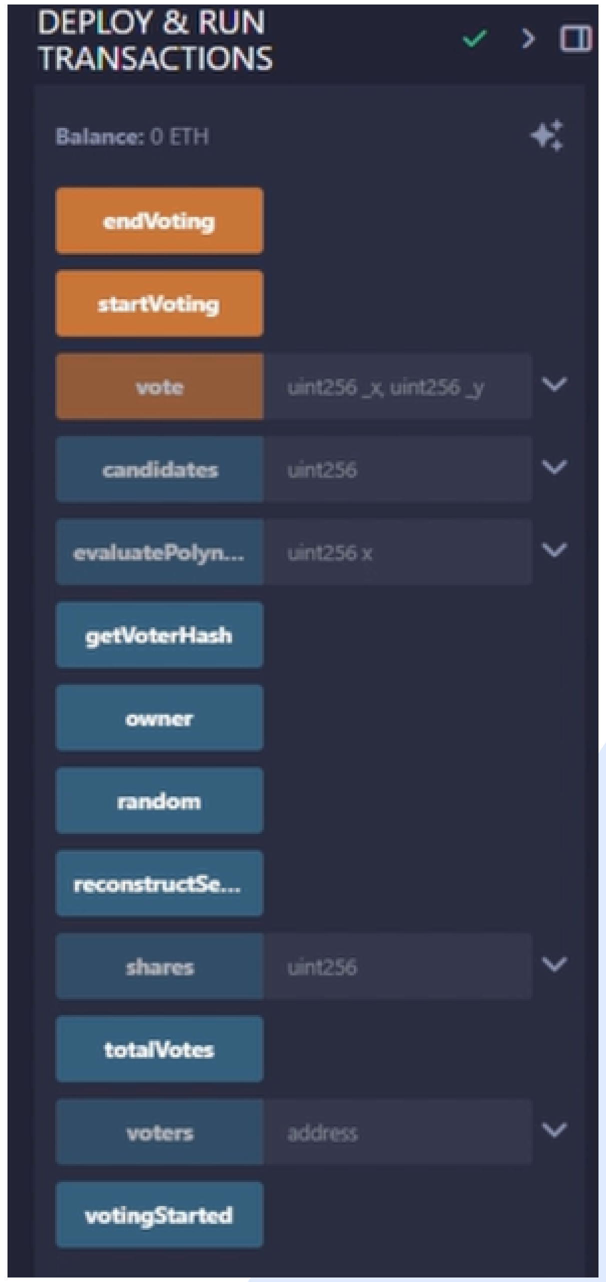
## VOTER

SHARE(1,17886430914)



_x: 1

_y: 17886430915



Balance: 0 ETH

Solidity

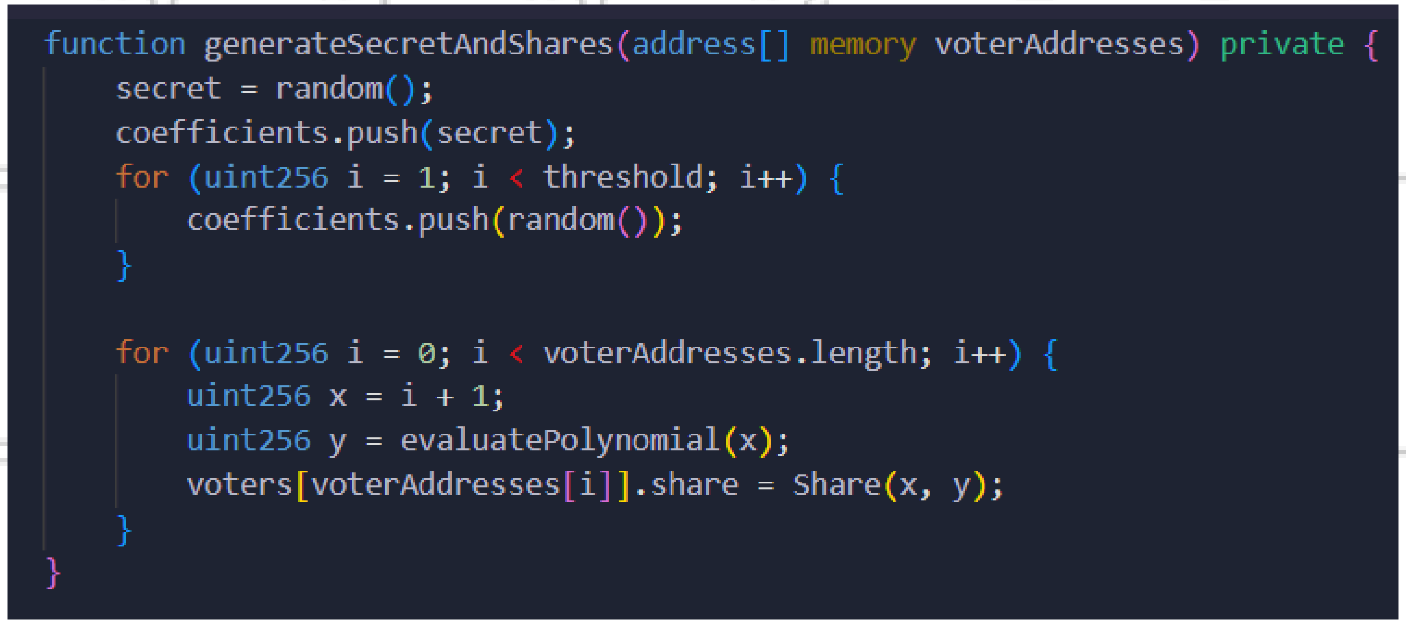
# E-VOTE

Co the Lagrange basis polynomials:

$$= \frac{x-x_1}{x_0-x_1}\cdot\frac{x-x_2}{x_0-x_2} = \frac{x-4}{2-4}\cdot\frac{x-5}{2-5} = \frac{1}{6}x^2 - \frac{3}{2}x + \frac{}{3}$$

$\ell_1(x) =$

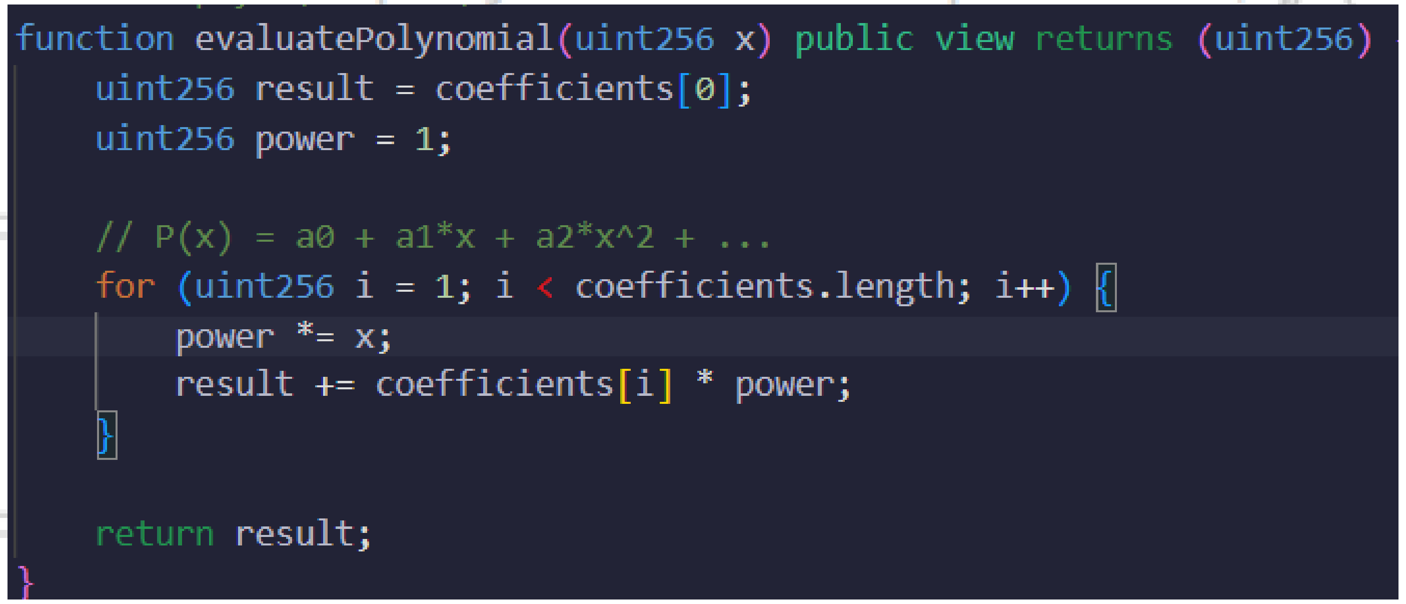
$\ell_2(x) =$



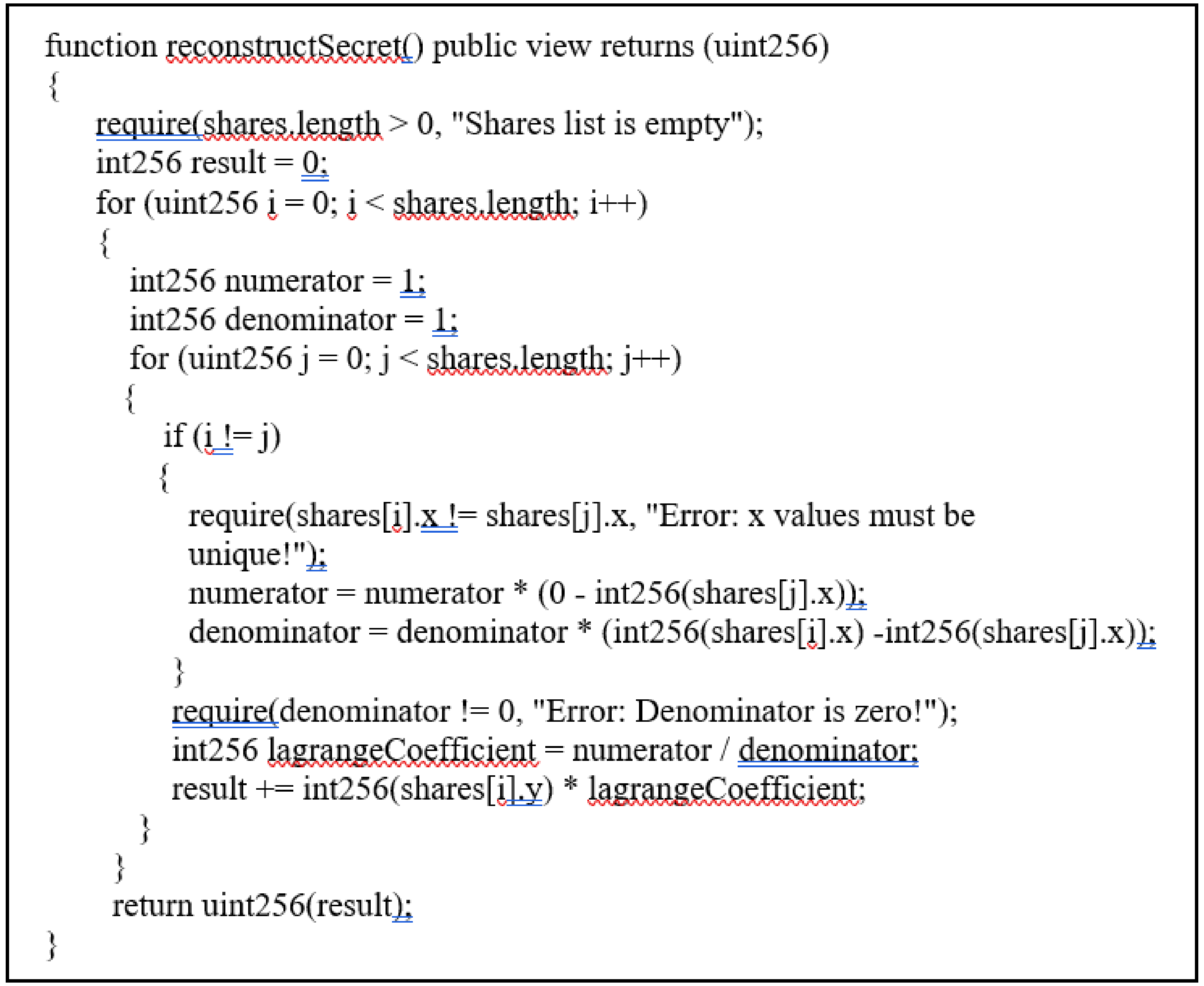
```
function generateSecretAndShares(address[] memory voterAddresses) private {
    secret = random();
    coefficients.push(secret);
    for (uint256 i = 1; i < threshold; i++) {
        coefficients.push(random());
    }

    for (uint256 i = 0; i < voterAddresses.length; i++) {
        uint256 x = i + 1;
        uint256 y = evaluatePolynomial(x);
        voters[voterAddresses[i]].share = Share(x, y);
    }
}
```

Using the formula for polynomial interpolation, $f(x)$ is:

$f(x) =$



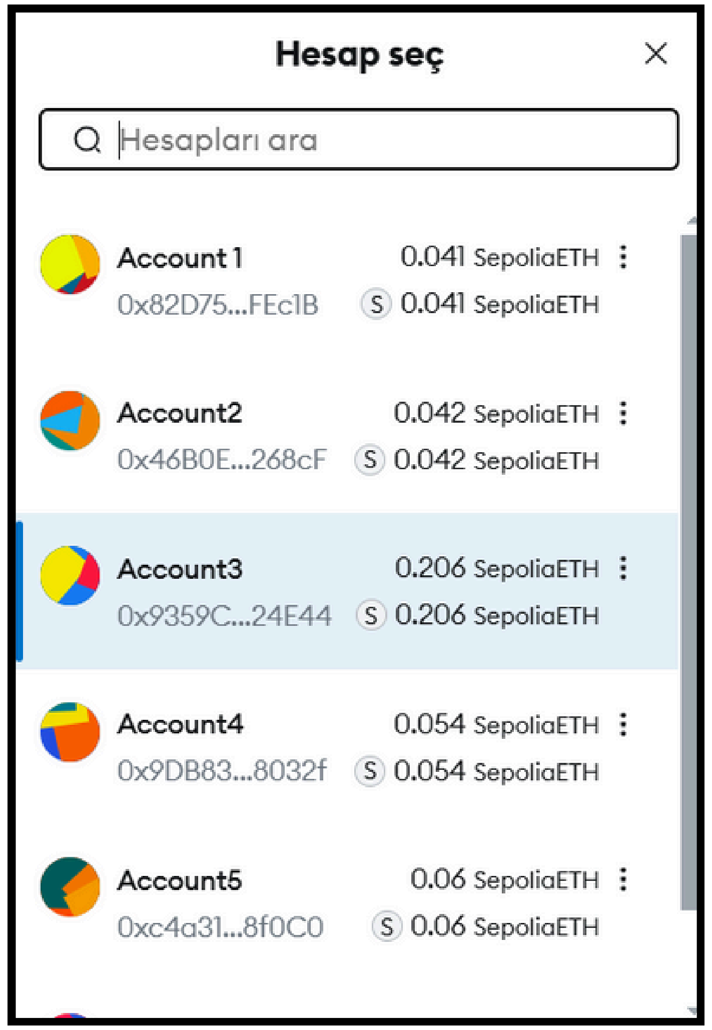
```
function evaluatePolynomial(uint256 x) public view returns (uint256)
    uint256 result = coefficients[0];
    uint256 power = 1;

    // P(x) = a0 + a1*x + a2*x^2 + ...
    for (uint256 i = 1; i < coefficients.length; i++) {
        power *= x;
        result += coefficients[i] * power;
    }

    return result;
}
```



```
function reconstructSecret() public view returns (uint256)
{
    require(shares.length > 0, "Shares list is empty");
    int256 result = 0;
    for (uint256 i = 0; i < shares.length; i++)
    {
      int256 numerator = 1;
      int256 denominator = 1;
      for (uint256 j = 0; j < shares.length; j++)
      {
        if (i != j)
        {
          require(shares[i].x != shares[j].x, "Error: x values must be
          unique!");
          numerator = numerator * (0 - int256(shares[j].x));
          denominator = denominator * (int256(shares[i].x) -int256(shares[j].x));
        }
        require(denominator != 0, "Error: Denominator is zero!");
        int256 lagrangeCoefficient = numerator / denominator;
        result += int256(shares[i].y) * lagrangeCoefficient;
      }
    }
    return uint256(result);
}
```

$$= 1942\left(\frac{1}{6}x^2 - \frac{3}{2}x + \frac{10}{3}\right) + 3402\left(-\frac{1}{2}x^2 + \frac{7}{2}x - 5\right) + 441$$

# E-VOTE



Hesap seç ×

Hesapları ara

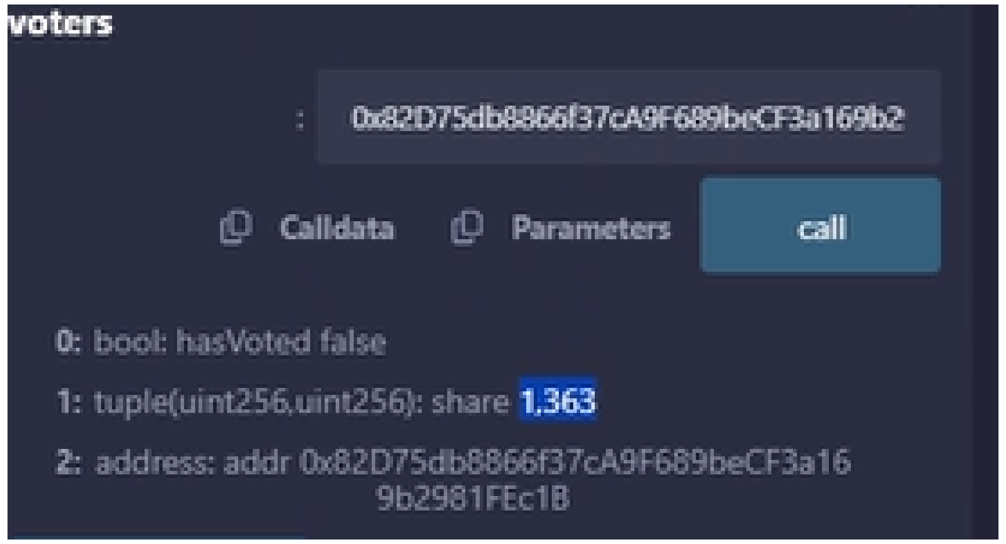
Account 1 0.041 SepoliaETH
0x82D75...FEc1B Ⓢ 0.041 SepoliaETH

Account2 0.042 SepoliaETH
0x46B0E...268cF Ⓢ 0.042 SepoliaETH

Account3 0.206 SepoliaETH
0x9359C...24E44 Ⓢ 0.206 SepoliaETH

Account4 0.054 SepoliaETH
0x9DB83...8032f Ⓢ 0.054 SepoliaETH

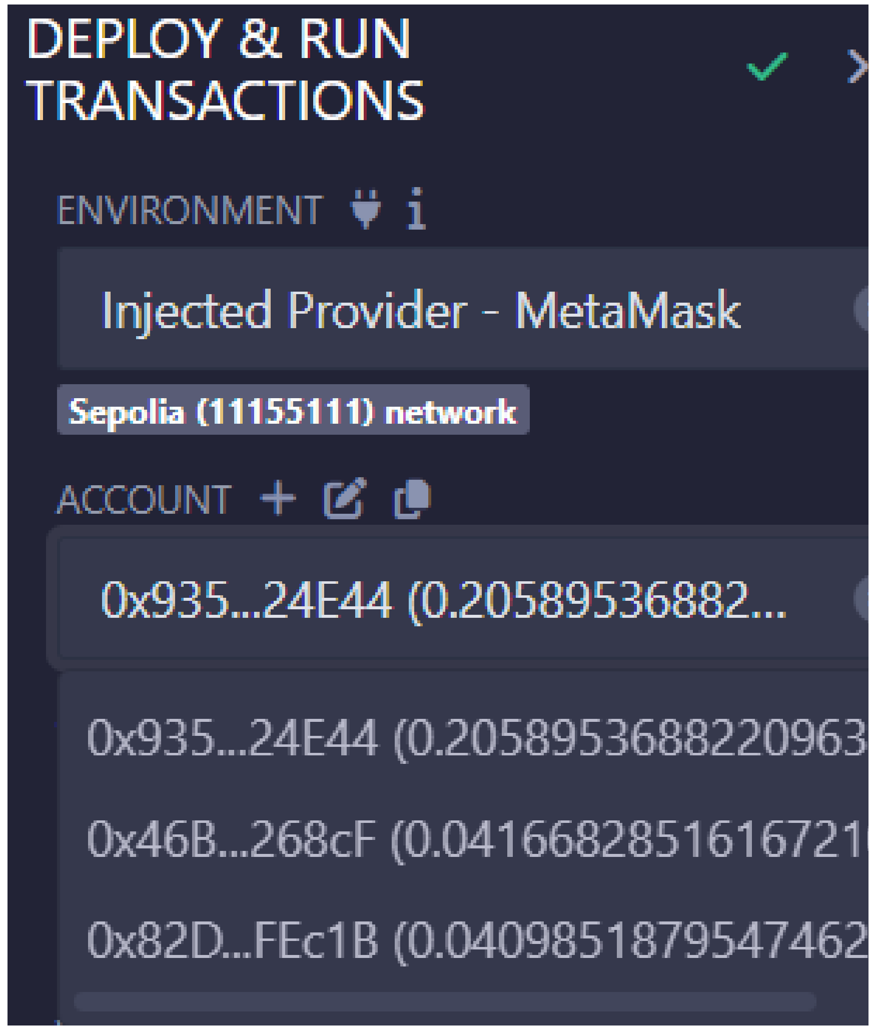
Account5 0.06 SepoliaETH
0xc4a31...8f0C0 Ⓢ 0.06 SepoliaETH



voters

: 0x82D75db8866f37cA9F689beCF3a169b2

```
0: bool: hasVoted false
1: tuple(uint256,uint256): share 1,363
2: address: addr 0x82D75db8866f37cA9F689beCF3a16
                9b2981FEc1B
```



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: Injected Provider - MetaMask

Sepolia (11155111) network

ACCOUNT: 0x935...24E44 (0.20589536882...

0x935...24E44 (0.2058953688220963
0x46B...268cF (0.0416682851616721
0x82D...FEc1B (0.0409851879547462

METAMASK

REMIX IDE

SepoliaETH

# E-VOTE

## ERROR-FREE VOTING

### TABLE OF SCENARIO 1

| Candidates | Number of Voter | Voter Indices | Expected Result | Actual Result | Incorrect Input | Gas Fee |
|---|---|---|---|---|---|---|
| Shamir, Blakely | 3 | 0 voters for shamir, 3 voters for blakely | "blakely" wins with majority votes | "blakely" wins with majority votes | No | 0,004409 Sepolia ETH |

## INVALID SECRET SHARE

### TABLE OF SCENARIO 2

| Candidates | Number of Voter | Voter Indices | Expected Result | Actual Result | Incorrect Input | Gas Fee |
|---|---|---|---|---|---|---|
| Crypto, Netsec | 3 | 0 voters for crypto, 3 voters for netsec | ERROR | ERROR Non-termination of the contract | Yes | 0,004300 Sepolia ETH |

# E-VOTE

## INVALID CANDIDATE INDEX

### TABLE OF SCENARIO 3

| Candidates | Number of Voter | Voter Indices | Expected Result | Actual Result | Incorrect Input | Gas Fee |
|---|---|---|---|---|---|---|
| Crypto[0], Netsec [1] | 3 | Two: 1, Invalid Index: 3 | "netsec" wins with majority votes | ERROR Then "netsec" wins with majority votes | Yes | 0,005170 Sepolia ETH |

## TIE

### TABLE OF SCENARIO 4

| Candidates | Number of Voter | Voter Indices | Expected Result | Actual Result | Incorrect Input | Gas Fee |
|---|---|---|---|---|---|---|
| new[0], old[1] | 4 | Two: 0, Two: 1 | "old" and “new” tie | tie | no | 0,0478 Sepolia ETH |

ANALYSIS

# Technical Complexity

- Initialization (Deployment): $O(v \cdot t)$, where v is the number of voters and t is the
- Voting:  Each voter is $O(1)$
- Finalization: The secret is reconstructed using Lagrange interpolation,$O(s^2)$,

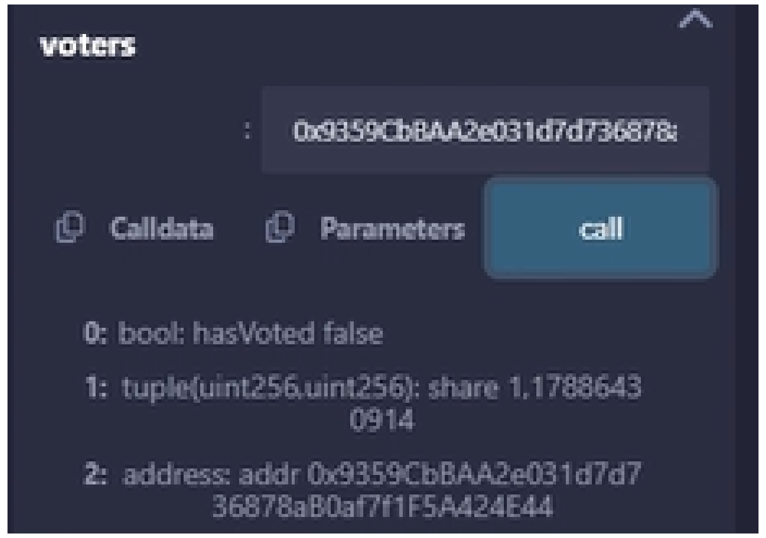
## Comparison:

ShamirVoting contract
$O(v \cdot t + s^2 + c))$

Simpler Ballot contract
$O(v + c)$

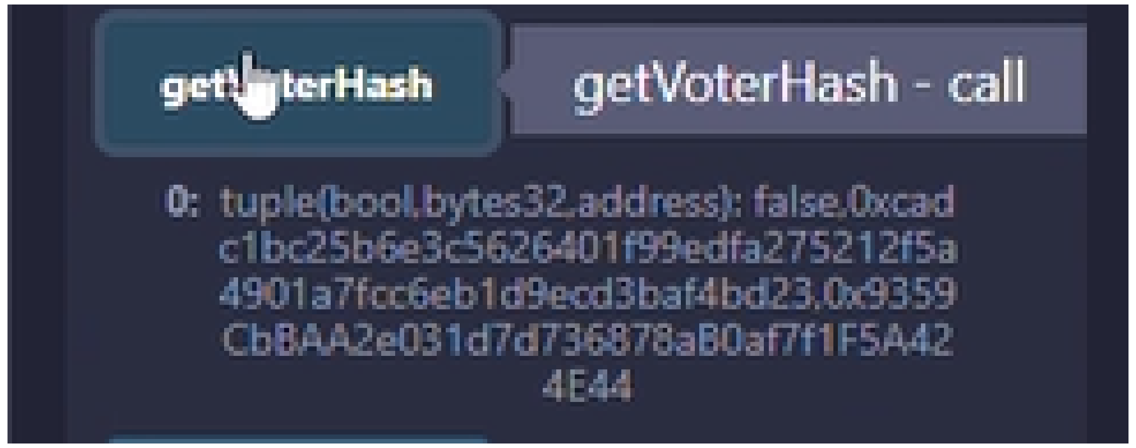
# ANALYSIS

# Privacy

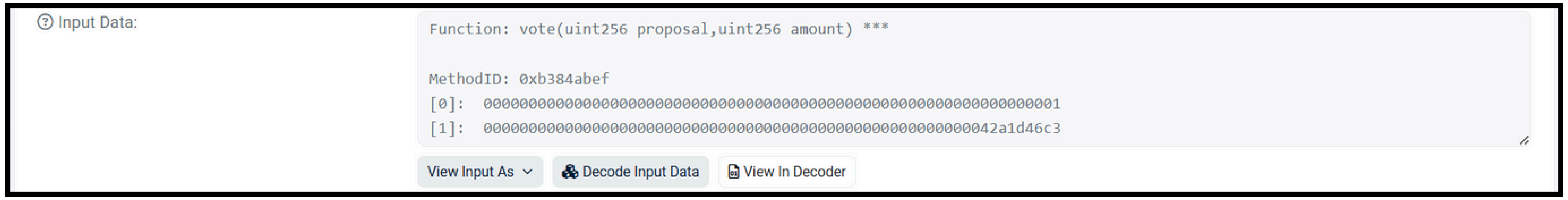


```
0x9359CbBAA2e031d7d736878a
```

```
0: bool: hasVoted false
1: tuple(uint256,uint256): share 1,1788643
                     0914
2: address: addr 0x9359CbBAA2e031d7d7
              36878aB0af7f1F5A424E44
```

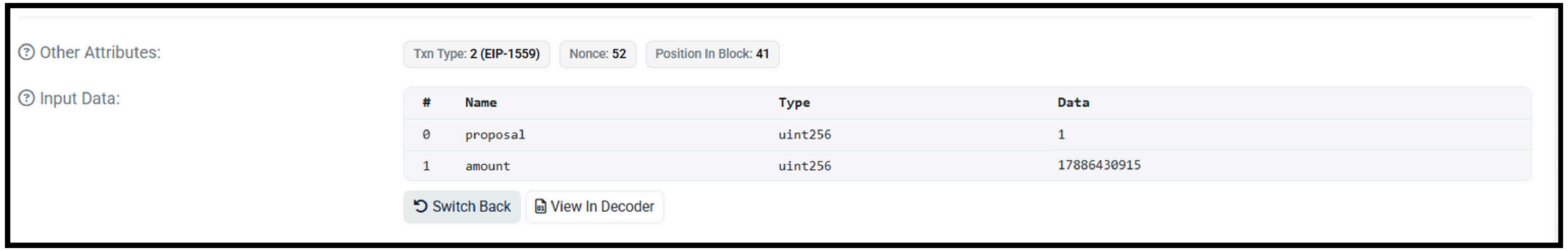


```
0: tuple(bool,bytes32,address): false,0xcad
   c1bc25b6e3c5626401f99edfa275212f5a
   4901a7fcc6eb1d9ecd3baf4bd23,0x9359
   CbBAA2e031d7d736878aB0af7f1F5A42
              4E44
```



Input Data:

```
Function: vote(uint256 proposal,uint256 amount) ***

MethodID: 0xb384abef
[0]:  0000000000000000000000000000000000000000000000000000000000000001
[1]:  000000000000000000000000000000000000000000000000000000042a1d46c3
```



Other Attributes: Txn Type: 2 (EIP-1559) Nonce: 52 Position In Block: 41

Input Data:

| # | Name | Type | Data |
|---|---|---|---|
| 0 | proposal | uint256 | 1 |
| 1 | amount | uint256 | 17886430915 |

ANALYSIS

# Cost and Accessability

**Contract Deployment**
0.00361796SepoliaETH

**StartVoting**
0.00007032SepoliaETH

3x
**Vote**
0.00017327SepoliaETH

**EndVote**
0.00012059SepoliaETH

= 0,0043277SepoliaETH

15,91 USD

**Contract Deployment**
0.0303191SepoliaETH

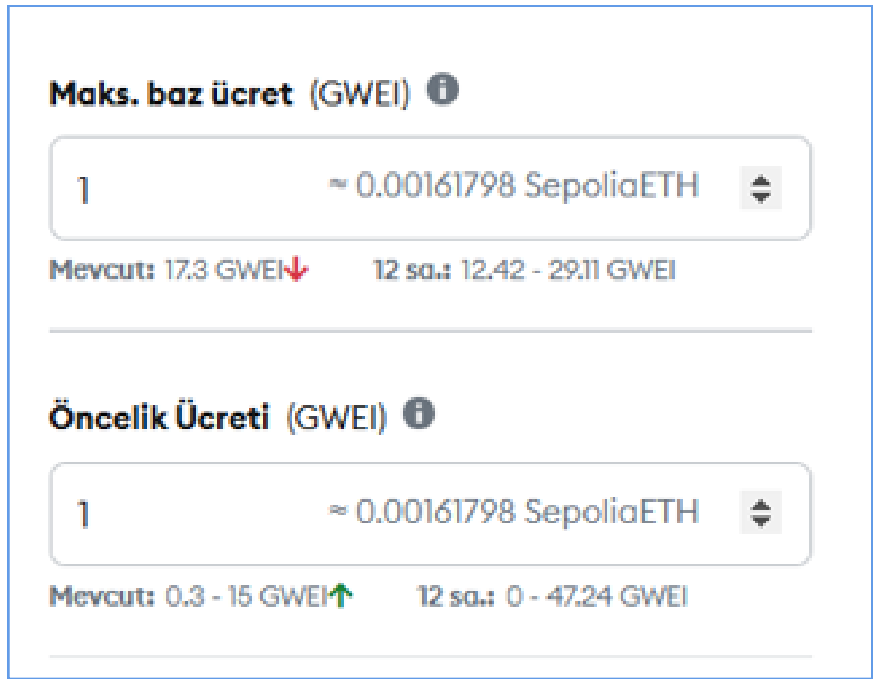
**StartVoting**
0.00040588SepoliaETH

3x
**Vote**
0.0018595SepoliaETH

**EndVote**
0.00124812SepoliaETH

= 0,0375516SepoliaETH

138,03 USD



**Maks. baz ücret** (GWEI)

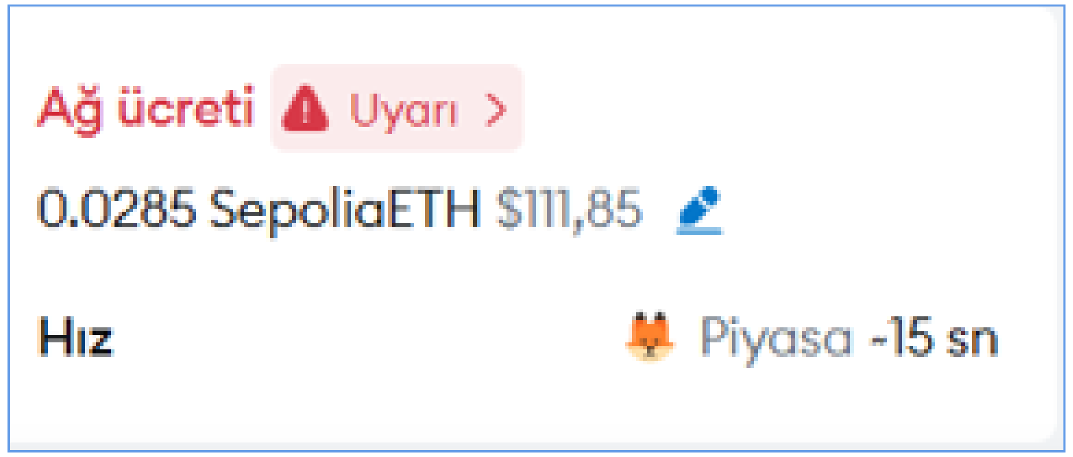
1 ≈ 0.00161798 SepoliaETH

**Mevcut:** 17.3 GWEI↓ **12 sa.:** 12.42 - 29.11 GWEI

**Öncelik Ücreti** (GWEI)

1 ≈ 0.00161798 SepoliaETH

**Mevcut:** 0.3 - 15 GWEI↑ **12 sa.:** 0 - 47.24 GWEI



Ağ ücreti ⚠ Uyarı >

0.0285 SepoliaETH $111,85

Hız 🦊 Piyasa -15 sn

# Conclusion

In blockchain systems based on transparency, external systems are needed to share data secretly. The current work is not sufficient in terms of security. It can be a suitable work for critical tasks by making improvements on privacy.

In terms of cost, it affects the number of people who will participate in the vote and the hours in which the vote will be held rather than the integration of the code. This also makes user access difficult.

Controls that can be done with code are sufficient for many scenarios. This feature, which was added for people to double-check, **if is  really necessary except for situations that require high multi-stage security.**

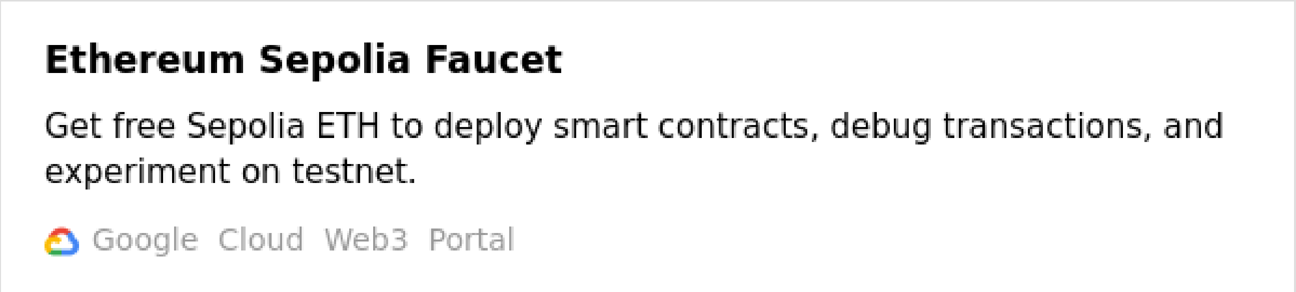
REFERENCES

# Thank you for listening.

## References

Esma, "LastShamirSecretSharingEvote.sol” GitHub Repository, 2025. [Online]. Available: https://github.com/Esma222/EvotWithShamir/blob/main/LastShamirsEVoteSmartContract.sol [Accessed: 06-Jan-2025].

Remix IDE, "Remix Ethereum IDE," [Online]. Available: https://remix.ethereum.org/#lang=en&optimize=false&runs=200&ev mVersion=null&version=soljson-v0.8.26+commit.8a97fa7a.js. [Accessed: 13-Nov-2024].



**Ethereum Sepolia Faucet**

Get free Sepolia ETH to deploy smart contracts, debug transactions, and experiment on testnet.

Google Cloud Web3 Portal