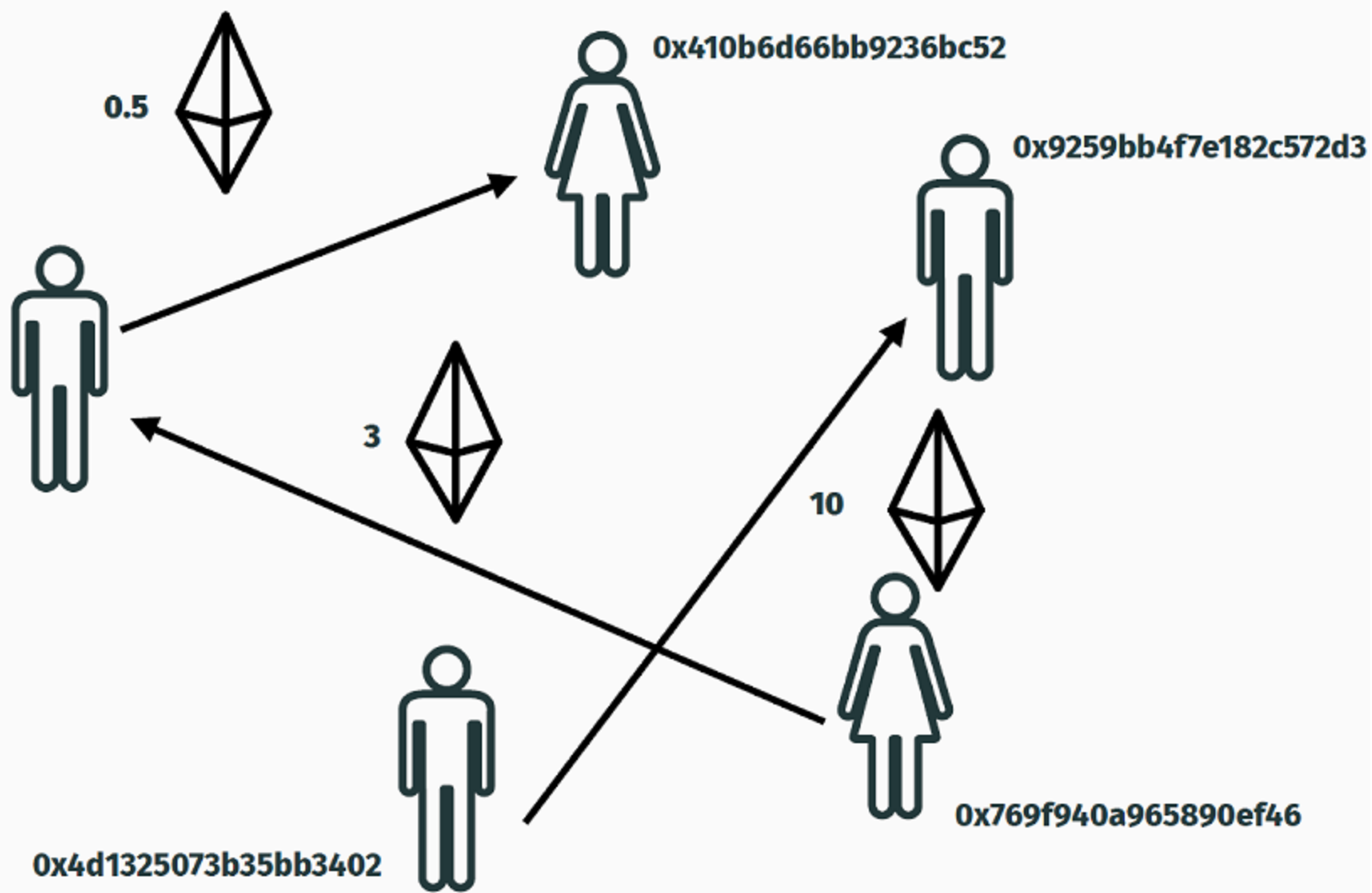


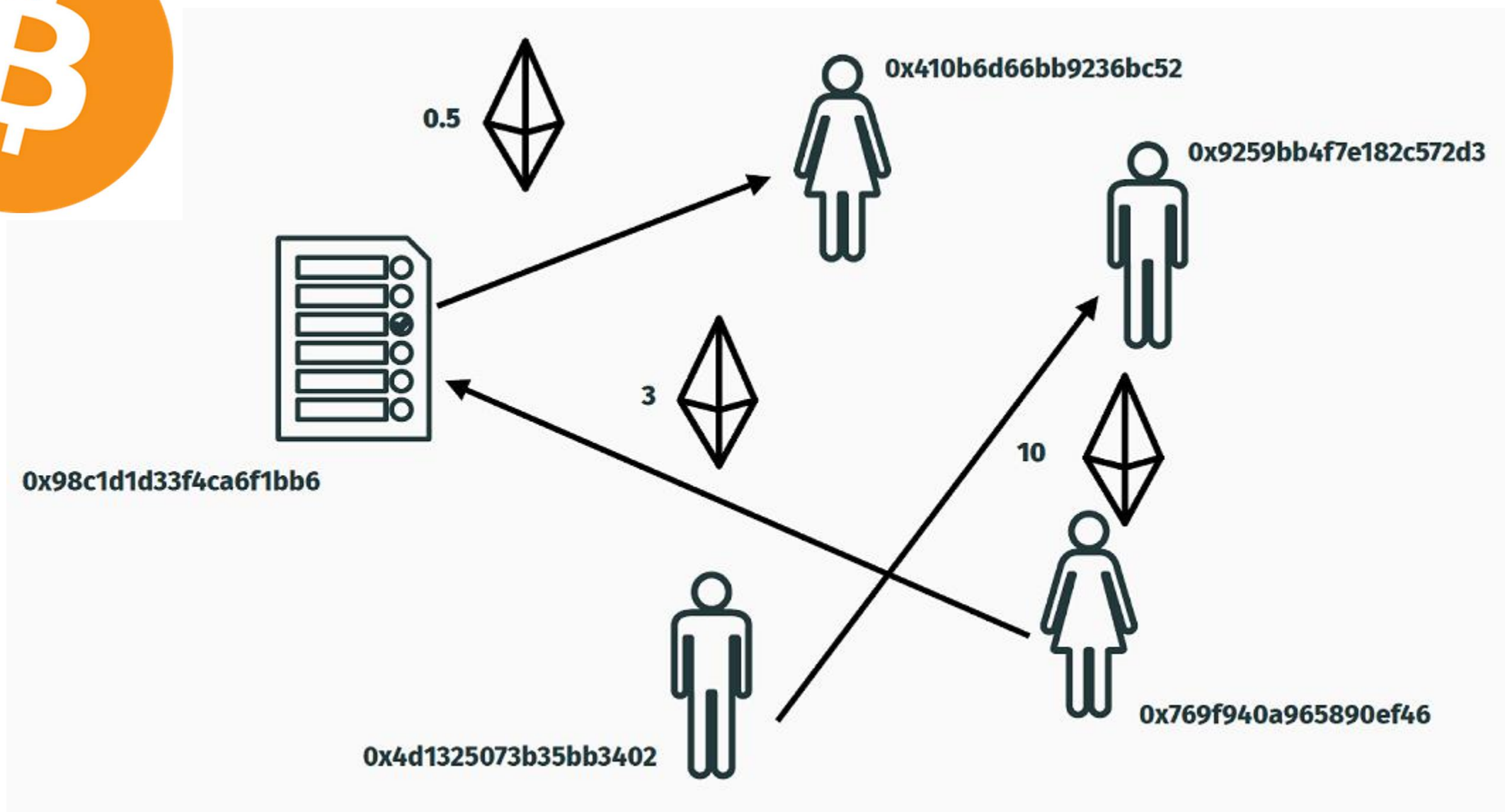
An Efficient and Scalable Smart Contract Checker

Shangyu Xie



0x98c1d1d33f4ca6f1bb6





If event X happens



0x410b6d66bb9236bc52



0x9259bb4f7e182c572d3



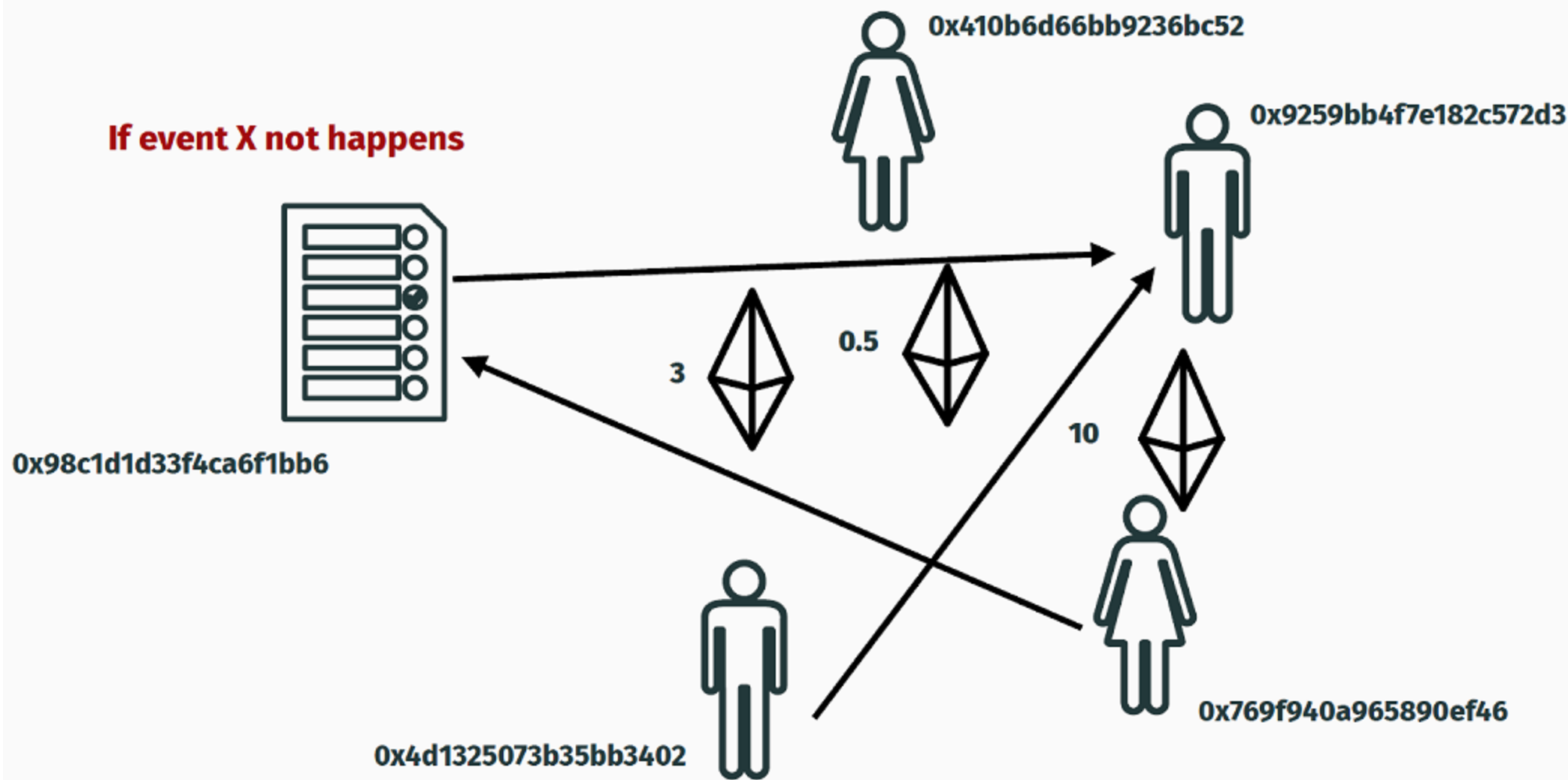
0x4d1325073b35bb3402



0x769f940a965890ef46

0x98c1d1d33f4ca6f1bb6

If event X not happens



A problem has been detected and Windows has been shut down to prevent damage to your computer.

The problem seems to be caused by the following file: kbdhid.sys

MANUALLY_INITIATED_CRASH

If this is the first time you've seen this stop error screen, restart your computer. If this screen appears again, follow these steps:

Check to make sure any new hardware or software is properly installed. If this is a new installation, ask your hardware or software manufacturer for any Windows updates you might need.

If problems continue, disable or remove any newly installed hardware or software. Disable BIOS memory options such as caching or shadowing. If you need to use safe mode to remove or disable components, restart your computer, press F8 to select Advanced Startup Options, and then select Safe Mode.

Technical Information:

*** STOP: 0x000000e2 (0x00000000, 0x00000000, 0x00000000, 0x00000000)

*** kbdhid.sys - Address 0x94efd1aa base at 0x94efb000 DateStamp 0x4a5bc705

A problem has been detected and Windows has been shut down to prevent damage to your computer.

A \$50 Million Hack Just Showed That the DAO Was All Too Human

Check to make sure any new hardware or software is properly installed. If this is a new installation, ask your hardware or software manufacturer for any Windows updates you might need.

If problems continue, disable or remove any newly installed hardware or software. Disable BIOS memory options such as caching or shadowing. If you need to use safe mode to remove or disable components, restart your computer, press F8 to select Advanced Startup Options, and then select Safe Mode.

Technical Information:

*** STOP: 0x000000e2 (0x00000000, 0x00000000, 0x00000000, 0x00000000)

*** kbdhid.sys - Address 0x94efd1aa base at 0x94efb000 DateStamp 0x4a5bc705

A problem has been detected and Windows has been shut down to prevent damage to your computer.

A \$50 Million Hack Just Showed That the DAO Was All Too Human

\$30 Million: Ether Reported Stolen Due to Parity Wallet Breach

your computer, press F8 to select Advanced Startup Options, and then select Safe Mode.

Technical Information:

*** STOP: 0x000000e2 (0x00000000, 0x00000000, 0x00000000, 0x00000000)

*** kbdhid.sys - Address 0x94efd1aa base at 0x94efb000 DateStamp 0x4a5bc705

A problem has been detected and Windows has been shut down to prevent damage to your computer.

A \$50 Million Hack Just Showed That the DAO Was All Too Human

\$30 Million: Ether Reported Stolen Due to Parity Wallet Breach

your computer, press F8 to select Advanced Startup Options, and then select

Techn

*** S

*** k

Someone 'Accidentally' Locked Away \$150M Worth of Other People's Ethereum Funds

... ..

Look closer---Parity Wallet Bug Example

```
1  contract WalletLibrary {
2      address[256] owners;
3      mapping(bytes => uint256) approvals;
4      function confirm(bytes32 _op) internal bool {
5          /* logic for confirmation */
6      }
7      function initWallet(address[] _owners) {
8          /* initialize the wallet owners */
9      }
10     function pay(address to, uint amount) {
11         if (confirm(keccak256(msg.data)))
12             to.transfer(amount);
13     }
14 }
15 contract Wallet {
16     address library = 0xAABB...;
17     // constructor
18     function Walle(address[] _owners) {
19         library.delegatecall("initWallet", _owners)
20     }
21     function() payable {
22         library.delegatecall(msg.data);
23     }
```

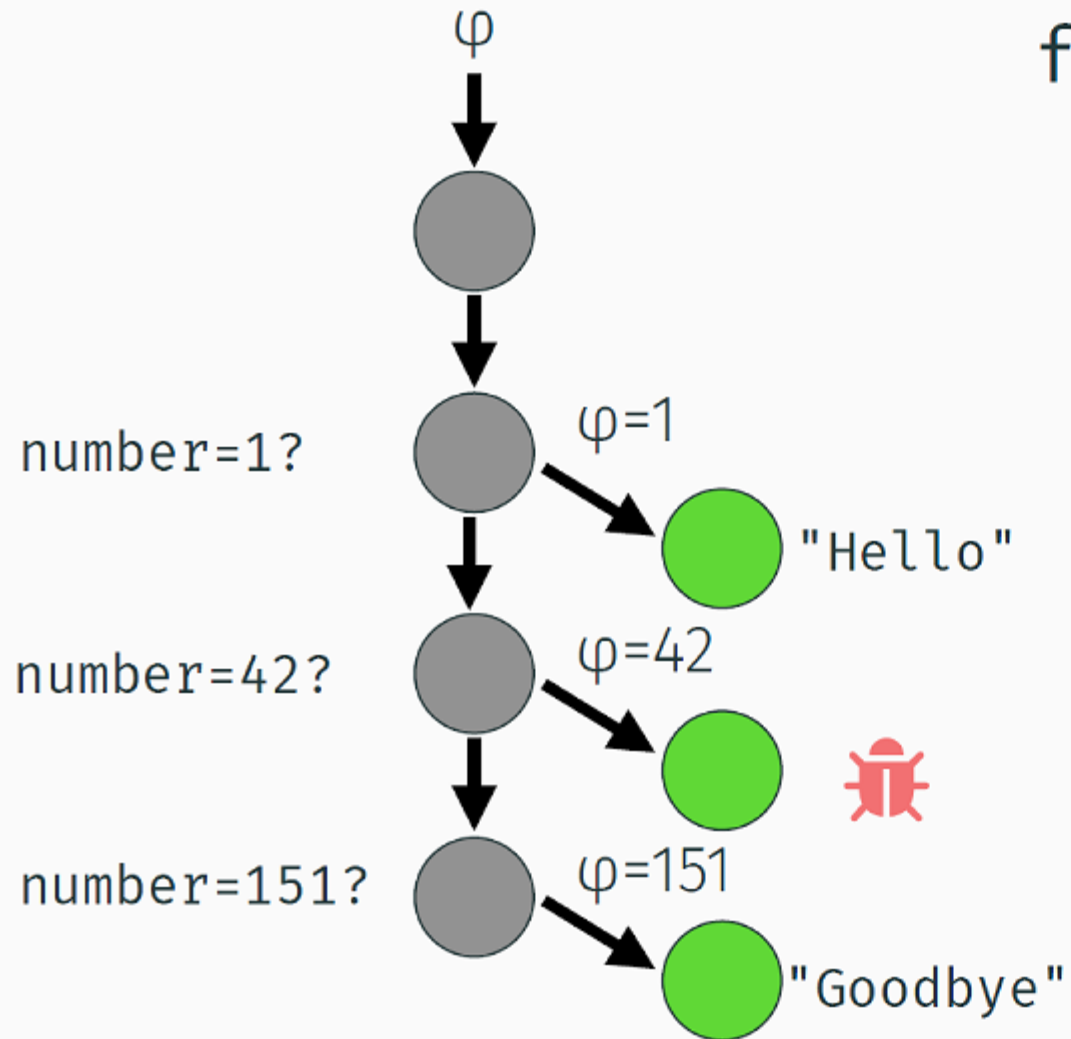


Adversaries may redirect the flow of contract to her addresses, e.g., extract all the money!


Project Goal


- How to efficiently detect the bug in the smart contract?
 - High computation costs for executing smart contract
 - A large number of smart contract programs
- How to accurately classify the bugged smart contract?
 - Machine learning techniques, decision trees, DNNs
 - More fine-grained feature extraction
 - Robust?

Symbolic Execution



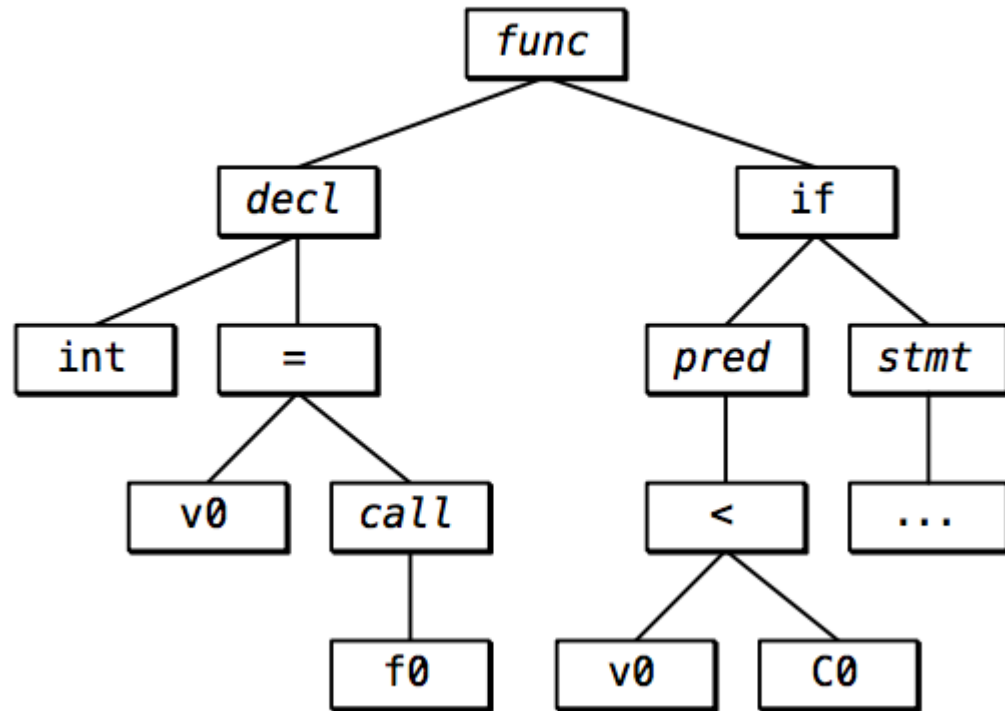
```
function(number):  
    if (number = 1) {  
        say "Hello"  
    }
```

```
    if (number = 42) {  
          
    }
```

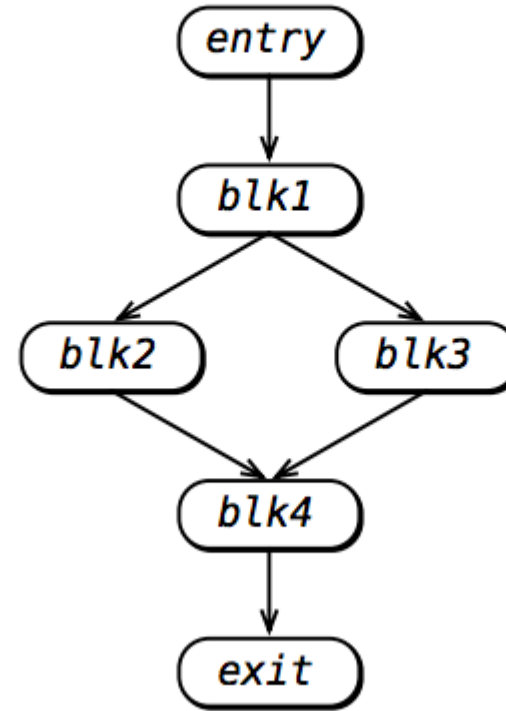
```
    if (number = 151) {   
        say "Goodbye"  
    }
```

Abstract syntax tree & Control-flow graph

Abstract syntax tree (AST)



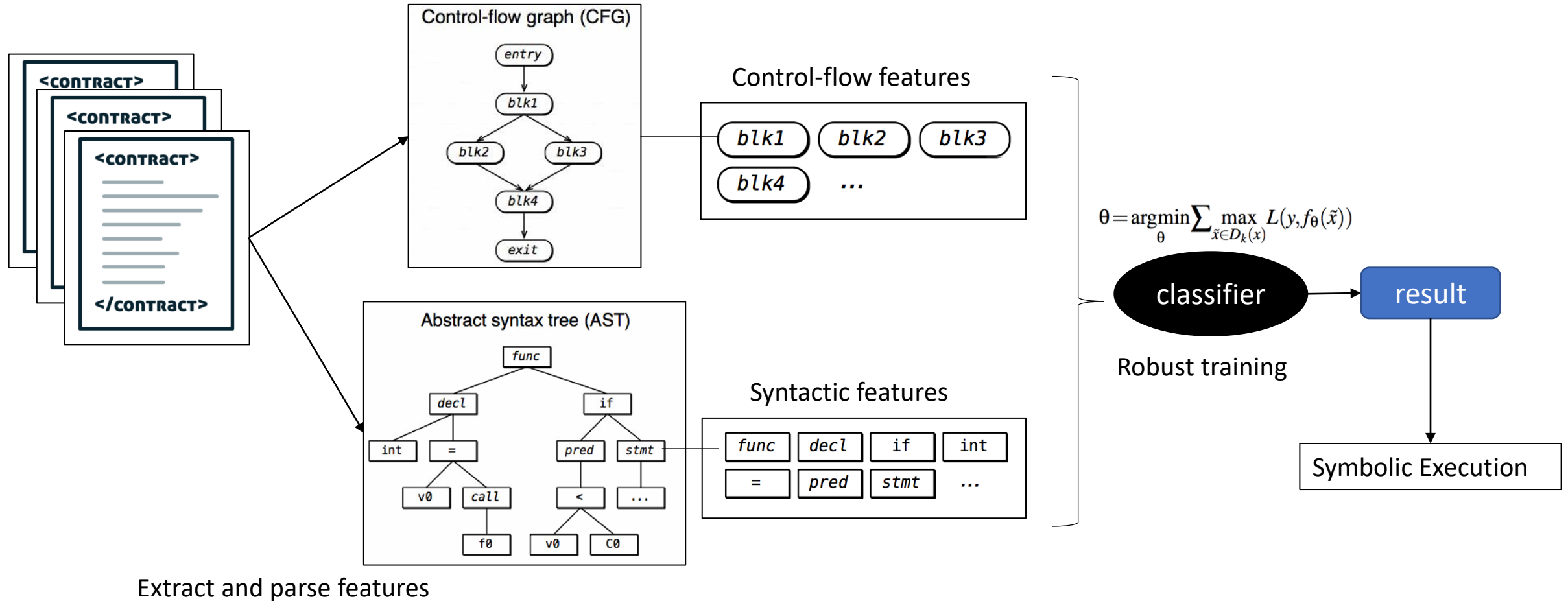
Control-flow graph (CFG)



Ideas

- Smart contract features extracted with AST and CFG
 - Local classifier to detect the bug
 - Compute the score of bug?
- Symbolic execution for further improvement
- Large-size smart contract?
 - Divide and conquer
 - Pipeline blocks

Framework



Implementation

- Ethereum Virtual Machine (EVM)
 - a special-purpose, stack-based virtual machine to determine the outcome of a smart contract execution.
- Training Phase
 - Optimization technique
 - Dimensionality reduction
- Symbolic Execution

Project plan

Time	Task
Week 1	Survey on Related work
Week 2~4	Framework Design
Week 5~7	Implementation and Experiments
Week 8	Write report