SNARKs for Ethereum

Sean Bowe	Zcash
Casey Detrio	Ethereum Foundation
Joshua Gancher	Cornell University
Yuncong Hu	Shanghai Jiao Tong University
Andrew Miller	IC3, UIUC, Zcash
Eran Tromer	Tel Aviv University, Columbia University, Zcash

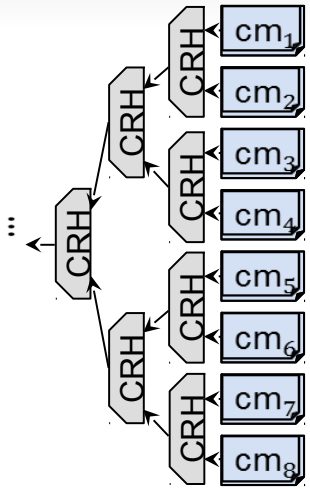
Bringing zero-knowledge SNARKs to Ethereum

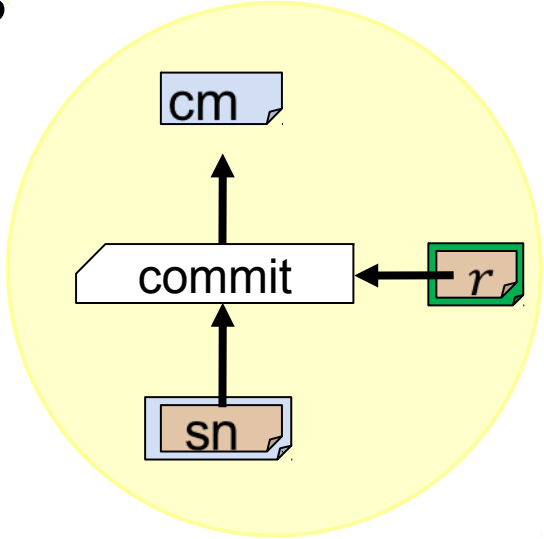
- Smart contracts where some of the computation is performed off-blockchain
  - without sacrificing integrity
  - with zero knowledge
- Scalability
- Privacy-preserving cryptocurrency
  - Zerocash over Ethereum (ZoE)
  - First milestone: Baby ZoE

Bringing zero-knowledge SNARKs to Ethereum

Baby ZoE

- Simple coin mixer contract
- Can deposit and later withdraw privately
- Fixed denomination
- Protocol based on commitments and Merkle trees
  [Sander Ta-Shma 1999]





Baby ZoE implementation components

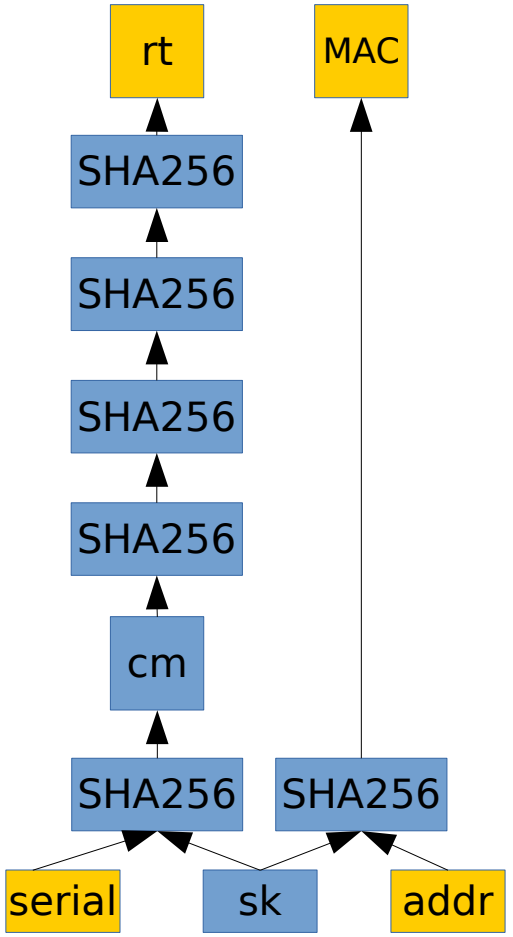
- Arithmetic circuit expressing the NP statement for the zkSNARK
- Contract
- Extend the Ethereum VM to support native SNARK verification
- Wallet

Native SNARK verification

- `snarkverify(vk, proof, public_input)`
  - Verifies the zkSNARK proof with the verification key, given the public input
- Parity EVM patch
- Rust wrapper for libsnark

ZoE Circuit

- Given **serial**, **addr**, **rt**, and **MAC**, there exists ***sk*** such that:
  - ***cm*** := SHA256(***sk*** || **serial**)
  - ***cm*** appears in the depth-4 merkle tree with root **rt**
  - **MAC** = SHA256(**addr** || ***sk***)



BabyZoE contract: Internal State

- Internal Merkle tree for commitments
- Verification key for zkSNARK
- List of spent serial numbers
- List of past Merkle tree roots

BabyZoE contract: Code

```
contract Mixer {
    SnarkPrecompile zksnark =
SnarkPrecompile(0x0000000000000000000000000000000000000005);

    mapping (bytes32 => bool) public serials;
    mapping (bytes32 => bool) public roots;

    struct Mtree {
        uint cur;
        bytes32[16] leaves;
    }

    Mtree public MT;
    bytes public vk;
```

BabyZoE contract: Code

```
function insert(bytes32 com) returns (bool res) {
    if (MT.cur == 16) {
        return false;
    }
    MT.leaves[MT.cur] = com;
    MT.cur++;
    return true;
}
```

BabyZoE contract: Code

```
function deposit(bytes32 com) returns (bool res) {
        if (msg.value != 1 ether) {
            msg.sender.send(msg.value);
            return false;
        }
        if (!insert(com)) {
            msg.sender.send(msg.value);
            return false;
        }
        bytes32 rt = getRoot();
        roots[rt] = true;
        return true;
    }
```

BabyZoE contract: Code

```
function withdraw(bytes32 serial, address addr, bytes32 rt, bytes32 mac, bytes proof) returns (bool success) {
        success = false;
        bytes20 addr_byte = bytes20(addr);
        bytes memory pub = new bytes(128);

        uint i;
        for (i = 0; i < 32; i++) pub[i] = serial[i];
        for (i = 0; i < 20; i++) pub[32 + i] = addr_byte[i];
        for (i = 20; i < 32; i++) pub[32 + i] = 0;
        for (i = 0; i < 32; i++) pub[32*2 + i] = rt[i];
        for (i = 0; i < 32; i++) pub[32*3 + i] = mac[i];

        if (roots[rt] == true) {
            if (!serials[serial]) {
                if (!zksnark.verify_proof(vk, proof, pub)) {
                    return false;
                }
                serials[serial] = true;
                if (!addr.send(1 ether)) {
                    throw;
                }
                else {
                    success = true;
                }
            }
            else {
                return;
            }
        }
        else {
            return;
        }
    }
```

Next steps

- Grown-up ZoE
  - Variable denomination
  - Splitting and joining coins
  - Payment destinations
- EIP for `snarkverify` precompile
- Full wallet
- Hawk over Ethereum
- Cool contracts!

Help baby ZoE grow and prosper

http://github.com/zcash/babyzoe