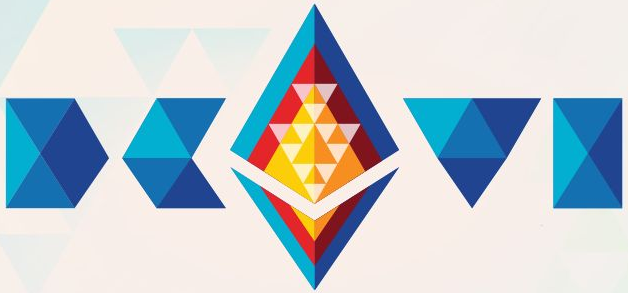


Threshold FHE for Blockchains

How to get confidential shared state

Wei Dai

Research Partner, Bain Capital Crypto

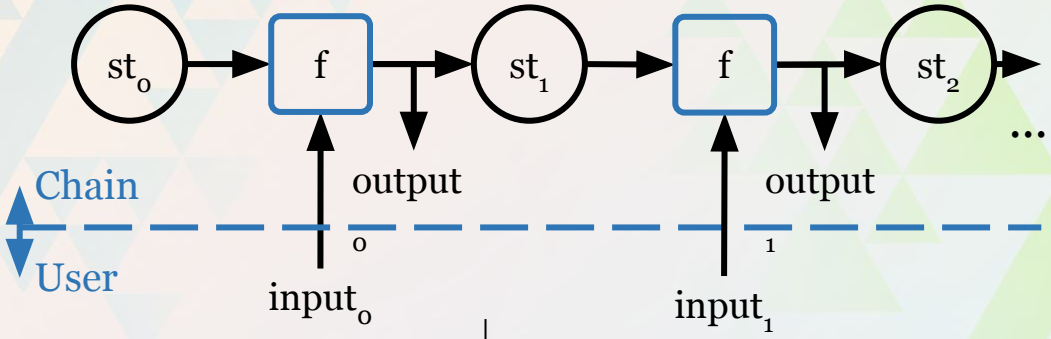
Does zero-knowledge solve ALL privacy problems for blockchains?

Zero-knowledge is NOT the full solution to blockchain privacy

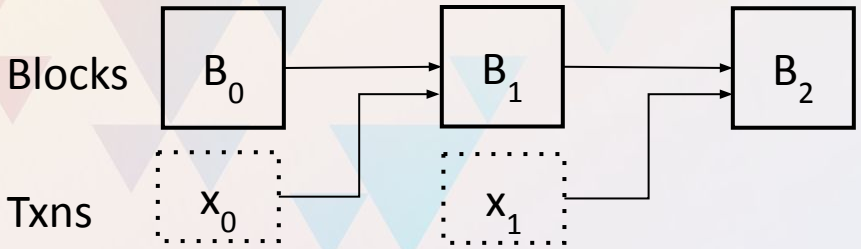
Blockchains are Public State Machines

State Machines

Transition function computes $(st_{i+1}, output_i) = f(st_i, input_i)$



Blockchains

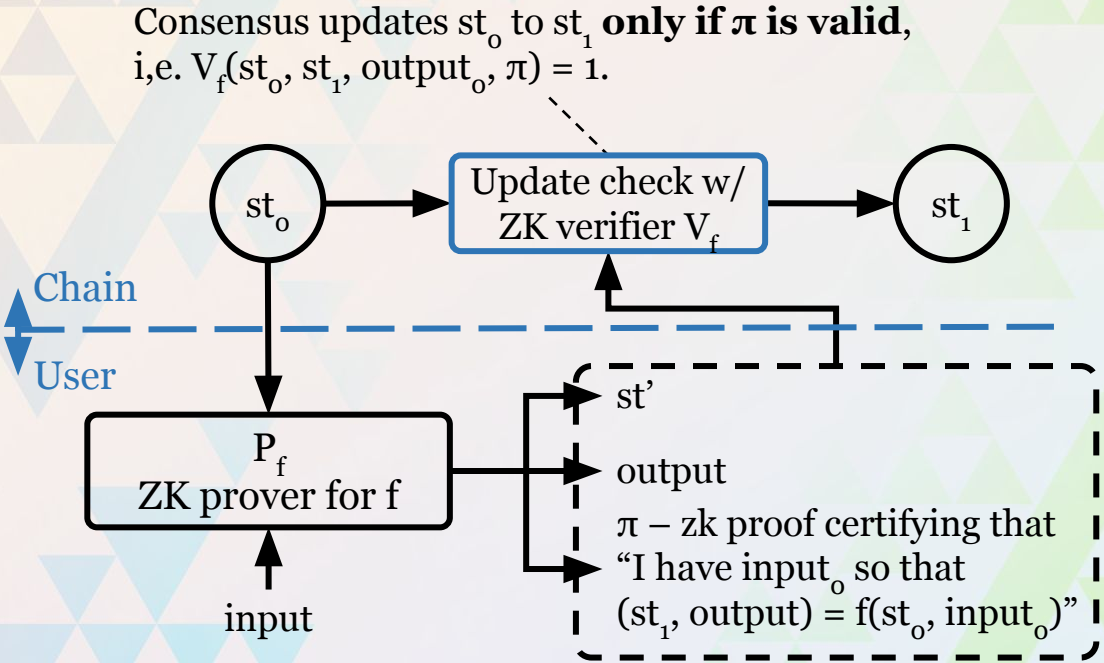


Blockchain State: $st_i = (B_0, ..., B_i)$

Smart Contracts

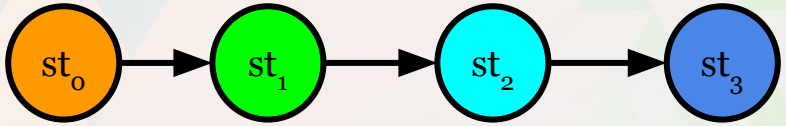
```
Contract LiquidityPool {
    uint public reserveX, reserveY;
    function swapXtoY( … ) public
{
        …
    }
}
```

ZK State Updates (Zexe / Aleo / Mina zkApps)

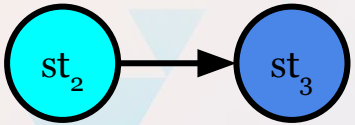


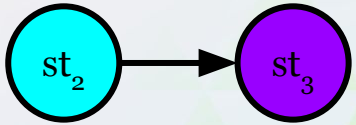
Problem: ZK Updates leads to Race Conditions

ZK Application State



User transactions





Only one state update can be performed.

ZKP smart contracts do not support **shared application state due to race conditions**

“ZK Uniswap” does not give privacy

research

Why you can’t build a private uniswap with ZKPs

Privacy

barryWhiteHat 1 Jul '20

Intro

A Note on Privacy in Constant Function Market Makers

Guillermo Angeris angeris@stanford.edu
Alex Evans alex@placeholder.vc
Tarun Chitra tarun@gauntlet.network

February 2021

But ZK-SNARKs cannot hold private state that nobody knows.
- Vitalik Buterin, “Some ways to use ZK-SNARKs for privacy” June. 2022

Best of both worlds?

Replicated on-chain
BFT-type trust

No privacy
Shared state

ZK off-chain
External trust

Private inputs
No shared state

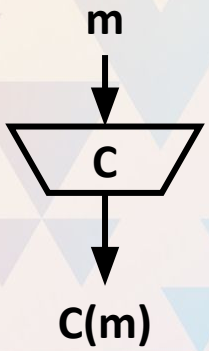
Private input to confidential shared state

Fully Homomorphic Encryption (FHE) to the rescue

Fully Homomorphic Encryption (FHE)

Kg -> (pk, dk)

Fixing any circuit C



m —C→ C(m)

Enc ↓ m

↑ Dec

Requires decryption key dk

Enc(m) —Eval(C, .)→ Enc(C(m))

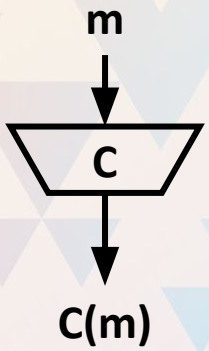
Done by
(Require

Single point of authority to decrypt data

Threshold Fully Homomorphic Encryption (thFHE)

Kg -> (pk, [dk_1, …, dk_n])

Fixing any circuit C



m —C→ C(m)

Enc ↓ m

Dec ↑ C(m)

Requires k out of [dk_1, …, dk_n]

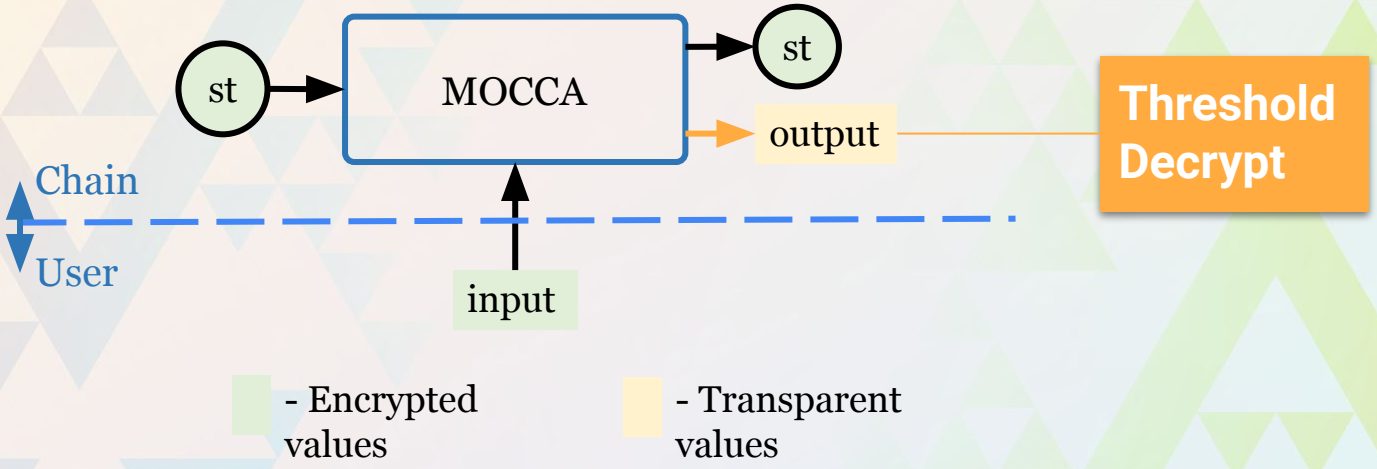
Enc(m) —Eval(C, .)→ Enc(C(m))

Done by (Require

Validator i holds a dk_i

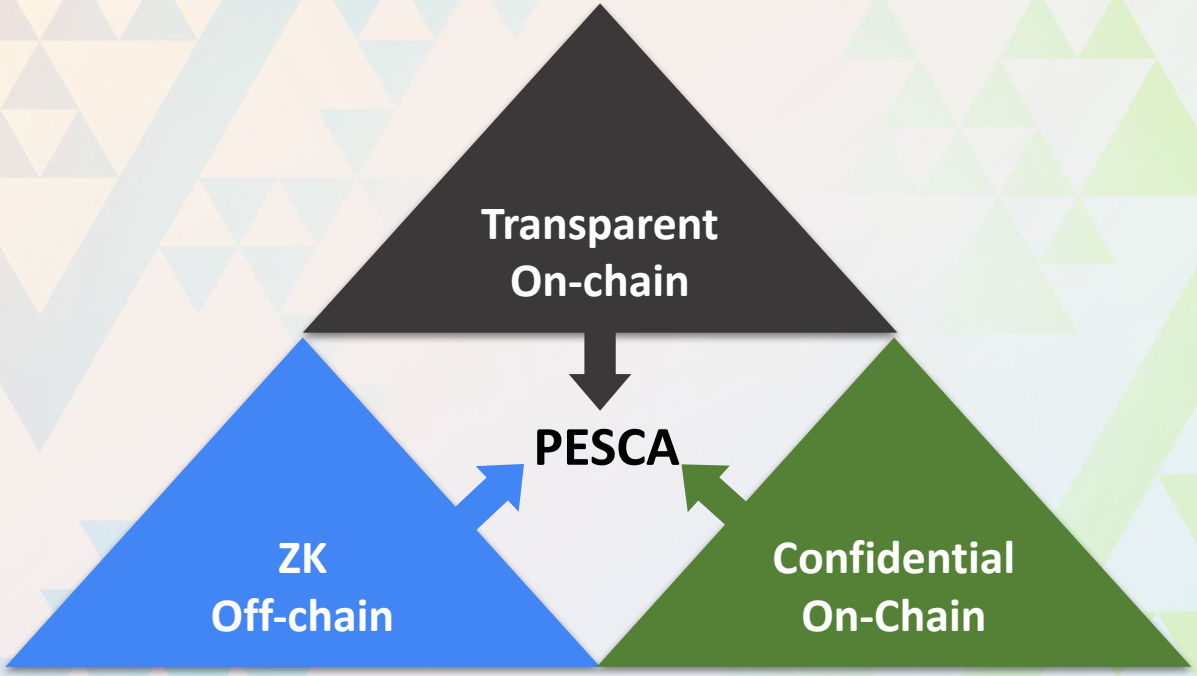
Third type of computation: Confidential On-chain

Magical On-chain Confidential Computing Apparatus



Can compute any function and selectively release information

PESCA: Privacy-Enhancing Smart-Contract Architecture



PESCA in a Nutshell

- **FHE** is slow and expensive
- **ZK** private state is well understood

- **Shared** confidential state (CFMM reservee) **FHE**
- **Private** user state (token balances) **ZK**
- **Connect** via **threshold decryption**

Extending on

PENUMBRA

Two applications:

- Sequentially-settled darkpool (privacy-preserving CFMM)
- First-price privacy-preserving auction

Thank you! & Final Remarks

- **Privacy** is important puzzle-piece to mainstream adoption
- **Privacy beyond anonymity** is possible with **FHE**
- **PESCA** extends on the Penumbra state model to **FHE**

Wei Dai

Research Partner, Bain Capital Crypto

w.dai@baincapital.com

@_weidai

Paper: ia.cr/2022/1119