smartFHE: Privacy-Preserving Smart Contracts from Fully Homomorphic Encryption

Ravital Solomon\textsuperscript{1}, Rick Weber\textsuperscript{1}, Ghada Almashaqbeh\textsuperscript{2}

\textsuperscript{1}Sunscreen, \textsuperscript{2}University of Connecticut

EuroS&P 2023
Big Dreams ...

Blockchain

Miners

Tx addr1 pays addr2 0.005 BTC
Limited functionality

No privacy

Is it all about currency transfer guarded by simple scripts??!!!

My activities can be tracked??!!!
Solutions Went Different Directions

Privacy

Public

Limited

Bitcoin

Functionality
Solutions Went Different Directions

- Privacy
- Public
- Limited

- Bitcoin
- Zcash, ...

Functionality
Solutions Went Different Directions

Privacy

Public

Limited

Bitcoin

Zcash, ...

Functionality
Ethereum was Born in 2015
Bigger Dreams ...

Privacy

Public

Private

Limited

Arbitrary

Zcash, ...

Bitcoin

Ethereum, ...

✔

✔

✔
Privacy-preserving Smart Contracts?

Private Inputs

Private Outputs
More Initiatives

Zether

Hawk

Kachina

Zexe

Zkay

Arbitrum

Ekiden
More Initiatives, But …

- Zether
- Hawk
- Kachina
- Zexe
- Ekiden
- Zkay
- Arbitrum

Limited Functionality!
Overload users!
ZKP-based Approach (Not Us)

Off-chain Private Computing

Compute + Hide & Prove + Verify & Update

Zero Knowledge Proofs (ZKPs)

Not suited for lightweight users 😞
Our Goal

On-chain Private Computing

Hide Inputs & Prove + Verify + Compute & Update

Fully Homomorphic Encryption (FHE) + Zero Knowledge Proof (ZKPs)
Contributions

A formal notion for privacy-preserving smart contracts (PPSCs) capturing arbitrary computation with I/O privacy.
Contributions

A formal notion for privacy-preserving smart contracts (PPSCs) capturing arbitrary computation with I/O privacy.

smartFHE framework: the first scheme to use FHE in the blockchain model!
Contributions

A formal notion for privacy-preserving smart contracts (PPSCs) capturing arbitrary computation with I/O privacy.

smartFHE framework: the first scheme to use FHE in the blockchain model!

smartFHE instantiation.
Contributions

A formal notion for privacy-preserving smart contracts (PPSCs) capturing arbitrary computation with I/O privacy.

smartFHE framework: the first scheme to use FHE in the blockchain model!

smartFHE instantiation.

Formal security proofs and implementation/benchmarks
  - The first library for short-discrete log proofs
smartFHE Framework

- Privacy extensions for a public smart contract-enabled blockchain, e.g., Ethereum
- Flexible and modular.
- Supports:
  - Private and public payments
  - Private and public smart contracts
- A user can have public and private accounts
smartFHE Framework

- **Network protocol operations:**
  - Private accounts: FHE keypairs and signature keypairs. Encrypted balance.
  - Private payments:
    - Shield
    - PrivTransfer
    - Deshield
  - Private smart contracts: contract-dependent, translated into FHE operations.
Several Challenges

- Working with FHE
- Combining FHE with ZKP
- Concurrency
smartFHE Instantiation

FHE: BFV scheme

+ 

ZKPs: Short discrete-log proofs + Bulletproofs

+ 

Signatures: ECDSA and/or Falcon
Implementation

- **Existing libraries:**
  - Microsoft SEAL for BFV
  - Dalek for Bulletproofs
  - OpenSSL for ECDSA

- **New library:**
  - First implementation of short-discrete log proofs with Apple Metal GPU-accelerated code.

- Benchmarks on Apple M2 Max with 64GB RAM
## Results - Benchmarks

### TABLE 1: Setup times (one time cost)

<table>
<thead>
<tr>
<th>Performed by</th>
<th>Operation</th>
<th>$d = 1024$</th>
<th>$d = 2048$</th>
<th>$d = 4096$</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>KeyGen</td>
<td>0.216 ms</td>
<td>0.375 ms</td>
<td>36.5 ms</td>
</tr>
<tr>
<td>System</td>
<td>ZKP setup</td>
<td>0.8 s</td>
<td>2.06 s</td>
<td>5.7 s</td>
</tr>
</tbody>
</table>

### TABLE 2: Private transaction costs for smartFHE’s instantiation—user side.

<table>
<thead>
<tr>
<th>$d = 1024$</th>
<th>Operation</th>
<th>Time (s)</th>
<th>Size (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Shield($tx_{shield}$)</td>
<td>0.0002</td>
<td>0.101</td>
</tr>
<tr>
<td></td>
<td>Deshield($tx_{deshield}$)</td>
<td>1.89</td>
<td>2.47</td>
</tr>
<tr>
<td></td>
<td>PrivTransfer($tx_{privtransf}$)</td>
<td>3.57</td>
<td>20.03</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$d = 2048$</th>
<th>Operation</th>
<th>Time (s)</th>
<th>Size (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Shield($tx_{shield}$)</td>
<td>0.0002</td>
<td>0.101</td>
</tr>
<tr>
<td></td>
<td>Deshield($tx_{deshield}$)</td>
<td>3.58</td>
<td>2.53</td>
</tr>
<tr>
<td></td>
<td>PrivTransfer($tx_{privtransf}$)</td>
<td>10.7</td>
<td>64.76</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$d = 4096$</th>
<th>Operation</th>
<th>Time (s)</th>
<th>Size (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Shield($tx_{shield}$)</td>
<td>0.0002</td>
<td>0.101</td>
</tr>
<tr>
<td></td>
<td>Deshield($tx_{deshield}$)</td>
<td>11.17</td>
<td>2.66</td>
</tr>
<tr>
<td></td>
<td>PrivTransfer($tx_{privtransf}$)</td>
<td>23.89</td>
<td>180.1</td>
</tr>
</tbody>
</table>
Results - Benchmarks

TABLE 3: Private transaction costs for smartFHE’s instantiation—miner side.

<table>
<thead>
<tr>
<th></th>
<th>Operation</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d = 1024$</td>
<td>VerifyShield</td>
<td>0.00017</td>
</tr>
<tr>
<td></td>
<td>VerifyDeshield</td>
<td>0.92</td>
</tr>
<tr>
<td></td>
<td>VerifyPrivTransfer</td>
<td>1.95</td>
</tr>
<tr>
<td>$d = 2048$</td>
<td>VerifyShield</td>
<td>0.00017</td>
</tr>
<tr>
<td></td>
<td>VerifyDeshield</td>
<td>1.92</td>
</tr>
<tr>
<td></td>
<td>VerifyPrivTransfer</td>
<td>6.37</td>
</tr>
<tr>
<td>$d = 4096$</td>
<td>VerifyShield</td>
<td>0.00017</td>
</tr>
<tr>
<td></td>
<td>VerifyDeshield</td>
<td>6.42</td>
</tr>
<tr>
<td></td>
<td>VerifyPrivTransfer</td>
<td>14.77</td>
</tr>
</tbody>
</table>
Results - Comparison

TABLE 4: Base private transaction costs for Veri-zexe.

<table>
<thead>
<tr>
<th>no. of inputs × no. of outputs</th>
<th>User generation time (s)</th>
<th>Miner verification time (ms)</th>
<th>Size (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 × 2</td>
<td>27.82</td>
<td>13.21</td>
<td>4.82</td>
</tr>
<tr>
<td>3 × 3</td>
<td>54.9</td>
<td>13.14</td>
<td>4.88</td>
</tr>
<tr>
<td>4 × 4</td>
<td>59</td>
<td>13.15</td>
<td>4.95</td>
</tr>
<tr>
<td>8 × 8</td>
<td>121</td>
<td>13.15</td>
<td>5.2</td>
</tr>
</tbody>
</table>

smartHE allows a user to issue payments at a rate 1.16x - 7.79x faster than Veri-zexe
### Results - Applications

**TABLE 5: Private smart contract application costs.**

<table>
<thead>
<tr>
<th>Application</th>
<th>Per user generation time (s)</th>
<th>Miner verification/computing time (s)</th>
<th>Size (KB) per user</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMM ($d = 2048$)</td>
<td>6.4</td>
<td>3.58</td>
<td>33.53</td>
</tr>
<tr>
<td>AMM ($d = 4096$)</td>
<td>20.88</td>
<td>12.64</td>
<td>91.4</td>
</tr>
<tr>
<td>Mean/variance ($d = 4096$)</td>
<td>20.89</td>
<td>62.9</td>
<td>91.45</td>
</tr>
<tr>
<td>Chi-squared ($d = 4096$)</td>
<td>23.89</td>
<td>44.39</td>
<td>26.95</td>
</tr>
</tbody>
</table>
Conclusion and Future Work

● This work
  ○ A privacy-preserving smart contract framework (and instantiation) from FHE and ZKP
  ○ Formal treatment
  ○ Implementation/testing

● Future work
  ○ Look into instantiations using other FHE/ZKP schemes
  ○ Addressing anonymity
  ○ Handling storage cost
Thank you!

Questions?