Programmable Cryptography

gubsheep (0xPARC) - DEVCON VI
What is 0xPARC?
At 0xPARC, we think a lot about applied ZK.
Here are two things that ZK allows us to do.
COOL THING #1:

ZK gives us an expressive language for claims
Example: zkSNARKs and membership proofs

Let's look at identity claims!
Example: zkSNARKs and membership proofs
Let’s look at identity claims!

🥳 I know a private key corresponding to Alice’s public key.
Example: zkSNARKs and membership proofs

Let’s look at identity claims!

😊 I know a private key corresponding to Alice, Bob, OR Charlie’s public keys.
Example: zkSNARKs and membership proofs
Let’s look at identity claims!

😊 I know a private key corresponding to Alice, Bob, OR Charlie’s public keys.

- myHash := mimc(secret)
- (myHash - hash1)(myHash - hash2)(myHash - hash3)... == 0
- msgAttestation := mimc(msg, secret)
Example: zkSNARKs and membership proofs

Let’s look at identity claims!

🤔 I know a private key corresponding to Alice, Bob, OR Charlie’s public keys, and the other two [can/can’t] prove that they did NOT generate this message.
Example: zkSNARKs and membership proofs

Let’s look at identity claims!

😊 I know a private key corresponding to Alice, Bob, OR Charlie’s public key, and I either possess a signed attestation from one of {David, Eve, Fred}, or during the block with header X, I knew the private key corresponding to an account with at least 32ETH, and...
zkSNARKs turn math problems into programming tasks.
COOL THING #2:

ZK adds interoperability to cryptographic systems
SNARK-friendly vs. SNARK-compatible

At least in the near-term, our most widely-used cryptographic systems will not be SNARK-friendly.

The underlying cryptography for many of these systems was invented before SNARK constructions were known!
Example: Key distribution and identity registries

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Email</th>
<th>Created</th>
<th>Fingerprint</th>
<th>Validity</th>
</tr>
</thead>
<tbody>
<tr>
<td>sec/pub</td>
<td>Alice</td>
<td><a href="mailto:alice@example.org">alice@example.org</a></td>
<td>31.07.17</td>
<td>D4CC A4B3 19AA CC70 B48A 89D2 D67F B78F FE84 21A7</td>
<td></td>
</tr>
<tr>
<td>pub</td>
<td>GPGTools Team</td>
<td><a href="mailto:team@gpgtools.org">team@gpgtools.org</a></td>
<td>19.08.10</td>
<td>E5E3 8F69 46B 44C1 EC9F B07B 7607 8F05 00D0 26C4</td>
<td></td>
</tr>
</tbody>
</table>

2 of 2 keys listed
Example: Key distribution and identity registries
Example: Key distribution and identity registries

View Wallet Info

YOUR ADDRESS

0x698042d6233042632711C86452A53A8E9637F585

PRIVATE KEY (UNENCRYPTED)

A2fc86c38a1a7fb6c0eaea9696d6434cd977dbef46fba3183ac99ad
Lots of existing cryptography can at least be made SNARK-compatible.
Both of these features are examples of the power of “programmable cryptography.”
Programmable cryptography is cryptography that can be “layered” on top of arbitrary computations.
Cryptography

For most of cryptography’s (short) history, the set of mechanisms we’ve been able to instantiate with it has been extremely narrow.

‣ This message originated from Alice.
‣ This message can only be read by Bob.

Every new mechanism needed a special-purpose-built mathematical protocol!
zkSNARKs

- This message originated from Alice.
- I know a private key corresponding to Alice, Bob, OR Charlie’s public key, and I either possess a signed attestation from one of {David, Eve, Fred}, or during the block with header X, I knew the private key corresponding to an account with at least 32ETH, and...
Witness Encryption

- Charlie has published some secret vote that only a coordinator can read.
- Charlie has committed to some secret vote, that only attestors with a certain permission level can decrypt today, but which a class of auditors with a lower permission level will be able to partially decrypt in one week.
Smart Contracts

- Bob can decrement his balance by 100 ether, to increment Alice’s balance by 100 ether.

- At block B, 100 ether will be available for withdrawal by Bob, so long as Bob has closed his position in X smart contract and no one has submitted a fraud challenge, though an early withdrawal may be initiated if 2 of the 3 solvency conditions are met...
Programmable Cryptography

zkSNARKs

- Proofs of specific claims $\rightarrow$ General-purpose claim language
Programmable Cryptography

zkSNARKs

- Proofs of specific claims $\rightarrow$ General-purpose claim language

Smart contracts

- Canonical data that can be modified in specific ways $\rightarrow$ General-purpose language for modifying canonical data
Programmable Cryptography

zkSNARKs
- Proofs of specific claims $\rightarrow$ General-purpose claim language

Smart contracts
- Canonical data that can be modified in specific ways $\rightarrow$ General-purpose language for modifying canonical data

Witness encryption
- Data that can be read by a specific set of people $\rightarrow$ Language for specifying arbitrary predicates for read permissions
Programmable Cryptography

zkSNARKs

- Proofs of specific claims $\rightarrow$ General-purpose claim language

Smart contracts

- Canonical data that can be modified in specific ways $\rightarrow$ General-purpose language for modifying canonical data

Witness encryption

- Data that can be read by a specific set of people $\rightarrow$ Language for specifying arbitrary predicates for read permissions

FHE, MPC, IO, ...
Programmable Cryptography and Blockchains
Ethereum: the global stream of consciousness

A 1gbps “coaxial cable” streaming canonical data: humanity’s promises, bets, secrets, debts, dreams,

...that any person or computing device in the world can hook into.
Ethereum: the global stream of consciousness

Right now, this stream is completely transparent.

This is currently necessary to build acceptance that the stream is canonical —“don’t trust, verify.”
Privacy is important, not just as a matter of ideology, but as a matter of mechanics.
Blockchains and Programmable Cryptography

rwX permissions on this canonical data stream are enabled by programmable cryptography.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>no permissions</td>
</tr>
<tr>
<td>1</td>
<td>-</td>
<td>-</td>
<td>x</td>
<td>only execute</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>w</td>
<td>-</td>
<td>only write</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>w</td>
<td>x</td>
<td>write and execute</td>
</tr>
<tr>
<td>4</td>
<td>r</td>
<td>-</td>
<td>-</td>
<td>only read</td>
</tr>
<tr>
<td>5</td>
<td>r</td>
<td>-</td>
<td>x</td>
<td>read and execute</td>
</tr>
<tr>
<td>6</td>
<td>r</td>
<td>w</td>
<td>-</td>
<td>read and write</td>
</tr>
<tr>
<td>7</td>
<td>r</td>
<td>w</td>
<td>x</td>
<td>read, write and execute</td>
</tr>
</tbody>
</table>
Player
Private, locally stored

(State 1)
s1

Network
Public, verifiable by anyone

C1 = hash(s1)
P1 = proof(s1, C1)

C2 = hash(s2)
P2 = proof(s1, C1, s2, C2)
“I walk into a store and perform a cryptographic handshake with the merchant and an identity provider. After verifying their identity, I give them one token that permissions them to access some specific data on my preferences for 60m, and another that allows them to transfer a limited amount from my balance. Then, I update my transaction history which is committed to on-chain but only visible to myself.”

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>no permissions</td>
</tr>
<tr>
<td>1</td>
<td>-</td>
<td>-</td>
<td>x</td>
<td>only execute</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>w</td>
<td>-</td>
<td>only write</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>w</td>
<td>x</td>
<td>write and execute</td>
</tr>
<tr>
<td>4</td>
<td>r</td>
<td>-</td>
<td>-</td>
<td>only read</td>
</tr>
<tr>
<td>5</td>
<td>r</td>
<td>-</td>
<td>x</td>
<td>read and execute</td>
</tr>
<tr>
<td>6</td>
<td>r</td>
<td>w</td>
<td>-</td>
<td>read and write</td>
</tr>
<tr>
<td>7</td>
<td>r</td>
<td>w</td>
<td>x</td>
<td>read, write and execute</td>
</tr>
</tbody>
</table>
As more of our social and economic activity move online, we’ll need digital “ender chests.”
@0xPARC
0xPARC.org