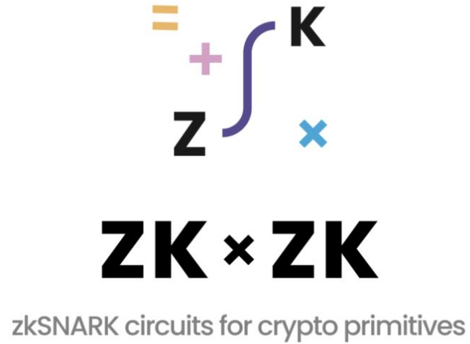
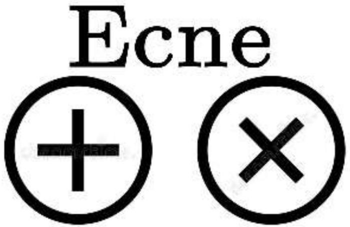
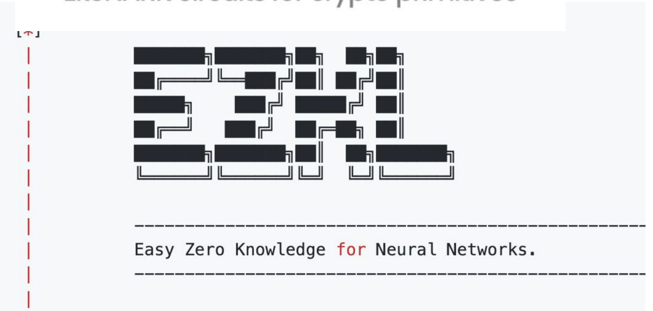
Programmable Cryptography

gubsheep (0xPARC) - DEVCON VI

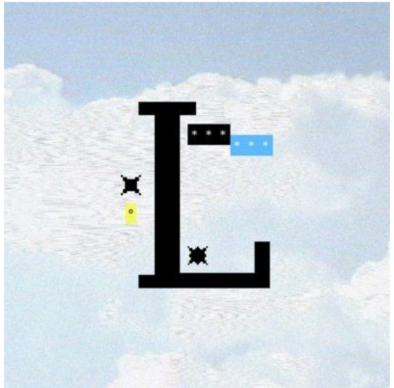
What is 0xPARC?







```
Easy Zero Knowledge for Neural Networks.
```



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:

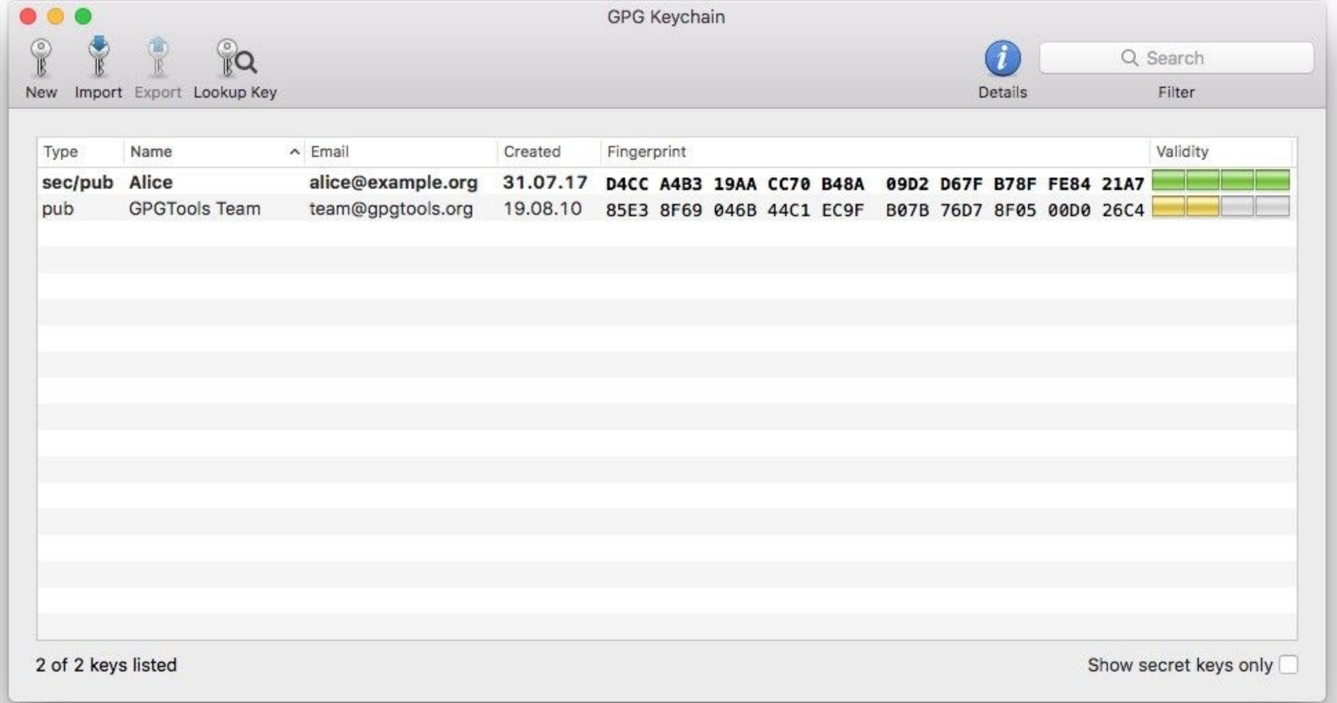
ZK adds interoperability to cryptographic systems

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

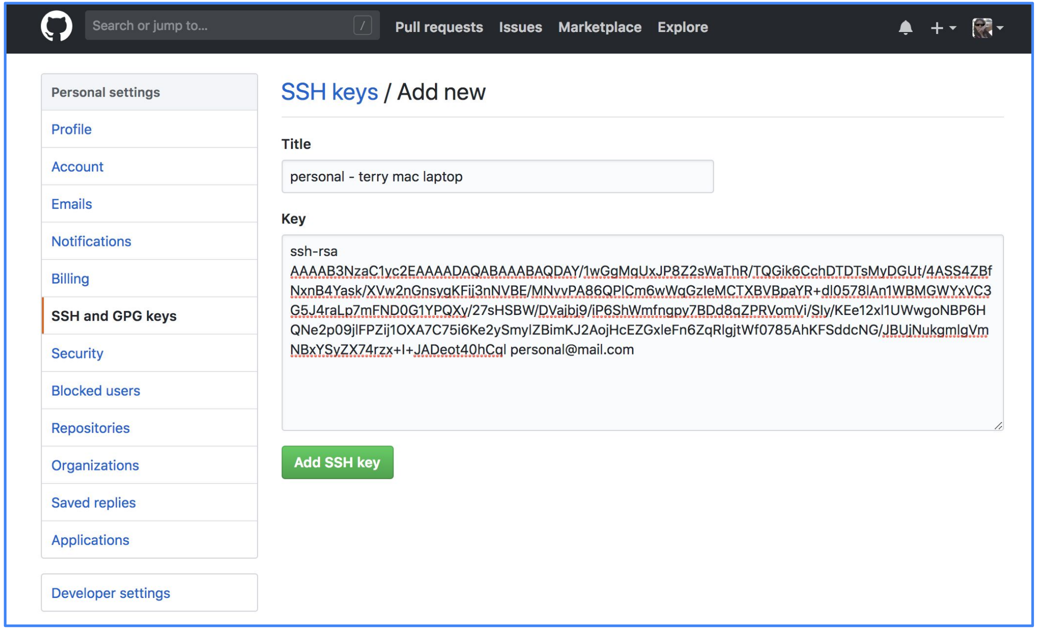


Type	Name	Email	Created	Fingerprint	Validity
sec/pub	**Alice**	**alice@example.org**	**31.07.17**	**D4CC A4B3 19AA CC70 B48A 09D2 D67F B78F FE84 21A7**	🟩🟩🟩🟩
pub	GPGTools Team	team@gpgtools.org	19.08.10	85E3 8F69 046B 44C1 EC9F B07B 76D7 8F05 00D0 26C4	🟨🟨⬜⬜

2 of 2 keys listed

Show secret keys only □

Example: Key distribution and identity registries



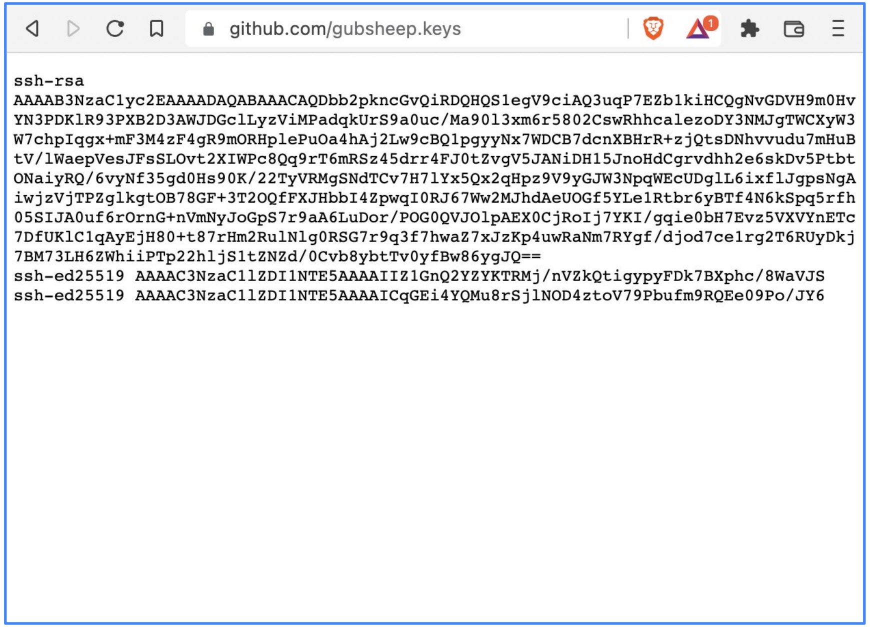
SSH keys / Add new

Title

personal - terry mac laptop

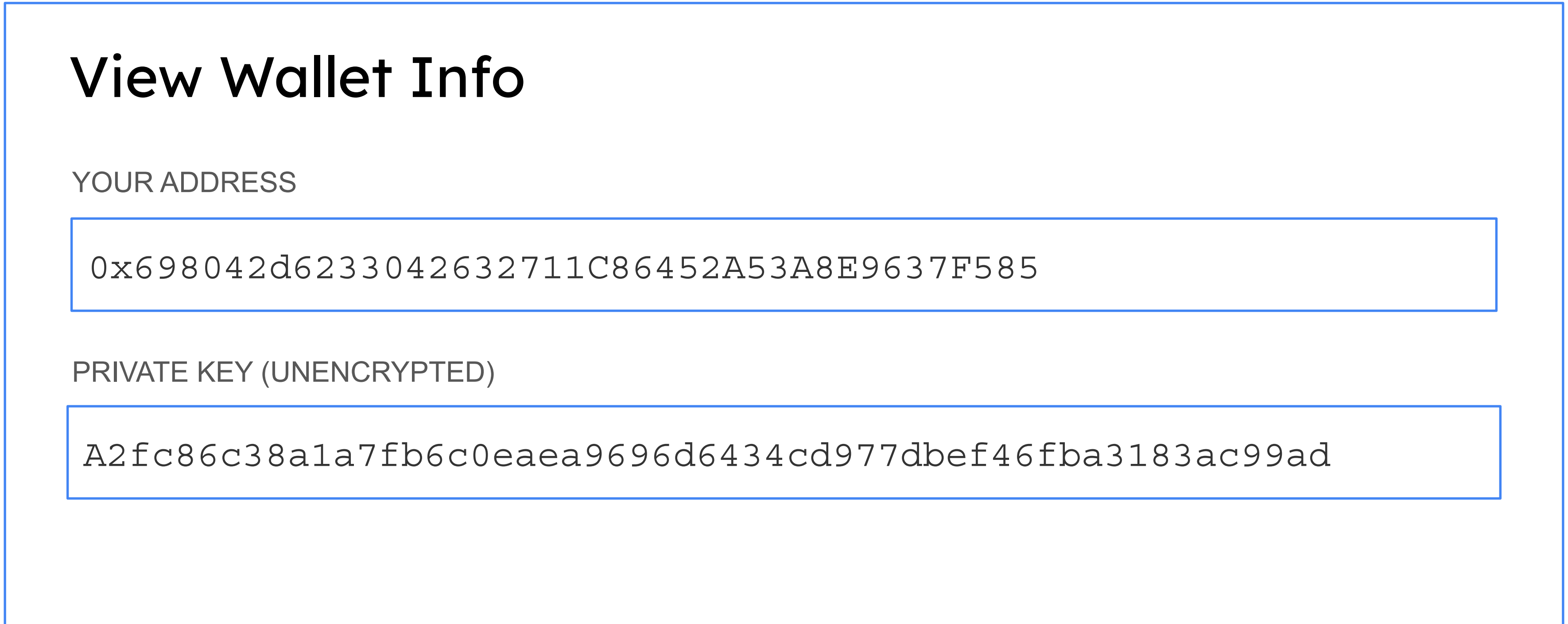
Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDAY/1wGgMqUxJP8Z2sWaThR/TQGik6CchDTDTsMyDGUt/4ASS4ZBf
NxnB4Yask/XVw2nGnsygKFij3nNVBE/MNvvPA86QPlCm6wWqGzIeMCTXBVBpaYR+dl0578lAn1WBMGWYxVC3
G5J4raLp7mFND0G1YPQXy/27sHSBW/DVajbj9/iP6ShWmfngpy7BDd8qZPRVomVi/Sly/KEe12xl1UWwgoNBP6H
QNe2p09jlFPZij1OXA7C75i6Ke2ySmylZBimKJ2AojHcEZGxleFn6ZqRlgjtWf0785AhKFSddcNG/JBUjNukgmlgVm
NBxYSyZX74rzx+I+JADeot40hCql personal@mail.com
```



```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAACAQDbb2pkncGvQiRDQHQS1egV9ciAQ3uqP7EZb1kiHCQgNvGDVH9m0Hv
YN3PDKlR93PXB2D3AWJDGclLyzViMPadqkUrS9a0uc/Ma90l3xm6r5802CswRhhcalezoDY3NMJgTWCXyW3
W7chpIqgx+mF3M4zF4gR9mORHplePuOa4hAj2Lw9cBQ1pgyyNx7WDCB7dcnXBHrR+zjQtsDNhvvudu7mHuB
tV/lWaepVesJFsSLOvt2XIWPc8Qq9rT6mRSz45drr4FJ0tZvgV5JANiDH15JnoHdCgrvdhh2e6skDv5Ptbt
ONaiyRQ/6vyNf35gd0Hs90K/22TyVRMgSNdTCv7H7lYx5Qx2qHpz9V9yGJW3NpqWEcUDglL6ixflJgpsNgA
iwjzVjTPZglkgtOB78GF+3T2OQfFXJHbbI4ZpwqI0RJ67Ww2MJhdAeUOGf5YLe1Rtbr6yBTf4N6kSpq5rfh
05SIJA0uf6rOrnG+nVmNyJoGpS7r9aA6LuDor/POG0QVJOlpAEX0CjRoIj7YKI/gqie0bH7Evz5VXVYnETc
7DfUKlC1qAyEjH80+t87rHm2RulNlg0RSG7r9q3f7hwaZ7xJzKp4uwRaNm7RYgf/djod7ce1rg2T6RUyDkj
7BM73LH6ZWhiiPTp22hljS1tZNZd/0Cvb8ybtTv0yfBw86ygJQ==
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIIZ1GnQ2YZYKTRMj/nVZkQtigypyFDk7BXphc/8WaVJS
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAICqGEi4YQMu8rSjlNOD4ztoV79Pbufm9RQEe09Po/JY6
```

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 → General-purpose claim language

Programmable Cryptography

zkSNARKs

- Proofs of specific claims → General-purpose claim language

Smart contracts

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

Programmable Cryptography

zkSNARKs

- Proofs of specific claims → General-purpose claim language

Smart contracts

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

Witness encryption

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

Programmable Cryptography

zkSNARKs

- Proofs of specific claims → General-purpose claim language

Smart contracts

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

Witness encryption

- Data that can be read by a specific set of people → 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

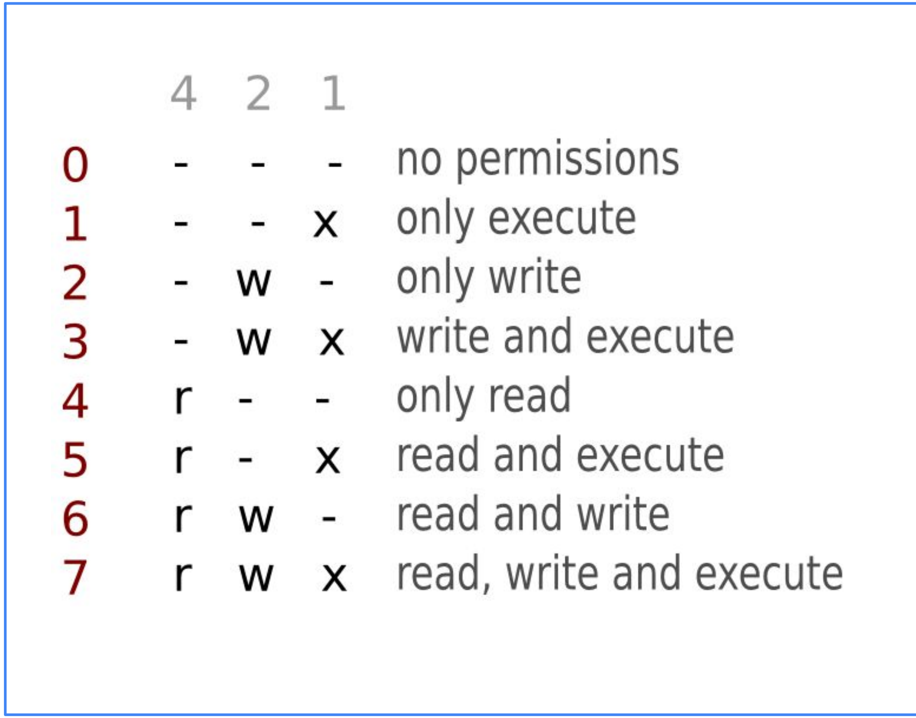
Right now, this stream is completely transparent.

This is currently necessary to build acceptance that the stream is canonical —“don’t trust, verify.”

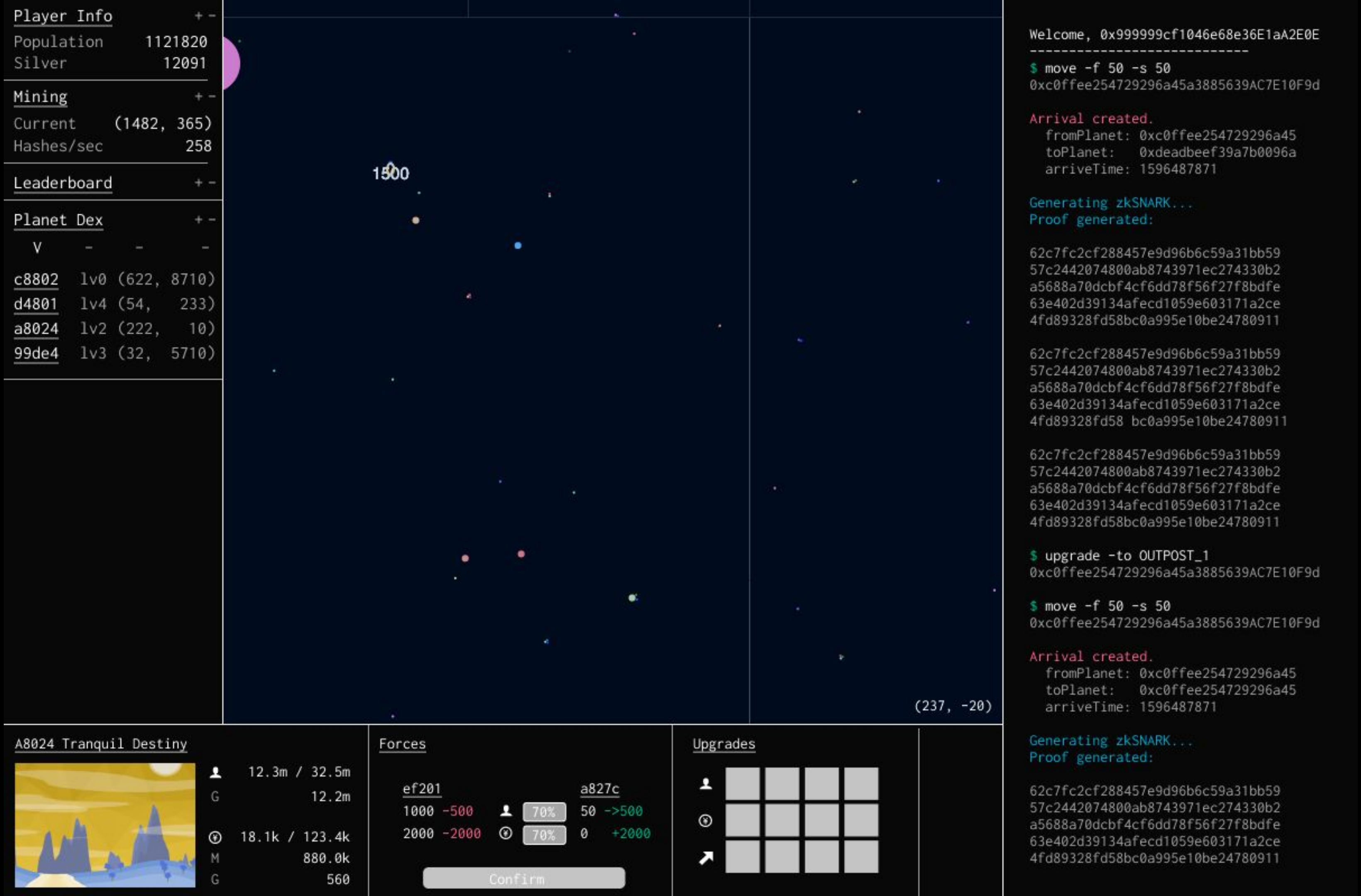
Privacy is important,
not just as a matter of ideology,
but as a matter of mechanics

Blockchains and Programmable Cryptography

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



	4	2	1	
0	-	-	-	no permissions
1	-	-	x	only execute
2	-	w	-	only write
3	-	w	x	write and execute
4	r	-	-	only read
5	r	-	x	read and execute
6	r	w	-	read and write
7	r	w	x	read, write and execute



Player Info

Population	1121820
Silver	12091

Mining

Current	(1482, 365)
Hashes/sec	258

Leaderboard

Planet Dex

V	-	-	-
c8802	lv0	(622,	8710)
d4801	lv4	(54,	233)
a8024	lv2	(222,	10)
99de4	lv3	(32,	5710)

A8024 Tranquil Destiny

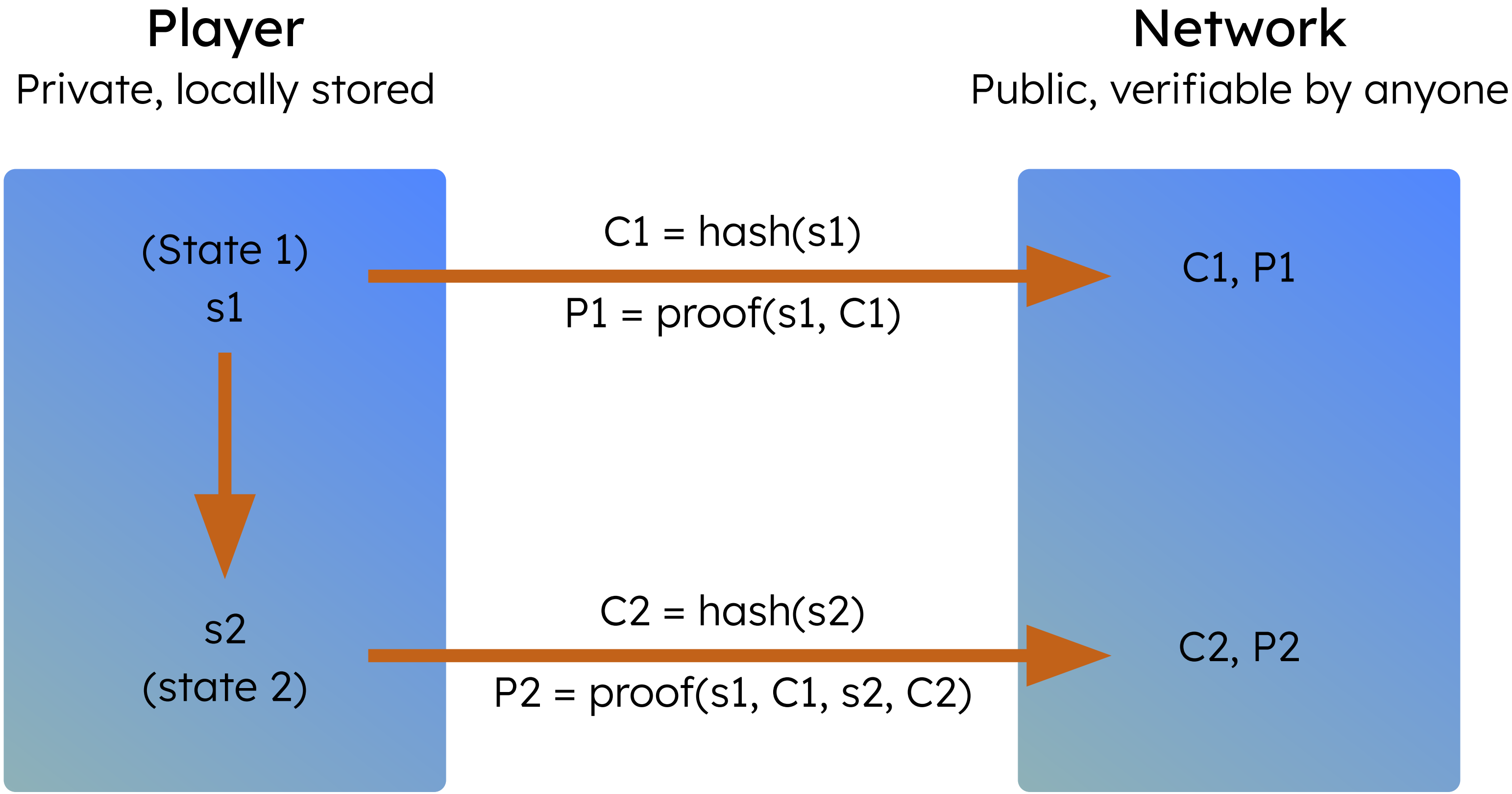
👤	12.3m / 32.5m
G	12.2m
ⓢ	18.1k / 123.4k
M	880.0k
G	560

Forces

ef201			a827c
1000 -500	👤	70%	50 ->500
2000 -2000	ⓢ	70%	0 +2000

Upgrades

```
Welcome, 0x999999cf1046e68e36E1aA2E0E
----------------------------
$ move -f 50 -s 50
0xc0ffee254729296a45a3885639AC7E10F9d

Arrival created.
  fromPlanet: 0xc0ffee254729296a45
  toPlanet:   0xdeadbeef39a7b0096a
  arriveTime: 1596487871

Generating zkSNARK...
Proof generated:

62c7fc2cf288457e9d96b6c59a31bb59
57c2442074800ab8743971ec274330b2
a5688a70dcbf4cf6dd78f56f27f8bdfe
63e402d39134afecd1059e603171a2ce
4fd89328fd58bc0a995e10be24780911

62c7fc2cf288457e9d96b6c59a31bb59
57c2442074800ab8743971ec274330b2
a5688a70dcbf4cf6dd78f56f27f8bdfe
63e402d39134afecd1059e603171a2ce
4fd89328fd58 bc0a995e10be24780911

62c7fc2cf288457e9d96b6c59a31bb59
57c2442074800ab8743971ec274330b2
a5688a70dcbf4cf6dd78f56f27f8bdfe
63e402d39134afecd1059e603171a2ce
4fd89328fd58bc0a995e10be24780911

$ upgrade -to OUTPOST_1
0xc0ffee254729296a45a3885639AC7E10F9d

$ move -f 50 -s 50
0xc0ffee254729296a45a3885639AC7E10F9d

Arrival created.
  fromPlanet: 0xc0ffee254729296a45
  toPlanet:   0xc0ffee254729296a45
  arriveTime: 1596487871

Generating zkSNARK...
Proof generated:

62c7fc2cf288457e9d96b6c59a31bb59
57c2442074800ab8743971ec274330b2
a5688a70dcbf4cf6dd78f56f27f8bdfe
63e402d39134afecd1059e603171a2ce
4fd89328fd58bc0a995e10be24780911
```



Player

Private, locally stored

Network

Public, verifiable by anyone

(State 1)

s1

C1 = hash(s1)

P1 = proof(s1, C1)

C1, P1

s2

(state 2)

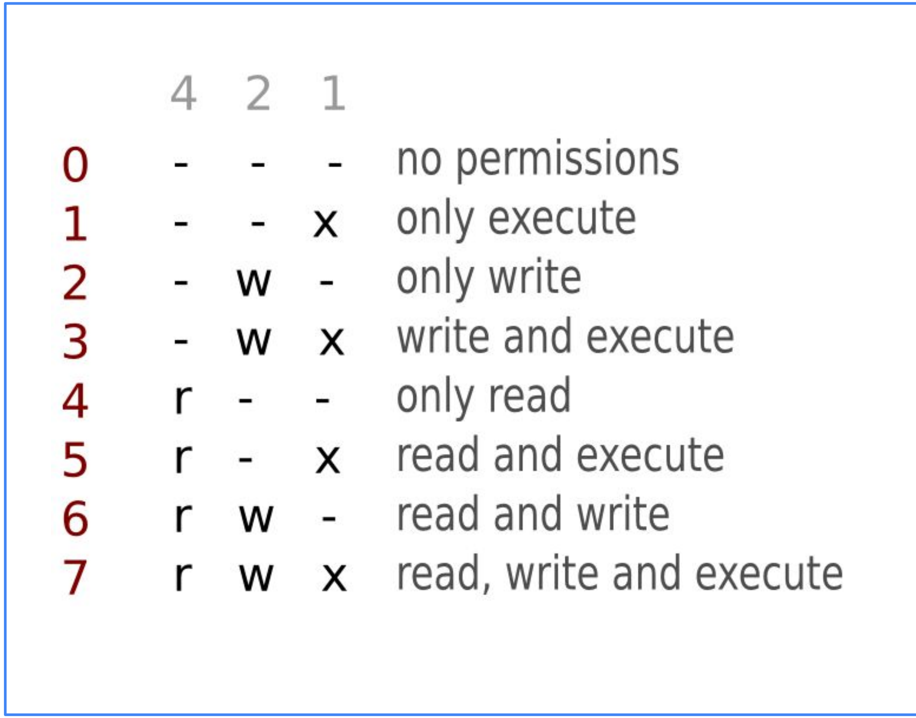
C2 = hash(s2)

P2 = proof(s1, C1, s2, C2)

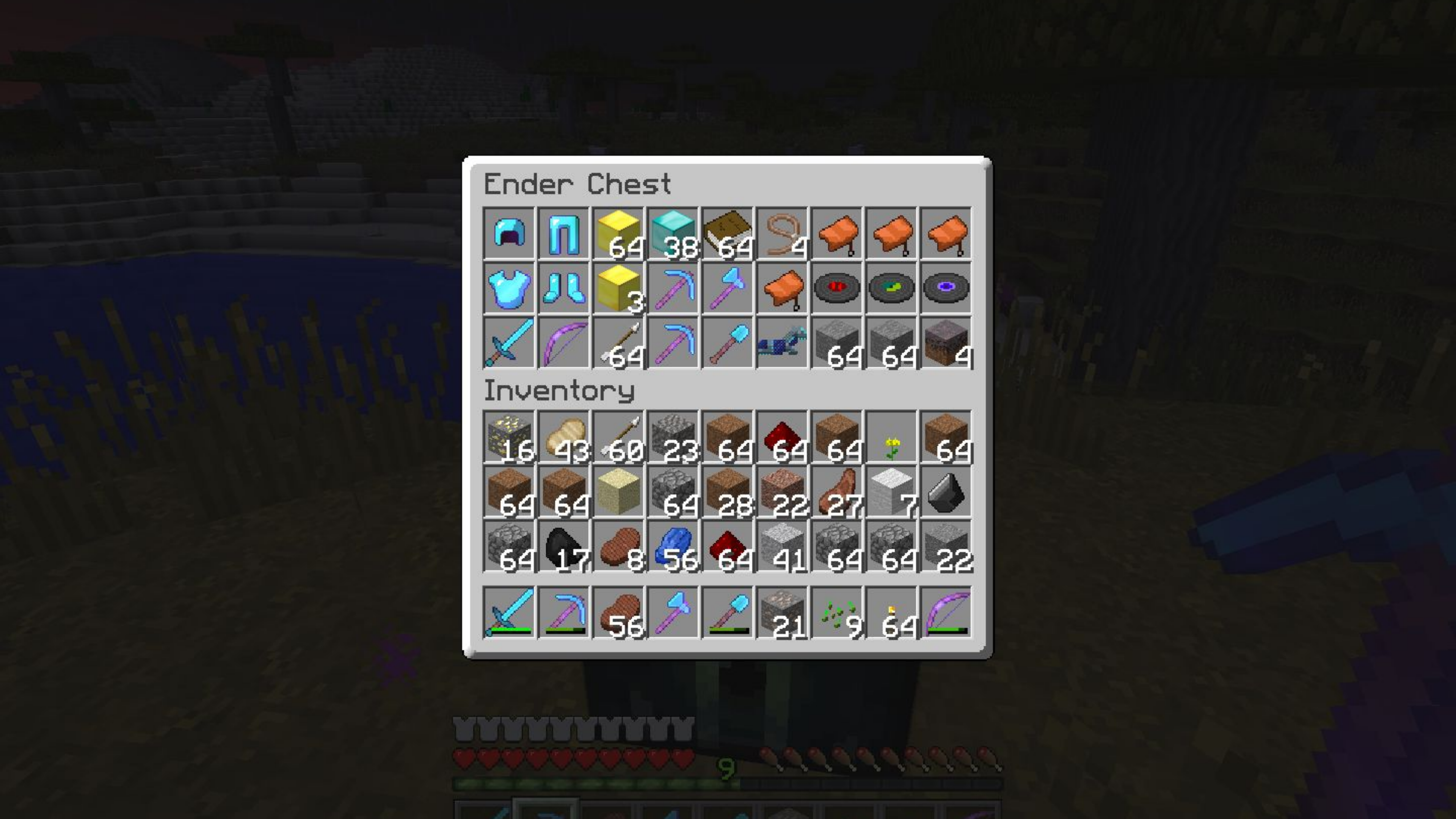
C2, P2

Blockchains and Programmable Cryptography

> “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.”



	4	2	1	
0	-	-	-	no permissions
1	-	-	x	only execute
2	-	w	-	only write
3	-	w	x	write and execute
4	r	-	-	only read
5	r	-	x	read and execute
6	r	w	-	read and write
7	r	w	x	read, write and execute


Ender Chest
64 38 64 4
3
64 64 64 4
Inventory
16 43 60 23 64 64 64 64
64 64 64 28 22 27 7
64 17 8 56 64 41 64 64 22
56 21 9 64
9

As more of our social and
economic activity move online,
we’ll need digital “ender chests.”

@0xPARC

0xPARC.org