TORNADO.CASH

By Simon CHEREL & Enguerrand DECLERCQ

Contents

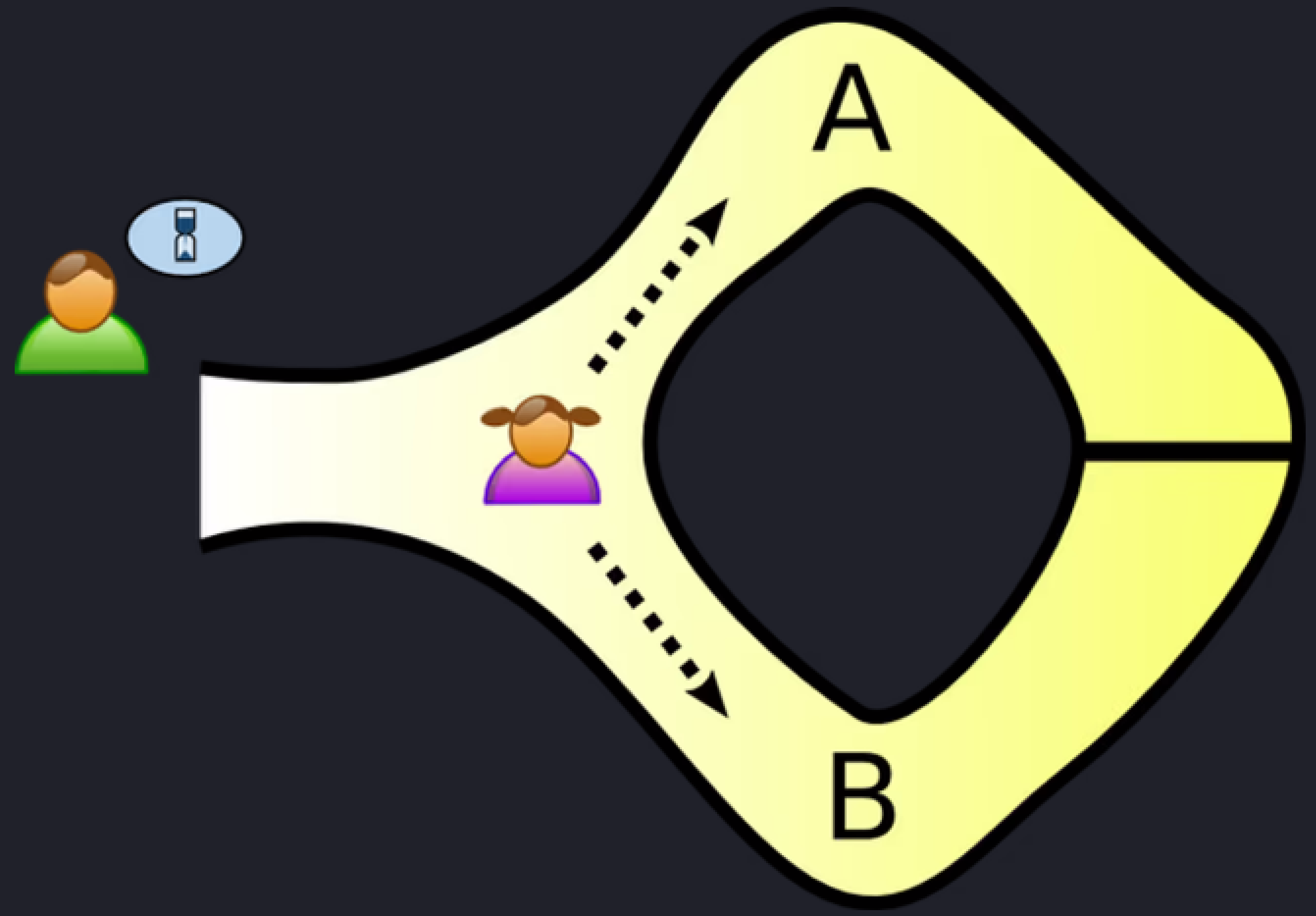
Topics to tackle

1. ZKP reminder
2. Deep down into zk-SNARK
3. Tornado
4. Questions

ZKP

A cryptosystem

- Prover and Verifier want to prove $\exists w \mid Q(x,w)=0$
- Anyone has access to the context x
- Prover has a secret w
- Verifier wants to verify $Q(x,w)=0$
- Prover and verifier cannot share w
- Prover sends proof π which does not reveal w but verifies $Q(x,w)=0$



An exemple of zkp

zk-SNARK

Succinct non-iterative argument of knowlege

- **Argument of knowledge:** V accepts π => P knows w | Q(x,w)=0;
- **Succint:** π is easy to compute and easy to verify (vital to minimize gas fees);
- **Non-iterative:** a single exchange between P and V is sufficient to transmit π.

Standard usage

zk-SNARKs usage

- Compliance
- Scalability
- Private transaction on public blockchain

Perks for public blockchains

zk-SNARKs perks

- Easy to compute
- Needs only one interaction
- Does not require storage

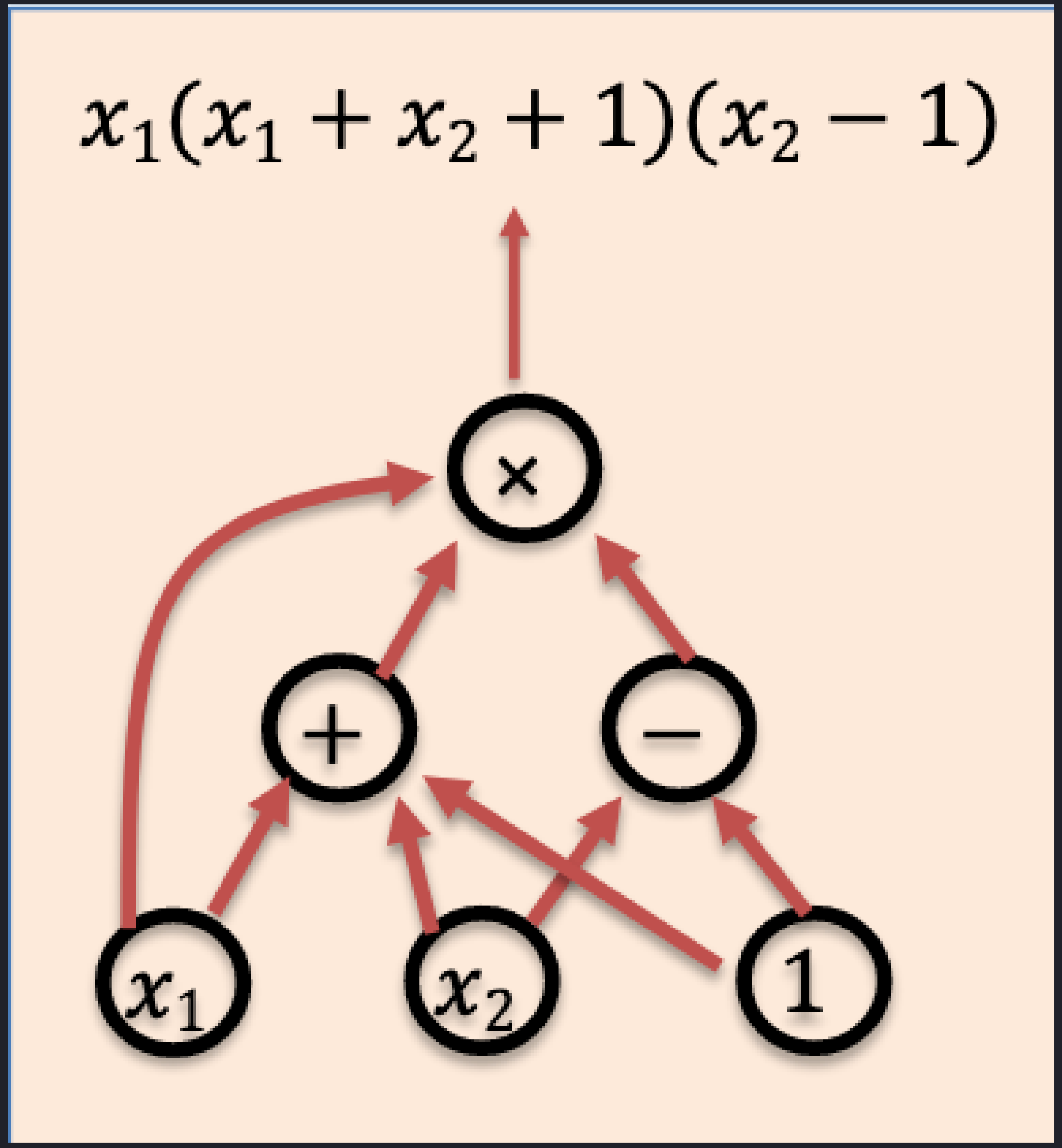
→

- Less gas fees
- Adapted to public blockchain where SC storage is public

Arithmetic circuits

A way to represents P problems

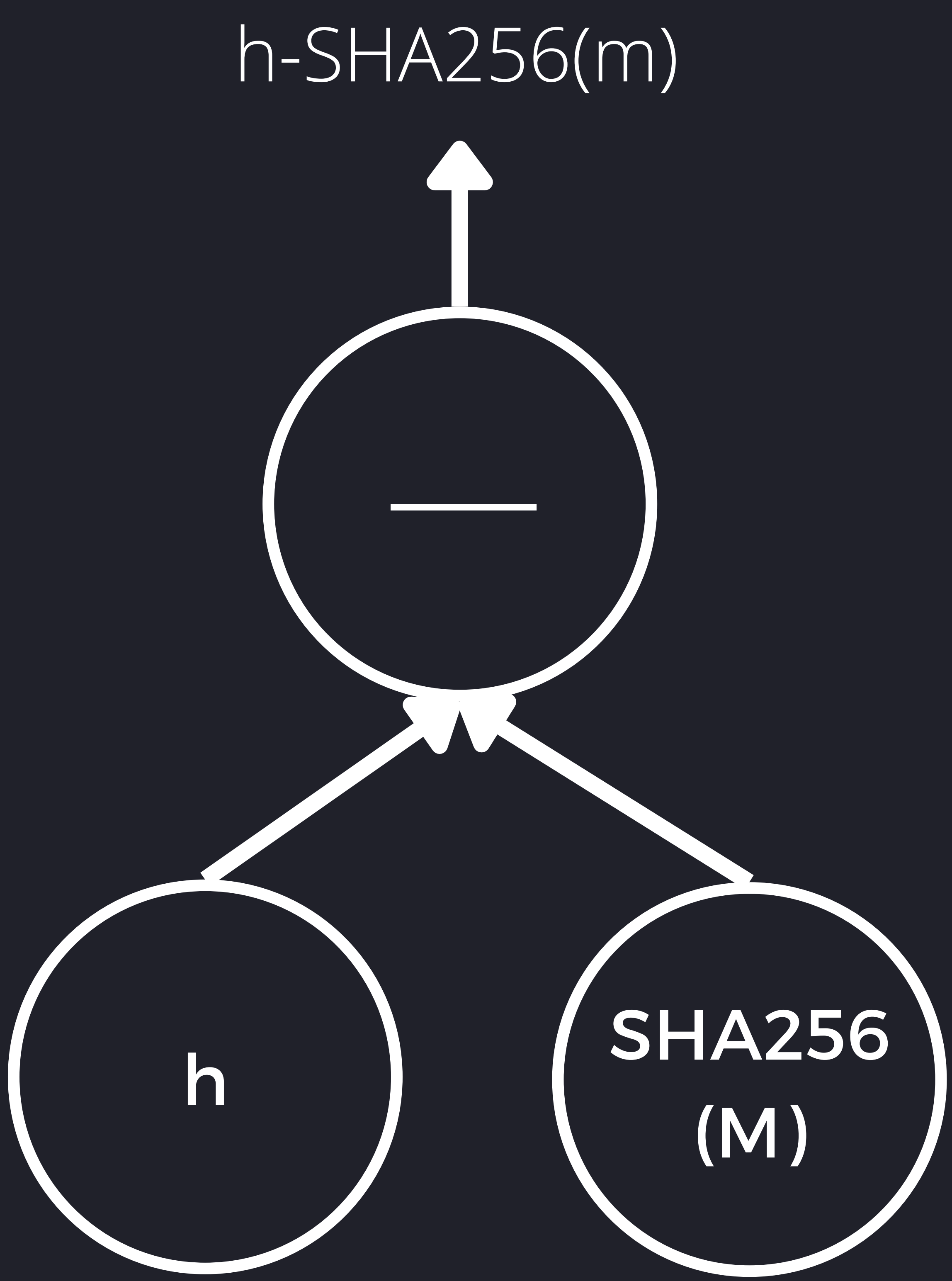
- P problem can be describe as an arithmetic circuits
- Directed acyclic graph
- Node operation : +, -, x
- Inputs : x1, x2, 1
- |C| = number of gates



Example circuits

Testing hash circuts and SHA256 circuts

- Test if SHA256(m)=h,
- C-testinghash(h,m)=(h-SHA256(m))
- |C-SHA256|=20K gates

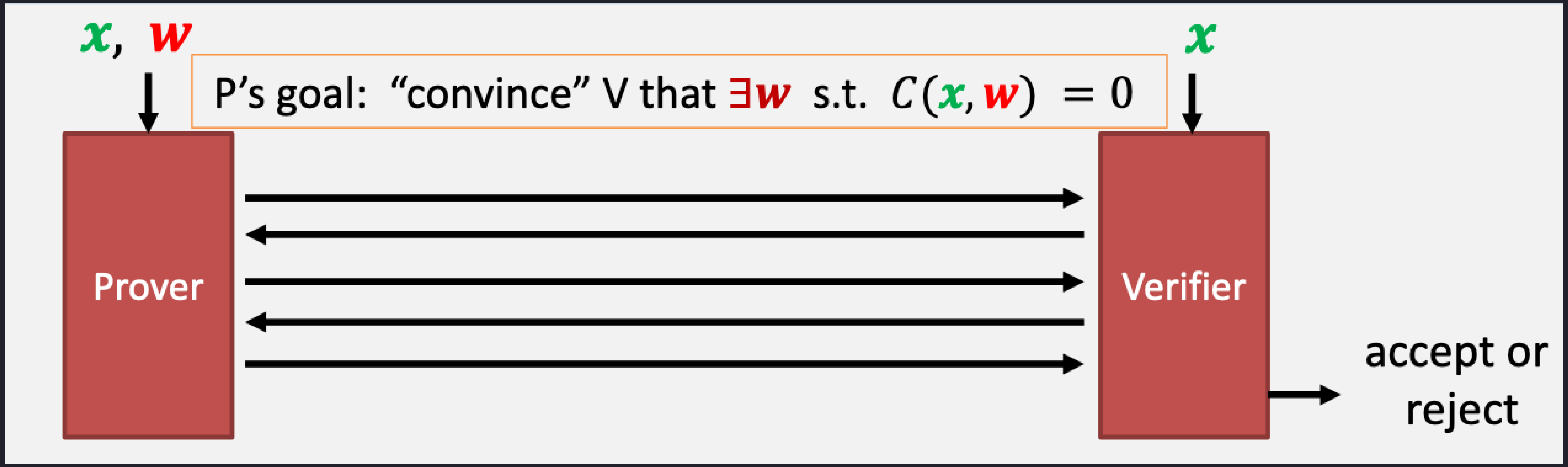


Argument systems

Definition

Reminder :

- *Arithmetic circuit : C(x,w)*
- *x public statement*
- *w secret witness*



Properties of argument system

Properties

Complete :

- $\forall$ x,w, C(x,w)=0 -> P[V(Sv,x, P(Sp,x,w))=accept]=1

Argument of knowledge :

- V accepts π => P knows w (C(x,w)=0)

Zero knowledge (optional):

- (Sv,x,π) reveals nothing about w

Reminder :

- *Arithmetic circuit : C(x,w)*
- *x public statement*
- *w secret witness*

Prepocessing argument systems

Definition

Preprocessing setup :

- public parameters (Sp, Sv)

To prove :

- P(Sp,x,w)-> proof π

To verify

- V(Sv,x,π)

Reminder :

- *Arithmetic circuit : C(x,w)*
- *x public statement*
- *w secret witness*

Succinct property

Properties

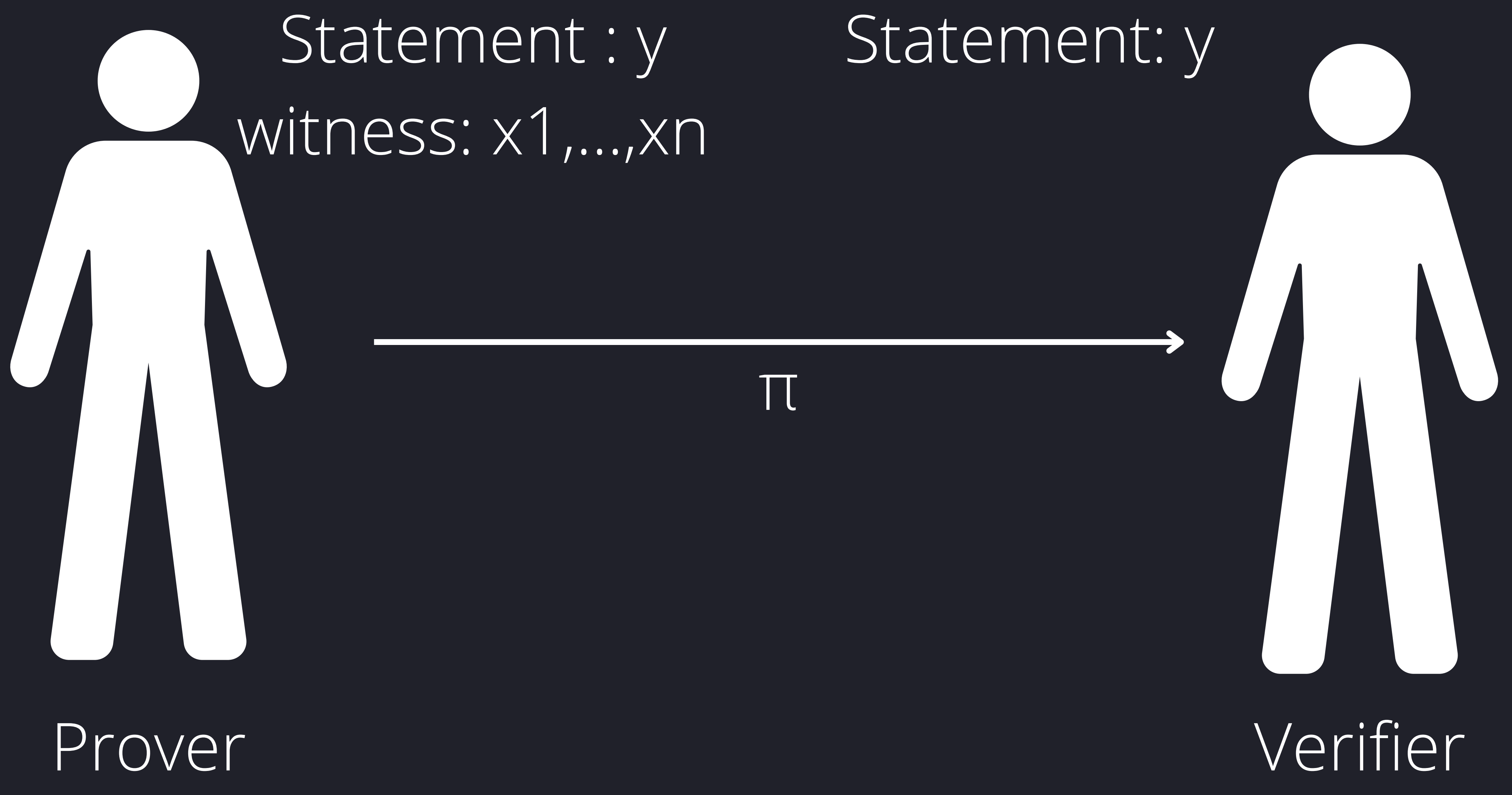
- Short proof π ($|\pi| = O(\log(|C|))$;
- Fast-to-verify proof π (time(V) = $O(\log(|C|))$;;
- The circuit preprocessing aims at providing a "summary" of the circuit so that the verifier does not have to read it all.

Reminder :

- *Arithmetic circuit : C(x,w)*
- *x public statement*
- *w secret witness*

Example

Size(π) and VerifyTime(π) is O(log(n))



Reminder :

- *Arithmetic circuit : C(x,w)*
- *x public statement*
- *w secret witness*

Types of preprocessing

3 models

Trusted setup : S(C,r), r a secret

- if r reveal -> false statement can be proved

Trusted but universal : secret r independent of C

- S(C)=(Sinit,Sindex):
- One time : Sinit(r)-> pk
- Every time : Sindex(pk,C)->(Sp,Sv)

Transparent setup:

- S(C) has no secret

Reminder :

- *Arithmetic circuit : C(x,w)*
- *x public statement*
- *w secret witness*

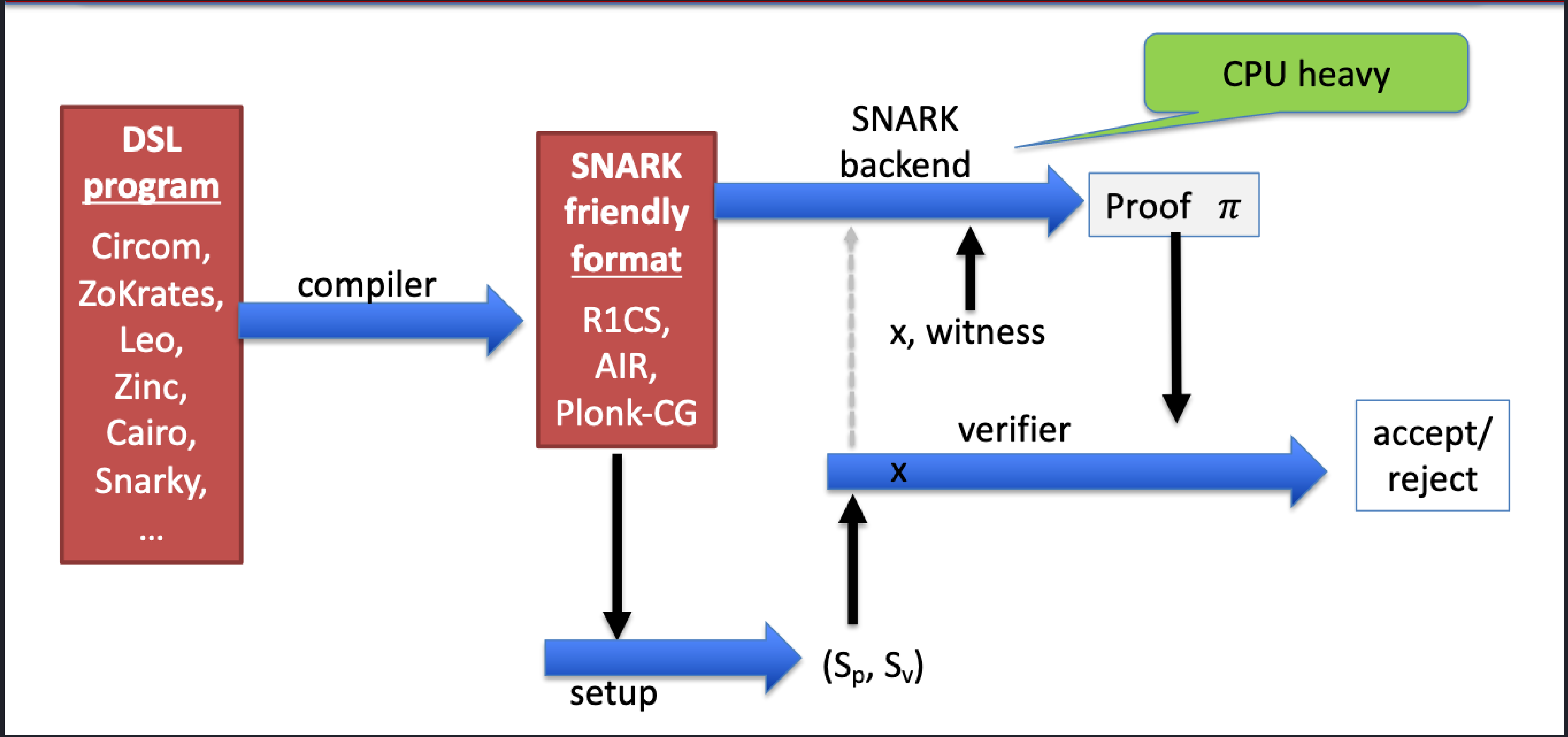
Types of SNARKs

Properties

	Size of proof π	Size of Sp	Verifier time	Trusted setup
Groth16	O(1)	O(|C|)	O(1)	Yes/per circuit
Plonk/Marlin	O(1)	O(|C|)	O(1)	Yes/universal
Bulletproofs	O(log(|C|))	O(1)	O(|C|)	No
STARK	O(log(|C|))	O(1)	O(log(|C|))	No
DARK	O(log(|C|))	O(1)	O(log(|C|))	No

SNARKs software

Architectures



Possible improvements

Fuzzing bots network

- Issue: the low amount of transactions passing through the mixer, which forces users to wait before withdrawing
- Network of off-chain bots
- Each bot sends multiple transactions/day to artificially grow the mixer's traffic
- Liquidity providers give money to the bots to play with
- Liquidity providers are financially incentivized (they earn a fee on each "organic" transaction)

Possible improvements

Simultaneous exit

- Issue : the amounts in the protocols are discrete
- Avoiding discret amounts by discretizing the exit
- To avoid temporal inference, sends batched transactions mixed with other users
- Drawbacks : gas fees