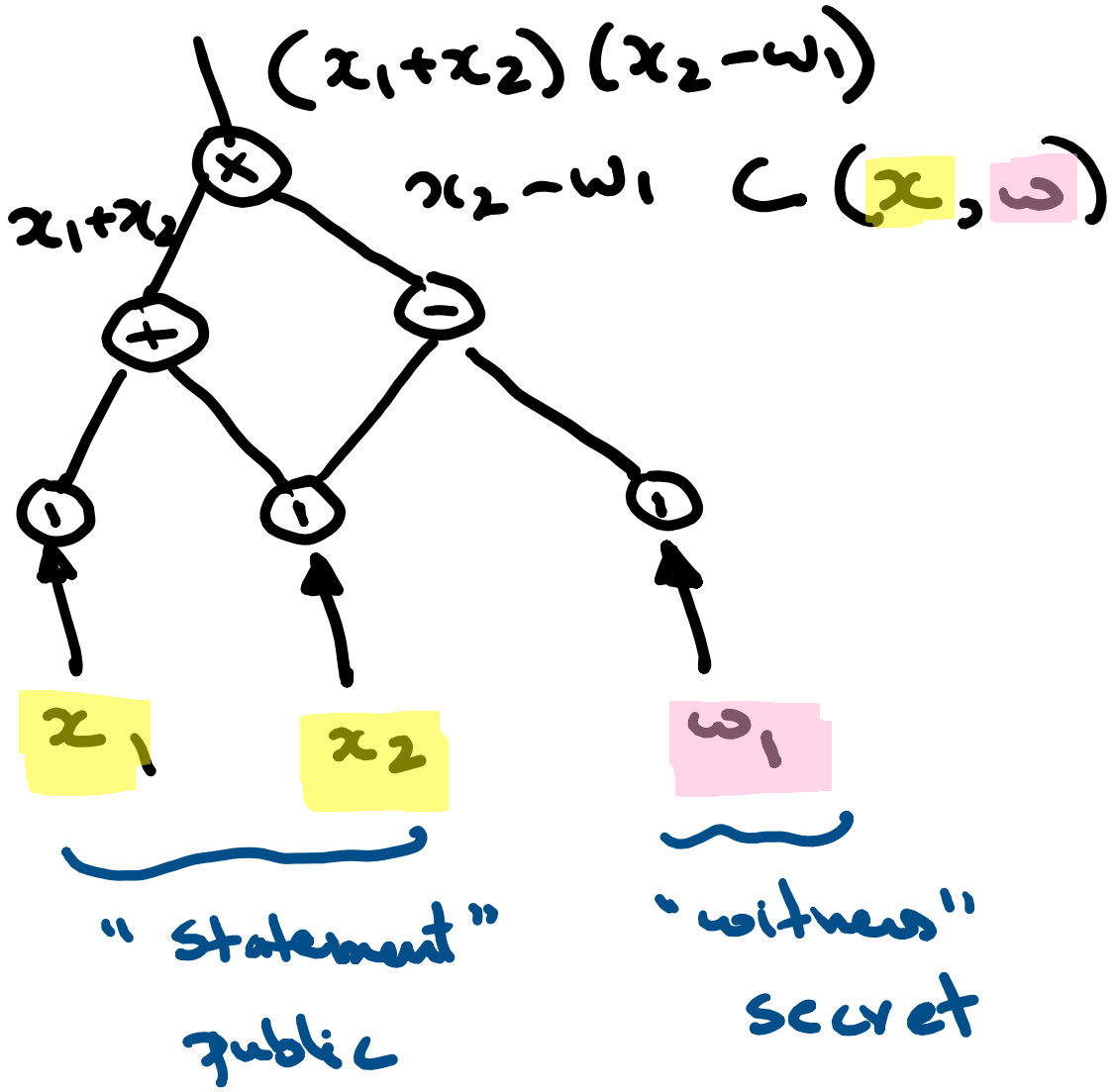
Overview of Modern SNARK Constructions

31-January-2023

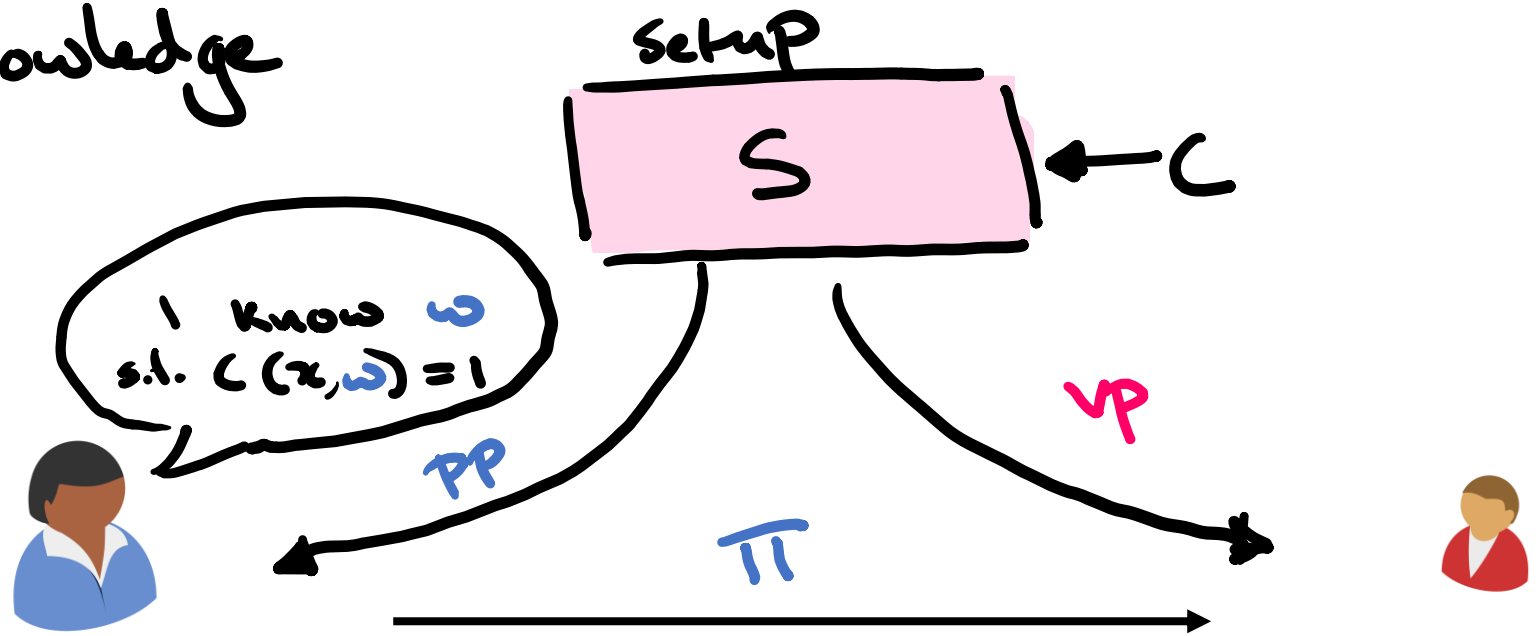
Model of Computation: Arithmetic Circuits

Field $\mathbb{F}$



Succinct Non-interactive Argument of Knowledge

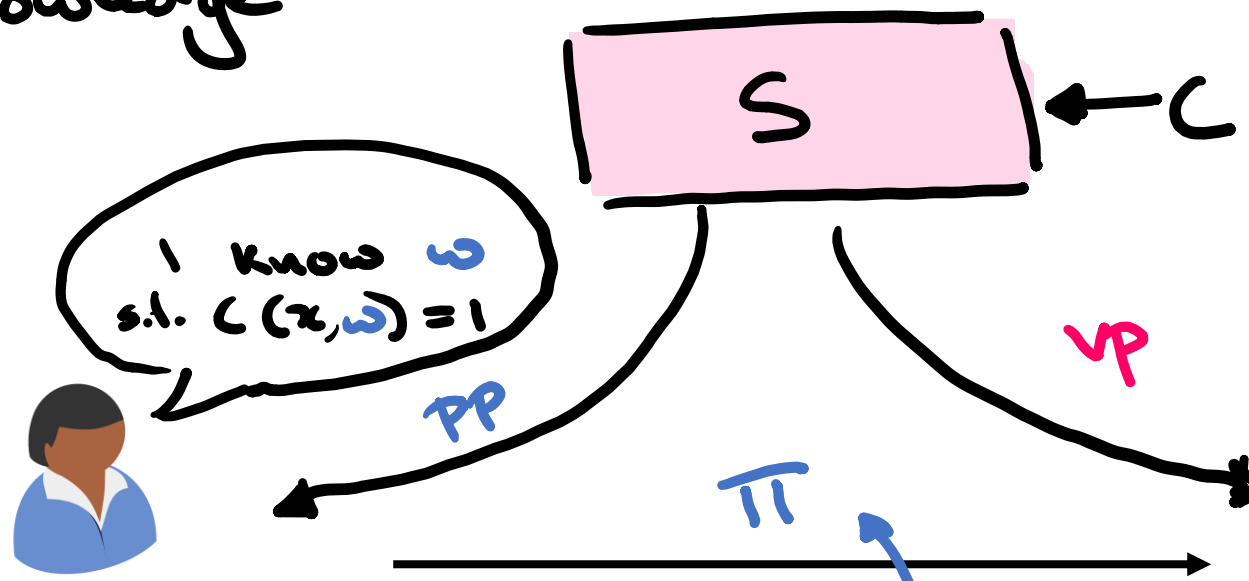
Random Oracle



$\pi \leftarrow P(C, x, \omega, pp)$

$V(C, x, \pi, vp) \rightarrow$ accept / reject

Succinct Non-interactive Argument of Knowledge

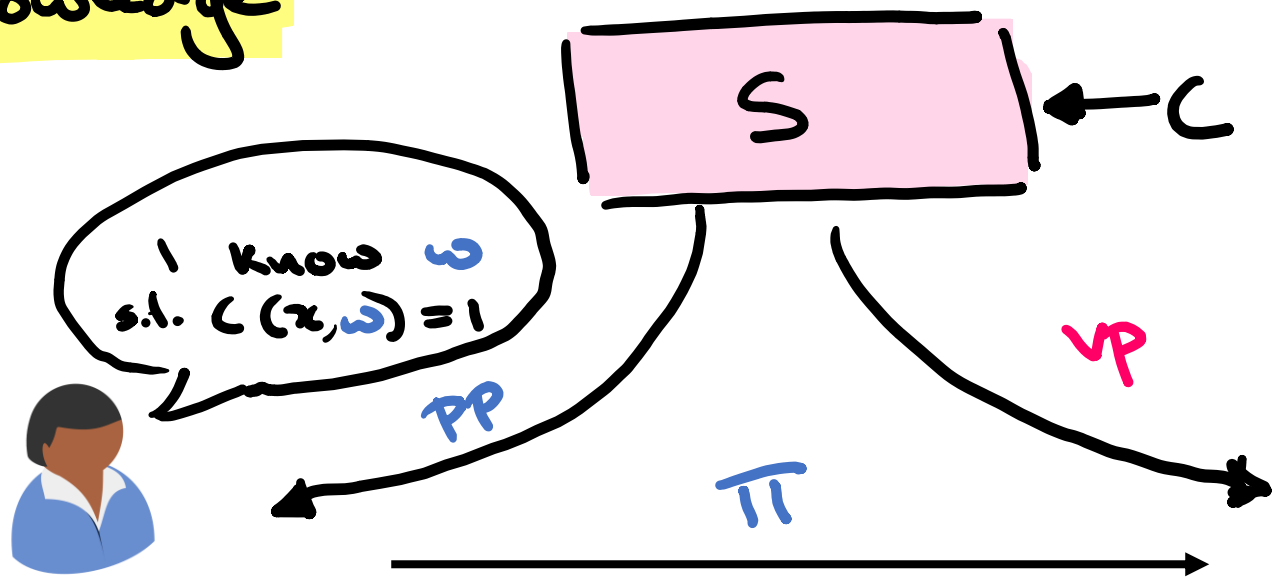


$\pi \leftarrow P(C, x, \omega, pp)$

single message

$V(C, x, \pi, vp) \rightarrow$ accept / reject

Succinct Non-interactive Argument of Knowledge



Soundness

Proof v/s Arguments (Computational Soundness)

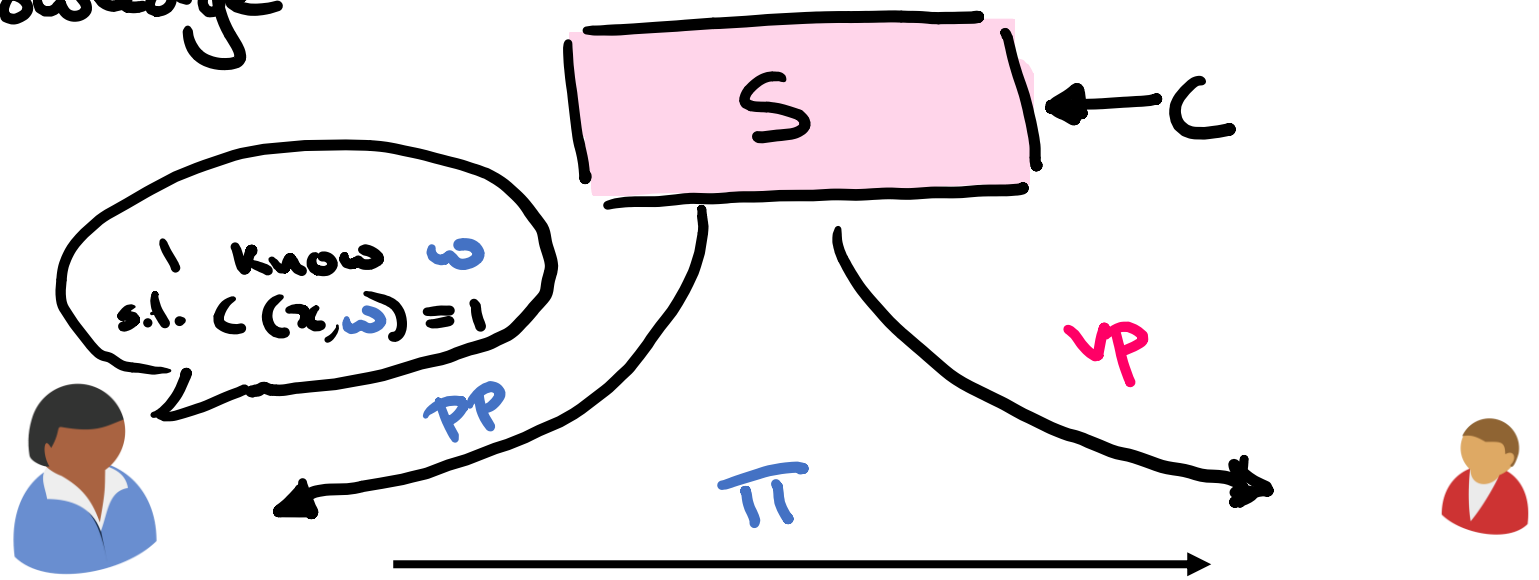
$\pi \leftarrow P(C, x, \omega, PP)$

$V(C, x, \pi, VP) \rightarrow$ accept / reject

$\forall$ PPT $\exists$ extractor E s.t.

if $V(C, x, \tilde{\pi}, VP) \rightarrow$ accept then $E(PP, \tilde{\pi}) \rightarrow \omega$ (witness)

Succinct Non-interactive Argument of Knowledge



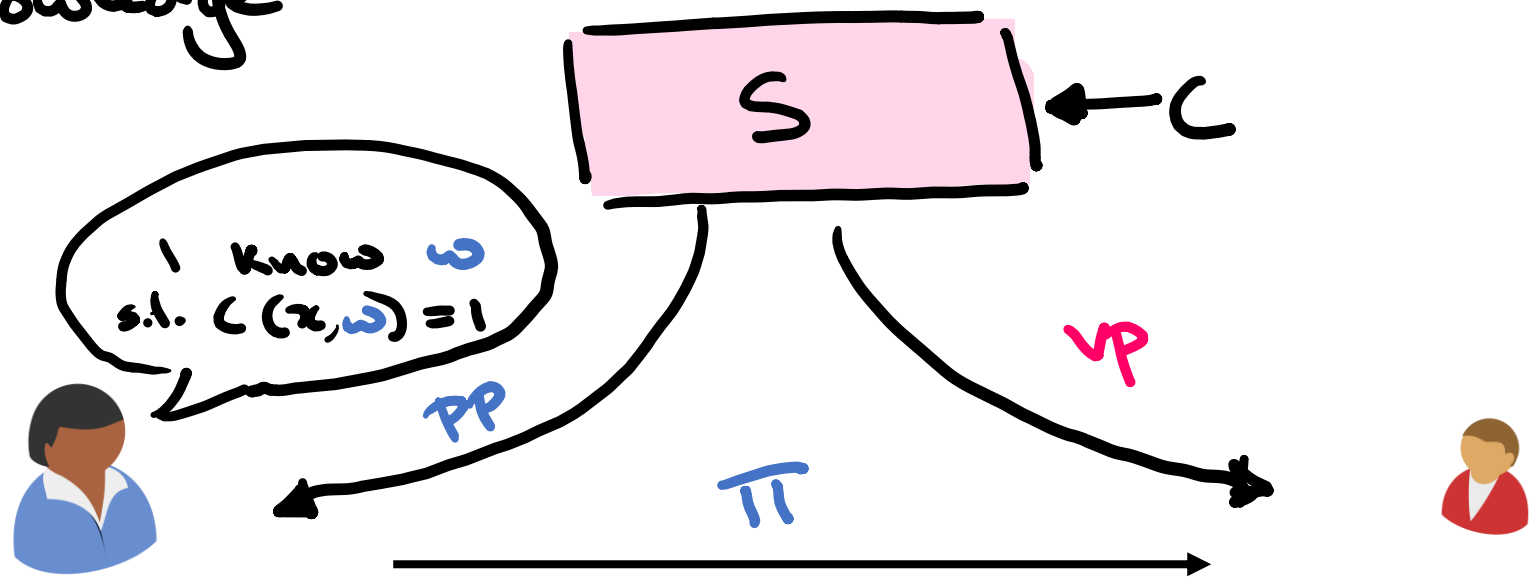
$\pi \leftarrow P(C, x, \omega, pp)$

$V(C, x, \pi, vp) \rightarrow$ accept / reject

$|\pi| = O(\log|C|)$
Proof size

Time = $O(|x|, \log|C|)$
verification time

Succinct Non-interactive Argument of Knowledge



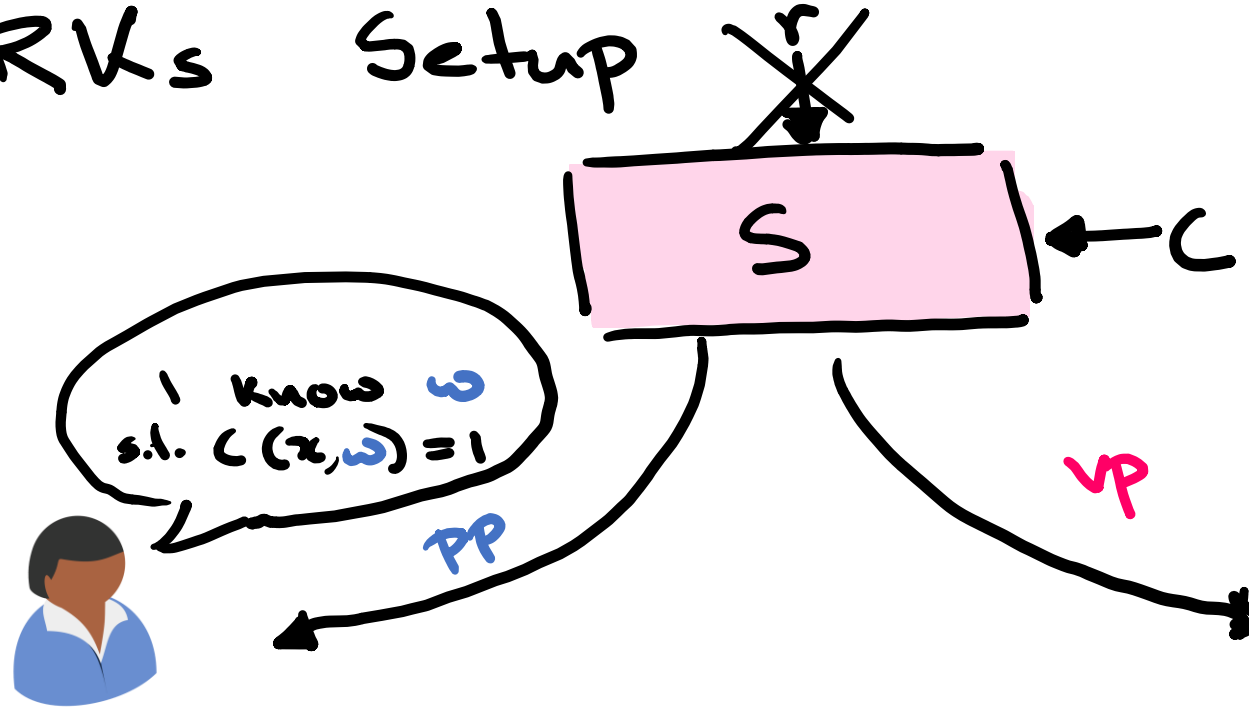
$\pi \leftarrow P(C, x, \omega, pp)$

$V(C, x, \pi, vp) \rightarrow$ accept / reject

Can be made Zero-Knowledge!

ZK-SNARKs

SNARKs Setup



requires secret r

① Trusted per circuit

② Trusted universal

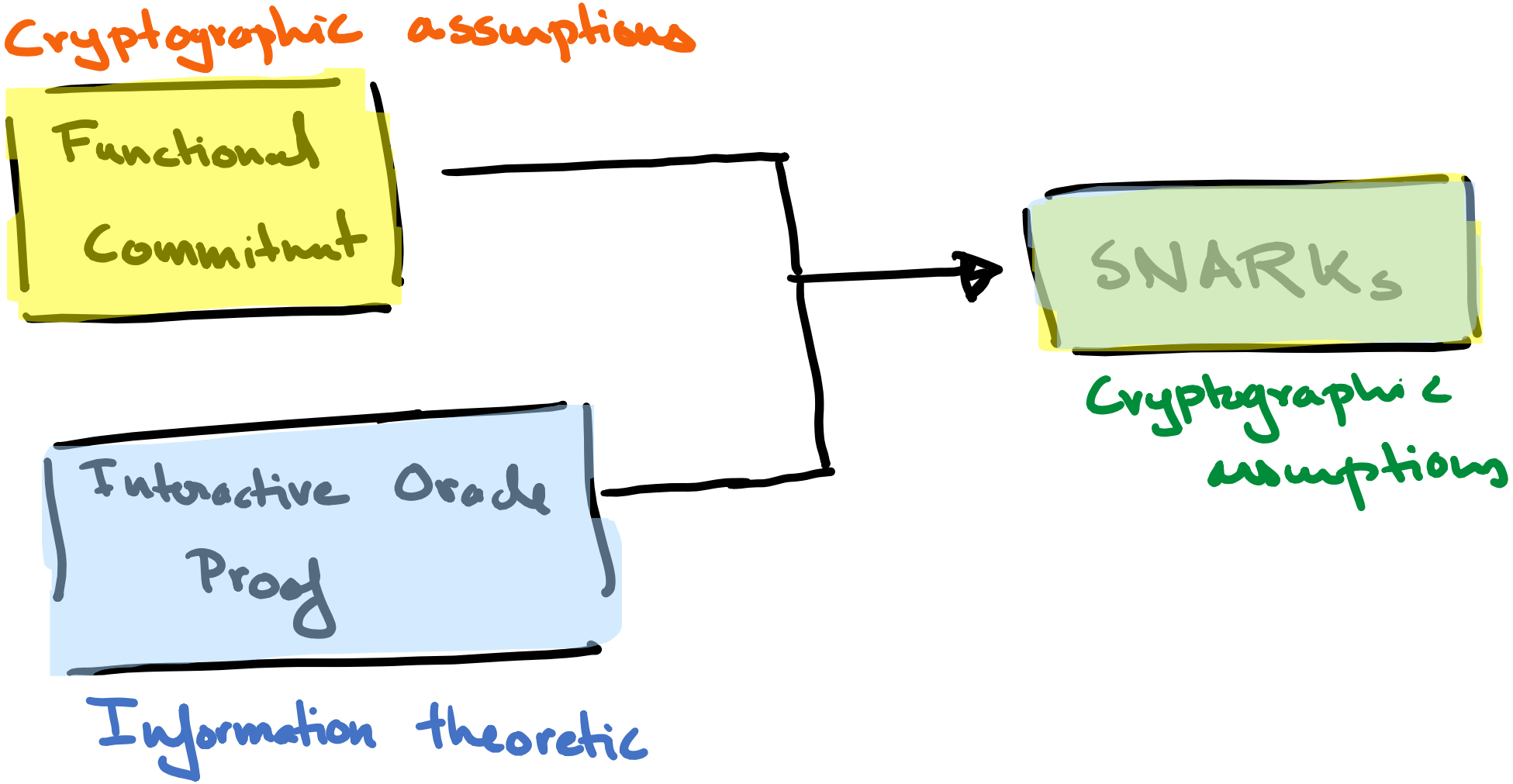
③ Transparent

$P \leftarrow S(r)$

$VP \leftarrow T(P, C)$

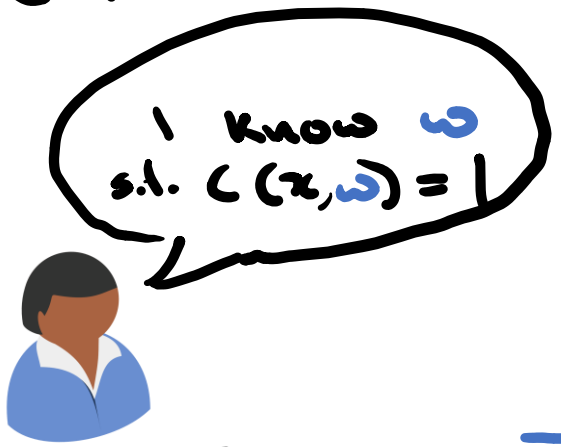
No secret

SNARK construction paradigm



Interactive Oracle Proofs (IOP)

Prover messages are oracles.



f_1

(C, x, ω, pp)

(C, x, π, vp)

r_1

$f_2(x) = e^{x+r}$

f_2

$V^{f_1, f_2}(r_1) \rightarrow$ accept / reject

$f_1(x) + f_2(y) \stackrel{?}{=} \alpha$

Functional Commitment for $\mathcal{F}$

Sender (f)

Commit phase

$\text{com}(f; r)$

x

Receiver

1. binding
2. hiding
3. succinct

π proves

1. $y = f(x)$
2. $f \in \mathcal{F}$

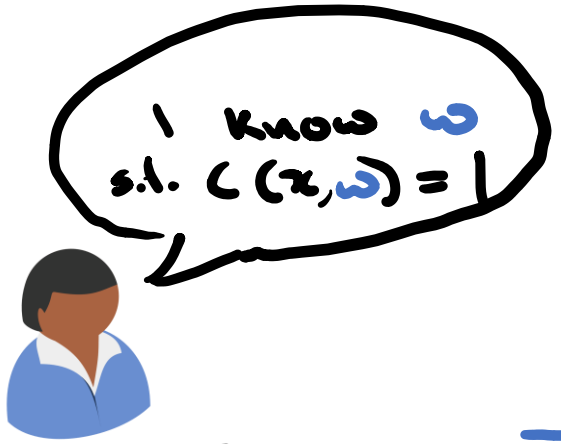
y, π

accept / Reject

Open / Eval phase

IOP + Functional Commitments

Prover messages are oracles.



(C, x, ω, pp)

d_1

r_1

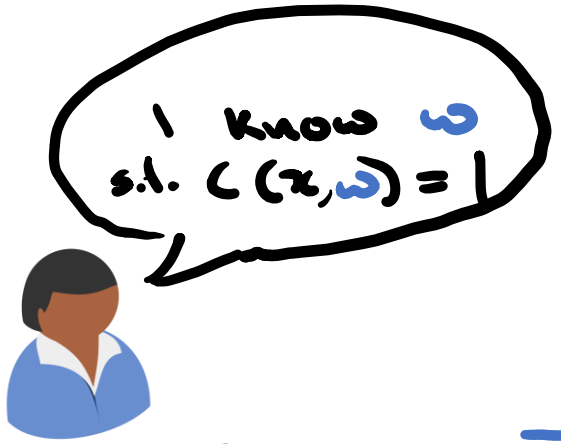
d_2

(C, x, π, vp)

$V^{d_1, d_2}(r_1) \rightarrow$ accept / reject

IOP + Functional Commitments

Prover messages are oracles.



(C, x, ω, pp)

$\text{Com}(f_1; \rho_1)$

r_1

$\text{Com}(f_2; \rho_2)$

x, y

a, π_1

b, π_2

(C, x, π, vp)

$a = f_1(x), \pi_1$

$b = f_2(y), \pi_2$

$V^{f_1, f_2}(r_1) \rightarrow$ accept / reject

$f_1(x) + f_2(y) = \alpha$

$a + b \stackrel{?}{=} \alpha$

Vector Commitment (Merkle Trees)

Commit to vector $x_1, \ldots, x_{2^k} \in \{0,1\}^n$

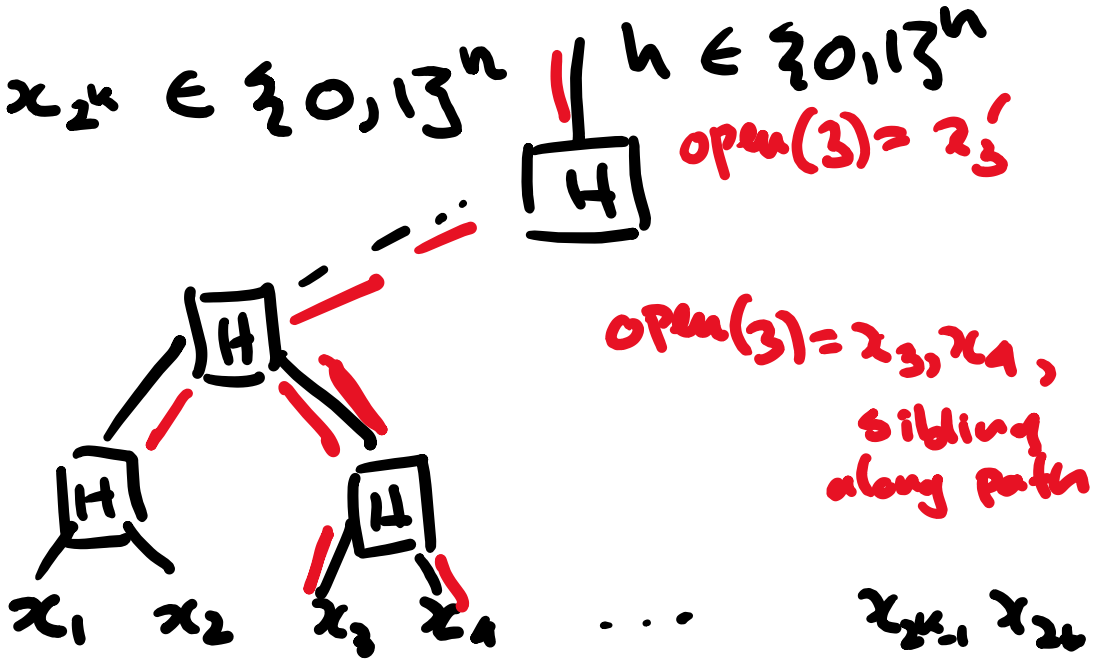
Size of opening (single element) $\approx O(k)$

Time to verify $\approx O(k)$

$H : \{0,1\}^{2n} \longrightarrow \{0,1\}^n$

"Hard to find" $y_1 \neq y_2$ s.t. $H(y_1) = H(y_2)$

Binding: Hard to open at index i to $x_i \neq x_i'$



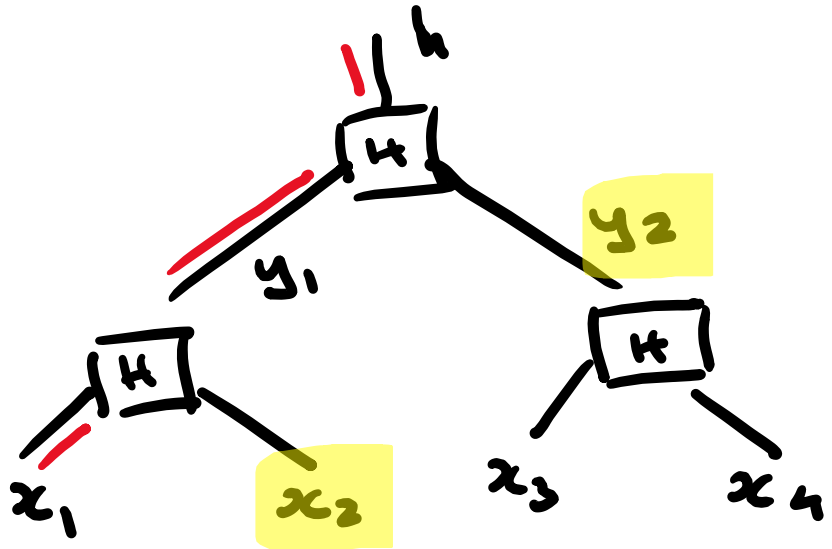
$Com(x_1, \ldots x_{2^k}) \rightarrow h$

$n \cdot 2^k$ bits → n bits

(optional) k-elements + k siblings } Merkle path

Vector Commitment (Merkle Trees)

Example



$Com(x_1, x_2, x_3, x_4) = h$

$open(1) = x_1, x_2, y_2$

$Verify(h, open(1)) \rightarrow h \stackrel{?}{=} H(H(x_1 \| x_2) \| y_2)$