#1. Why ZKP? zero-knowledge proof

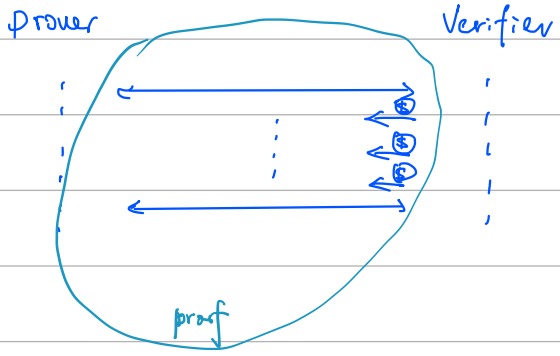
(π) proof ~> Theorem / Statement / Proposition

Two-Party Interaction

[Prover —π→ Verifier]

① Completeness (honest prover)
② Soundness (malicious prover)
③ Zero-knowledge (malicious verifier)

} Meta Theorems



Blockchain: Non-interactive zkP.

Prover —π→ Verifier (on-chain)

Fiat-Shamir Transformation (public-coin protocol ~> non-interactive protocol)

Random-Oracle

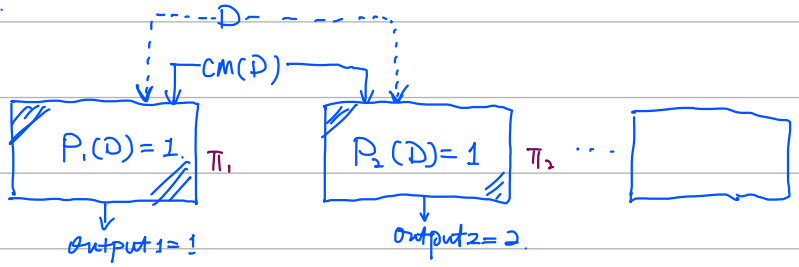
RO (Cryptographic hash function)

prover —π→ Verifier (on-chain)

Commitment: { O(1) size, Binding, Hiding
e.g. 10GB file
$cm(D_1) \neq cm(D_2)$ if $D_1 \neq D_2$
} hash, merkle-tree, pedersen-commitment, KZG10, ⋮

Blockchain: put commitments on-chain (EIP-4844, Rollup-Blob)

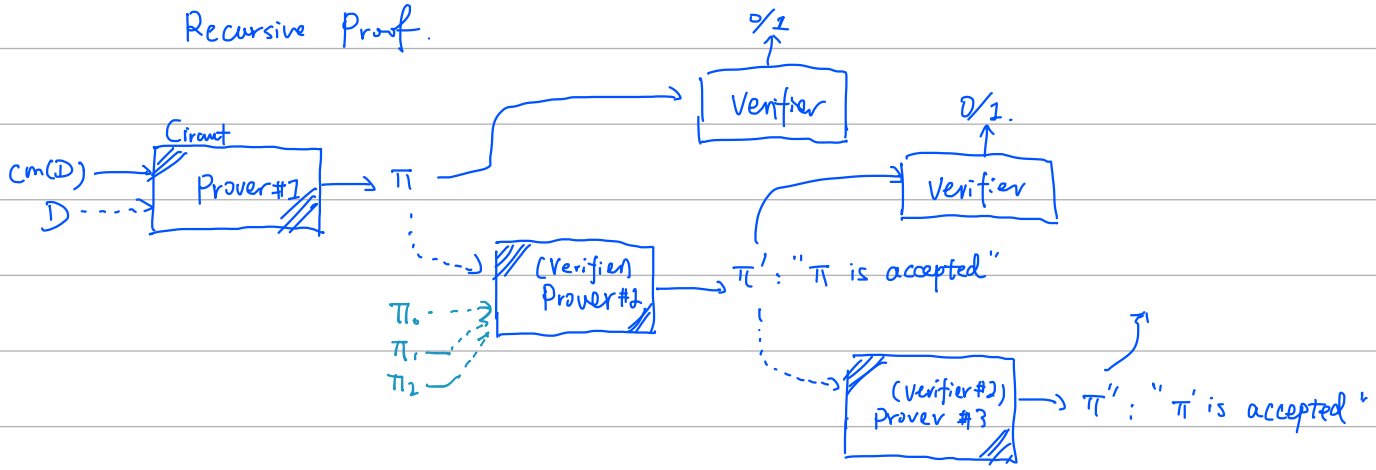
Commit-and-prove.
(plonk)



#2. Why zkSNARK? → Proof of knowledge → Succinctness
- proof size — sublinear / logarithmic
- verifier cost — sublinear / poly-logarithmic

Blockchain: implement verifier on-chain (computation compression)

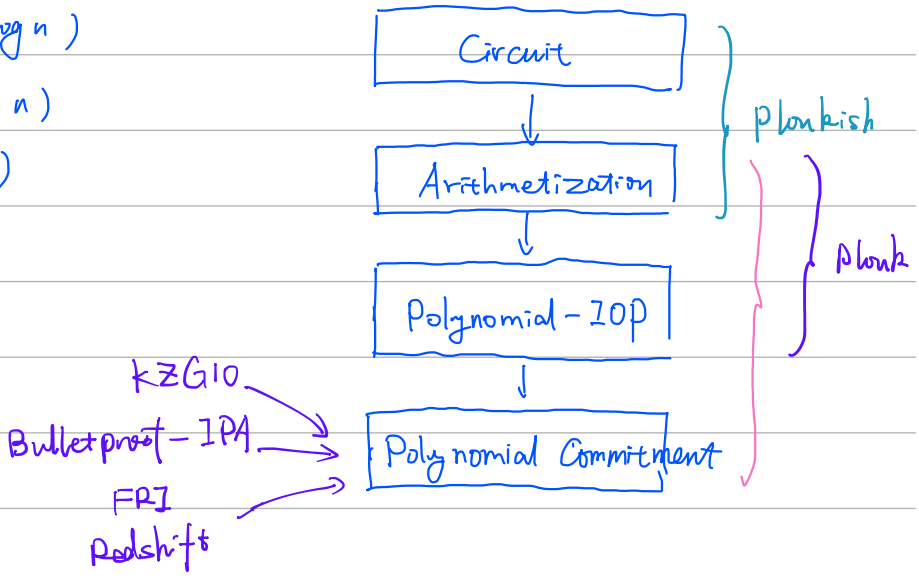
Recursive Proof.



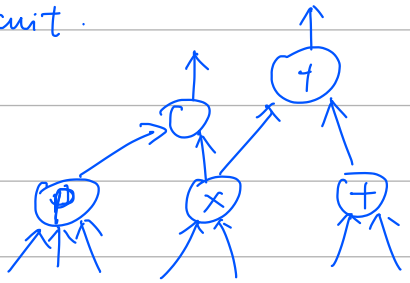
#3. Why plonk? (2019) (Groth16)

① Universal Trusted Setup. (KZG10)
(Transparent Setup)

② Proving $O(n \cdot \log n)$
Verifier $O(\log n)$
Proof-size $O(1)$

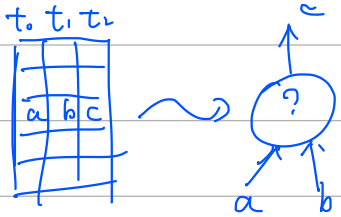


3.1. Circuit.



Arithmetic Gates.

Polynomial Gates
(high-degree gates)
(custom gates)

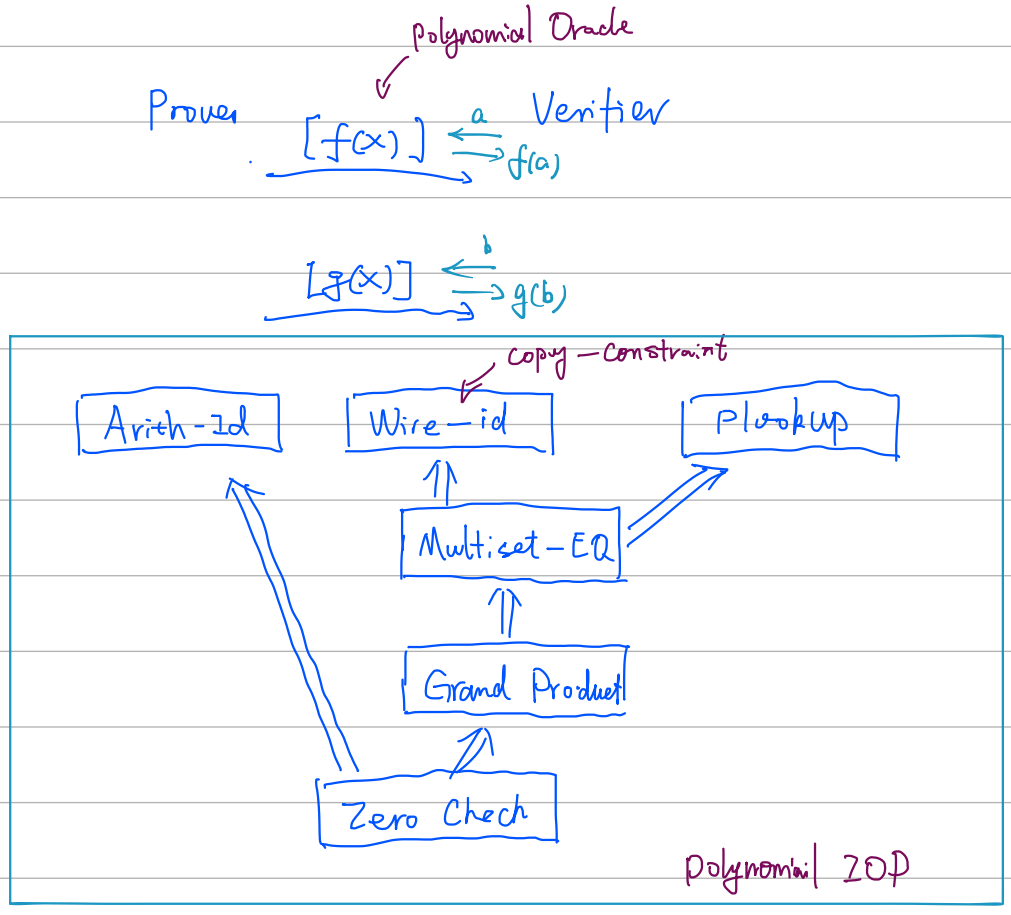


Lookup Gates

⇓

[polynomials] [polynomials] --- [----]

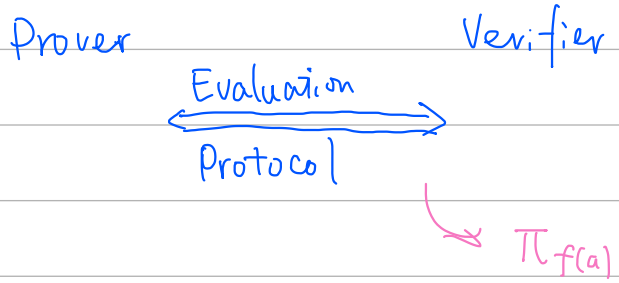
3.2 Polynomial IOP.



#3.3. polynomial commitment

(KZG10, IPA, FRI ···)

$f(x) \rightsquigarrow [f(x)]$



#3.4. Hyper-Plonk ···· (multi-variate polynomial)