Zero Knowledge Proofs (ZKP)

- Interactive Proofs
- Zero Knowledge & Simula^n
- Interactive Digital Signatures
- Non-interactive ZKP
- Succintness etc. [ZKSNARKS]

Definition of a Proof

Proof ≡ String c

Class P

∃ poly-time TM M s.t ∀ w∈L ⇒ M(w) = accept
∀ w∉L ⇒ M(w) = reject
} poly time deciders

then L ∈ P

Class NP

∃ poly-time TM M s.t
∀ w∈L ∃c s.t M(w,c) = accept
∀ w∉L ∄c s.t M(w,c) = accept
} poly time verifiers (c is called certificate)

we now allow margin of error
this gives us BPP

Class BPP

∃ PPTM M s.t
∀ w∈L P[M(w) = accept] ≥ 2/3
∀ w∉L P[M(w) = reject] ≥ 2/3

Alternate def of NP

prover — c — verifier
P — V
∀ P

∀ w∈L ∃P s.t (P ⟶ V) accepts
∀ w∉L ∀P* P* ⟶ V reject

Class IP/NP

∀ w∈L ∃P P[(P ⟶ V) = accept] ≥ k
∀ w∉L ∀P* P[(P* ⟶ V) = accept] ≤ k' (k < k')

for non Prover-Verifier communication. is no-way if verifier can communicate with Prover ⟹ we get IP

IP = PSPACE ⊇ NPSPACE
↳ PSPACE should be ⊃ NP (is not yet proven)
more powerful than NP (∵ IP ⊇ co-NP)

Intuition

Consider a Game

winning strategy

∃m₁ ∀m₂ ∃m₃ ∀m₄ - - - f(m⃗) = 1
(∃ moves, ∀ moves)

TQBF Problem

hard to get a certificate (need to give entire decision)
∉ NP (not proven yet)
easy to solve using interaction (just play a game with V)
(needs more strategy - Arithmetization [Convert Boolean formula to arithmetic formula])

ZKP

completeness : ∀ w∈L ∃P P((P ⟶ V) = accept) ≥ k
Soundness : ∀ w∉L ∀P* P((P* ⟶ V) = accept) ≤ k' (k > k')
Zero Knowledge : ∃ PPTM S that 'simulates' interaction with P
or
V does not get any certificate

verifier interacting with P
≡
verifier interacting with Sim S

NP ⊆ ZKP

Proof

Consider an NP Problem

say, Graph 3-coloring (NP-complete)
Input : Graph G = (V, E)
Output : Yes if G is 3-colorable (tripartite)

① Prover permutes the 3 colors (R, G, B) and generates another valid 3-coloring of G

② ∀ nodes, send a locked box containing the color values

P — [c₁] [c₂] [c₃] [c₄] - - - [cₙ] — V

③ Verifier asks for keys of the boxes of endpts of any random edge in G
(i, j) ∈ G ∀ cᵢ & cⱼ requested

④ if cᵢ = cⱼ ⟶ accept
cᵢ ≠ cⱼ ⟶ reject

Completeness
→ if coloring is valid ; color of endpts will always be different
∴ Pr[(P ⟷ V) = accepts] = 1

Soundness
Pr[(P* ⟷ V) = accepts] = 1 − 1/|E| < 1
P* can cheat by adding another color ; but will be caught with a non-zero prob.

Zero-Knowledge
for (i, j) ∈ G if valid 3-coloring exists
(cᵢ, cⱼ) will be assigned
can be (R, G), (R, B), (G, R), (B, G), (G, B), (B, R)
with equal probs.
simulator S creates (R, G), (R, B), (G, R), (B, G), (G, B), (B, R)
with equal probs.
∴ S & P are indistinguishable to V

How to make boxes? Use Bit commitment
(so that there is guarantee that color inside box won't be changed when giving keys)
Use commit phase & reveal phase

Digital Signatures are non-interactive + Zero Knowledge Proof
- non-interactive: Prover sends signature to verifier
- Zero Knowledge Proof: Verifier can verify Prover is the author of the message without knowing the message

Interactive Digital Signatures ⟶ not Impactful

SNARKS problem

ZKP for DLP

y ≡ g^x mod p ⟶ Prover says he knows (x)

b = rand()
P — V
g^b mod p ①
r ②
g^x ③
A = rx + b
Verifier verifies g^A = (g^x)^r · g^b
g^A
g^x

Here system is Interactive
(V is sending back r)

Use Fiat-Shamir Heuristic
H(y, g, r, p) = r
(Hashes are equivalent to random oracle)
∴ r = random

random values can be sent either way
∴ non-interactive mode
P — V
g^b
r = H(y, g, r, p)
g^A

non-interactive (only Prover sending) ≡ Digital signature

Succintness

since hash-and-sign paradigm allows large messages to have small digital signature. We define succintness as the amount of compression of a proof (non-interactive)