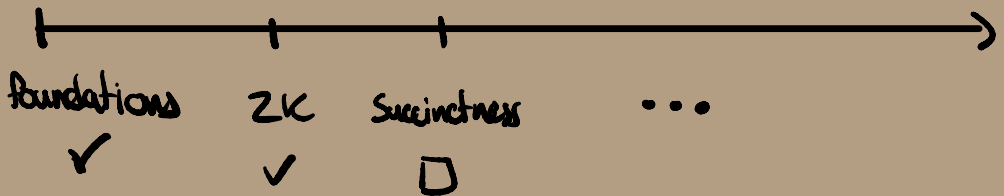


lec10: Succinctness

Course



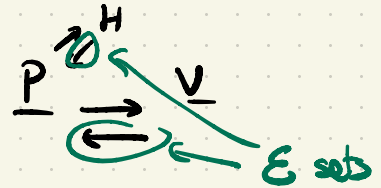
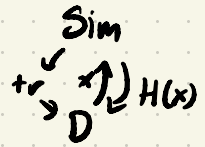
Today:

- ① Succinctness
- ② Probabilistically Checkable Proofs (PCPs)
- ③ Merkle Trees
- ④ Kilian-Micali Proofs

10 Last Time:

• ROM: Random Oracle Model

- $H(m, r)$ is a commitment
- NIZKs are possible
 - Fiat-Shamir: replace V with H .
 - ZK: Sim implements H
 - PoK: $\mathcal{E}$ programs H
- controversial, but successful



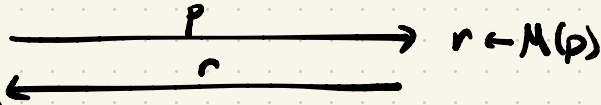
1 Motivation: verifiable outsourcing

- Let M be a BIG open-source model,

Bob (small phone)

Amazon (AWS)

- prompt p



Is $r \stackrel{?}{=} M(p)$?

- What if Amazon could prove this to Bob?

- Amazon: Prover P , Bob: Verifier V
- Relation (attempt): $\phi(x=(p, M, r), w=\phi) \stackrel{\Delta}{=} r=M(p)$
 - Problem: M is too BIG for Bob's phone - use a hash!
- Relation: $\phi(x=(p, h, r), w=M) \stackrel{\Delta}{=} h=H(M) \wedge r=M(p)$
 - Now, x is small!
 - Bob can get h from Hugging Face

- Proof requirements (informal)

- Interactive is ok, non-interactive is better
- Communication/proof size should be small
- V should be fast
- privacy is not necessary

- Definition: Succinctness for an IP $\Pi = (P, V)$

- Let $\phi(x, w)$ be an NP relation (e.g. circuit)
- Let $|\phi|$ be the time to check ϕ

- Π is succinct if:

- communication is $O(|x|) + o(|\phi| + |w|)$
- V -time is $O(|x|) + o(|\phi| + |w|)$

Warning: the details here are subtle and vary from source to source.

- f is $o(g)$ if $\lim_{\lambda \rightarrow \infty} \frac{f(\lambda)}{g(\lambda)} = 0$

- $\log n, \sqrt{n}, n^{0.99}$ are all $o(n)$

- typically succinct proofs seek $\log |\phi|$ or $\sqrt{|\phi|}$

- in practice, constants matter

- since these are $O(|x|)$, the defⁿ is most useful when $|x| = o(|\phi|)$

- hence, representing M with $h = H(M)$

- This seems impossible!

- It is possible! But, with weaker soundness (in theory)

- Defⁿ: Computational Soundness:

- $\forall x \notin \mathcal{L}_\phi, \forall \text{PPT } P^*, \Pr[(P^*, V)(x) = 1] \leq \text{negl.}$

- Computational knowledge soundness also quantifies over PPT P^*

- If Π is computationally (knowledge) sound, it is an "argument" not a "proof".

Terminology:

- technically Π is a "proof" if it is (knowledge) sound
- an "argument" — computationally (knowledge) sound
- SNARG - Succinct, Noninteractive ARGument
- SNARK - " ARGument of Knowledge

Today, SNARGs via crazy theory

[2] Probabilistically Checkable Proofs (PCPs)

- PCPs are related to succinctness/IPs but a little different
 - randomized, assumption-free, non-interactive, specific model
- Let L_ϕ be an NP language with $\phi(x, \omega)$
- How to prove $x \in L_\phi$?
 - Option: send ω , V checks $\phi(x, \omega) = 1$
 - V reads all of ω
 - Does V really need to do this?
- Analogy: grading homework
 - Student submits ω
 - Does the TA need to read all of ω ?
 - No, instead, check some random parts of ω
 - $\sim$ local consistency testing $\rightarrow$ imperfect, but pretty good
 - relatedly: Gradescope has regrade requests
 - use interaction to help us get you the right grade. $\ddot{\smile}$
- To make this work, we'll change the proof format
 - $\omega \rightarrow \Pi$

- PCP definition:
 - Prove $(x, w) \rightarrow \pi$
 - Verify $\pi(x) \rightarrow \{0, 1\}$: V only queries π , bit-by-bit (poly) (poly)
 - soundness is unconditional, approximate ($< 1/3$ error)
- PCP theorem:
 - Let $L_p \in NP$. Then L has a PCP with 3 queries. !?!
 - This should shock you
 - it shocked me and got me into cryptography
 - (story time)

- The PCP theorem is the most lauded theory result of the last 4 decades.
 - 2001: Arora, Feige, Goldwasser, Lund, Lovász, Motwani, Safra, Sudan, Szegedy win Gödel prize for it with $\log(n)$ bits.
 - Gödel prize: CS prize for best theory result
 - Proof is 3 papers, 100s of pages
 - 2011: Håstad wins it again for 3-bits
 - 2019: Dinur wins it again for a 'simple' proof
 - based on gap amplification
 - "only" 30 pages
- Proof is hard - we won't do it.
- We'll just use PCPs ☺
- Our task: PCP $\rightarrow$ SNARG
 - SNARG: P sends V bits...
 - PCP: P sends V an oracle

$$\underline{P(x, w)} \longrightarrow \boxed{\pi} \longleftrightarrow \underline{V(x)}$$

$\downarrow$
 $\{0, 1\}$

- idea: P sends $O(1)$ -sized commitment to π
 - problem P can never open all of π - too big.
 - solution: open single bits!
 - need a vector commitment

3 Merkle Trees

- A MT is a vector commitment
- A special $\text{Commit}(m \in \{0, 1\}^n) \rightarrow c$ that can be opened to any m :
 - it has a way to prove

$$\phi(c, i, v, m) \stackrel{\Delta}{=} \text{Commit}(m) = c \wedge m_i = v$$

$\uparrow \quad \uparrow$
index value

Syntax:

- $\text{Commit}(m \in \{0, 1\}^n) \rightarrow c$
- $\text{IdxOpen}(m, i) \rightarrow \pi$
- $\text{IdxVerify}(c, i, m_i, \pi) \rightarrow \{0, 1\}$
- Efficiency: $|\pi|, \text{IdxVerify}$ are $O(\log |\pi|)$
- Properties: correctness: $\forall m, v_i, \text{IdxVerify}(\text{Commit}(m), i, m_i, \text{IdxOpen}(m, i)) = 1$

Index Binding $\forall$ PPT A ,

$$P_r \left[\begin{array}{l} c, i, v, v', \pi, \pi' \leftarrow A() \\ \wedge v \neq v' \\ \wedge \text{IdxVerify}(c, i, v, \pi) = 1 \\ \wedge \text{IdxVerify}(c, i, v', \pi') = 1 \end{array} \right] \leq \text{negl}.$$

- Index Binding + correctness $\Rightarrow$ Binding (for Commit)

- extensions:

- you can add r and define hiding for Commit too
- you can define hiding for IdxOpen

- Merkle Trees are a vector commitment.

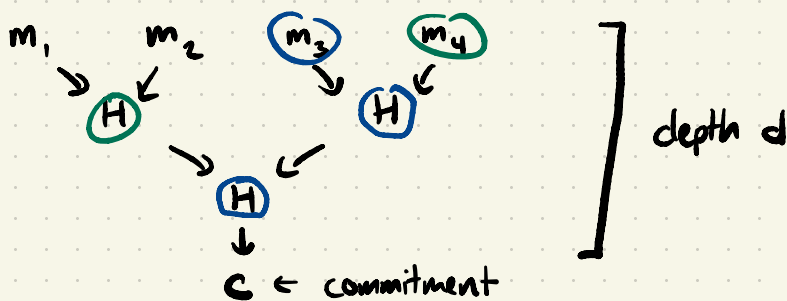
- They use a $2 \rightarrow 1$ hash function $H(\{0,1\}^A, \{0,1\}^A) \rightarrow \{0,1\}^A$

- Assume collision resistance (on H).

- hard to find $(x,y) \neq (x',y')$ s.t. $H(x,y) = H(x',y')$

- Idea: binary trees!

- Define $\text{Commit}(m \in \{0,1\}^{2^d}) \rightarrow \{0,1\}^A$



- How to open m_i :

- Walk path $c \rightarrow m_i$ (example: $i=3$)

- add siblings to π (example: $\pi = (H(m_1, m_2), m_4)$)

- Idx Verify checks d H 's along path

- example: $\text{IdxVerify}(c, 3, m_3, \pi)$:

$$(h, m_4) \leftarrow \pi$$

$$l_1 \leftarrow H(m_3, m_4)$$

$$l_2 \leftarrow H(h, l_1)$$

$$l_2 \stackrel{?}{=} c$$

- Security intuition: index binding break: some collision on the path (HW).

- Very nice idea, widely used in practice to authenticate random-access data

- Examples: Certificate Transparency (TLS)

- Git

- IPFS (peer-to-peer file system)

- Ethereum

- E2E Encrypted Messaging Key Transparency

- Signal, Meta, Apple, ...

- Ralph Merkle - the inventor - is quite an inventive guy.

- In 1974, as a under grad, he takes a sec. class.

- proposes public-key exchange as a course project

- clear problem statement

- good vision of its significance

- and a solution: "Merkle Puzzles" - impractical but correct

- Bad luck! Merkle is at Berkeley, and the Prof. hates the project idea.

- Merkle drops the class.

- 1975: (independently) Diffie + Hellman @ Stanford write "New Directions in Cryptography"

- 1976: Merkle joins them for his PhD

4 Kilian-Micali Proofs: Merkle + PCP = SNARG

• Fix a PCP for ϕ :

• Prove $(x, w) \rightarrow \pi$

• Query $(x) \rightarrow i_1, i_2, i_3$

• Decide $(x, \pi_{i_1}, \pi_{i_2}, \pi_{i_3})$

• The Kilian succinct proof for ϕ is P_K, V_K : [Kilian'92]

$P_K(x, w)$

$V_K(x)$

$\pi \leftarrow \text{Prove}(x, w)$

$C \leftarrow \text{Commit}(\pi)$

$\xrightarrow[\substack{C \\ i}]{} i_1, i_2, i_3 \leftarrow \text{Query}(x)$

for $i_j \in \vec{i}$:

$p_j \leftarrow \text{IdxOpen}(\pi, i_j) \mid (b_i, p_j)_{j=1}^3$
 $b_j \leftarrow \pi_{i_j}$

$\wedge \text{IdxVerify}(C, i_j, b_j, p_j)$

$\wedge \text{Decide}(x, b_1, b_2, b_3)$

• Efficiency:

• comm. $|C| + 3|p_j| + 3 + 3 \cdot \log |\pi|$

$O(1) + O(\log |\pi|)$

$O(\log |\phi|)$

Merkle: $|p_j| = \log |\pi|$

PCP: Prove is poly

• computation:

$O(1 \times 1) + O(\log |\phi|)$

Decide IdxVerify

• Security

• Complete: PCP + Merkle completeness

• Sound: (computationally): Let the PCP has error ϵ . Π_K is

• intuition: if P^* wins with "wrong" b_i , $(\epsilon + \text{negl})$ -sound;
 it could break index binding assuming idx binding

• proof challenge: defining "right" b_i - defining π

• Fix $x \notin \mathcal{L}_\phi$, PPT P^*, r_P

• $\Pr_{\text{Query}} [\langle P^*, v_k \rangle(x) = 1]$

define $p_{i,b} = \Pr_i [\exists j \text{ } i_j = i \wedge b_j = b \wedge p_j \text{ valid}]$
passes Idk Verify

define $\pi_i^* = \operatorname{argmax}_b p_{i,b}$

('canonical' π^* - most likely)

define $\text{Bad}(\vec{i}) \triangleq \exists j \text{ } b_j \neq \pi_{i_j}^* \wedge p_j \text{ valid}$

• $\leq \Pr_i [\text{Decide}(x, \pi_{i_1}^*, \pi_{i_2}^*, \pi_{i_3}^*) = 1] + \Pr_i [\text{Bad}(\vec{i})]$

$\underbrace{\hspace{15em}}_s \quad \underbrace{\hspace{10em}}_{\leq \text{negl. why?}}$

informally:

• If not, we break index binding.

• Consider B , an index-binding adversary.

• runs P^*, v_k for $\vec{i}$ s.t. $\text{Bad}(\vec{i}) @ j$

• runs P^*, v_k for $\vec{i}'$ containing non-negl
 $i'_j = i_j$ with valid p'_j and $\pi_{i'_j}^* = b_j$

• this is $\perp$ and non-negl

b/c argmax in π^* 's definition.

• gives a non-negl attack on index-binding:

$\neg (c, b_j, b'_j, p_j, p'_j) \quad i$

• Last thing: Get a SNARG via Fiat-Shamir [Micali '94]

WOW!

Computationally sound proofs
with $O(\log |\phi|)$ size!

- Next 3 lectures: making this practical
by removing the PCPs!