CONSTANT-ROUND CZK PROOFS for NP

ALON ROSEN

IDC HERZLIYA

FACT: FOUNDATIONS & APPLICATIONS of CRYPTOGRAPHIC THEORY

The Goal

Goal: construct proof for every $L \in \mathrm{NP}$

- in computational ZK
- with negligible soundness
- and a constant number of rounds

Need to address:

- malleability
- aborts in simulation

Recall: CZK proof for HAM

- $G_0 = \pi(w)$
- $G_1 = \pi, \pi(G)$

P **V**

$\boldsymbol{c} = Com(G_0, G_1)$ $\longrightarrow$

$\longleftarrow$ $b = V^*(\boldsymbol{c})$

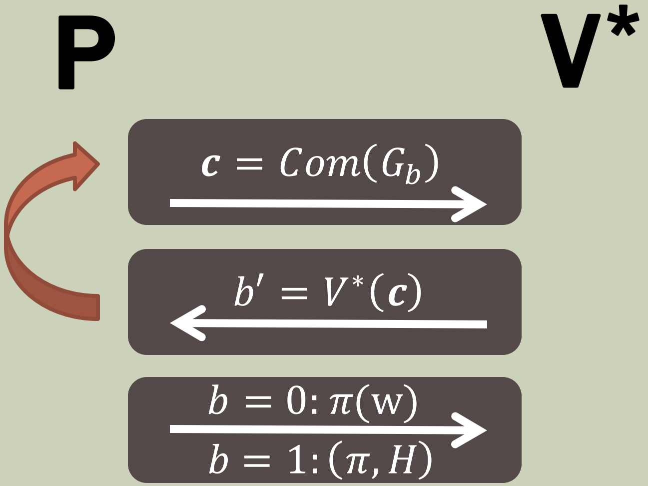
$b = 0: \pi(\mathrm{w})$ $\longrightarrow$
$b = 1: (\pi, H)$

- **<u>Prover</u>: commit to** G_0, G_1
- **<u>Verifier</u>: send** $b \in_R \{0,1\}$
- **<u>Prover</u>: decommit to** G_b

- **<u>Completeness</u>: can always make sure that** G_0, G_1 **are <u>valid</u>**
- **<u>Soundness</u>: either** G_0 **or** G_1 **is <u>invalid</u>**
- **<u>Zero-Knowledge</u>: given** b **can always ensure that** G_b **is <u>valid</u>**

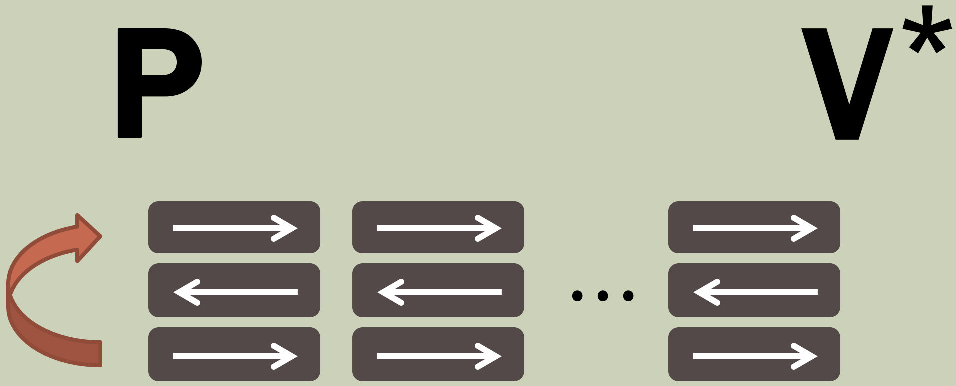
Zero-Knowledge

- $G_0 = \pi(w)$
- $G_1 = \pi, \pi(G)$



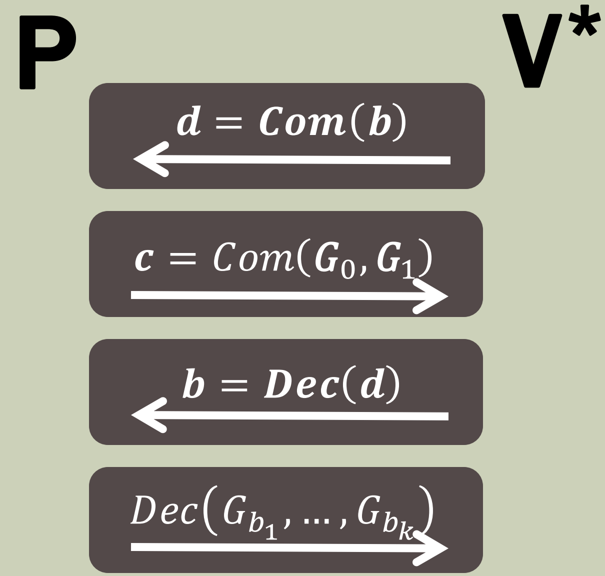
- **<u>Simulator</u>: sample** $b \in_R \{0,1\}$
- **<u>Simulator</u>: commit to** G_0, G_1 **so that** G_b **is <u>valid</u>**
- **<u>Verifier</u>*: send** $b' = V^*(\boldsymbol{c})$
- **<u>Simulator</u>: if** $b' = b$ **decommit to** G_b**, otherwise repeat**

Parallel repetition



- To reduce soundness error – repeat k times in parallel
- <u>Problem</u>: V^*'s challenge is now a string $\boldsymbol{b} \in_R \{0,1\}^k$
- Simulator's expected number of guessing attempts is 2^k
- <u>Solution</u>: Let verifier commit to $\boldsymbol{b}$ in advance

Parallel HAM



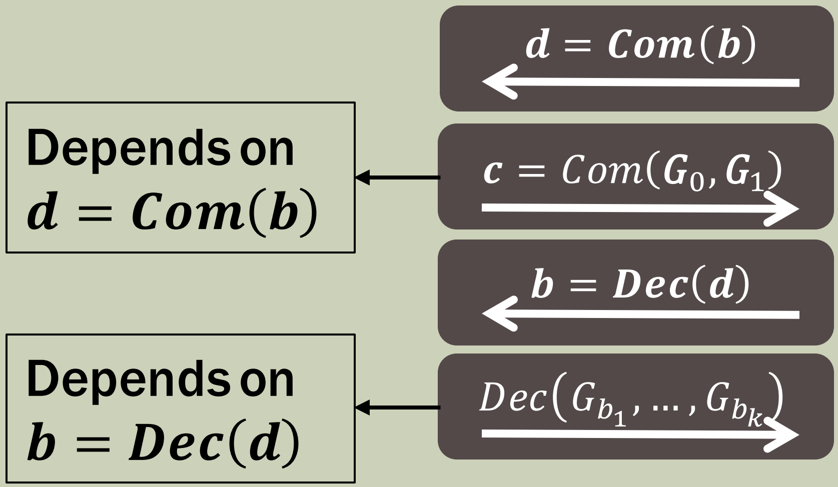
- $G_0 = \pi_1(w), \dots, \pi_k(w)$
- $G_1 = \pi_1, \pi_1(G), \dots, \pi_k, \pi_k(G)$

- <u>Verifier</u>: commit to $b \in_R \{0,1\}^k$
- <u>Prover</u>: commit to G_0, G_1
- <u>Verifier</u>: decommit to b
- <u>Prover</u>: decommit to $G_{b_1}, \dots, G_{b_k}$

- <u>Soundness</u>:
  - Relies on hiding of Com
  - Probability that $G_{b_1}, \dots, G_{b_k}$ are <u>all valid</u> is at most 2^{-k}
- <u>Zero-Knowledge</u>: given b_i can ensure that G_{b_i} is <u>valid</u>

Malleability of Prover Commitment

- $\boldsymbol{Com}$ must be statistically hiding
- Otherwise P can generate $\boldsymbol{c} = Com(\boldsymbol{G_0}, \boldsymbol{G_1})$ that depends on $\boldsymbol{d} = \boldsymbol{Com}(\boldsymbol{b})$ so that upon seeing $\boldsymbol{b} = Dec(\boldsymbol{d})$ he can generate valid $\boldsymbol{Dec}\left(G_{b_1}, \dots, G_{b_k}\right)$



- Succeeding in doing so would not necessitate P to violate the (computational) hiding property of $\boldsymbol{Com}$
- "Man-in-the-middle" attacks are feasible and devastating
- This "malleability" issue is averted by using $\boldsymbol{Com}$ that is statistically hiding

Statistically-hiding Commitments

<u>Definition</u>: A <u>statistically-hiding</u> (Com, Dec) satisfies:

<u>Statistical hiding</u>: $\forall R^* \; \forall m_1, m_2$

$$Com(m_1) \cong_s Com(m_2)$$

<u>Computational binding</u>: $\forall PPT \; C^* \; \forall m_1 \neq m_2$

$$Pr[C^* \text{ wins the binding game}] \leq neg(n)$$

- Can also consider commitments that are simultaneously computationally hiding and binding
- <u>Exercise</u>: There do not exist commitments that are simultaneously statistically hiding and binding
- Instance-dependent: hiding for $x \in L$, binding for $x \notin L$

Examples (statistically-hiding)

- **Pedersen (assuming DL):**

$$\boldsymbol{Com}_{g,h}(m,r) = h^r \cdot g^m$$

- **Any CRH** $H: \{0,1\}^* \rightarrow \{0,1\}^n$**:**

$$\boldsymbol{Com}_H(m,r) = \ (H(r), h(r) \oplus m)$$

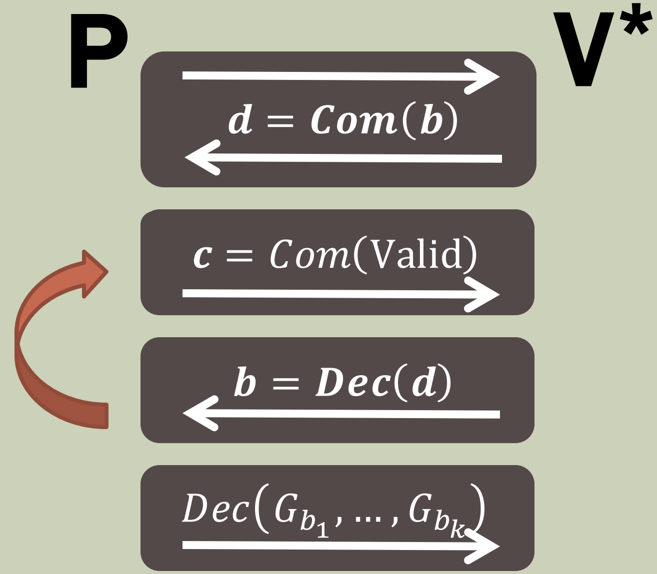
- **"Random oracle"** $H: \{0,1\}^* \rightarrow \{0,1\}^n$**:**

$$\boldsymbol{Com}(m) = H(m)$$

- **Any OWF:** $poly(n)$ **rounds of interaction**

Zero-Knowledge (attempt)

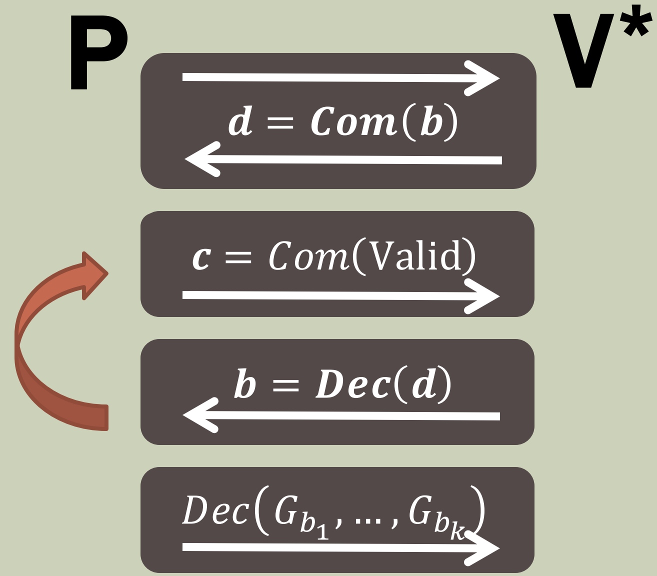
(garbage: all 0's string)



- Verifier: commit to $\boldsymbol{b} \in_R \{0,1\}^k$
- Simulator: commit to garbage
- Verifier*: decommit to $\boldsymbol{b}$
- Simulator: rewind and adjust garbage to be valid
- $\boldsymbol{Com}$ is comp. binding so V^* cannot decommit to $\boldsymbol{b}' \neq \boldsymbol{b}$
- But what if V^* refuses to decommit altogether?
  - V^* might ABORT w/ unknown probability $0 \le p \le 1$
  - Simulator needs to generate the correct distribution

A Naïve Simulator

(<u>garbage</u>: all 0's string)



<u>Naïve simulator</u>:

- commit to <u>garbage</u>
- If $V^*(c) =$ ABORT, halt
- If $V^*(c) \neq$ ABORT,
    - a) rewind and adjust <u>garbage</u> to be <u>valid</u>
    - b) obtain decommitment to b from V^*
    - c) Repeat (a),(b) until $V^*(c) \neq$ ABORT again

<u>The problem</u>: $Pr[V^* \neq \text{ABORT}]$ may change depending on whether simulator committed to <u>garbage</u> or to <u>valid</u>

The Issue

Let

$$s(n) = Pr[V^* \neq \textbf{ABORT} | \text{ garbage }]$$
$$t(n) = Pr[V^* \neq \textbf{ABORT} | \text{ valid }]$$

then

$$\mathbb{E}[\#\text{repetitions } of\ (a), (b)] = s(n)/t(n)$$

Suppose that for infinitely many n's

$$s(n) = 2^{-n}$$
$$t(n) = 2^{-2n}$$

Then for these n's, $s(n)/t(n)$ is too large!

Fixing the Naïve Simulator

Theorem [GK'91]: If statistically-hiding commitments exist then every $L \in \mathrm{NP}$ has a ZK proof with soundness error 2^{-k}

Round-optimal[K'12]: if a language L has a four-round zero-knowledge proof then $L \in \mathrm{coMA}$

The GK solution:

- **have the simulator first obtain an estimate $\tilde{t}(n)$ on $t(n)$**
- **achieved by rewinding with valid commitment until $m(n)$ successful decommits occur for some $m(n) = poly(n)$**
- **In step (c), the simulator then repeats (a),(b) up to some $poly(n)/\tilde{t}(n)$ repetitions, unless $V^*(\boldsymbol{c}) \neq$ ABORT again**

A Simpler Solution

The idea [R'04]: V^* commits to challenge $\boldsymbol{b}$ in a way that allows extraction of $\boldsymbol{b}$ before $\boldsymbol{c}$ is even sent

Stage I:

- Verifier: commit to $\boldsymbol{b} \in_R \{0,1\}^k$ and to

  $\left\langle \begin{matrix} \boldsymbol{b}_1^0, \boldsymbol{b}_2^0 & \ldots & , \boldsymbol{b}_n^0 \\ \boldsymbol{b}_1^1, \boldsymbol{b}_2^1 & & , \boldsymbol{b}_n^1 \end{matrix} \right\rangle$ so that $\forall i \in [n],\ \boldsymbol{b}_i^0 \oplus \boldsymbol{b}_i^1 = \boldsymbol{b}$
- Prover: send n random bits $r_1, \ldots, r_n \in_R \{0,1\}^n$
- Verifier: decommit to $\boldsymbol{b}_1^{r_1}, \boldsymbol{b}_2^{r_2}, \ldots, \boldsymbol{b}_n^{r_n}$

Stage II:

- Run 3-round protocol for HAM (parallel version) with $\boldsymbol{b}$ as challenge (V decommits to $\boldsymbol{b}$ and $\boldsymbol{b}_1^{1-r_1}, \ldots, \boldsymbol{b}_n^{1-r_n}$)

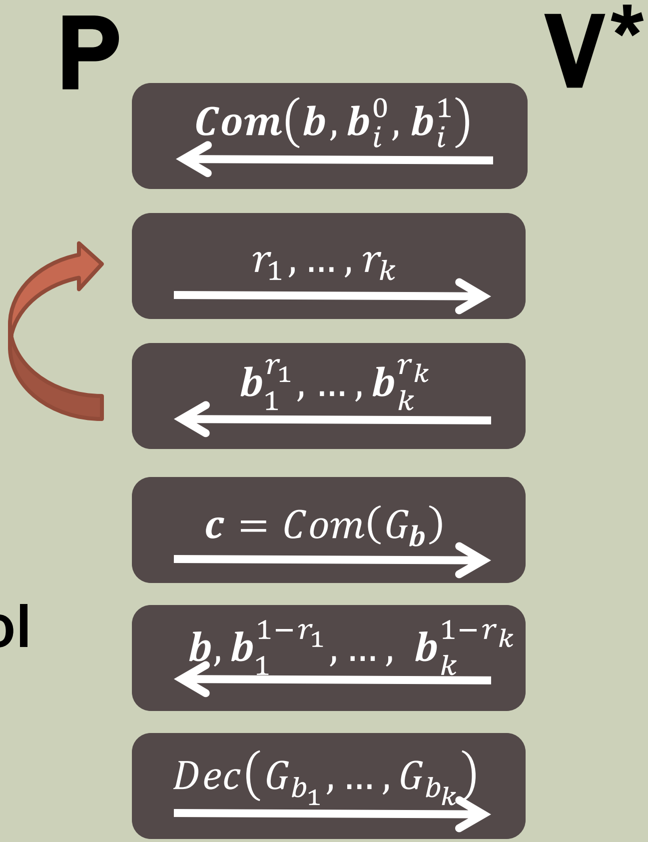
Simulating the protocol

Simulator:

- **Learn $\boldsymbol{b}$ using naïve rewinding by learning $\boldsymbol{b}_i^0, \boldsymbol{b}_i^1$ for some $i \in [n]$**

$$\left\langle \begin{matrix} \boldsymbol{b}_1^0, & \boldsymbol{b}_2^0 & \ldots & , \boldsymbol{b}_n^0 \\ \boldsymbol{b}_1^1, & \boldsymbol{b}_2^1 & & , \boldsymbol{b}_n^1 \end{matrix} \right\rangle \to \boldsymbol{b}_2^0 \oplus \boldsymbol{b}_2^1 = \boldsymbol{b}$$

- **Given $\boldsymbol{b}$ can simulate 3-round protocol**

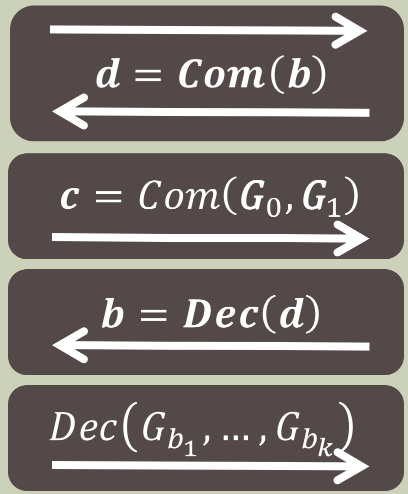


The point:

- **rewindings are non adaptive ($r_1, \ldots, r_k$ are random)**
- $s(n) = t(n)$ **by definition**

What about Proof of Knowledge?

The 5-round protocol seems to not be a POK:



- in order to extract, one must obtain different responses from the prover relative to the same first message $\boldsymbol{c}$
- However, V (and thus extractor) is bound to $\boldsymbol{b}$ before P commits to $\boldsymbol{c}$, and the value of $\boldsymbol{c}$ may depend on V's commitment to $\boldsymbol{b}$
- Thus the extractor cannot change the query $\boldsymbol{b}$ without P changing $\boldsymbol{c}$

The Solution

- G_0, G_1 as before

P **V**

$Com(\boldsymbol{G_0}, \boldsymbol{G_1}) \rightarrow$

$\leftarrow \boldsymbol{d} = \boldsymbol{Com}(\boldsymbol{b}_1)$

$\boldsymbol{e} = Com(\boldsymbol{b}_2) \rightarrow$

$\leftarrow \boldsymbol{b}_1 = \boldsymbol{Dec}(\boldsymbol{d})$

$Dec(\boldsymbol{b}_2, G_{c_1}, \ldots, G_{c_k}) \rightarrow$

- <u>Prover</u>: commit to $\boldsymbol{G}_0, \boldsymbol{G}_1$
- <u>Verifier</u>: commit to $\boldsymbol{b}_1 \in_R \{0,1\}^k$
- <u>Prover</u>: commit to $\boldsymbol{b}_2 \in_R \{0,1\}^k$
- <u>Verifier</u>: decommit to $\boldsymbol{b}_1$
- <u>Prover</u>: decommit to $\boldsymbol{b}_2$ and $G_{c_1}, \ldots, G_{c_k}$ where $\boldsymbol{c} = \boldsymbol{b}_1 \oplus \boldsymbol{b}_2$

<u>Theorem [L'12]</u>: If <u>statistically-hiding</u> commitments exist then every $L \in \mathrm{NP}$ has a ZKPOK with soundness error 2^{-k}

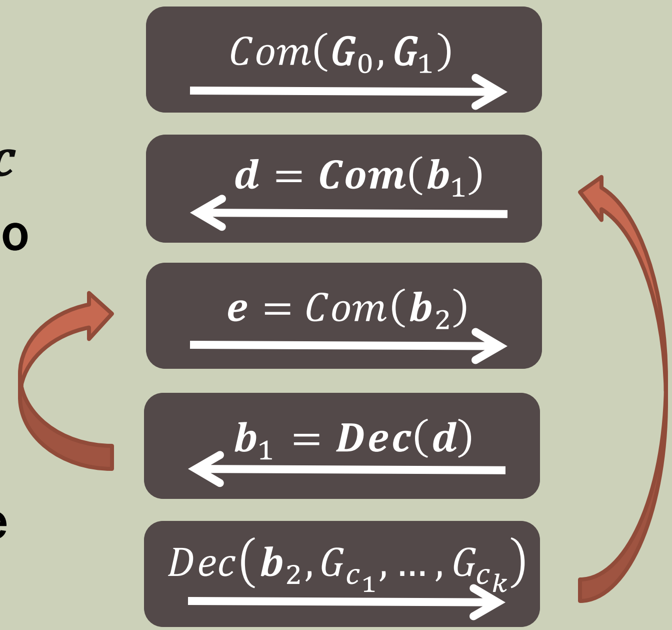
ZK and POK

<u>Zero-knowledge</u>:

- Simulator guesses ahead of time a string $\boldsymbol{c}$
- It then obtains $\boldsymbol{b}_1$, and rewinds V in order to set $\boldsymbol{b}_2$ such that $\boldsymbol{b}_1 \oplus \boldsymbol{b}_2 = \boldsymbol{c}$

<u>Proof of knowledge</u>:

- Extractor rewinds P multiple times relative to the same first message
- it obtains multiple openings with different strings $\boldsymbol{c} = \boldsymbol{b}_1 \oplus \boldsymbol{b}_2$
- This enables extraction from the HAM protocol, albeit with some complications



Summary

Saw:

- CZK proof of knowledge $\forall L \in \mathrm{NP}$
- with negligible soundness
- and a constant number of rounds

Issues addressed:

- malleability
- aborts in simulation

Issues still to be addressed:

- public-coin
- Strict polynomial-time simulation

History

Ariel Kahan

Jonathan Katz

Yehuda Lindell

Alon Rosen

Ivan Damgård

Torben Pryds Pedersen

Birgit Pfitzmann

Ramarathnam Venkatesan

Moti Yung

Iftach Haitner

Omer Reingold

The End

Questions?