Lower Bound on SNARGs in the Random Oracle Model

Daniel Nukrai

Joint work with
Iftach Haitner & Eylon Yogev
Succinct Non-Interactive Arguments
SNARGS
SNARGs

Prover $\pi$ Verifier

$x \in L$
SNARGs

Verifier

Prover

CRS

\( \pi \)

\( x \in L \)
SNARGs in the ROM

SNARG: **Succinct** Non-interactive **Argument**

ROM: **Random Oracle Model**

\[ \phi \in \mathcal{L} \]

Prover

Verifier
SNARGs in the ROM

SNARG: **Succinct Non-interactive Argument**
ROM: **Random Oracle Model**

\[ \phi \in \mathcal{L} \]

\[ \zeta : \{0,1\}^* \rightarrow \{0,1\}^\lambda \]

Prover

Verifier

\[ \pi \]
SNARGs in the ROM

SNARG: **Succinct** Non-interactive Argument
ROM: **Random** Oracle Model

\[ \zeta : \{0,1\}^* \rightarrow \{0,1\}^\lambda \]

Prover \rightarrow \phi \in L \rightarrow \pi

Verifier
SNARGs in the ROM

SNARG: Succinct Non-interactive Argument

ROM: Random Oracle Model

\( \phi \in L \)
SNARGs in the ROM

SNARG: Succinct Non-interactive Argument
ROM: Random Oracle Model

\[ \zeta: \{0,1\}^* \rightarrow \{0,1\}^\lambda \]

Prover \[\pi\] Verifier

\[ \phi \in L \]
SNARGs in the ROM

SNARG: **Succinct Non-interactive Argument**

ROM: **Random Oracle Model**

- Soundness against (computationally unbounded) **query bounded provers**
SNARGs in the ROM

SNARG: Succinct Non-interactive Argument
ROM: Random Oracle Model

- Soundness against (computationally unbounded) query bounded provers
- $2^\lambda \gg$ instance size ($n$) and cheating prover running time ($t$)
Completeness

\[ \xi: \{0, 1\}^* \to \{0, 1\}^d \]

\[ \phi \in L \]

Prover \[\pi\] Verifier
Completeness

$\alpha$-completeness: for every $\phi \in L$:

$$\Pr_{\zeta}\left[V_{\zeta}(\phi, \pi) = 1 : \pi \leftarrow P_{\zeta}\right] \geq \alpha$$
\((t, \epsilon)\)-soundness

\[\zeta : \{0, 1\}^* \rightarrow \{0, 1\}^\lambda\]

Verifier

Prover

\[\phi \in L\]

\[\pi\]
\[(t, \varepsilon)\text{-soundness}\]

\[(t, \varepsilon)\text{-soundness: for any } \phi \notin L \text{ and } t\text{-query (comp. unbounded) } \tilde{P}:\]

\[
\Pr_{\zeta} \left[ V_{\zeta}(\phi, \pi) = 1 : \pi \leftarrow \tilde{P}^\zeta \right] \leq \varepsilon
\]
Importance of the ROM
Importance of the ROM

- **Simple** information-theoretic model
Importance of the ROM

- **Simple** information-theoretic model
- Supports many well-known constructions
Importance of the ROM

- **Simple** information-theoretic model
- Supports many well-known constructions
- Supports many well-known lower bounds
Importance of the ROM

- **Simple** information-theoretic model
- Supports many well-known **constructions**
- Supports many well-known **lower bounds**
- ROM **huristic**: ROM is instantiated via **lightweight** crypto (e.g. SHA-256)
Importance of the ROM

- **Simple** information-theoretic model
- Supports many well-known constructions
- Supports many well-known lower bounds
- **ROM heuristic**: ROM is instantiated via **lightweight** crypto (e.g. SHA-256)

Constructions in ROM heuristic are:
Importance of the ROM

- **Simple** information-theoretic model
- Supports many well-known constructions
- Supports many well-known lower bounds
- ROM heuristic: ROM is instantiated via lightweight crypto (e.g. SHA-256)

Constructions in ROM heuristic are:

- Fast to compute
Importance of the ROM

- **Simple** information-theoretic model
- Supports many well-known constructions
- Supports many well-known lower bounds
- **ROM heuristic**: ROM is instantiated via lightweight crypto (e.g. SHA-256)

Constructions in ROM heuristic are:

- Fast to compute
- No trusted setup
Importance of the ROM

- **Simple** information-theoretic model
- Supports many well-known **constructions**
- Supports many well-known **lower bounds**
- **ROM heuristic:** ROM is instantiated via **lightweight** crypto (e.g. SHA-256)

Constructions in ROM heuristic are:

- Fast to compute
- No trusted setup
- Potentially **post-quantum** …
Importance of the ROM

- **Simple** information-theoretic model
- Supports many well-known constructions
- Supports many well-known lower bounds
- ROM heuristic: ROM is instantiated via **lightweight** crypto (e.g. SHA-256)

Constructions in ROM heuristic are:

- Fast to compute
- No trusted setup
- Potentially **post-quantum** ...
- Widely used in practice
Known ROM-SNARGS constructions
Known ROM-SNARGS constructions

- Micali’94, BCS’16:
  - Proof length: $O\left(\left(\log\frac{t}{\epsilon}\right)^2 \cdot \log n\right)$
  - # verifier queries: $\Theta\left(\log\frac{t}{\epsilon}\right)$
Known ROM-SNARGS constructions

- Micali’94, BCS’16:
  - Proof length: $O\left(\left(\frac{\log t}{\epsilon}\right)^2 \cdot \log n\right)$
  - # verifier queries: $\Theta\left(\frac{\log t}{\epsilon}\right)$

- CY’21:
  - Proof length: $O\left(\frac{\log t}{\epsilon} \cdot \log t \cdot \log n\right)$
  - # verifier queries: $\Theta\left(\frac{\log t}{\epsilon}\right)$
Known ROM-SNARGS constructions

- Micali’94, BCS’16:
  - Proof length: $O\left(\left(\frac{\log t}{\epsilon}\right)^2 \cdot \log n\right)$
  - # verifier queries: $\Theta\left(\frac{t}{\epsilon}\right)$

- CY’21:
  - Proof length: $O\left(\frac{t}{\epsilon} \cdot \log t \cdot \log n\right)$
  - # verifier queries: $\Theta\left(\frac{t}{\epsilon}\right)$
Proof length

\[ \Theta\left(\left(\frac{\log t}{\epsilon}\right)^2 \cdot \log n\right) \]

\[ \Theta\left(\frac{\log t}{\epsilon} \cdot \log t \cdot \log n\right) \]

\[ \tilde{O}\left(\frac{\log t}{\epsilon}\right) \]

Micali

CY’21

Lower bound
Proof length

\[ \Theta\left(\left(\log\frac{t}{\varepsilon}\right)^2 \cdot \log n\right) \]

Micali

\[ \Theta\left(\frac{\log t}{\varepsilon} \cdot \log t \cdot \log n\right) \]

CY’21

Open

\[ \tilde{O}\left(\log\frac{t}{\varepsilon}\right) \]

Lower bound
Our lower bound
Our lower bound

**Thm:** Assuming rnd ETH, any “natural” ROM-SNARG \((P, V)\) of

\((t, \epsilon)\)-soundness has proof size \(\Omega\left(\frac{\log t}{\epsilon} \cdot \log t/\log q_P\right)\)

Tight up to \(\log n \cdot \log q_P\) term ([CY’21] proof size is \(\Theta\left(\frac{\log t}{\epsilon} \cdot \log t \cdot \log n\right)\) )
Our lower bound

**Thm:** Assuming rnd ETH, any “natural” ROM-SNARG \((P, V)\) of 

\[(t, \epsilon)-\text{soundness has proof size } \Omega\left(\frac{\log t}{\epsilon} \cdot \log t / \log q_P\right)\]

Tight up to \(\log n \cdot \log q_P\) term (\([CY'21]\) proof size is \(\Theta\left(\frac{\log t}{\epsilon} \cdot \log t \cdot \log n\right)\))

**Natural constructions:**
Our lower bound

**Thm:** Assuming rnd ETH, any “natural” ROM-SNARG \((P, V)\) of

\[(t, \epsilon)\)-soundness has proof size \(\Omega \left( \log \frac{t}{\epsilon} \cdot \log t / \log q_P \right)\)

Tight up to \(\log n \cdot \log q_P\) term ([CY'21] proof size is \(\Theta \left( \log \frac{t}{\epsilon} \cdot \log t \cdot \log n \right)\))

**Natural** constructions:

1. Non-adaptive deterministic verifier
Our lower bound

**Thm:** Assuming rnd ETH, any “natural” ROM-SNARG \((P, V)\) of (\(t, \epsilon\))-soundness has proof size \(\Omega\left(\log \frac{t}{\epsilon} \cdot \log \frac{t}{\log q_P}\right)\)

Tight up to \(\log n \cdot \log q_P\) term ([CY'21] proof size is \(\Theta\left(\log \frac{t}{\epsilon} \cdot \log t \cdot \log n\right)\))

**Natural constructions:**

1. Non-adaptive deterministic verifier
2. Salted soundness
Our lower bound

**Thm:** Assuming rnd ETH, any “natural” ROM-SNARG \((P, V)\) of

\((t, \epsilon)\)-soundness has proof size \(\Omega\left(\frac{\log t}{\epsilon} \cdot \log t / \log q_P\right)\)

Tight up to \(\log n \cdot \log q_P\) term ([CY’21] proof size is \(\Theta\left(\frac{\log t}{\epsilon} \cdot \log t \cdot \log n\right)\))

**Natural** constructions:

1. Non-adaptive deterministic verifier
2. Salted soundness
3. Reasonable \(q_P\) and \(q_V\) \((P/V\) query complexity) as functions of \(n\)
Our lower bound

**Thm:** Assuming rnd ETH, any “natural” ROM-SNARG \((P, V)\) of

\((t, \epsilon)\)-soundness has proof size \(\Omega\left(\frac{\log t}{\epsilon} \cdot \log t / \log q_P\right)\)

Tight up to \(\log n \cdot \log q_P\) term ([CY’21] proof size is \(\Theta\left(\frac{\log t}{\epsilon} \cdot \log t \cdot \log n\right)\))

**Natural constructions:**

1. Non-adaptive deterministic verifier
2. Salted soundness
3. Reasonable \(q_P\) and \(q_V\) (\(P/V\) query complexity) as functions of \(n\)

All known (non-contrived) constructions are natural
Proof size for natural constructions

\( \Theta \left( \left( \frac{\log t}{\epsilon} \right)^2 \cdot \log n \right) \)

\( \Theta \left( \frac{\log t \cdot \log t \cdot \log n}{\epsilon} \right) \)

\( \Omega \left( \frac{\log t \cdot \log t / \log q_P}{\epsilon} \right) \)

\( \Omega \left( \frac{\log t}{\epsilon} \right) \)

Micali

CY’21

This work

Lower bound
Lower bound on ROM SubVector Commitment

Subvector commitment (SVC) – non-interactive cmt with local opening.
Lower bound on ROM SubVector Commitment

Subvector commitment (SVC) – **non-interactive** cmt with local opening.

- \((t, \varepsilon)\)-binding in ROM
Lower bound on ROM SubVector Commitment

Subvector commitment (SVC) – non-interactive cmt with local opening.

- \((t, \epsilon)\)-binding in ROM
- \(\alpha\) – commitment length
Subvector commitment (SVC) – non-interactive cmt with local opening.

- $(t, \epsilon)$-binding in ROM
- $\alpha$ – commitment length
- $\beta(m)$ – length of opening $m$ elements.
Lower bound on ROM SubVector Commitment

Subvector commitment (SVC) – non-interactive cmt with local opening.

- \((t, \epsilon)\)-binding in ROM
- \(\alpha\) – commitment length
- \(\beta(m)\) – length of opening \(m\) elements.

**Thm:** Assuming rnd ETH, any “natural” ROM-SVC \((S, R)\) of 
\((t, \epsilon)\)-binding has 
\[
\alpha + \beta\left(\log \frac{t}{\epsilon}\right) \in \Omega\left(\log \frac{t}{\epsilon} \cdot \log t/\log q_S\right)
\]
Lower bound on ROM SubVector Commitment

Subvector commitment (SVC) – **non-interactive** cmt with **local** opening.

- \((t, \epsilon)\)-binding in ROM
- \(\alpha\) – commitment length
- \(\beta(m)\) – length of opening \(m\) elements.

**Thm:** Assuming rnd ETH, any “natural” ROM-SVC \((S, R)\) of 
\((t, \epsilon)\)-binding has \(\alpha + \beta\left(\log \frac{t}{\epsilon}\right) \in \Omega\left(\log \frac{t}{\epsilon} \cdot \log t / \log q_S\right)\)

- Tight bound up to \(\log n \cdot \log q_S\) term \((n\) is commited string length)
Lower bound on ROM SubVector Commitment

Subvector commitment (SVC) – non-interactive cmt with local opening.

- \((t, \epsilon)\)-binding in ROM
- \(\alpha\) – commitment length
- \(\beta(m)\) – length of opening \(m\) elements.

**Thm:** Assuming rnd ETH, any “natural” ROM-SVC \((S, R)\) of

\((t, \epsilon)\)-binding has \(\alpha + \beta\left(\log \frac{t}{\epsilon}\right) \in \Omega\left(\log \frac{t}{\epsilon} \cdot \log \frac{t}{\log q_S}\right)\)

- Tight bound upto \(\log n \cdot \log q_S\) term (\(n\) is commited string length)

**How to prove:** SVC + PCP \(\rightarrow\) SNARG
Lower bound on ROM SubVector Commitment

Subvector commitment (SVC) – **non-interactive** cmt with **local** opening.

- \((t, \epsilon)\)-binding in ROM
- \(\alpha\) – commitment length
- \(\beta(m)\) – length of opening \(m\) elements.

**Thm:** Assuming rnd ETH, any “natural” ROM-SVC \((S, R)\) of
\((t, \epsilon)\)-binding has \(\alpha + \beta \left( \log \frac{t}{\epsilon} \right) \in \Omega \left( \log \frac{t}{\epsilon} \cdot \log \frac{t}{\log q_S} \right)\)

- Tight bound upto \(\log n \cdot \log q_S\) term \((n\) is committed string length)  

**How to prove:** SVC + PCP \(\rightarrow\) SNARG
Salted Soundness

Malicious prover can resample queries, and choose the answers he likes
Malicious prover can **resample** queries, and choose the answers he likes
Malicious prover can resample queries, and choose the answers he likes.
Salted Soundness

Malicious prover can resample queries, and choose the answers he likes
Salted Soundness

Malicious prover can resample queries, and choose the answers he likes
Malicious prover can resample queries, and choose the answers he likes
Salted Soundness

Malicious prover can resample queries, and choose the answers he likes.
Salted Soundness

Malicious prover can resample queries, and choose the answers he likes
Malicious prover can resample queries, and choose the answers he likes

- All known constructions have salted soundness
Malicious prover can resample queries, and choose the answers he likes

- All known constructions have salted soundness
- Easy to construct a SNARG that has no salted soundness
Salted Soundness

Malicious prover can resample queries, and choose the answers he likes
- All known constructions have salted soundness
- Easy to construct a SNARG that has no salted soundness
- Seems hard to get rid of w/o making the verifier adaptive
Short SNARGS to Fast Algorithms

SNARG with small proof length $\Rightarrow$ SNARG with small query complexity

CY’20
Short SNARGS to Fast Algorithms

- SNARG with small proof length
- SNARG with small query complexity
- Fast algorithm for SAT

CY’20
Short SNARGS to Fast Algorithms

- SNARG with small proof length
- SNARG with small query complexity
- Fast algorithm for SAT

- Proof size is unchanged
Short SNARGS to Fast Algorithms

SNARG with small proof length $\implies$ SNARG with small query complexity $\implies$ Fast algorithm for SAT

- Proof size is unchanged
- Soundness is unchanged
Short SNARGS to Fast Algorithms

SNARG with small proof length $\Rightarrow$ SNARG with small query complexity $\Rightarrow$ Fast algorithm for SAT

- Proof size is unchanged
- Soundness is unchanged
- Nontrivial completeness
Short SNARGS to Fast Algorithms

- SNARG with small proof length
- SNARG with small query complexity
- Fast algorithm for SAT

- Proof size is unchanged
- Soundness is unchanged
- Nontrivial completeness
- Verifier running time: $t^{1/C}$
Short SNARGS to Fast Algorithms

SNARG with small proof length \Rightarrow SNARG with small query complexity \Rightarrow Fast algorithm for SAT

• Proof size is unchanged
• Soundness is unchanged
• Nontrivial completeness
• Verifier running time: \( t^{1/C} \)
• Query complexity \( \log \frac{t}{\epsilon} \)
Short SNARGs to Low-query SNARGs
Short SNARGs to Low-query SNARGs

Given SNARG \((P, V)\), we modify to \(\tilde{V}\) as follows (\(P\) is unchanged):
Short SNARGs to Low-query SNARGs

Given SNARG \((P, V)\), we modify to \(\tilde{V}\) as follows (\(P\) is unchanged):

Oracle: \(\zeta: \{0,1\}^* \to \{0,1\}^\lambda\)
Short SNARGs to Low-query SNARGs

Given SNARG \((P, V)\), we modify to \(\tilde{V}\) as follows (\(P\) is unchanged):

Oracle: \(\zeta: \{0,1\}^* \rightarrow \{0,1\}^\lambda\)
Input: \((\phi, \pi)\)
Short SNARGs to Low-query SNARGs

Given SNARG \((P, V)\), we modify to \(\tilde{V}\) as follows (\(P\) is unchanged):

Oracle: \(\zeta: \{0,1\}^* \rightarrow \{0,1\}^\lambda\)
Input: \((\phi, \pi)\)

1. Let \(u_1, \ldots, u_m\) denote the queries of \(V(\phi, \pi)\) (Recall, \(V\) is non-adaptive)
Short SNARGs to Low-query SNARGs

Given SNARG \((P, V)\), we modify to \(\tilde{V}\) as follows (\(P\) is unchanged):

Oracle: \(\zeta: \{0,1\}^* \rightarrow \{0,1\}^\lambda\)
Input: \((\phi, \pi)\)

1. Let \(u_1, \ldots, u_m\) denote the queries of \(V(\phi, \pi)\) (Recall, \(V\) is non-adaptive)
2. Sample uniform \(k\)-size subset \(J \subseteq [m]\)
Short SNARGs to Low-query SNARGs

Given SNARG $(P, V)$, we modify to $\tilde{V}$ as follows ($P$ is unchanged):

Oracle: $\zeta: \{0,1\}^* \rightarrow \{0,1\}^\lambda$

Input: $(\phi, \pi)$

1. Let $u_1, \ldots, u_m$ denote the queries of $V(\phi, \pi)$ (Recall, $V$ is non-adaptive)
2. Sample uniform $k$-size subset $J \subseteq [m]$
3. For each $j \in J$: let $a_j = \zeta(u_j)$
Given SNARG \((P, V)\), we modify to \(\tilde{V}\) as follows (\(P\) is unchanged):

**Oracle**: \(\zeta: \{0,1\}^* \rightarrow \{0,1\}^k\)

**Input**: \((\phi, \pi)\)

1. Let \(u_1, \ldots, u_m\) denote the queries of \(V(\phi, \pi)\) (Recall, \(V\) is non-adaptive)

2. Sample uniform \(k\)-size subset \(J \subseteq [m]\)

3. For each \(j \in J\): let \(a_j = \zeta(u_j)\)

4. For each \(j \notin J\): uniformly sample \(2^r\) candidate answers \(\{a_{j,d}\}\)
Short SNARGs to Low-query SNARGs

Given SNARG \((P, V)\), we modify to \(\widetilde{V}\) as follows (\(P\) is unchanged):

Oracle: \(\zeta: \{0,1\}^* \rightarrow \{0,1\}^d\)

Input: \((\phi, \pi)\)

1. Let \(u_1, \ldots, u_m\) denote the queries of \(V(\phi, \pi)\) (Recall, \(V\) is non-adaptive)
2. Sample uniform \(k\)-size subset \(J \subseteq [m]\)
3. For each \(j \in J\): let \(a_j = \zeta(u_j)\)
4. For each \(j \notin J\): uniformly sample \(2^r\) candidate answers \(\{a_{j,i}\}\)
5. Accept if any combination of these answers makes \(V(\phi, \pi)\) accept
Short SNARGs to Low-query SNARGs

Given SNARG \((P, V)\), we modify to \(\tilde{V}\) as follows (\(P\) is unchanged):

Oracle: \(\zeta: \{0,1\}^* \to \{0,1\}^d\)

Input: \((\phi, \pi)\)

1. Let \(u_1, \ldots, u_m\) denote the queries of \(V(\phi, \pi)\) (Recall, \(V\) is non-adaptive)
2. Sample uniform \(k\)-size subset \(J \subseteq [m]\)
3. For each \(j \in J\): let \(a_j = \zeta(u_j)\)
4. For each \(j \notin J\): uniformly sample \(2^\gamma\) candidate answers \(\{a_{j, a}\}\)
5. Accept if any combination of these answers makes \(V(\phi, \pi)\) accept

\(\gamma \approx \log t\) and \(k \approx |\pi|/\gamma\) (hence, \(|\pi| < \log(t/\epsilon) \cdot \log t \rightarrow q_{\tilde{V}} < \log(t/\epsilon))\)
Short SNARGs to Low-query SNARGs

Given SNARG \((P, V)\), we modify to \(\tilde{V}\) as follows (\(P\) is unchanged):

\[
\begin{align*}
\text{Oracle: } & \zeta: \{0,1\}^* \to \{0,1\}^d \\
\text{Input: } & (\phi, \pi) \\
1. & \text{Let } u_1, \ldots, u_m \text{ denote the queries of } V(\phi, \pi) \quad \text{(Recall, } V \text{ is non-adaptive)} \\
2. & \text{Sample uniform } k \text{-size subset } J \subseteq [m] \\
3. & \text{For each } j \in J: \text{ let } a_j = \zeta(u_j) \\
4. & \text{For each } j \notin J: \text{ uniformly sample } 2^{\gamma} \text{ candidate answers } \{a_{j,i}\} \\
5. & \text{Accept if any combination of these answers makes } V(\phi, \pi) \text{ accept}
\end{align*}
\]

- \(\gamma \approx \log t \text{ and } k \approx |\pi|/\gamma \quad \text{(hence, } |\pi| < \log(t/\epsilon) \cdot \log t \to q_{\tilde{V}} < \log(t/\epsilon))\)

- \((P, V)\) has \((t, \epsilon)\)-salted-soundness \(\to (P, \tilde{V})\) has \((t, \epsilon)\)-soundness
Short SNARGs to Low-query SNARGs

Given SNARG \((P, V)\), we modify to \(\tilde{V}\) as follows (\(P\) is unchanged):

Oracle: \(\zeta: \{0,1\}^* \rightarrow \{0,1\}^\lambda\)

Input: \((\phi, \pi)\)

1. Let \(u_1, \ldots, u_m\) denote the queries of \(V(\phi, \pi)\) (Recall, \(V\) is non-adaptive)
2. Sample uniform \(k\)-size subset \(J \subseteq [m]\)
3. For each \(j \in J\): let \(a_j = \zeta(u_j)\)
4. For each \(j \notin J\): uniformly sample \(2^\gamma\) candidate answers \(\{a_{j,\ell}\}\)
5. Accept if any combination of these answers makes \(V(\phi, \pi)\) accept

- \(\gamma \approx \log t\) and \(k \approx |\pi|/\gamma\) (hence, \(|\pi| < \log(t/\epsilon) \cdot \log t \rightarrow q_{\tilde{V}} < \log(t/\epsilon))\)
- \((P, V)\) has \((t, \epsilon)\)-salted-soundness \(\rightarrow (P, \tilde{V})\) has \((t, \epsilon)\)-soundness
- \((P, \tilde{V})\) has completeness \((\gamma \cdot q_V \cdot \left(\frac{q_V}{k}\right))^{-1}\)
Motivating examples
Motivating examples

Consider the two following argument systems with $\zeta: \{0,1\}^* \rightarrow \{0,1\}^\lambda$: 
Motivating examples

Consider the two following argument systems with $\zeta: \{0,1\}^* \to \{0,1\}^\lambda$:

$$y = \zeta(q_1)$$
Motivating examples

Consider the two following argument systems with $\zeta: \{0,1\}^* \rightarrow \{0,1\}^\lambda$:

\[
y = \zeta(q_1)
\]

Local information: $\pi = \zeta(0)$ and $V$ verifies that

$V$ accepts if $j \in J$ ($j$ is the index of 0)

Short $\pi$ cannot contain many such oracle answers
Motivating examples

Consider the two following argument systems with \( \zeta: \{0,1\}^* \rightarrow \{0,1\}^\lambda \):

\[
y = \zeta(q_1)
\]

\[
y = \zeta(q_1) \oplus \ldots \oplus \zeta(q_k)
\]

**Local** information: \( \pi = \zeta(0) \) and \( V \) verifies that \( \tilde{V} \) accepts if \( j \in J \) (\( j \) is the index of 0)

Short \( \pi \) cannot contain many such oracle answers
Motivating examples

Consider the two following argument systems with $\zeta: \{0,1\}^* \rightarrow \{0,1\}^\lambda$:

**Local** information: $\pi = \zeta(0)$ and $V$ verifies that $\widetilde{V}$ accepts if $j \in J$ ($j$ is the index of 0)

Short $\pi$ cannot contain many such oracle answers

**Global** information: $\pi = \zeta(0) \oplus \zeta(1) \ldots \oplus \zeta(k)$

$\widetilde{V}$ samples $\approx 2^\gamma$ options for each $\zeta(q_i)$

If $\gamma > \lambda/k$, then $\widetilde{V}$ accepts whp.
Motivating examples

Consider the two following argument systems with $\zeta: \{0,1\}^* \to \{0,1\}^\lambda$:

**Local** information: $\pi = \zeta(0)$ and $V$ verifies that

$\tilde{V}$ accepts if $j \in J$ (where $j$ is the index of 0)

Short $\pi$ cannot contain many such oracle answers

**Global** information: $\pi = \zeta(0) \oplus \zeta(1) \ldots \oplus \zeta(k)$

$\tilde{V}$ samples $\approx 2^\gamma$ options for each $\zeta(q_i)$

If $\gamma > \lambda/k$, then $\tilde{V}$ accepts whp.
Completeness
Completeness

- The lemma shows $V$ must make $\approx |\pi|/\gamma$ queries, and the rest can be completed by uniform sampling with some probability $\mathcal{V} \approx \frac{|\pi|}{\gamma}$.
Completeness

- The lemma shows $V$ must make $\approx |\pi|/\gamma$ queries, and the rest can be completed by uniform sampling with some probability
- The probability $V$ guesses correctly the important queries is small, yet nontrivial as $|\pi|$ is small
Completeness

- The lemma shows $V$ must make $\approx |\pi|/\gamma$ queries, and the rest can be completed by uniform sampling with some probability.

- The probability $V$ guesses correctly the important queries is small, yet nontrivial as $|\pi|$ is small.

- This yields completeness slightly larger than the soundness error, $\epsilon$. 
Hitting High-Entropy Events Lemma
**Hitting High-Entropy Events Lemma**

**Lemma** [Hitting High Entropy Events, Informal]:
Let $X = X_1, \ldots, X_n$ be variables over $\left(\{0,1\}^{\lambda}\right)^n$, with $H(X) \geq \lambda \cdot n - \ell$.
Lemma [Hitting High Entropy Events, Informal]:
Let $X = X_1, \ldots, X_n$ be variables over $\left(\{0,1\}^\lambda\right)^n$, with $H(X) \geq \lambda \cdot n - \ell$

Then, $x \leftarrow X$ consist of $O(\ell'/\gamma)$ binding coordinates, when the rest can be completed using uniform sampling of size $2^\gamma$
Hitting High-Entropy Events Lemma

Lemma [Hitting High Entropy Events, Informal]:
Let $X = X_1, \ldots, X_n$ be variables over $\{0,1\}^n$, with $H(X) \geq \lambda \cdot n - \ell$
Then, $x \leftarrow X$ consist of $O(\ell'/\gamma)$ binding coordinates, when the rest can be completed using uniform sampling of size $2^\gamma$

- We first show that for $x \leftarrow X$, exists $B \subseteq [n]$ such that for
Hitting High-Entropy Events Lemma

**Lemma** [Hitting High Entropy Events, Informal]:
Let $X = X_1, \ldots, X_n$ be variables over $(\{0,1\}^\lambda)^n$, with $H(X) \geq \lambda \cdot n - \ell$

Then, $x \leftarrow X$ consist of $O(\ell/\gamma)$ binding coordinates, when the rest can be completed using uniform sampling of size $2^\gamma$

- We first show that for $x \leftarrow X$, exists $B \subseteq [n]$ such that for

  $$X' = (X_{[n] \setminus B} | X_B = x_B)$$

  and all $I \subseteq [n - |B|]$, $H(X'_I) \geq (\lambda - \gamma) \cdot |I|$
Hitting High-Entropy Events Lemma

**Lemma [Hitting High Entropy Events, Informal]:**

Let $X = X_1, \ldots, X_n$ be variables over $\left( \{0,1\}^\lambda \right)^n$, with $H(X) \geq \lambda \cdot n - \ell$

Then, $x \leftarrow X$ consist of $O(\ell / \gamma)$ binding coordinates, when the rest can be completed using uniform sampling of size $2^\gamma$

- We first show that for $x \leftarrow X$, exists $B \subseteq [n]$ such that for

  $$X' = (X_{[n] \setminus B} | X_B = x_B)$$

  and all $I \subseteq \left[ n - |B| \right]$, $H(X'_I) \geq (\lambda - \gamma) \cdot |I|$

- Then we show that for such $B$, sampling $S \leftarrow (\{0,1\}^{\gamma})^n$ intersects the support of $X'$ with high probability.
Hitting High-Entropy Events Lemma

**Lemma** [Hitting High Entropy Events, Informal]:
Let $X = X_1, \ldots, X_n$ be variables over $(\{0,1\}^\lambda)^n$, with $H(X) \geq \lambda \cdot n - \ell$

Then, $x \leftarrow X$ consist of $O(\ell/\gamma)$ binding coordinates, when the rest can be completed using uniform sampling of size $2^\gamma$

- We first show that for $x \leftarrow X$, exists $B \subseteq [n]$ such that for

  $$X' = (X_{[n] \setminus B} \mid X_B = x_B)$$

  and all $I \subseteq \left[ n - |B| \right]$, $H(X'_I) \geq (\lambda - \gamma) \cdot |I|

- Then we show that for such $B$, sampling $S \leftarrow (\{0,1\}^\gamma)^n$ intersects the support of $X'$ with high probability

- We conclude by showing that the expected size of $B$ is $O(\ell/\gamma)$
Soundness
Soundness

Given malicious $P'$ that fools $\tilde{V}$, we construct $P$ that wins the salted soundness game:
Soundness

Given malicious $P'$ that fools $\tilde{V}$, we construct $P$ that wins the salted soundness game:

1. $P$ simulates $P'$ to obtain a proof $\pi$
Soundness

Given malicious $P'$ that fools $\tilde{V}$, we construct $P$ that wins the salted soundness game:

1. $P$ simulates $P'$ to obtain a proof $\pi$
2. Then, $P$ emulates $\tilde{V}(\pi)$
Soundness

Given malicious $P'$ that fools $\tilde{V}$, we construct $P$ that wins the salted soundness game:

1. $P$ simulates $P'$ to obtain a proof $\pi$
2. Then, $P$ emulates $\tilde{V}(\pi)$
   - $\tilde{V}$'s queries are emulated by queries in the game
Soundness

Given malicious $P'$ that fools $\tilde{V}$, we construct $P$ that wins the salted soundness game:

1. $P$ simulates $P'$ to obtain a proof $\pi$
2. Then, $P$ emulates $\tilde{V}(\pi)$
   - $\tilde{V}$'s queries are emulated by queries in the game
3. $P$ chooses the answers that made $\tilde{V}$ accept
Soundness

Given malicious $P'$ that fools $\tilde{V}$, we construct $P$ that wins the salted soundness game:

1. $P$ simulates $P'$ to obtain a proof $\pi$
2. Then, $P$ emulates $\tilde{V}(\pi)$
   - $\tilde{V}'$'s queries are emulated by queries in the game
3. $P$ chooses the answers that made $\tilde{V}$ accept

- Notice the similarity of the salted soundness game to the definition of $\tilde{V}$
Soundness

Given malicious $P'$ that fools $\tilde{V}$, we construct $P$ that wins the salted soundness game:

1. $P$ simulates $P'$ to obtain a proof $\pi$
2. Then, $P$ emulates $\tilde{V}(\pi)$
   - $\tilde{V}$’s queries are emulated by queries in the game
3. $P$ chooses the answers that made $\tilde{V}$ accept

- Notice the similarity of the salted soundness game to the definition of $\tilde{V}$
Conclusions and open problems
Conclusions and open problems

SNARGs in the ROM:
Conclusions and open problems

SNARGs in the ROM:

• Have optimal size between $O\left(\frac{t}{\epsilon} \cdot \log t \cdot \log n\right)$ and $O\left(\frac{\log t}{\epsilon}\right)$
Conclusions and open problems

SNARGs in the ROM:

- Have optimal size between $O\left(\log\frac{t}{\epsilon} \cdot \log t \cdot \log n\right)$ and $O\left(\log\frac{t}{\epsilon}\right)$
- Have size $\Omega\left(\log\frac{t}{\epsilon} \cdot \log t / \log p\right)$ for “natural” constructions
Conclusions and open problems

SNARGs in the ROM:

• Have optimal size between \( O\left( \log\frac{t}{\epsilon} \cdot \log t \cdot \log n \right) \) and \( O\left( \log\frac{t}{\epsilon} \right) \)

• Have size \( \Omega\left( \log\frac{t}{\epsilon} \cdot \log t / \log q_p \right) \) for “natural” constructions
Conclusions and open problems

SNARGs in the ROM:

- Have optimal size between $O\left(\frac{\log t}{\epsilon} \cdot \log t \cdot \log n\right)$ and $O\left(\frac{\log t}{\epsilon}\right)$
- Have size $\Omega\left(\frac{\log t}{\epsilon} \cdot \frac{\log t}{\log q_P}\right)$ for “natural” constructions

Open questions:
Conclusions and open problems

SNARGs in the ROM:

• Have optimal size between $O\left(\frac{t}{\epsilon} \cdot \log t \cdot \log n\right)$ and $O\left(\log \frac{t}{\epsilon}\right)$

• Have size $\Omega\left(\frac{t}{\epsilon} \cdot \log t / \log p\right)$ for “natural” constructions

Open questions:

• General lower bound (for adaptive verifier or without salted soundness)
Conclusions and open problems

SNARGs in the ROM:
- Have optimal size between $O\left(\log\frac{t}{\epsilon} \cdot \log t \cdot \log n\right)$ and $O\left(\log\frac{t}{\epsilon}\right)$
- Have size $\Omega\left(\log\frac{t}{\epsilon} \cdot \log t / \log q_P\right)$ for “natural” constructions

Open questions:
- General lower bound (for adaptive verifier or without salted soundness)
- Build an improved SNARG without salted soundness
Conclusions and open problems

SNARGs in the ROM:
- Have optimal size between $O\left(\frac{\log t}{\epsilon} \cdot \log \cdot \log n\right)$ and $O\left(\frac{\log t}{\epsilon}\right)$
- Have size $\Omega\left(\frac{\log t}{\epsilon} \cdot \log t / \log q_p\right)$ for “natural” constructions

Open questions:
- General lower bound (for adaptive verifier or without salted soundness)
- Build an improved SNARG without salted soundness

Thank You!