Succinct Arguments from MIPs and their Efficiency Benefits

Nir Bitansky  Alessandro Chiesa
How quickly can we verify the result of long computations?

∃ w s.t. M(x, w) = 1 in ≤ T steps?

Proof w checkable in T time

L = \text{Lang}(M) ∈ \text{NP} ⇒ T_L = \text{poly}_L(|x|)
Succinct Arguments for NP

A (computationally-sound) proof for NP where verifier’s time complexity is independent of the time complexity $T_L$ required to check membership in the language.

\[ \exists w \text{ s.t. } M(x, w) = 1 \text{ in } \leq T_L \text{ steps?} \]

\[ \text{poly}_u(k + T_L) \quad P \quad \vdash \quad V \quad \text{poly}_u(k + |x|) \]

Exist under standard assumptions (CRHs)

[Succinct arguments enable us to delegate ``NP'']
Non-Interactive Succinct Arguments (Of Knowledge) \(\equiv\) SNARKs

\[
\begin{align*}
\text{poly}_u(k + T_L) & \overset{\pi}{\rightarrow} (M, x, T) \overset{\sigma}{\rightarrow} G \\
& \quad \overset{\tau}{\rightarrow} \text{poly}_u(k + |x|)
\end{align*}
\]

- \(\tau\) must be secret = \textit{designated-verifier}
- \(\tau\) can be published = \textit{publicly-verifiable}

\begin{itemize}
\item \(\text{TIME}(G) = \text{poly}_u(k)\) \text{ fully-succinct}
\item \(\text{TIME}(G) = \text{poly}_u(k + T_L)\) \text{ preprocessing}
\end{itemize}
Non-Interactive Succinct Arguments (Of Knowledge) \(\equiv\) SNARKs

\[
\begin{array}{c}
P \\
\sigma \\
G \\
\tau \\
V
\end{array}
\]

\[
\begin{array}{c}
\text{reference string} \\
\text{poly}_u(k + T_L) \\
(M, x, T) \\
\text{verication key} \\
\text{poly}_u(k + |x|)
\end{array}
\]

**KNOWN:**

[Gentry Wichs 11] can’t prove secure via black-box reduction to falsifiable assumptions (for “hard enough NP language”)

[BCCT11] fully-succinct BUT designated-verifier

[DHF11] from extractable collision-resistant hashes

[GLR11] 

[Groth10] publicly-verifiable BUT preprocessing

[Lipmaa11] from knowledge of exponent assumptions

[GGPR12]
Verifier runs fast, gets strong guarantee.

BUT...

What about the prover?

The verifier might be paying the prover for his work!

**ADDITIONAL GOAL:**
minimize prover’s complexity!

Where do we stand?
2 Approaches for Succinct Arguments for NP

PCP-based
4-msg from CRH [Kil92, Mic94, BG02]
2-msg from PIR+ECRH [BCCT11, DHF11, GLR11]

bilinear maps + KEA
[Groth10, Lipmaa11, GGPR12]

\( \tilde{O}(T) \) time BUT need \( \Omega(T) \) space!
NOT EFFICIENT ENOUGH!

For a T-time S-space RAM computation:

<table>
<thead>
<tr>
<th></th>
<th>preprocessing time</th>
<th>prover time</th>
<th>prover space</th>
<th>verifier time</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Kil92] ...</td>
<td>poly(k)</td>
<td>$T \cdot \text{poly}(k)$</td>
<td>$T \cdot \text{poly}(k)$</td>
<td>poly(k)</td>
</tr>
<tr>
<td>[GGPR12]</td>
<td>$T \cdot \text{poly}(k)$</td>
<td>$T \cdot \text{poly}(k)$</td>
<td>$T \cdot \text{poly}(k)$</td>
<td>poly(k)</td>
</tr>
</tbody>
</table>

QUESTIONS

Are there COMPLEXITY-PRESERVING
• succinct arguments from standard assumptions?
• SNARKs from reasonable assumptions?
Yes and Yes
Theorem 1
MIP + FHE $\Rightarrow$ complexity-preserving 4-msg succinct argument

new tool:
succinct function commitment

[not public coin]

Theorem 2
MIP + FHE $\Rightarrow$ complexity-preserving SNARK w/ knowledge

new (non-standard) assumption:
FHE with extractable homomorphism

[designated verifier]

Why do MIPs pop up here?
The Role of MIPs
What is the problem with PCP+CRH?

Let $f(i)$ compute the $i$-th bit of PCP. Committing to PCP requires $|\text{PCP}| = \Omega(T)$ evaluations of $f$.

How to compute all these evaluations?

naively: $\Omega(T^2)$ time

[BCGT12]: $\tilde{O}(T)$ time via FFT methods BUT $\Omega(T)$ space

BUT: verifier asks only $q \overset{\text{def}}{=} \text{polylog}(T)$ evaluations!

Can we save on evaluations when committing?

If so, we may hope for better efficiency...
we treat $f$ as a string because Merkle trees are a succinct STRING commitment

ALTERNATIVE: treat $f$ as a function

More concretely:

**STEP 1:** give a time-and-space-efficient construction in a model where the verifier sends one query to each of $q$ identical functions $\equiv \text{MIP}$

**STEP 2:** implement model in a complexity-preserving way

just as good: not-necessarily-identical

**CHALLENGES**

1. sufficiently-efficient MIP construction?
2. how to implement MIP model (w/ ONE prover)?
Essentially-Optimal MIPs

**Thm:** \( \exists \) a 1-round MIP where to check that a \( T \)-time \( S \)-space RAM \( M \) accepts \((x, w)\) for some \( w \),

(i) the MIP verifier runs in time \( \tilde{O}(|x|) \)

(ii) each MIP prover runs in time \( \tilde{O}(T) \) & space \( \tilde{O}(S) \)

NOTE: PCPs with the above efficiency not known!

Tackled first challenge. ✔️
Succinct Function Commitment (SFC)

given $T$-time $S$-space functions $(f_1, \ldots, f_\ell): A \rightarrow A$,

**Commit**

sender \hspace{1cm} : \hspace{1cm} receiver

**Decommit**

sender \hspace{1cm} : \hspace{1cm} receiver

\[ q_1, \ldots, q_\ell \]

\[ f_1(q_1), \ldots, f_\ell(a_\ell) \]

**time:** \[ \ell \cdot T \cdot \text{poly}(k) \]

\[ \ell \cdot \log |A| \cdot \text{poly}(k) \]

**space:** \[ \ell \cdot S \cdot \text{poly}(k) \]

\[ \ell \cdot \log |A| \cdot \text{poly}(k) \]
Succinct Function Commitment (SFC)

- [Ishai, Kushilevitz, Ostrovsky, CCC ‘07]
  linear hom. enc. ⇒ SFC for linear functions
Succinct Function Commitment (SFC)

**Thm:** FHE $\Rightarrow$ 4-msg SFC for ANY polytime function

**IDEA:**

**STEP 1:** start from the delegation scheme of [CKV10]...
Succinct Function Commitment (SFC)

**Thm:** FHE $\Rightarrow$ 4-msg SFC for ANY polytime function

**IDEA:**

**STEP 1:** ... and “delegate” its preprocessing phase

Diagram:
- **Sender** sends $E(0)$ to the **receiver**.
- The receiver receives $\hat{E}(f(0))$.
- The receiver sends $E(0), \hat{E}(f(0))$ back to the sender.
- The sender receives $E(0), E(x)$.
- The sender sends $\hat{E}(f(0)), \hat{E}(f(x))$ to the receiver.
- The receiver computes $f(x)$ online.
Succinct Function Commitment (SFC)

**Thm:** FHE $\Rightarrow$ 4-msg SFC for ANY polytime function

**IDEA:**

**STEP 1:** ... and “delegate” its preprocessing phase
Succinct Function Commitment (SFC)

**Thm:** FHE $\Rightarrow$ 4-msg SFC for ANY polytime function

**IDEA:**

**STEP 1:** ... and “delegate” its preprocessing phase

**STEP 2:** amplify with parallel repetition [Hai09,CL10]

Tackled second challenge.
The Role of MIPs

**Thm:** $\text{MIP} + \text{SFC} \Rightarrow \text{complexity-preserving 4-msg succinct arguments}$

What about SNARKs?

[Dwork et al., ’04]: $\text{MIP} + \text{PIR unlikely to work}$

**Thm:** $\text{MIP} + \text{FHE}^* \Rightarrow \text{complexity-preserving SNARKs}$

($\text{FHE}^* \approx \text{FHE}$ where homomorphic ops. are extractable)

In fact, can “squash” any public-coin interactive argument (and not just proofs as in [KR09])
Follow-Up

[Bitansky, Canetti, Chiesa, Tromer, EPRINT 12]

Any SNARK $\Rightarrow$ complexity-preserving SNARK & proof-carrying data

Even if has expensive preprocessing!

Want More?

See paper for details & interesting open problems!
THANKS!
http://eprint.iacr.org/2012/461