An Algebraic Framework for Updatable and Universal SNARKs

Carla Ràfols and Arantxa Zapico

Crypto 2021
Pairing-Based (zk)SNARKs

State of the art

Interactive Proof-Systems [GMR89] → ZK proofs for all NP [GMW] →...

Trusted Setup!!!

Multiparty Computation (Zcash Ceremony)

One ceremony per relation!!!
Updatable and Universal SNARKs

• Multiparty Computation Model:

• Updatable Model:
Updatable and Universal (zk)SNARKs

Common Design Principle

Information
Theoretical Object + Cryptographic Compiler = SNARK

Polynomial Holographic:

- **Indexer** computes relation-dependent polynomials
- **Prover’s** messages include polynomials
- **Verifier** has oracle access to both sets of polynomials, can do degree checks, etc.

\[
P_0 = \{ P_{00}(x), \ldots, P_{0n_0}(x) \}
\]

\[
P_1 \xrightarrow{G_1} \]

\[
P_2 \xrightarrow{G_2} \]

\[
\deg P_{23} \leq d ?
\]

\[
P_{12}(y) ?
\]
Motivation

Can we break further the information theoretical object?

1. Extract
2. Compare
3. Combine
4. Improve
Algebraic Intuition

Circuit Satisfiability

• quadratic: \[ a_i b_i = c_i \quad \forall \ i = 1, \ldots, m \]

• linear: \[ a_i = \sum_{j=1}^{m} f_{ij} c_j, \quad b_i = \sum_{j=1}^{m} g_{ij} c_j \quad \forall \ i = 1, \ldots, m \]

\[ \exists \ \tilde{a}, \tilde{b}, \tilde{c} \in \mathbb{F}^m \text{ s.t. for given } F, G \in \mathbb{F}^{m \times m}: \]

1. \[ \tilde{a} \circ \tilde{b} = \tilde{c} \]

2. \[ \begin{pmatrix} 1 & 0 & -F \\ 0 & 1 & -G \end{pmatrix} \begin{pmatrix} \tilde{a} \\ \tilde{b} \\ \tilde{c} \end{pmatrix} = \begin{pmatrix} W_a & W_b & W_c \end{pmatrix} \begin{pmatrix} \tilde{a} \\ \tilde{b} \\ \tilde{c} \end{pmatrix} = W \begin{pmatrix} \tilde{a} \\ \tilde{b} \\ \tilde{c} \end{pmatrix} = \tilde{0} \]
Algebraic Intuition

Circuit Satisfiability

\[
\begin{pmatrix}
1 & 0 & -F \\
0 & 1 & -G
\end{pmatrix}
\begin{pmatrix}
\bar{a} \\
\bar{b} \\
\bar{c}
\end{pmatrix}
= 
\begin{pmatrix}
w_a & w_b & w_c
\end{pmatrix}
\begin{pmatrix}
\bar{a} \\
\bar{b} \\
\bar{c}
\end{pmatrix}
= W
\begin{pmatrix}
\bar{a} \\
\bar{b} \\
\bar{c}
\end{pmatrix}
= \tilde{0}
\]

\[
(\tilde{W}_a, \tilde{W}_b, \tilde{W}_c)_i \cdot (\bar{a}, \bar{b}, \bar{c}) = 0
\]

- Sample a random vector \((\tilde{d}_a, \tilde{d}_b, \tilde{d}_c)\) in the rowspace of \(W\).

\[
(\tilde{d}_a, \tilde{d}_b, \tilde{d}_c) = \sum_{i=1}^{2m} \alpha_i (\tilde{W}_a, \tilde{W}_b, \tilde{W}_c)_i
\]

- Check one inner product \((\tilde{d}_a, \tilde{d}_b, \tilde{d}_c) \cdot (\bar{a}, \bar{b}, \bar{c}) = 0\).
“Compressed” Linear Algebra

Let \( \mathbb{H} = \{h_1, \ldots, h_m\} \subset \mathbb{F}_p^* \).

\[
\lambda_i(X) = \prod_{j \neq i} \frac{(X - h_j)}{(h_i - h_j)}, \quad t(X) = \prod_j (X - h_j)
\]

<table>
<thead>
<tr>
<th>Linear Algebra World</th>
<th>Polynomial World</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tilde{y} = (y_1, \ldots, y_m) )</td>
<td>( Y(X) = \sum_{i=1}^{m} y_i \lambda_i(X) )</td>
</tr>
</tbody>
</table>
From Vectors to Polynomials

• Sample \((\tilde{d}_a, \tilde{d}_b, \tilde{d}_c)\) and compute \(D_a(X), D_b(X), D_c(X)\)

• From \(A(X), B(X), C(X)\) and \(D_a(X), D_b(X), D_c(X)\):

1. \(\vec{a} \cdot \vec{b} = \vec{c}\)

\[
A(X)B(X) - C(X) = t(X)H_1(X)
\]

2. \((\tilde{d}_a, \tilde{d}_b, \tilde{d}_c) \cdot (\vec{a}, \vec{b}, \vec{c}) = 0\)

\[
(D_a(X), D_b(X), D_c(X)) \cdot (A(X), B(X), C(X)) = XR(X) + t(X)H_2(X)
\]
Checkable Subspace Sampling (CSS)

1. $\tilde{\alpha}W$

   $\tilde{\alpha}(Y) = (\alpha_1(Y), \ldots, \alpha_{2m}(Y))$

   $\tilde{\alpha} = \tilde{\alpha}(y)$, for $y$ sent by the verifier

2. $\tilde{D}(X) = (D_a(X), D_b(X), D_c(X))$

   $\tilde{D}(X) = \sum_{j=1}^{3m} (d_a, d_b, d_c)_j \lambda_j(X) = \sum_{j=1}^{3m} \left( \sum_{i=1}^{2m} \alpha_i(W_a, W_b, W_c)_{ij} \right) \lambda_j(X)$

   $= \sum_{j=1}^{3m} \sum_{i=1}^{2m} \alpha_i(y) (W_a, W_b, W_c)_{ij} \lambda_j(X)$

   $= W(X, y)$
Checkable Subspace Sampling

Definition

- **Offline phase:** Indexer outputs polynomials describing matrix $W$.
- **Online phase:**
  - **Sampling:**
  - **Prove Sampling:**
  - **Decision phase:** Verifier accepts only if $D(X)$ encodes vector in the rowspace of $W$ sampled according to $x$. 
Checkable Subspace Sampling

*State-of-the-art*

- **Sonic:**
  - Signature of correct computation: evaluation of $s(X, Y)$.
  - Complexity grows according to decomposition $W$ as a sum of permutation matrices.
  - Amortized CSS: efficient and unrestricted.

- **Marlin, Lunar:**
  - Linear encoding for sparse matrices, relatively large SRS.

- **Our work:**
  - Extended Vandermonde Sampling.
  - Reduce SRS size by decomposing Marlin’s CSS into simpler building blocks.
  - Limited fan-out.
Take away details
Why?

• *Decomposing* constructions of universal and updatable SNARKs into blocks that have a well defined *algebraic* meaning.

• *Captures* several constructions.

• In fact, CSS is the main *bottleneck* in efficiency/generality in these constructions.

• *Isolating* this component allows to focus on improvements.

• *Mix and match:* we can combine different CSS arguments.
Thank you!

https://eprint.iacr.org/2021/590

carla.rafols@upf.edu, arantxa.zapico@upf.edu