Hyperproofs: Aggregating and Maintaining Proofs in Vector Commitments

Shravan Srinivasan  
Alexander Chepurnoy  
Charalampos Papamanthou  
Alin Tomescu  
Yupeng Zhang  

University of Maryland  
Ergo Platform  
Yale University  
Aptos Labs  
Texas A&M University

USENIX Security ’22
Vector Commitment (VC)

- Short commitment to an ordered sequence of values [CF13]
- Open the commitment at specific positions with small proofs
- Should be *position binding*
- Example: Merkle tree

Useful in verifiable storage, stateless blockchains, SNARKs, and more
Problem statement

- **Aggregatable**: combine many proofs into a short proof
- **Maintainable**: update all proofs in sublinear time
- $O(1)$-sized proofs
- aggregatable
- hard-to-maintain: $O(n)$
- RSA: [CF13, BBF19, LM19, CFG$^+$ 20, TXN20]
- Bilinear: [KZG10, CF13, CDHK15, LM19, GRWZ20, TAB$^+$ 20]

Use SNARKs to aggregate the Merkle tree?

- 10-92M constraints
- Large memory overhead

2
Problem statement

- **Aggregatable**: combine many proofs into a short proof
- **Maintainable**: update all proofs in sublinear time
- \( \mathcal{O}(1) \)-sized proofs
- \( \mathcal{O}(\log n) \)-sized proofs
- aggregatable
- not efficiently aggregatable
- hard-to-maintain: \( \mathcal{O}(n) \)
- easy-to-maintain: \( \mathcal{O}(\log n) \)
- RSA: [CF13, BBF19, LM19, CFG\(^+\) 20, TXN20]
- Tree-based: [Mer87, PSTY13, TCZ\(^+\) 20, Tom20, CNR\(^+\) 22]
- Bilinear: [KZG10, CF13, CDHK15, LM19, GRWZ20, TAB\(^+\) 20]
Problem statement

- **Aggregatable**: combine many proofs into a short proof
- **$O(1)$-sized proofs**
- **aggregatable**
- **hard-to-maintain: $O(n)$**
- **RSA**: [CF13, BBF19, LM19, CFG$^+$ 20, TXN20]
- **Bilinear**: [KZG10, CF13, CDHK15, LM19, GRWZ20, TAB$^+$ 20]

- **Maintainable**: update all proofs in sublinear time
- **$O(\log n)$-sized proofs**
- **not efficiently aggregatable**
- **easy-to-maintain: $O(\log n)$**
- **Tree-based**: [Mer87, PSTY13, TCZ$^+$ 20, Tom20, CNR$^+$ 22]

Use SNARKs to aggregate the Merkle tree?

- 10-92M constraints
- Large memory overhead
Problem statement

- **Aggregatable**: combine many proofs into a short proof
  - $O(1)$-sized proofs
  - aggregatable
  - hard-to-maintain: $O(n)$

- **Maintainable**: update all proofs in sublinear time
  - $O(\log n)$-sized proofs
  - not *efficiently* aggregatable
  - easy-to-maintain: $O(\log n)$

Can we build a VC that is both *efficiently* aggregatable and maintainable?
Why do we care about aggregatability and maintainability?

1. Efficiently aggregatable and maintainable VC is an open problem
   1.1 Inefficient SNARK-based Merkle aggregation is the alternative

2. Application to stateless blockchain validation
Our contributions

• First *efficiently* aggregatable and *maintainable* VC
• Aggregation 10× to 41× faster than Merkle based alternatives
  - 2 mins vs 20 mins to 1.4 hrs
• Maintainable: 2.58 ms per update
• Small $O(\log n)$ sized proofs (1.44 KiB)
• Aggregated proof is 52 KiB
• Needs $O(n)$-sized trusted public parameters
Vector as Multi Linear Extension (MLE)

vector $a = [a_0, \ldots, a_7]$

$$f(x_3, x_2, x_1) = a_0 \cdot (1 - x_3) (1 - x_2) (1 - x_1) +$$
Vector as Multi Linear Extension (MLE)

vector $\mathbf{a} = [a_0, \ldots, a_7]$

\[
f(x_3, x_2, x_1) = a_0 \cdot (1 - x_3)(1 - x_2)(1 - x_1) + \\
a_1 \cdot (1 - x_3)(1 - x_2)x_1 + \\
\]

For any index $i$ with binary expansion $(i_3, i_2, i_1)$, we have:
Vector as Multi Linear Extension (MLE)

vector \( \mathbf{a} = [a_0, \ldots, a_7] \)

\[
f(x_3, x_2, x_1) = a_0 \cdot (1 - x_3) (1 - x_2) (1 - x_1) + \\
a_1 \cdot (1 - x_3) (1 - x_2) x_1 + \\
a_2 \cdot (1 - x_3) x_2 (1 - x_1) + \\]
Vector as **Multi Linear Extension (MLE)**

vector $\mathbf{a} = [a_0, \ldots, a_7]$

$$f(x_3, x_2, x_1) = a_0 \cdot (1 - x_3) (1 - x_2) (1 - x_1) +$$

$$a_1 \cdot (1 - x_3) (1 - x_2) x_1 +$$

$$a_2 \cdot (1 - x_3) x_2 (1 - x_1) +$$

$$a_3 \cdot (1 - x_3) x_2 x_1$$
Vector as **Multi Linear Extension (MLE)**

vector $a = [a_0, \ldots, a_7]$

\[
f(x_3, x_2, x_1) = a_0 \cdot (1 - x_3) (1 - x_2) (1 - x_1) + a_4 \cdot x_3 (1 - x_2) (1 - x_1) + a_1 \cdot (1 - x_3) (1 - x_2) x_1 + a_2 \cdot (1 - x_3) x_2 (1 - x_1) + a_3 \cdot (1 - x_3) x_2 x_1 +
\]
Vector as **Multi Linear Extension (MLE)**

\[
\text{vector } \mathbf{a} = [a_0, \ldots, a_7]
\]

\[
f(x_3, x_2, x_1) = a_0 \cdot (1 - x_3)(1 - x_2)(1 - x_1) + a_4 \cdot x_3 (1 - x_2)(1 - x_1) + \\
a_1 \cdot (1 - x_3)(1 - x_2) x_1 + a_5 \cdot x_3 (1 - x_2) x_1 + \\
a_2 \cdot (1 - x_3) x_2 (1 - x_1) + \\
a_3 \cdot (1 - x_3) x_2 x_1 +
\]
Vector as Multi Linear Extension (MLE)

\[
\text{vector } \mathbf{a} = [a_0, \ldots, a_7]
\]

\[
f(x_3, x_2, x_1) = a_0 \cdot (1 - x_3) (1 - x_2) (1 - x_1) + a_4 \cdot x_3 (1 - x_2) (1 - x_1) +
\]
\[
a_1 \cdot (1 - x_3) (1 - x_2) x_1 + a_5 \cdot x_3 (1 - x_2) x_1 +
\]
\[
a_2 \cdot (1 - x_3) x_2 (1 - x_1) + a_6 \cdot x_3 x_2 (1 - x_1) +
\]
\[
a_3 \cdot (1 - x_3) x_2 x_1 +
\]
Vector as **Multi Linear Extension** (MLE)

**vector** \( \mathbf{a} = [a_0, \ldots, a_7] \)

\[
f(x_3, x_2, x_1) = a_0 \cdot (1 - x_3)(1 - x_2)(1 - x_1) + a_4 \cdot x_3(1 - x_2)(1 - x_1) + \\
a_1 \cdot (1 - x_3)(1 - x_2)x_1 + a_5 \cdot x_3(1 - x_2)x_1 + \\
a_2 \cdot (1 - x_3)x_2(1 - x_1) + a_6 \cdot x_3x_2(1 - x_1) + \\
a_3 \cdot (1 - x_3)x_2x_1 + a_7 \cdot x_3x_2x_1
\]

For any index \( i \) with binary expansion \((i_3, i_2, i_1)\), we have:
Vector as Multi Linear Extension (MLE)

vector \( a = [a_0, \ldots, a_7] \)

\[
f(x_3, x_2, x_1) = a_0 \cdot (1 - x_3) (1 - x_2) (1 - x_1) + a_4 \cdot x_3 (1 - x_2) (1 - x_1) + \\
a_1 \cdot (1 - x_3) (1 - x_2) x_1 + a_5 \cdot x_3 (1 - x_2) x_1 + \\
a_2 \cdot (1 - x_3) x_2 (1 - x_1) + a_6 \cdot x_3 x_2 (1 - x_1) + \\
a_3 \cdot (1 - x_3) x_2 x_1 + a_7 \cdot x_3 x_2 x_1
\]

For any index \( i \) with binary expansion \((i_3, i_2, i_1)\), we have:

\[
f(i_3, i_2, i_1) = a_i
\]
Commitment to MLE polynomial [PST13][ZGK\textsuperscript{+}17][ZGK\textsuperscript{+}18]

\[
\text{pp} := g^{s_1}, g^{s_2}, g^{s_3}, g^{s_2s_1}, g^{s_3s_2}, g^{s_3s_1}, g^{s_3s_2s_1}
\]

Commit:

\[
\text{pst}(f) = g^{f(s_3,s_2,s_1)}
\]
Commitment to MLE polynomial [PST13][ZGK+17][ZGK+18]

$$\text{pp} := g^{s_1}, g^{s_2}, g^{s_3}, g^{s_2s_1}, g^{s_3s_2}, g^{s_3s_1}, g^{s_3s_2s_1}$$

Commit: 

$$\text{pst}(f) = g^{f(s_3, s_2, s_1)}$$

Homomorphism: 

$$\text{pst}(f + f') = \text{pst}(f) \cdot \text{pst}(f')$$
Commitment to MLE polynomial [PST13][ZGK+17][ZGK+18]

\[ pp := g^{s_1}, g^{s_2}, g^{s_3}, g^{s_2 s_1}, g^{s_3 s_2}, g^{s_3 s_1}, g^{s_3 s_2 s_1} \]

Commit:

\[ \text{pst}(f) = g^{f^{(s_3, s_2, s_1)}} \]

Homomorphism:

\[ \text{pst}(f + f') = \text{pst}(f) \cdot \text{pst}(f') \]

Commitment to vector \( a \) in Hyperproof is \( \text{pst}(f) \)
Multi Linear Tree (MLT)

\[ f \]
Multi Linear Tree (MLT)
Multi Linear Tree (MLT)
Multi Linear Tree (MLT)

\[
\begin{align*}
f &= f_0 + f_1 \\
f_0 &= f_{00} + f_{01} \\
f_{00} &= a_0 \\
f_{01} &= a_1 \\
f_{000} &= a_0 \\
f_{001} &= a_1
\end{align*}
\]
Multi Linear Tree (MLT)

\[
\begin{align*}
    f & = a_0 \\
    f_0 & = a_1 \\
    f_{00} & = a_2 \\
    f_{000} & = a_3 \\
    f_1 & \\
    f_{01} & \\
    f_{001} & \\
    f_{010} & \\
    f_{011} & \\
    f_{10} & \\
    f_{100} & \\
    f_{101} & \\
    f_{11} & \\
    f_{110} & \\
    f_{111} &
\end{align*}
\]
Multi Linear Tree (MLT)

\[ f_0 = a_0, \quad f_00 = a_0, \quad f_01 = a_1, \quad f_000 = a_0, \quad f_{001} = a_1 \]

\[ f_1 = a_2, \quad f_10 = a_4, \quad f_{100} = a_4, \quad f_{101} = a_5 \]

\[ f_{11} = a_7, \quad f_{110} = a_6, \quad f_{111} = a_7 \]
Multi Linear Tree (MLT)

\[
\begin{align*}
\text{MLT} & = a_0 \quad a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6 \quad a_7 \\
\end{align*}
\]

\[
\begin{align*}
f & = f_1 - f_0 \\
\end{align*}
\]
Multi Linear Tree (MLT)

\[
\begin{align*}
f_0 &= f_00 \quad f_01 \\
f_1 &= f_{00} \quad f_{01} \quad f_{10} \quad f_{11} \\
&= a_0 \quad a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6 \quad a_7 \\
f_1 - f_0 &= f_{00} \\
\end{align*}
\]
Multi Linear Tree (MLT)

\[\begin{align*}
f & = f_0 + f_1 \\
f_0 & = f_{00} + f_{01} \\
f_1 & = f_{10} + f_{11} \\
f_{00} & = a_0 \\
f_{01} & = a_1 \\
f_{010} & = a_2 \\
f_{011} & = a_3 \\
f_{100} & = a_4 \\
f_{101} & = a_5 \\
f_{110} & = a_6 \\
f_{111} & = a_7 \\
f_{01} & = f_{01} - f_{00} \\
f_{11} & = f_{11} - f_{10}
\end{align*}\]
Multi Linear Tree (MLT)

\[
f = a_0 \quad f_0 = a_2 \quad f_00 = a_4 \quad f_010 = a_6 \\
= a_1 \quad f_001 = a_3 \quad f_10 = a_5 \quad f_110 = a_7 \\
\]

\[
f = f_1 - f_0 \quad f_101 = f_1 - f_00 \\
\]

\[
f_11 = f_1 - f_10 \\
\]

\[
f_001 - f_000 \\
\]

\[
f_01 - f_00 \quad f_10 - f_01 \\
\]

\[
f_000 = a_0 \quad f_001 = a_1 \\
\]

\[
f_010 = a_2 \quad f_011 = a_3 \quad f_100 = a_4 \quad f_101 = a_5 \\
\]

\[
f_110 = a_6 \quad f_111 = a_7 \\
\]
Multi Linear Tree (MLT)

\[
\begin{align*}
    f & = a_0 \\
    f_0 & = a_1 \\
    f_00 & = a_2 \\
    f_000 & = a_3 \\
    f_01 & = a_4 \\
    f_010 & = a_5 \\
    f_011 & = a_6 \\
    f_001 & = a_7
\end{align*}
\]
Multi Linear Tree (MLT)
Multi Linear Tree (MLT)

\[
f = a_0 \cdot f_{00} + a_1 \cdot f_{001} + a_2 \cdot f_{010} + a_3 \cdot f_{011} + a_4 \cdot f_{100} + a_5 \cdot f_{101} + a_6 \cdot f_{110} + a_7 \cdot f_{111} = f_0 - f_{00} + f_{01} - f_0 + f_{10} - f_{00} + f_{11} - f_{00}
\]
Multi Linear Tree (MLT)

\[
\begin{align*}
f & = a_0 + a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7 \\
f_{00} & = a_0 \\
f_{01} & = a_1 \\
f_{10} & = a_2 \\
f_{11} & = a_3 \\
f_{000} & = a_4 \\
f_{001} & = a_5 \\
f_{010} & = a_6 \\
f_{011} & = a_7 \\
\end{align*}
\]
Multi Linear Tree (MLT)

\[ f_1 - f_0 \]

\[ f_{01} - f_{00} \]

\[ f_{001} - f_{000} \]

\[ f_{011} - f_{010} \]

\[ f_{101} - f_{100} \]

\[ f_{111} - f_{110} \]

\[ a_0 \quad a_1 \]

\[ a_2 \quad a_3 \]

\[ a_4 \quad a_5 \]

\[ a_6 \quad a_7 \]
MLT: A tree of proofs

\[ \text{pst}(f_1 - f_0) \]

\[ \text{pst}(f_{01} - f_{00}) \]

\[ \text{pst}(f_{001} - f_{000}) \]

\[ a_0 \]

\[ a_1 \]

\[ \text{pst}(f_{011} - f_{010}) \]

\[ a_2 \]

\[ a_3 \]

\[ \text{pst}(f_{101} - f_{100}) \]

\[ a_4 \]

\[ a_5 \]

\[ \text{pst}(f_{111} - f_{110}) \]

\[ a_6 \]

\[ a_7 \]

MLT computation \( O(n \log n) \) time
Proof of $a_5$

\[ \text{pst}(f_{1} - f_{0}) \]

\[ \text{pst}(f_{01} - f_{00}) \]

\[ \text{fst}(f_{001} - f_{000}) \]

\[ \text{pst}(f_{01} - f_{00}) \]

\[ \text{pst}(f_{011} - f_{010}) \]

\[ \text{pst}(f_{101} - f_{100}) \]

\[ \text{pst}(f_{11} - f_{10}) \]

\[ \text{pst}(f_{111} - f_{110}) \]

\[ a_0 \quad a_1 \]

\[ a_2 \quad a_3 \]

\[ a_4 \quad a_5 \]

\[ a_6 \quad a_7 \]

\( O(\log n) \) proof size and verification time
Maintainability

\[
pst(f_1 - f_0) = \begin{align*}
pst(f_{01} - f_{00}) & = \begin{align*}
pst(f_{01} - f_{00}) & = \begin{align*}
pst(f_{001} - f_{000}) & = a_0 \quad a_1 \\
pst(f_{011} - f_{010}) & = a_2 \quad a_3 \\
\end{align*}
\end{align*}
\end{align*}
\]

\[
pst(f_{11} - f_{10}) = \begin{align*}
pst(f_{101} - f_{100}) & = \begin{align*}
pst(f_{101} - f_{100}) & = \begin{align*}
pst(f_{111} - f_{110}) & = a_4 \quad a_5 \\
a_6 \quad a_7 \\
\end{align*}
\end{align*}
\end{align*}
\]
Maintainability

\[ \text{pst}(f_1 - f_0) = \text{pst}(f_{01} - f_{00}) + \text{pst}(f_{11} - f_{10}) \]

\[ \text{pst}(f_{001} - f_{000}) = a_0 \cdot a_1 \]

\[ \text{pst}(f_{011} - f_{010}) = a_2 \cdot a_3 \]

\[ \text{pst}(f_{101} - f_{100}) = a_4 \cdot (a_5 + \delta) \]

\[ \text{pst}(f_{111} - f_{110}) = a_6 \cdot a_7 \]
Maintainability

\[ \text{pst}(f_1 - f_0) \]

\[ \text{pst}(f_{01} - f_{00}) \]

\[ \text{pst}(f_{001} - f_{000}) \]

\[ \text{pst}(f_{011} - f_{010}) \]

\[ \text{pst}(\delta) \cdot \text{pst}(f_{101} - f_{100}) \]

\[ \text{pst}(f_{11} - f_{10}) \]

\[ \text{pst}(f_{111} - f_{110}) \]

\[ a_0 \quad a_1 \]

\[ a_2 \quad a_3 \]

\[ a_4 \quad a_5 + \delta \]

\[ a_6 \quad a_7 \]
Maintainability

\[ \text{pst}(f_1 - f_0) \]

\[ \text{pst}(f_{01} - f_{00}) \]

\[ \text{pst}(f_{001} - f_{000}) \quad \text{pst}(f_{011} - f_{010}) \]

\[ \begin{array}{c}
\text{pst}(\delta) \cdot \text{pst}(f_{101} - f_{100}) \\
\text{pst}(\delta \cdot x_1) \cdot \text{pst}(f_{11} - f_{10})
\end{array} \]

\[ \begin{array}{c}
\text{pst}(f_{001} - f_{000}) \\
\text{pst}(f_{011} - f_{010})
\end{array} \]

\[ \begin{array}{c}
a_0 \quad a_1 \\
a_2 \quad a_3
\end{array} \]

\[ \begin{array}{c}
a_4 \quad a_5 + \delta \\
a_6 \quad a_7
\end{array} \]
Maintainability

\[
pst(\delta \cdot (1 - x_2)x_1) \cdot pst(f_1 - f_0)
\]

\[
pst(f_{01} - f_{00})
\]

\[
pst(f_{001} - f_{000})
\]

\[
pst(f_{011} - f_{010})
\]

\[
pst(\delta) \cdot pst(f_{101} - f_{100})
\]

\[
pst(f_{111} - f_{110})
\]

\[
pst(\delta \cdot x_1) \cdot pst(f_{11} - f_{10})
\]

\[
pst(\delta) \cdot pst(f_{01} - f_{00})
\]

\[
pst(\delta) \cdot pst(f_{001} - f_{000})
\]

\[
pst(\delta) \cdot pst(f_{011} - f_{010})
\]

\[
pst(\delta)
\]

\[
pst(f_{111} - f_{110})
\]

Update all proofs in \(O(\log n)\) time
Aggregation

\[
pst(f_{1} - f_{0}) = \left(\left(a_{4} - a_{0}\right)\left(1 - x_{2}\right)\left(1 - x_{1}\right) + \left(a_{5} - a_{1}\right)\left(1 - x_{2}\right)x_{1} + \left(a_{6} - a_{2}\right)x_{2}\left(1 - x_{1}\right) + \left(a_{7} - a_{3}\right)x_{2}x_{1}\right)
\]

\[
pst(f_{01} - f_{00}) = \left(\left(a_{2} - a_{0}\right)\left(1 - x_{1}\right) + \left(a_{3} - a_{1}\right)x_{1}\right)
\]

\[
pst(f_{001} - f_{000}) = \left(\left(a_{5} - a_{0}\right)\left(1 - x_{2}\right) + \left(a_{7} - a_{1}\right)x_{2}\right)
\]

\[
pst(f_{11} - f_{10}) = \left(\left(a_{6} - a_{4}\right)\left(1 - x_{1}\right) + \left(a_{7} - a_{5}\right)x_{1}\right)
\]

Inner-pairing product between the proof and the selectors
Aggregation

\[
pst(f_1 - f_0) = pst((a_4 - a_0)(1 - x_2)(1 - x_1) + (a_5 - a_1)x_1 + (a_6 - a_2)x_2(1 - x_1) + (a_7 - a_3)x_2x_1)
\]

\[
pst(f_{01} - f_{00}) = pst(a_2 - a_0)(1 - x_1) + (a_3 - a_1)x_1
\]

\[
pst(f_{001} - f_{000}) = pst(a_1 - a_0)
\]

\[
pst(f_{011} - f_{010}) = pst(a_3 - a_2)
\]

\[
pst(f_{11} - f_{10}) = pst((a_6 - a_4)(1 - x_1) + (a_7 - a_5)x_1)
\]

\[
pst(f_{101} - f_{100}) = pst(a_5 - a_4)
\]

\[
pst(f_{111} - f_{110}) = pst(a_7 - a_6)
\]

\[
e(pst(f)/g^{a_5}, g) = e(w_3, ) \cdot e(w_2, ) \cdot e(w_1, )
\]
Aggregation

\[ \text{pst}(f_1 - f_0) \]

\[ \text{pst}(f_{01} - f_{00}) \]

\[ \text{pst}(f_{001} - f_{000}) \quad \text{pst}(f_{011} - f_{010}) \]

\[ a_0 \quad a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6 \quad a_7 \]

\[ e(\text{pst}(f)/g^{a_5}, g) = e(w_3, g^{(s_3-1)}) \cdot e(w_2, g^{(s_2-0)}) \cdot e(w_1, g^{(s_1-1)}) \]
Aggregation

\[ \text{pst}(f_1 - f_0) = \text{pst}(a_4 - a_0)(1 - x_2)(1 - x_1) + \text{pst}(a_5 - a_1)(1 - x_2)x_1 + \text{pst}(a_6 - a_2)x_2(1 - x_1) + \text{pst}(a_7 - a_3)x_2x_1 \]

\[ \text{pst}(f_{01} - f_{00}) = \text{pst}(a_2 - a_0)(1 - x_1) + \text{pst}(a_3 - a_1)x_1 \]

\[ \text{pst}(f_{001} - f_{000}) = \text{pst}(a_1 - a_0) \]

\[ \text{pst}(f_{011} - f_{010}) = \text{pst}(a_3 - a_2) \]

\[ \text{pst}(f_{11} - f_{10}) = \text{pst}(a_6 - a_4)(1 - x_1) + \text{pst}(a_7 - a_5)x_1 \]

\[ \text{pst}(f_{101} - f_{100}) = \text{pst}(a_5 - a_4) \]

\[ \text{pst}(f_{111} - f_{110}) = \text{pst}(a_7 - a_6) \]

\[ e(\text{pst}(f)/g^{a_5}, g) = e(w_3, g^{(s_3-1)}) \cdot e(w_2, g^{(s_2-0)}) \cdot e(w_1, g^{(s_1-1)}) \]

Inner-pairing product between the proof and the selectors
Proof aggregation

Say, $\ell$ is the height of MLT and the batch size is $k$

\[
\begin{bmatrix} w_{i,1}, \cdots, w_{i,\ell} \end{bmatrix}
\]

\[
e(pst(f)/g^{a_i}, g) = \prod_{i=1}^{\ell} e(w_{i,\ell}, g^{s-i_\ell})
\]

- Prove knowledge of all $w_{i,j}$'s using inner-product argument (IPA) based on [BMM$^+21$]
- Random linear combination of the pairing equations

$O(\log (k \cdot \ell))$ aggregated proof size and $O(k \cdot \ell)$ aggregation time
Proof aggregation

Say, $\ell$ is the height of MLT and the batch size is $k$

$$\left\{ [w_{i,1}, \cdots, w_{i,\ell}] \right\}_{i \in [k]} \quad \left\{ e\left(\frac{\text{pst}(f)}{g^{a_i}}, g\right) = \prod_{i=1}^{\ell} e\left(w_{i,\ell}, g^{s-i_{\ell}}\right) \right\}_{i \in [k]}$$

- Prove knowledge of all $w_{i,j}$’s using inner-product argument (IPA) based on [BMM$^{+}$21]
- Random linear combination of the pairing equations

$O(\log (k \cdot \ell))$ aggregated proof size and $O(k \cdot \ell)$ aggregation time
Benchmarks

# transactions = 1024, height of the tree = 30, single threaded

<table>
<thead>
<tr>
<th>Operation</th>
<th>Hyperproofs</th>
<th>Merkle w/ Poseidon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agg</td>
<td>123 s</td>
<td>20 min</td>
</tr>
<tr>
<td>Aggr. Ver</td>
<td>17.4 s</td>
<td>0.11 s</td>
</tr>
<tr>
<td>Aggr. proof size</td>
<td>52 KiB</td>
<td>192 B</td>
</tr>
</tbody>
</table>
Benchmarks

# transactions = 1024, height of the tree = 30, single threaded

<table>
<thead>
<tr>
<th>Operation</th>
<th>Hyperproofs</th>
<th>Merkle w/ Poseidon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agg</td>
<td>123 s</td>
<td>20 min</td>
</tr>
<tr>
<td>Aggr. Ver</td>
<td>17.4 s</td>
<td>0.11 s</td>
</tr>
<tr>
<td>Aggr. proof size</td>
<td>52 KiB</td>
<td>192 B</td>
</tr>
<tr>
<td>Block proposal (P)</td>
<td>2.23 min</td>
<td>81 min</td>
</tr>
<tr>
<td>Block validation (V)</td>
<td>17.5 s</td>
<td>0.18 s</td>
</tr>
<tr>
<td>Proof maintenance (M)</td>
<td>5.14 s</td>
<td>4.7 s</td>
</tr>
<tr>
<td>Total (P + (20 · V) + M)</td>
<td>8 min</td>
<td>81 min</td>
</tr>
</tbody>
</table>

Faster end-to-end performance than Merkle tree
Conclusion

- VC with efficient aggregation + maintainability + unstealability
- Unstealability incentivizes proof computation
- End-to-end performance is $10 \times$ to $41 \times$ faster than Merkle
- Algebraic alternative for Merkle tree
References I


References II

Composable and Modular Anonymous Credentials: Definitions and Practical Constructions.

[CF13] Dario Catalano and Dario Fiore.
Vector Commitments and Their Applications.
References III


Vector Commitment Techniques and Applications to Verifiable Decentralized Storage.
https://eprint.iacr.org/2020/149.

Linear-map Vector Commitments and their Practical Applications.
References IV


References V

Subvector Commitments with Application to Succinct Arguments.

[Mer87] Ralph C. Merkle.
A Digital Signature Based on a Conventional Encryption Function.
References VI


References VII


References VIII

How to Keep a Secret and Share a Public Key (Using Polynomial Commitments).

Authenticated Dictionaries with Cross-Incremental Proof (Dis)aggregation.
References IX
