Efficient All-but-one Random Vector Commitment from Block Cipher *and More*

ePrint 2024/097 & Recent progress by Prof. Guo

March 5, 2024· Presented by Hongrui Cui

Motivations

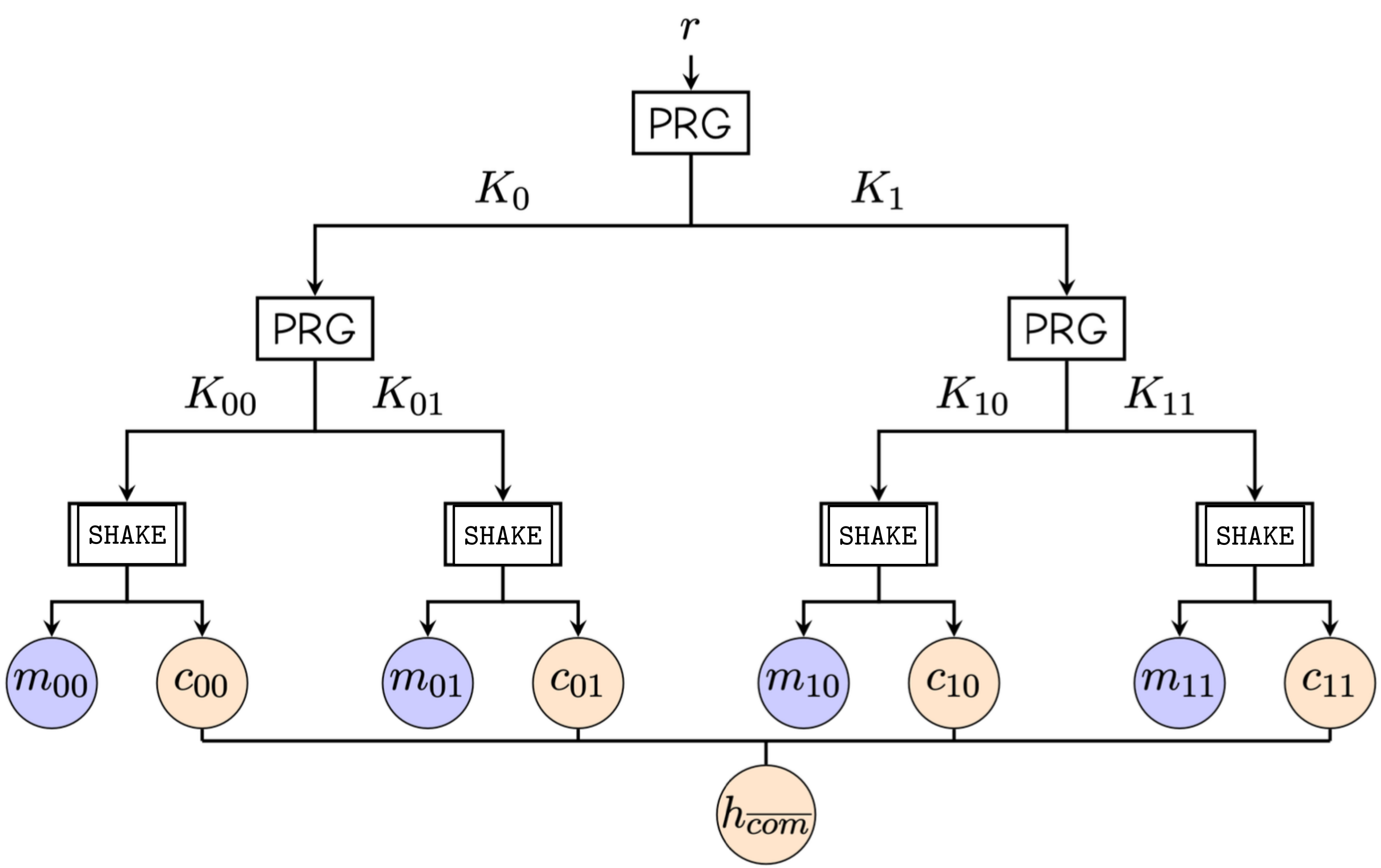
Applications that require all-but-one random vector commitment

- Post-quantum signatures:
  - VOLEitH ($\texttt{FAEST}$, $\texttt{ReSolveD}$),...
  - MPCitH ($\texttt{SDitH}$, $\texttt{Banquet}$),...

Applications that require AES-based CCR Hash for $\lambda \in \{192, 256\}$

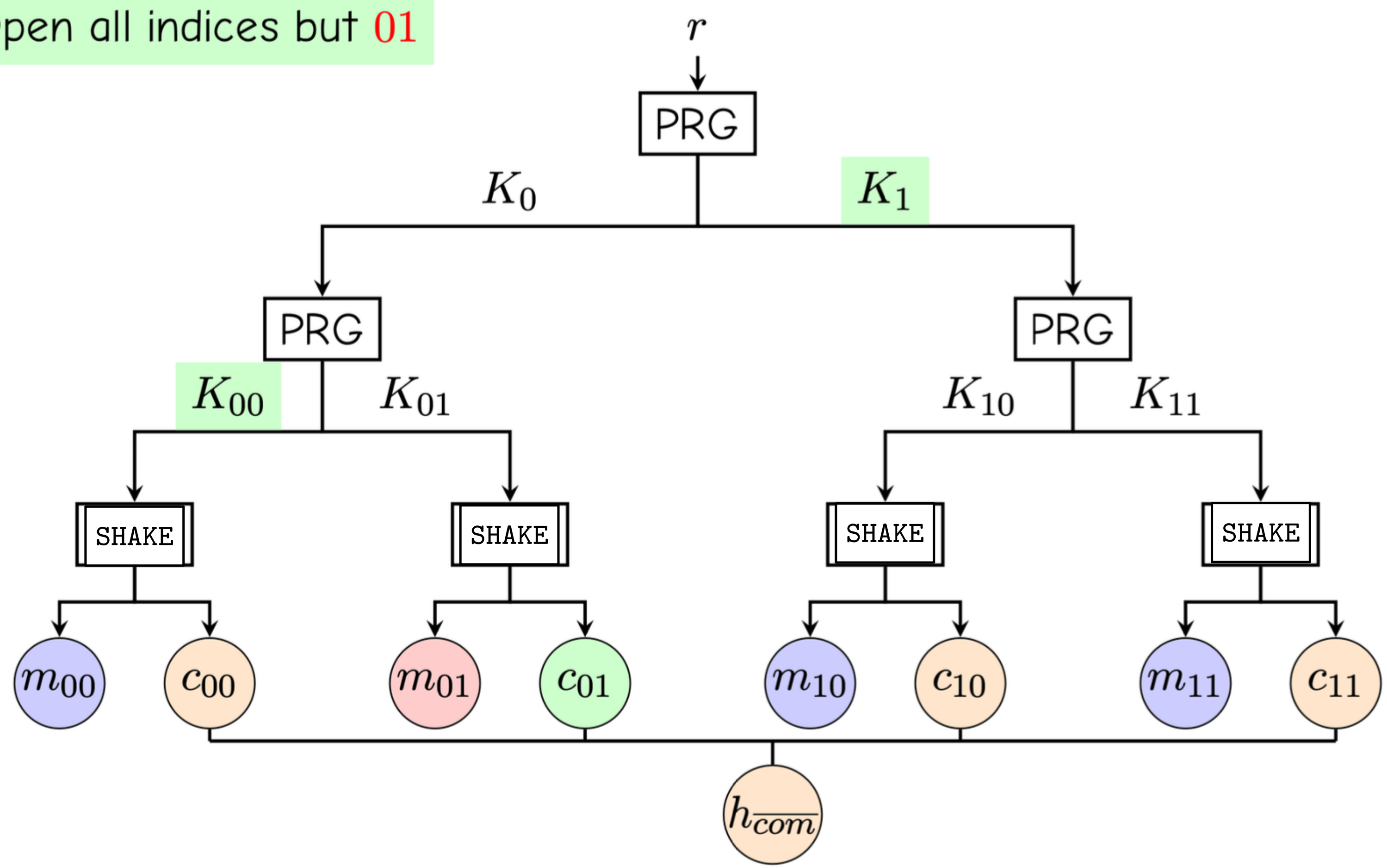
- 2PC: Half-tree (correlated GGM, psudorandom correlated GGM),...

Current Construction from PRG and RO



Current Construction from PRG and RO

Open all indices but 01



Completeness

A vector commitment scheme VC is (perfectly) correct if for all $\lambda \in \mathbb{N}$ and $N = \mathsf{poly}(\lambda)$ the following condition holds.

$$\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, N), (\mathsf{com}, \mathsf{decom}, (m_0, ..., m_{N-1})) \leftarrow \mathsf{Commit}(\mathsf{crs}), \forall \alpha \in [0, N)$$
$$\mathsf{decom}_\alpha \leftarrow \mathsf{Open}(\mathsf{crs}, \mathsf{decom}, \alpha) : \mathsf{Verify}(\mathsf{crs}, \mathsf{com}, \alpha, \mathsf{decom}_\alpha) = (m_i)_{i \in [0,N), i \neq \alpha}.$$

Hiding

The adaptive hiding experiment for VC with $N = 2^k = \mathrm{poly}(\lambda)$ and stateful $\mathcal{A}$ is defined as follows.

1. $\mathrm{crs} \leftarrow \mathrm{Setup}(1^\lambda, N), b^* \leftarrow \{0, 1\}$
2. $(\mathrm{com}, \mathrm{decom}, (m_0^*, \ldots, m_{N-1}^*)) \leftarrow \mathrm{Commit}(\mathrm{crs})$
3. $\alpha \leftarrow \mathcal{A}(1^\lambda, \mathrm{crs}, \mathrm{com})$
4. $\mathrm{decom}_\alpha \leftarrow \mathrm{Open}(\mathrm{crs}, \mathrm{decom}, \alpha)$
5. Let $m_i = m_i^*$ for $i \in [0, N), i \neq \alpha$
6. For $i = \alpha$, set $m_i = \begin{cases} m_i^* & \text{if } b^* = 0 \\ \text{random} & \text{if } b^* = 1 \end{cases}$
7. $b \leftarrow \mathcal{A}((m_i)_{i \in [0,N)}, \mathrm{decom}_\alpha)$
8. Output 1 (success) if $b = b^*$, else 0 (failure).

- In the selective hiding experiment, $\mathcal{A}$ must choose α prior to receiving com.
- $\mathrm{AdvAdpHide}^{\mathrm{VC}}(\mathcal{A}) = \left| \Pr[\mathcal{A} \text{ wins}] - \frac{1}{2} \right|$

Binding

Let $\mathsf{Ext}(\mathsf{crs}, \mathsf{com}, Q_E) \rightarrow (m_i)_{i \in [0,N)}$ be a extraction function.

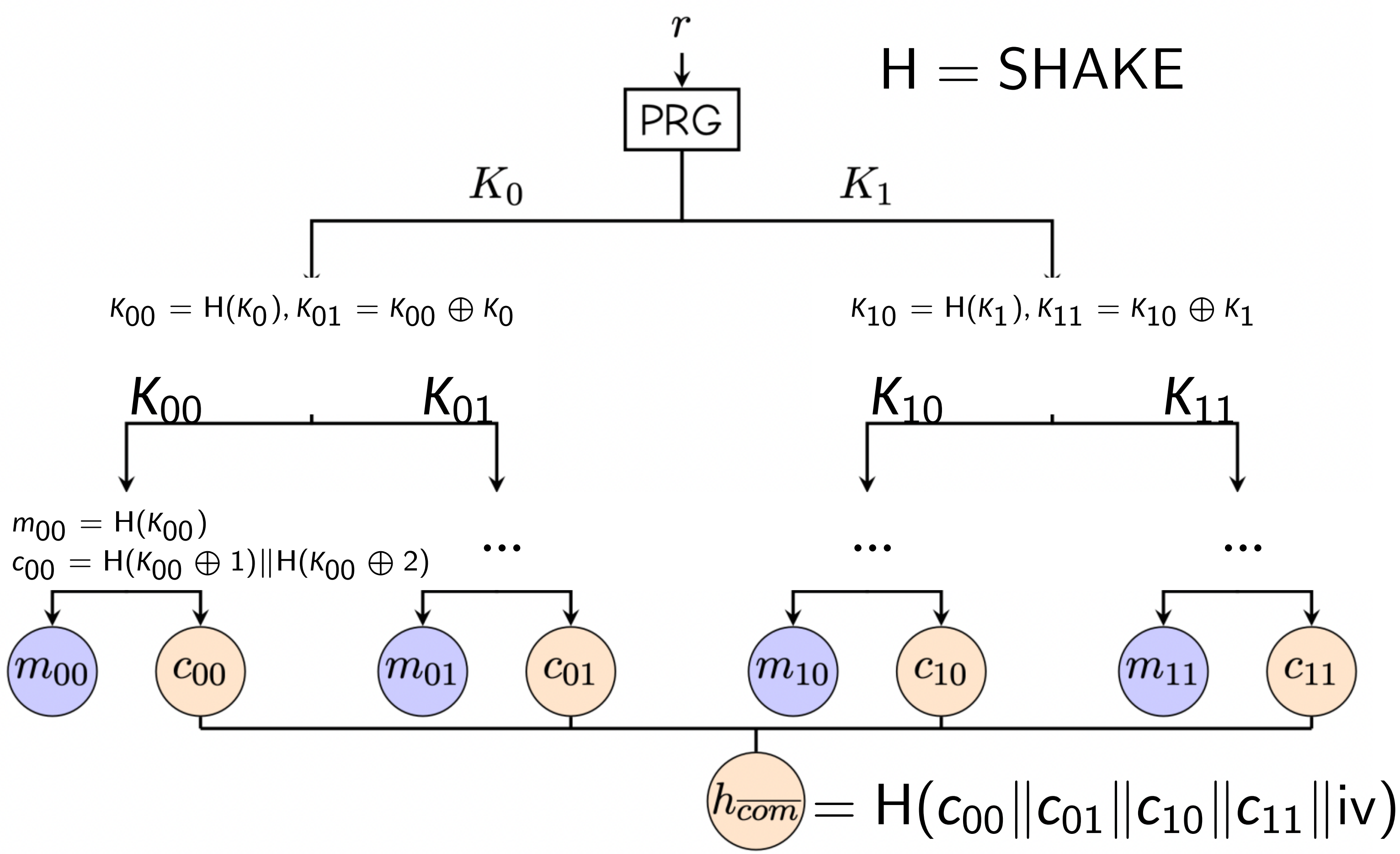
For $N = 2^k = \mathsf{poly}(\lambda)$, define the following extractable binding game for a stateful adversary $\mathcal{A}$.

1. $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, N)$
2. $\mathsf{com} \leftarrow \mathcal{A}(\mathsf{crs})$
3. $(m_i^*)_{i \in [0,N)} = \mathsf{Ext}(\mathsf{crs}, \mathsf{com}, Q_E)$
4. $(\mathsf{decom}_\alpha, \alpha) \leftarrow \mathcal{A}()$
5. Output 1 (success) if $\mathsf{Verify}(\mathsf{com}, \alpha, \mathsf{decom}_\alpha) = (m_i)_{i \in [0,N), i \neq \alpha}$ but $m_i \neq m_i^*$ for some $i \in [0, N), i \neq \alpha$. Otherwise, output 0 (failure).

- $\mathsf{AdvEB}^{\mathsf{VC}}(\mathcal{A}) = \Pr[\mathcal{A} \text{ wins}]$

A CCR-based Construction (2024/097)

- Let $\pi : \{0,1\}^\lambda \to \{0,1\}^\lambda$ be random permutation
- According to [GKYW20], $\mathsf{H}(x) := \pi(\sigma(x)) \oplus \sigma(x)$ is a $(t, q, \rho, \frac{2tq}{2^\rho} + \frac{q^2}{2^{\lambda+1}})$-CCR



Proof of Adaptive Hiding

- Step 1: Generate $h_{\mathsf{com}} \leftarrow \{0,1\}^{2\lambda}$

Security loss: $\frac{|Q_{\mathsf{RO}}|}{2^{2\lambda}}$

- Step 2: Use $\mathcal{O}^{\mathsf{ccr}}$ to simulate opening

Security loss: $\varepsilon_{\mathsf{ccr}}$

- $\mathsf{AdvAdpHide}^{\mathsf{VC}}(\mathcal{A}) \leq \frac{|Q_{\mathsf{RO}}|}{2^{2\lambda}} + \varepsilon_{\mathsf{ccr}}$

Proof of Binding

- Step 1: Extract $(\mathsf{com}_1, \ldots, \mathsf{com}_N)$ s.t. $\mathsf{SHAKE}(\mathsf{com}_1 \| \ldots \| \mathsf{com}_N \| \mathsf{iv}) = h_{\mathsf{com}}$

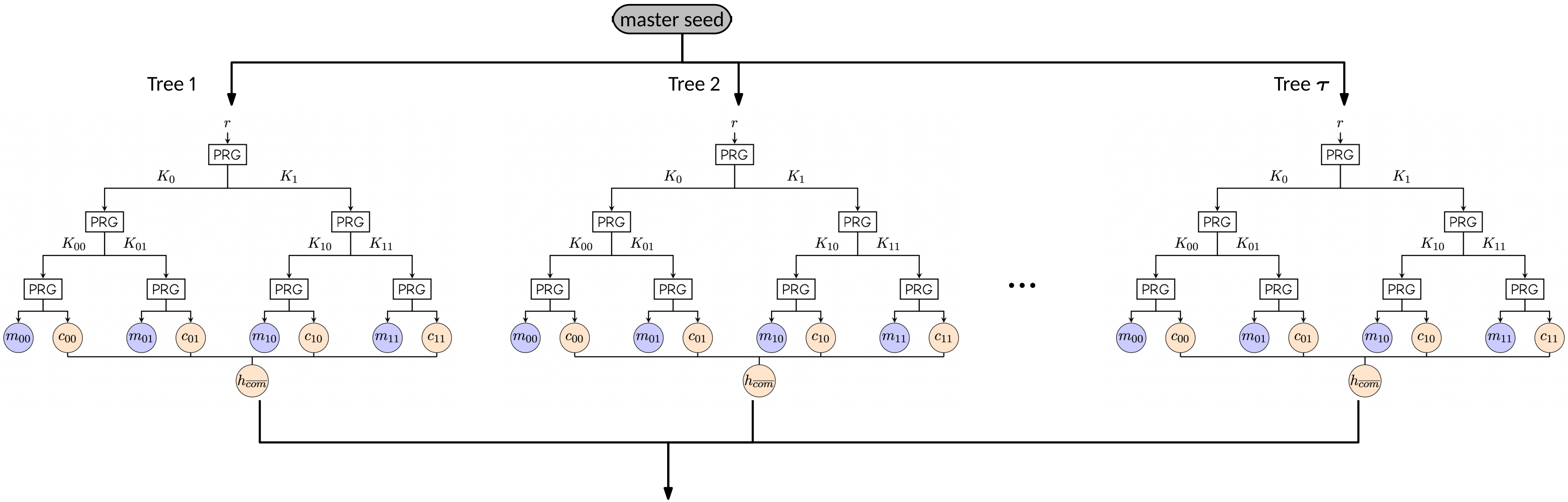
$$\Pr[\text{Ext fails}] = \Pr[\text{SHAKE collision}] \leq \frac{|Q_{\mathsf{RO}}|}{2^{2\lambda}}$$

- Step 2: For $i \in [N]$, extract r_i s.t. $\mathsf{H}(r_i \oplus 1) \| \mathsf{H}(r_i \oplus 2) = \mathsf{com}_i$

$$\Pr[\text{Ext fails}] \leq \frac{|Q_\pi| \cdot (|Q_\pi| - 1)}{2} \cdot \Pr\begin{bmatrix} \pi(\sigma(r_i \oplus 1)) \oplus \sigma(r_i \oplus 1) = \pi(\sigma(r_j \oplus 1)) \oplus \sigma(r_j \oplus 1) \\ \wedge \\ \pi(\sigma(r_i \oplus 2)) \oplus \sigma(r_i \oplus 2) = \pi(\sigma(r_j \oplus 2)) \oplus \sigma(r_j \oplus 2) \end{bmatrix}$$

- $\mathsf{AdvEB}^{\mathsf{VC}}(\mathcal{A}) \leq \frac{|Q_{\mathsf{RO}}|}{2^{2\lambda}} + \frac{|Q_\pi| \cdot (|Q_\pi| - 1)}{2} \cdot \frac{1}{2^{2\lambda}}$

Multi-Tree VC



$h_{\text{com}} = \text{SHAKE}(h^1_{\text{com}} \| \ldots \| h_{\text{com}^{\tau}})$

Discussion

- Motivation 1 (VC) is a strict super set of Motivation 2 (Half-tree cGGM)
- Problem with 2024/097: π only has 128-bit block size with AES-NI

Algorithm $\mathsf{H}_{\mathsf{ccr}}(1^\lambda, r)$

- If $\lambda = 128$: return $\mathsf{AES\text{-}128}\big(\mathsf{C}_0, \sigma(r)\big) \oplus \sigma(r)$
- If $\lambda = 192$:
  1. $r_L \leftarrow \mathsf{left}_{128}(r), r_R \leftarrow \mathsf{right}_{64}(r)$
  2. return $\mathsf{AES\text{-}192}\big(r_R \| \mathsf{C}_0, \sigma(r_L)\big) \oplus \sigma(r_L)$ $\Big\| \mathsf{left}_{64}\Big(\mathsf{AES\text{-}192}\big(r_R \| \mathsf{C}_1, \sigma(r_L)\big) \oplus \sigma(r_L)\Big)$
- If $\lambda = 256$:
  1. $r_L \leftarrow \mathsf{left}_{128}(r), r_R \leftarrow \mathsf{right}_{128}(r)$
  2. return $\mathsf{AES\text{-}256}\big(r_R \| \mathsf{C}_0, \sigma(r_L)\big) \oplus \sigma(r_L) \big\| \mathsf{AES\text{-}256}\big(r_R \| \mathsf{C}_1, \sigma(r_L)\big) \oplus \sigma(r_L)$

Leaf Derivation

Algorithm $\mathsf{H}_{\text{leaf}}(1^\lambda, r, \ell)$

- If $\lambda = 128$:
  1. For $i = 0$ to $\ell - 1$ do
     - $y_i \leftarrow \text{AES-128}(C_2, r + i) \oplus (r + i)$
  2. $\text{com}_L \leftarrow \text{AES-128}(C_3, r) \oplus r$, $\text{com}_R \leftarrow \text{AES-128}(C_3, r \oplus 1) \oplus r$
  3. $\text{com} \leftarrow \text{com}_L \| \text{com}_R$
  4. return $(y_0 \| \ldots \| y_{\ell-1}, \text{com})$
- If $\lambda = 192$:
  1. $r_L \leftarrow \text{left}_{128}(r)$, $r_R \leftarrow \text{right}_{64}(r)$
  2. For $i = 0$ to $\ell - 1$ do
     - $y_i \leftarrow \text{AES-192}(r_R \| C_2, r_L + i) \oplus (r_L + i)$
  3. $\text{com}_1 \leftarrow \text{AES-192}(r_R \| C_3, r_L) \oplus r_L$, $\text{com}_2 \leftarrow \text{AES-192}(r_R \| C_3, r_L \oplus 1) \oplus r_L \oplus 1$, $\text{com}_3 \leftarrow \text{AES-192}(r_R \| C_3, r_L \oplus 2) \oplus r_L \oplus 2$
  4. $\text{com} \leftarrow \text{com}_1 \| \text{com}_2 \| \text{com}_3$
  5. return $(y_0 \| \ldots \| y_{\ell-1}, \text{com})$
- If $\lambda = 256$:
  1. $r_L \leftarrow \text{left}_{128}(r)$, $r_R \leftarrow \text{right}_{128}(r)$
  2. For $i = 0$ to $\ell - 1$ do
     - $y_i \leftarrow \text{AES-256}(r_R \| C_2, r_L + i) \oplus (r_L + i)$
  3. $\text{com}_1 \leftarrow \text{AES-256}(r_R \| C_3, r_L) \oplus r_L$, $\text{com}_2 \leftarrow \text{AES-256}(r_R \| C_3, r_L \oplus 1) \oplus r_L \oplus 1$, $\text{com}_3 \leftarrow \text{AES-256}(r_R \| C_3, r_L \oplus 2) \oplus r_L \oplus 2$, $\text{com}_4 \leftarrow \text{AES-256}(r_R \| C_3, r_L \oplus 3) \oplus r_L \oplus 3$
  4. $\text{com} \leftarrow \text{com}_1 \| \text{com}_2 \| \text{com}_3 \| \text{com}_4$
  5. return $(y_0 \| \ldots \| y_{\ell-1}, \text{com})$

Theoretical Model

- λ: security parameter, $\lambda \in \{128, 192, 256\}$
- n: block-size of E, $n = 128$
- κ: key-size of E, $\kappa \in \{128, 192, 256\}$
- Note that $|r| = \lambda$ (internal node)

Algorithm $\mathsf{H}^{\mathsf{E}}_{\mathsf{ccr}}(1^\lambda, 1^n, 1^\kappa, r)$

- If $n \leq \lambda \leq 2n \ll n + \kappa$:
  1. $r_L \leftarrow \mathsf{left}_n(r), r_R \leftarrow \mathsf{right}_{\lambda-n}(r)$
  2. $z_L \leftarrow \mathsf{E}\big(r_R \| [0]_{\kappa+n-\lambda}, \sigma(r_L)\big) \oplus \sigma(r_L)$
  3. $z_R \leftarrow \mathsf{left}_{\lambda-n}\big(\mathsf{E}\big(r_R \| [1]_{\kappa+n-\lambda}, \sigma(r_L)\big) \oplus \sigma(r_L)\big)$ // omit if $n = \lambda$
  4. return $z_L \| z_R$

Theoretical Model

Algorithm $\mathsf{H}_{\mathsf{leaf}}(1^\lambda, 1^n, 1^\kappa, r, \ell)$
// λ: security parameter; n: block-size of E; κ: key-size of E.
// Note that $|r| = \lambda$.

- If $n \leq \lambda \leq 2n \ll n + \kappa$:
  1. $r_L \leftarrow \mathsf{left}_n(r), r_R \leftarrow \mathsf{right}_{\lambda-n}(r)$
  2. For $i = 0$ to $\ell - 1$ do
     $y_i \leftarrow \mathsf{E}(r_R \| [2]_{\kappa+n-\lambda}, r_L + i) \oplus (r_L + i)$
  3. $w \leftarrow \lceil 2\lambda/n \rceil$
  4. For $i = 0$ to $w - 1$ do
     $\mathsf{com}_i \leftarrow \mathsf{E}(r_R \| [3]_{\kappa+n-\lambda}, r_L \oplus [i]_n) \oplus (r_L \oplus [i]_n)$
  5. $y \leftarrow y_0 \| \ldots \| y_{\ell-1}$, $\mathsf{com} \leftarrow \mathsf{com}_0 \| \ldots \| \mathsf{com}_{w-1}$
  6. return (y, com)