SPED: Syndrome-Decoding Signature based on VOLE-in-the-Head

Inspired by

Publicly Verifiable Zero-Knowledge and Post-Quantum Signatures From VOLE-in-the-Head

Carsten Baum, TU Denmark, Aarhus University
Lennart Braun, Aarhus University
Cyprien Delpech de Saint Guilhem, KU Leuven
Michael Klooß, Aalto University
Emmanuela Orsini, Bocconi University
Lawrence Roy, Aarhus University
Peter Scholl, Aarhus University

Crypto'23 (ia.cr/2023/996)

Motivations

- VOLE-in-the-Head $\geq$ MPC-in-the-Head
- FAEST Signature

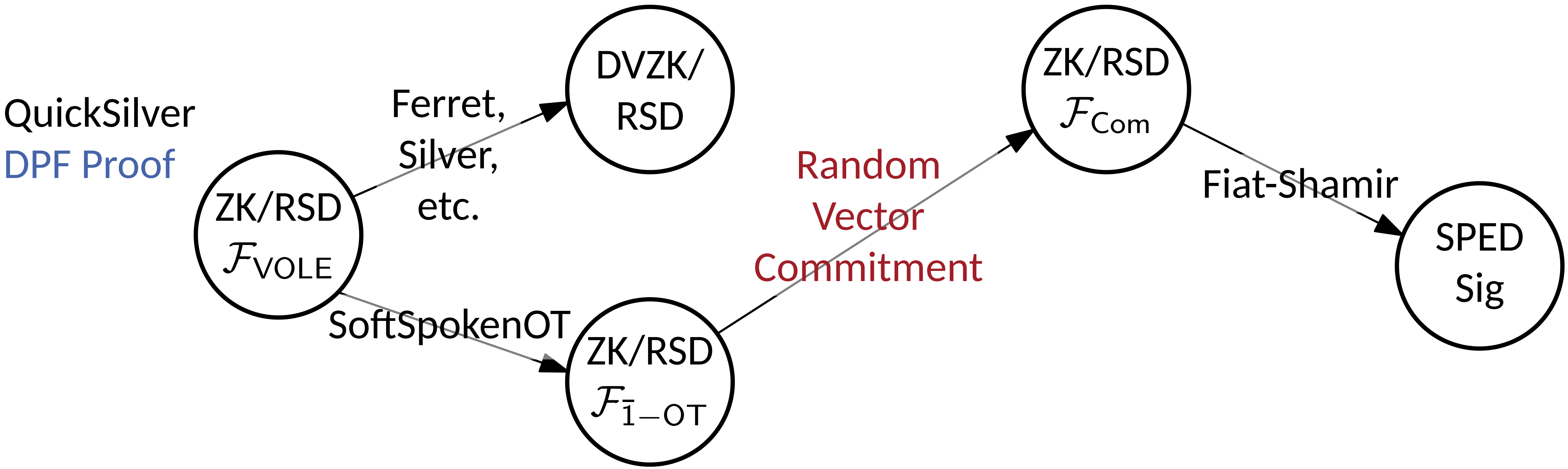
- P.S. Landscape of Efficient Zero Knowledge

	zk-SNARK, GKR, etc.	GCZK	DVZK	DVZK*
$\mathcal{P}$ Comp.	$\Omega(\lvert C\rvert)$	$O(\lvert C\rvert)$	$O(\lvert C\rvert)$	$O(\lvert C\rvert \log \lvert C\rvert)$
$\mathcal{P}$ Mem.	$\Omega(\lvert C\rvert)$	$O(1)$	$O(1)$	$O(\lvert C\rvert^{1/4})$
Proof Size	$O(\log(\lvert C\rvert))$	$O(\kappa \cdot \lvert C\rvert)$	$O(\lvert C\rvert)$ or $O(\lvert w\rvert + d)$	$O(\lvert C\rvert^{3/4})$
$\mathcal{V}$ Type	Universal	Designated	Designated	Designated
Advantage	Low-Bandwidth Medium Circuit	High-Bandwidth Large Circuit	High-Bandwidth Large Circuit Polynomials	High-Bandwidth Large Circuit

Main techniques (of DVZK):

- Random (subfield) VOLE
- Low-Degree Test

Contributions



- Contribution 1: Combine DPF proof with VOLE-in-the-Head
- Contribution 2: Use half-tree to optimize computational performance

Performance of the ZK Compilers

Table 1. Comparison of linear-size zero-knowledge proof systems

Protocol	Field*	Model	Comm./gate†	Assumption
VOLE-ZK [YSWW21]‡	$\mathbb{F}_2$	deg-d constraints	1	LPN
VOLE-ZK [DIO21, YSWW21]‡	$\mathbb{F}_p$	deg-d constraints	1	LPN
Limbo [dOT21]	$\mathbb{F}_2$	Circuits (free XOR)	42 (11)	Hash
Limbo [dOT21]	$\mathbb{F}_p$	Circuits (free add)	40 (11)	Hash
VOLE-in-the-head (§E.3)	$\mathbb{F}_2$	deg-d constraints	16 (5)	Hash
VOLE-in-the-head (§5.1)	$\mathbb{F}_p$	deg-d constraints	3 (2)	Hash

* $p \approx 2^{64}$

† Soundness error at most 2^{-128} (2^{-40}). Cost is average number of field elements sent per AND/mult. gate, for a circuit with 2^{20} such gates.

‡ Designated-verifier only

Preliminary: VOLE as IT-MAC (Linear Commitment)

Receiver/$\mathcal{V}$

$\Delta \in \mathbb{F}_{p^r}$
(global key)
$\mathsf{K}[\mathbf{u}] \in \mathbb{F}_{p^r}^n$
(MAC Key)

$\mathcal{F}_{\mathsf{rVOLE}}$

Sender/$\mathcal{P}$

$\mathbf{u} \in \mathbb{F}_{p^r}^n$ (message)
$\mathsf{M}[\mathbf{u}] \in \mathbb{F}_{p^r}^n$ (MAC Tag)

Sender commits to $\mathbf{x}$ by sending $\mathbf{d} := \mathbf{x} - \mathbf{u}$

IT-MAC $[\mathbf{x}] := (\mathbf{x}, \mathsf{M}[\mathbf{x}], \mathsf{K}[\mathbf{x}])$ subject to $\mathsf{M}[\mathbf{x}] = \mathsf{K}[\mathbf{x}] + \mathbf{x} \cdot \Delta$

- Linear Homomorphism: $[x] + [y] \mapsto [x + y]$
- Open($[x]$): $\mathcal{P} \to \mathcal{V} : (x, \mathsf{M}[x])$, $\mathcal{V}$ checks $\mathsf{M}[x] = \mathsf{K}[x] + x \cdot \Delta$
- Batched Open:

Receiver/$\mathcal{V}$ Sender/$\mathcal{P}$

$\{\chi_i\}$

$\mathbf{x}$, Open($[\sum_i \chi_i \cdot \mathbf{x}_i]$)

- $\mathcal{P}$ opens a different value $\to$ $\mathcal{P}$ guesses Δ
- Soundness error $= \frac{1}{p^r}$

Starting Point: DVZK for Quadratic Relations

Prove $a_1 \times a_2 = a_3$ $\quad \underbrace{\mathsf{K}[\mathbf{a}] = \Delta}_{\mathcal{V}} \cdot \underbrace{\mathbf{a} + \mathsf{M}[\mathbf{a}]}_{\mathcal{P}}$

Starting Point: DVZK for Quadratic Relations

Prove $a_1 \times a_2 = a_3$ $\quad \underbrace{\mathsf{K}[\mathbf{a}] = \Delta}_{\mathcal{V}} \cdot \underbrace{\mathbf{a} + \mathsf{M}[\mathbf{a}]}_{\mathcal{P}}$

$$\underbrace{\mathsf{K}[a_1] \cdot \mathsf{K}[a_2] + \Delta \cdot \mathsf{K}[a_3]}_{B} = (\mathsf{M}[a_1] - a_1 \cdot \Delta) \cdot (\mathsf{M}[a_2] - a_2 \cdot \Delta) + \Delta \cdot (\mathsf{M}[a_3] - a_3 \cdot \Delta)$$

$$= (a_1 \cdot a_2 - a_3)\Delta^2 + (\underbrace{-a_1\mathsf{M}[a_2] - a_2\mathsf{M}[a_1] + \mathsf{M}[a_3]}_{A_1})\Delta + \underbrace{\mathsf{M}[a_1]\mathsf{M}[a_2]}_{A_0}$$

Starting Point: DVZK for Quadratic Relations

Prove $a_1 \times a_2 = a_3$ $\quad \underbrace{\mathsf{K}[\mathbf{a}] = \Delta}_{\mathcal{V}} \cdot \underbrace{\mathbf{a} + \mathsf{M}[\mathbf{a}]}_{\mathcal{P}}$

$$\underbrace{\mathsf{K}[a_1] \cdot \mathsf{K}[a_2] + \Delta \cdot \mathsf{K}[a_3]}_{B} = (\mathsf{M}[a_1] - a_1 \cdot \Delta) \cdot (\mathsf{M}[a_2] - a_2 \cdot \Delta) + \Delta \cdot (\mathsf{M}[a_3] - a_3 \cdot \Delta)$$

$$= (a_1 \cdot a_2 - a_3)\Delta^2 + (\underbrace{-a_1\mathsf{M}[a_2] - a_2\mathsf{M}[a_1] + \mathsf{M}[a_3]}_{A_1})\Delta + \underbrace{\mathsf{M}[a_1]\mathsf{M}[a_2]}_{A_0}$$

- $\mathcal{P}$ sends A_1, A_0 to $\mathcal{V}$
- $\mathcal{V}$ checks that $A_1 \cdot \Delta + A_0 = B$

Starting Point: DVZK for Quadratic Relations

Prove $a_1 \times a_2 = a_3$ $\quad \underbrace{\mathsf{K}[\mathbf{a}] = \Delta}_{\mathcal{V}} \cdot \underbrace{\mathbf{a} + \mathsf{M}[\mathbf{a}]}_{\mathcal{P}}$

$$\underbrace{\mathsf{K}[a_1] \cdot \mathsf{K}[a_2] + \Delta \cdot \mathsf{K}[a_3]}_{B} = (\mathsf{M}[a_1] - a_1 \cdot \Delta) \cdot (\mathsf{M}[a_2] - a_2 \cdot \Delta) + \Delta \cdot (\mathsf{M}[a_3] - a_3 \cdot \Delta)$$

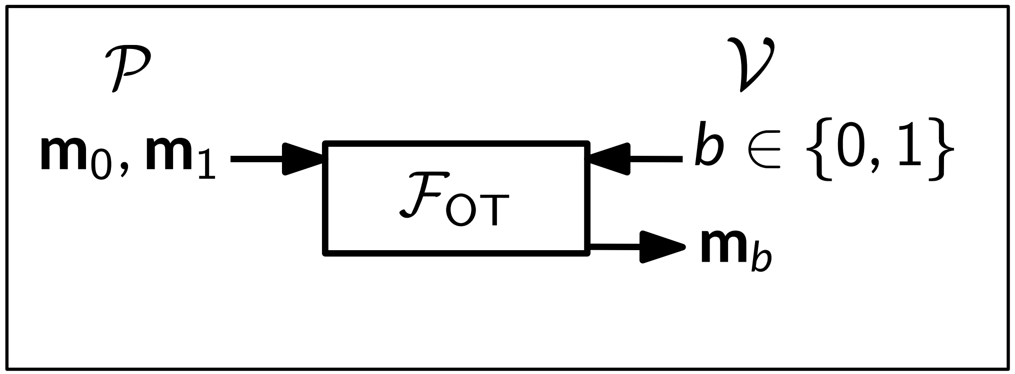
$$= (a_1 \cdot a_2 - a_3)\Delta^2 + (\underbrace{-a_1\mathsf{M}[a_2] - a_2\mathsf{M}[a_1] + \mathsf{M}[a_3]}_{A_1})\Delta + \underbrace{\mathsf{M}[a_1]\mathsf{M}[a_2]}_{A_0}$$

- $\mathcal{P}$ sends A_1, A_0 to $\mathcal{V}$
- $\mathcal{V}$ checks that $A_1 \cdot \Delta + A_0 = B$

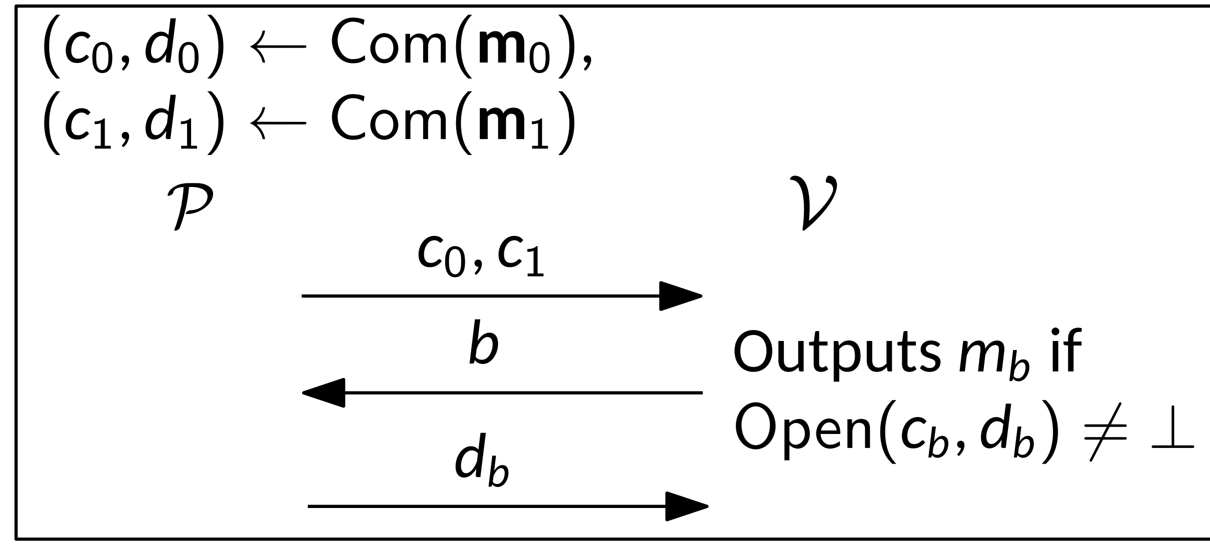
- We can prove multiple quadratic relations using random linear combination
- Sample $\boldsymbol{\chi} = (\chi^{(1)}, ..., \chi^{(\ell)})$
- Compute $A_1 = \sum_i \chi^{(i)} A_1^{(i)}, A_0 = \sum_i \chi^{(i)} A_0^{(i)}, B = \sum_i \chi^{(i)} B^{(i)}$
- Soundness loss $= \frac{1}{|\mathbb{F}|}$

Starting Point: Public Coin $\mathcal{F}_{\mathrm{OT}}$ by Com&Open

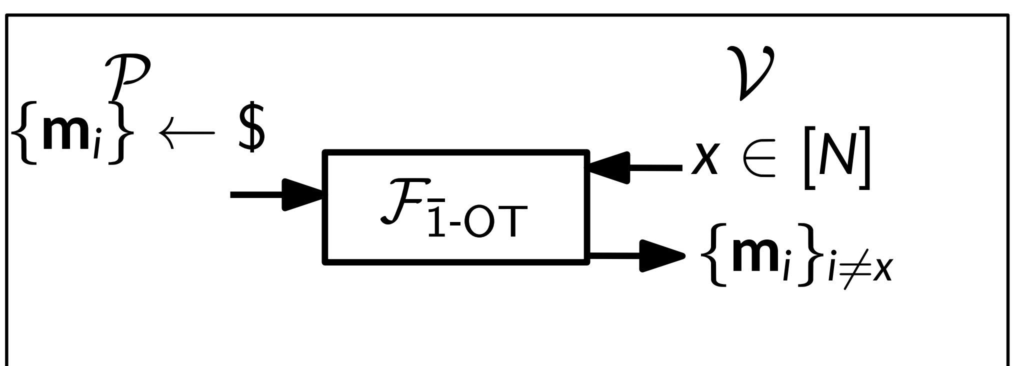
- For public-coin $\mathcal{V}$, we have public-coin $\binom{2}{1}$-OT



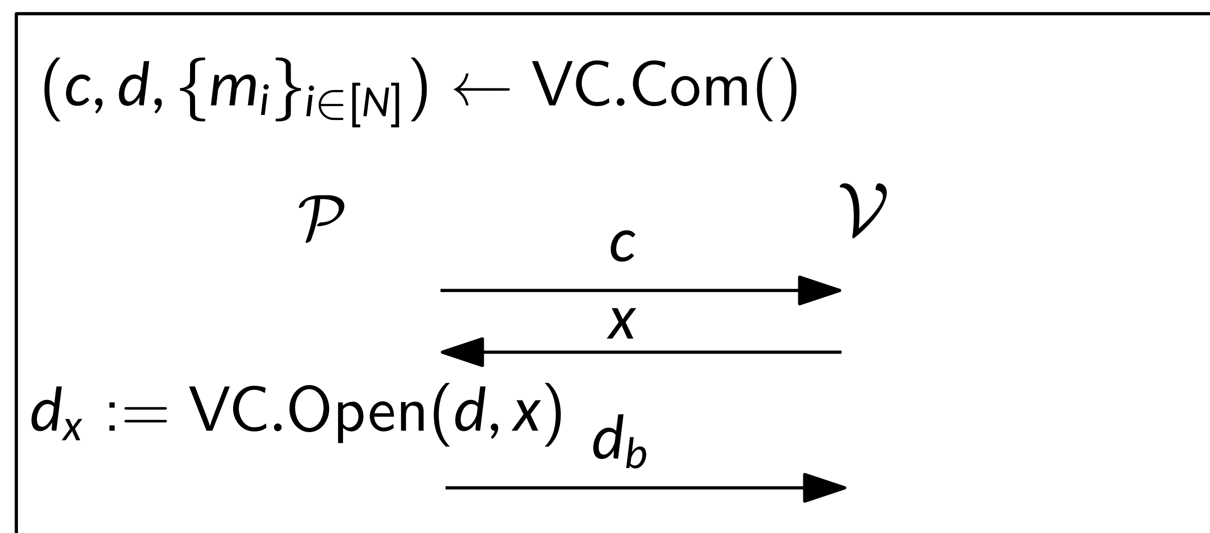
$\equiv$



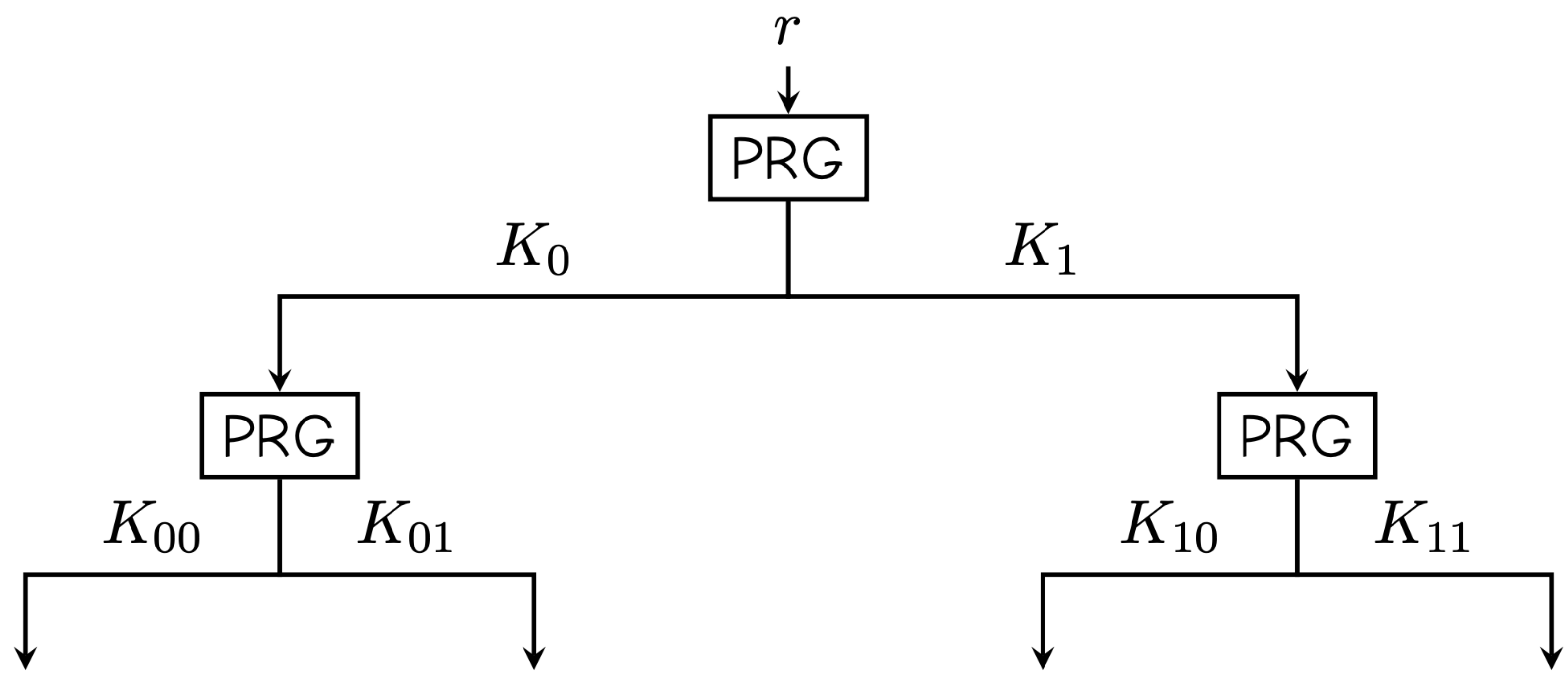
- In particular, we have public-coin **random** $\binom{N}{N-1}$-OT with $O(\log N)$ comm.



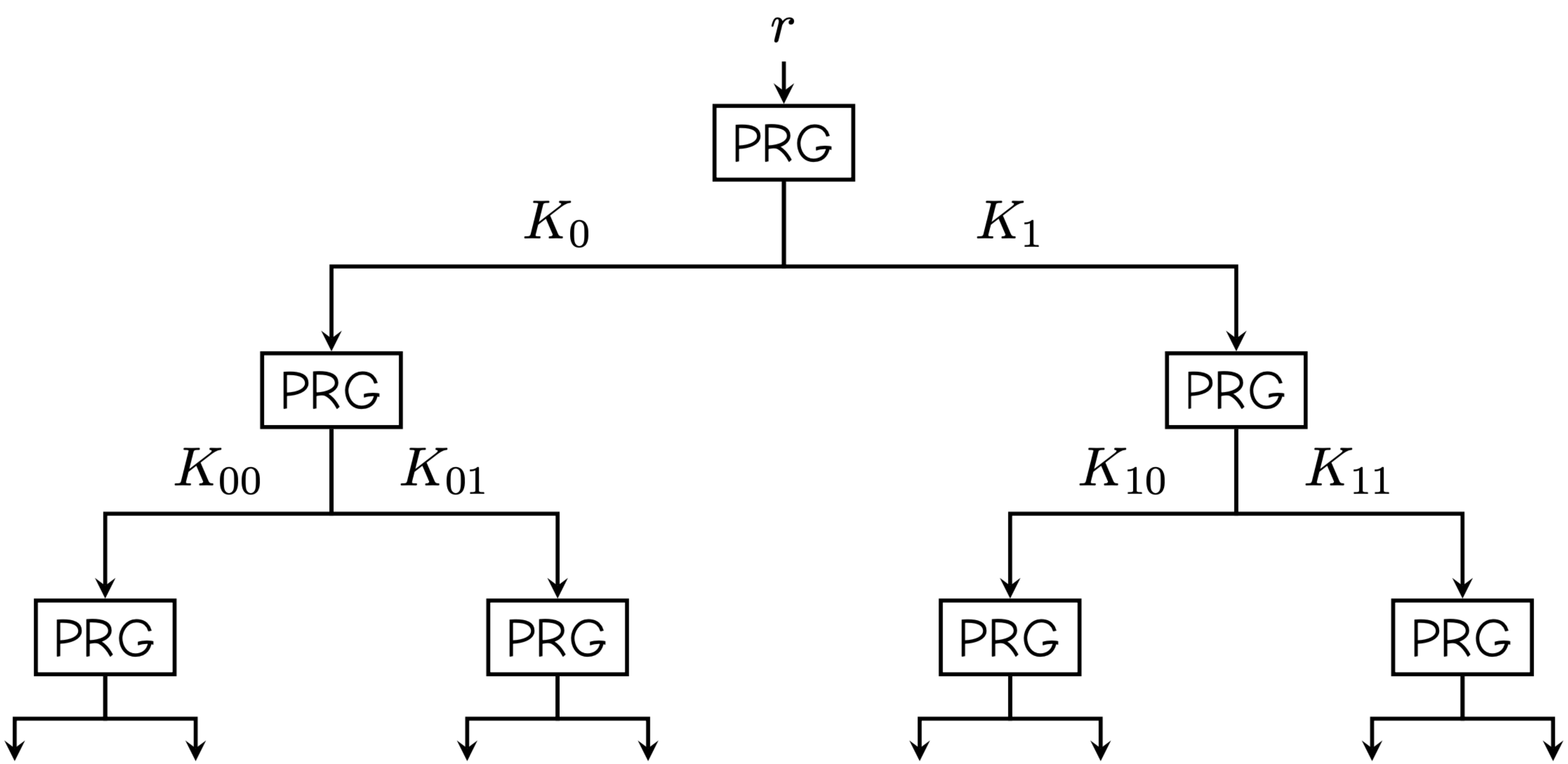
$\equiv$



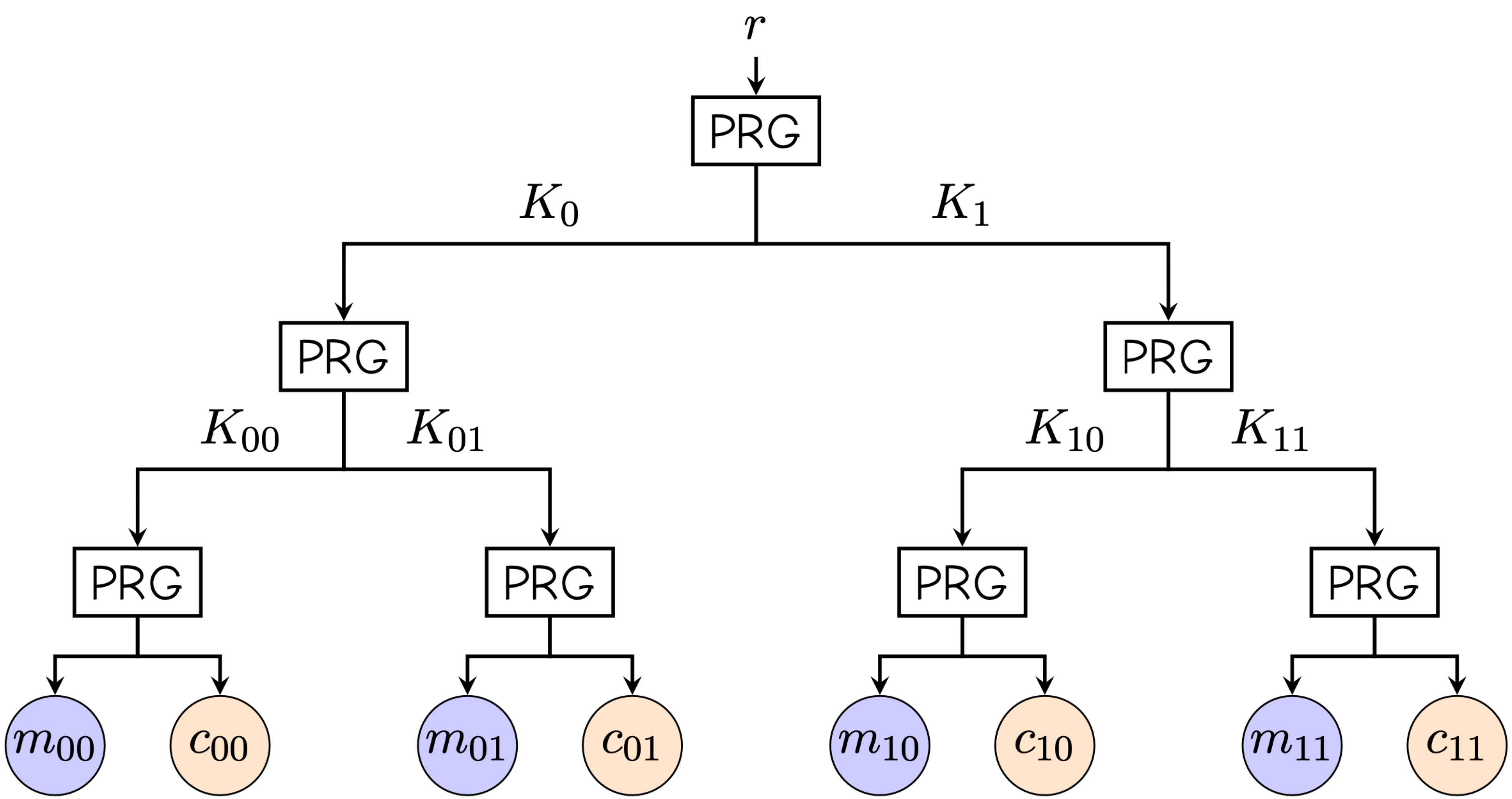
Random all-but-one VCOM as all-but-one OT



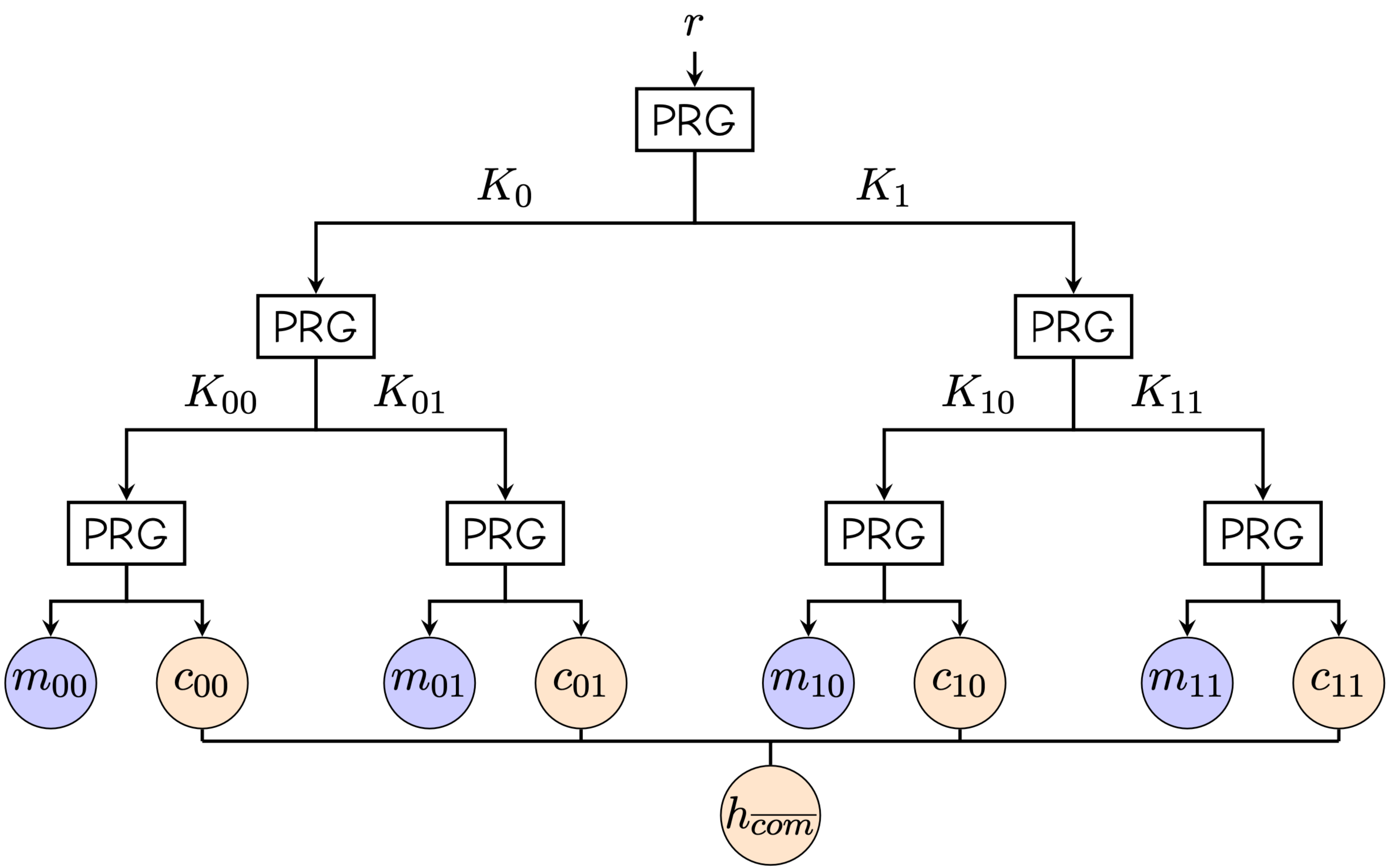
Random all-but-one VCOM as all-but-one OT



Random all-but-one VCOM as all-but-one OT

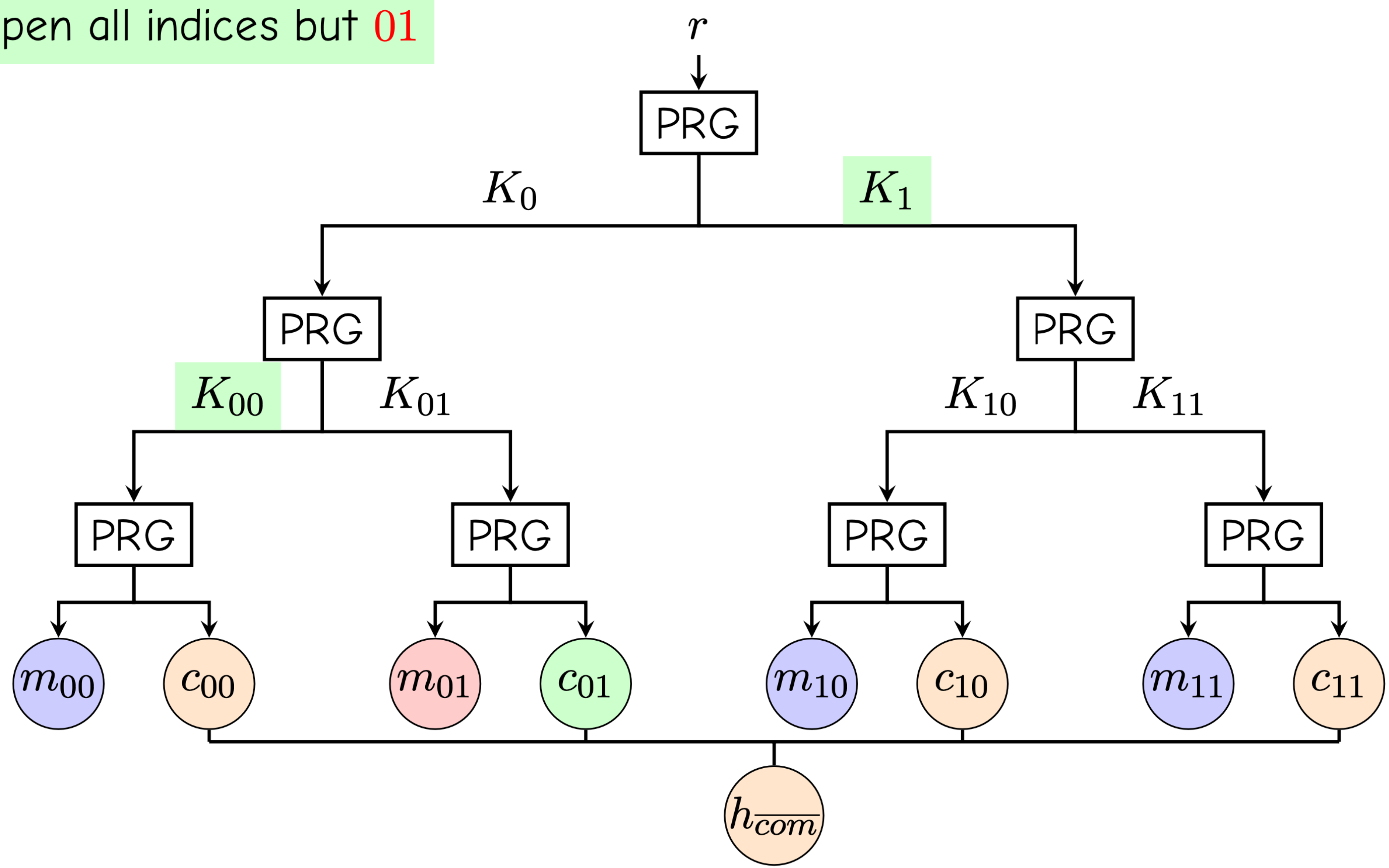


Random all-but-one VCOM as all-but-one OT



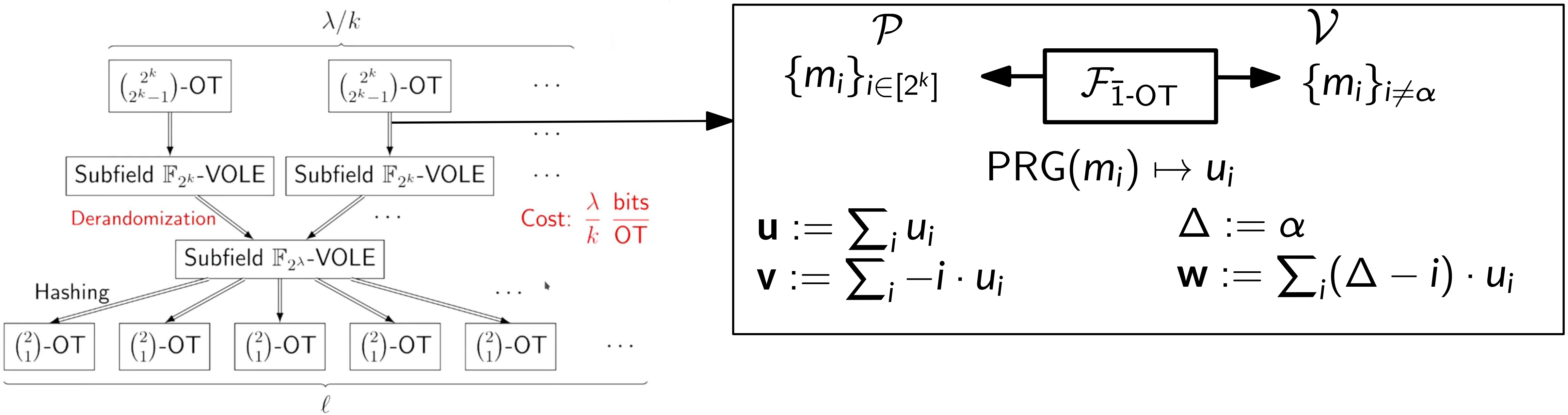
Random all-but-one VCOM as all-but-one OT

Open all indices but 01



14.5

Next Step: From $\mathcal{F}_{\bar{1}\text{-OT}}$ to Subspace $\mathcal{F}_{\text{VOLE}}$ (SoftSpokenOT)



$$\mathbf{w} = \mathbf{v} - \mathbf{u} \times \Delta \quad \in \mathbb{F}_{2^k}$$

Repeat $\tau = \lceil \kappa/k \rceil$ times

$$\mathbf{W}' = \mathbf{V} - \mathbf{U}' \times \begin{pmatrix} \Delta_1 & & \\ & \ddots & \\ & & \Delta_\tau \end{pmatrix}$$

κ-bit entropy

From $\mathcal{F}_{\bar{1}\text{-OT}}$ to Subspace $\mathcal{F}_{\text{VOLE}}$ (SoftSpokenOT), Continued

- Goal: $2^{-\kappa}$-sound IT-MAC

$$\underbrace{\mathbf{W}'}_{\tau} = \underbrace{\mathbf{V}}_{\tau} - \underbrace{\mathbf{u}}_{1} \times \underbrace{\begin{bmatrix}1 \ldots 1\end{bmatrix}}_{\tau} \times \begin{bmatrix}\Delta_1 & & \\ & \ddots & \\ & & \Delta_\tau\end{bmatrix}$$

$$\mathbb{F}_{2^\kappa} \equiv \mathbb{F}_2^\kappa$$

$$\Delta \in \mathbb{F}_{2^\kappa} \equiv \begin{bmatrix}1 \ldots 1\end{bmatrix} \times \begin{bmatrix}\Delta_1 & & \\ & \ddots & \\ & & \Delta_\kappa\end{bmatrix}$$

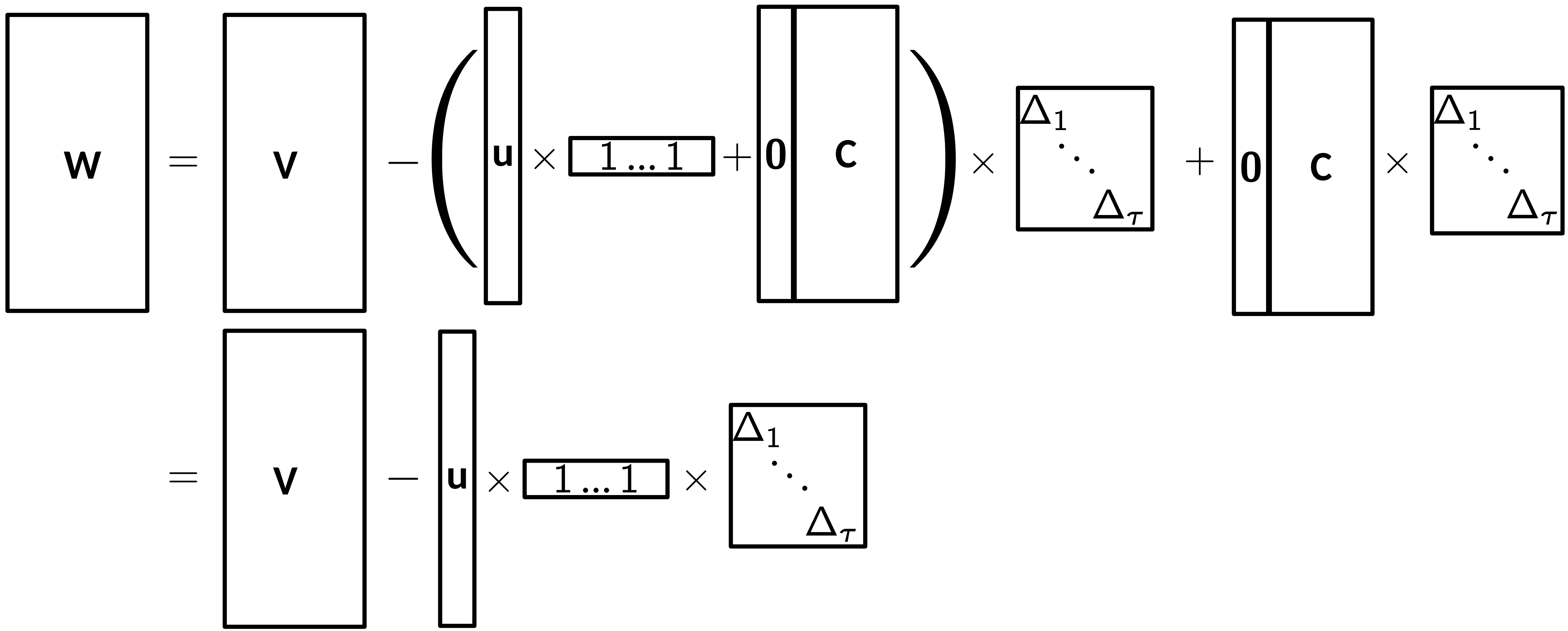
- Send Syndrome

$$\mathbf{W}' = \mathbf{V} - \mathbf{U}' \times \begin{bmatrix}\Delta_1 & & \\ & \ddots & \\ & & \Delta_\tau\end{bmatrix}$$

$$\mathbf{U}' \longrightarrow \underbrace{\mathbf{u}}_{1} \times \underbrace{\begin{bmatrix}1 \ldots 1\end{bmatrix}}_{\tau} + \begin{bmatrix}\underbrace{\mathbf{0}}_{1} & \underbrace{\mathbf{C}}_{\tau - 1}\end{bmatrix}$$

From $\mathcal{F}_{\bar{1}\text{-OT}}$ to Subspace $\mathcal{F}_{\text{VOLE}}$ (SoftSpokenOT), Continued

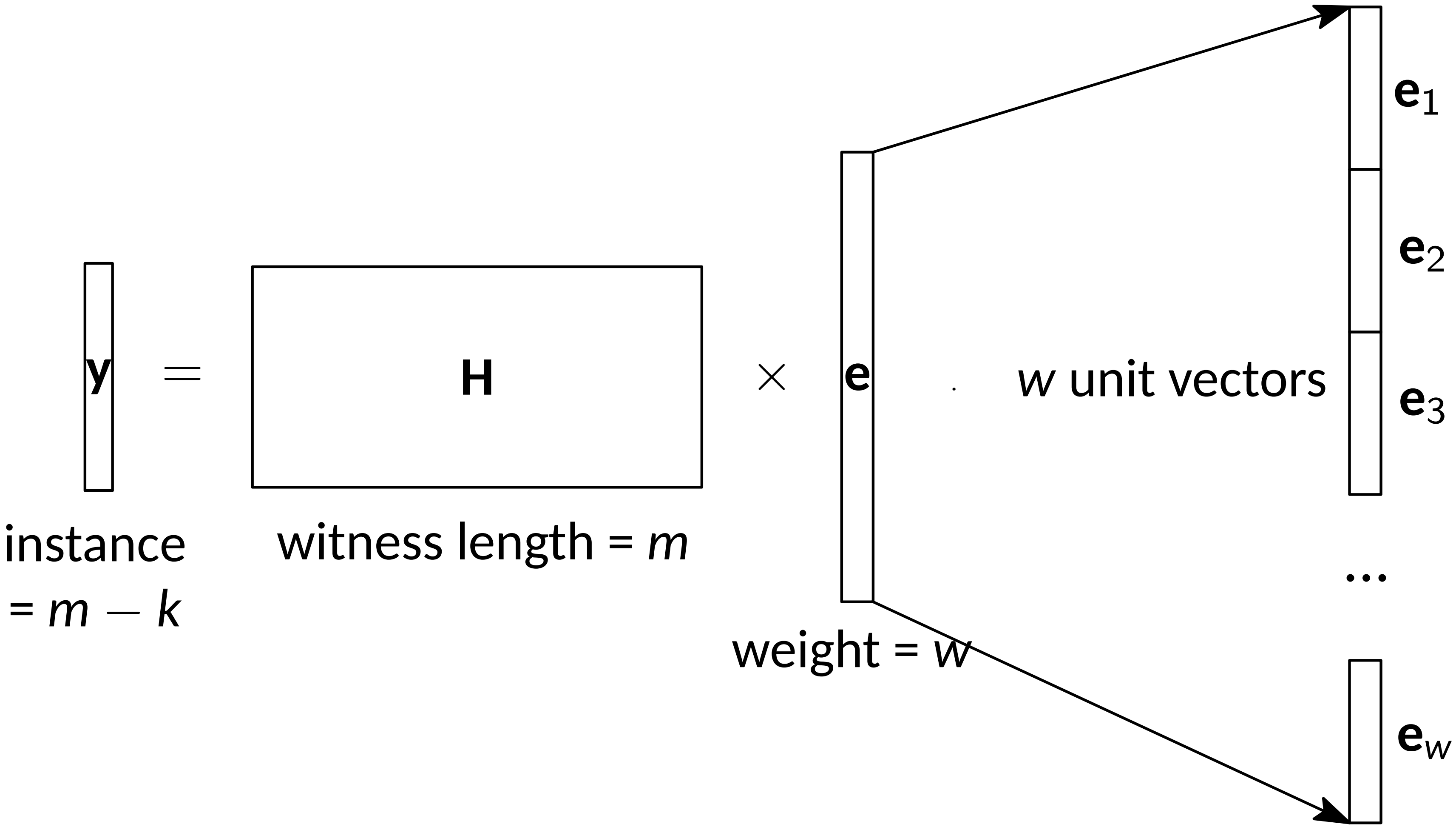
- $\mathcal{V}$ locally sets $\mathbf{W} = \mathbf{W}' + [0\|\mathbf{C}] \cdot \text{diag}(\mathbf{\Delta})$



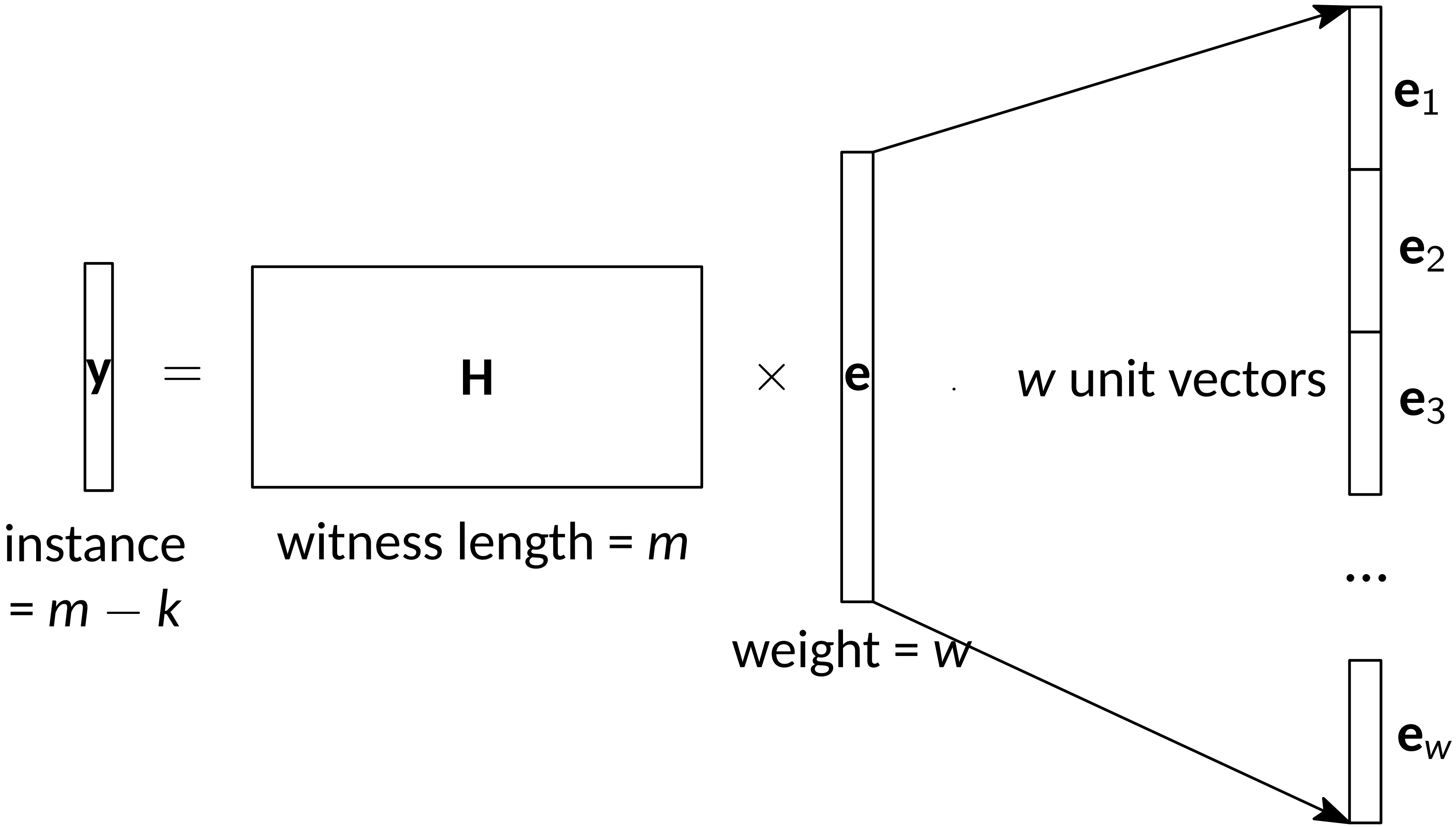
- Consistency Check: Use Linear-UHF to hash and reveal some rows to check $\mathcal{C}$-Δ-relations

Theorem 2. Protocol Π_{sVOLE} securely realizes $\mathcal{F}_{\text{sVOLE}}$ with distinguishing advantage $\binom{n_{\mathcal{C}}}{k_{\mathcal{C}}+1} \cdot \varepsilon$

Regular Syndrome Decoding



Regular Syndrome Decoding



- Systematic Form: $\mathbf{H} = [I_{m-k} \| \mathbf{H_B}]$
- $\mathbf{y} = \mathbf{H} \cdot \mathbf{e} = \mathbf{e}_A + \mathbf{H_B} \cdot \mathbf{e}_B$
- We only commit and get $[\mathbf{e}_B]$ and reconstruct $[\mathbf{e}] = [\mathbf{y} - \mathbf{H_B} \cdot \mathbf{e}_B \| \mathbf{e}_B]$

Sketching Technique

$$\left\langle \underset{\mathbf{r}_1}{\boxed{}}, \underset{\mathbf{e}}{\boxed{}} \right\rangle \times \left\langle \underset{\mathbf{r}_2}{\boxed{}}, \underset{\mathbf{e}}{\boxed{}} \right\rangle = \left\langle \underset{\mathbf{r}_1 \circ \mathbf{r}_2}{\boxed{}}, \underset{\mathbf{e}}{\boxed{}} \right\rangle$$

Sketching Technique

$$\left\langle \underset{\mathbf{r}_1}{\boxed{}}, \underset{\mathbf{e}}{\boxed{}} \right\rangle \times \left\langle \underset{\mathbf{r}_2}{\boxed{}}, \underset{\mathbf{e}}{\boxed{}} \right\rangle = \left\langle \underset{\mathbf{r}_1 \circ \mathbf{r}_2}{\boxed{}}, \underset{\mathbf{e}}{\boxed{}} \right\rangle$$

- If $\|\mathbf{e}\|_0 > 1$, there will be cross terms, soundness error $= \frac{2}{|\mathbb{F}|}$

Sketching Technique

$$\langle \mathbf{r}_1, \mathbf{e} \rangle \times \langle \mathbf{r}_2, \mathbf{e} \rangle = \langle \mathbf{r}_1 \circ \mathbf{r}_2, \mathbf{e} \rangle$$

- If $\|\mathbf{e}\|_0 > 1$, there will be cross terms, soundness error $= \frac{2}{|\mathbb{F}|}$

$$\langle \mathbf{1}, \mathbf{e} \rangle = 1$$

Sketching Technique

$$\left\langle \underset{\mathbf{r}_1}{\boxed{}}, \underset{\mathbf{e}}{\boxed{}} \right\rangle \times \left\langle \underset{\mathbf{r}_2}{\boxed{}}, \underset{\mathbf{e}}{\boxed{}} \right\rangle = \left\langle \underset{\mathbf{r}_1 \circ \mathbf{r}_2}{\boxed{}}, \underset{\mathbf{e}}{\boxed{}} \right\rangle$$

- If $||\mathbf{e}||_0 > 1$, there will be cross terms, soundness error $= \frac{2}{|\mathbb{F}|}$

$$\left\langle \underset{\mathbf{1}}{\boxed{}}, \underset{\mathbf{e}}{\boxed{}} \right\rangle = 1$$

- Use IT-MAC opening to check that $\langle \mathbf{1}, \mathbf{e} \rangle = 1$

Choosing Parameters for RSD

- Linearization Attack
- Information Set Decoding Attack
- Generalized Birthday Paradox Attack

- From SD-in-the-Head Specifications

4.1 Selection of the SD parameters

To select the parameters relative to the syndrome decoding problem, we estimate the cost of the best known algorithms to solve this problem. There exists two main families of such algorithms: the *Information Set Decoding* (ISD) algorithms and the *Generalized Birthday Algorithms* (GBA) [TS16; BBC+19]. The SD parameters are chosen such that both types of SD solving algorithms have complexity at least 2^κ corresponding to the complexity of breaking AES by exhaustive search (in the gate-count metric). In practice, we take κ equal to 143, 207 and 272 respectively for categories I (AES-128), III (AES-192) and V (AES-256) in accordance to [NIS22].

https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/spec-files/SDitH-spec-web.pdf

Parameters

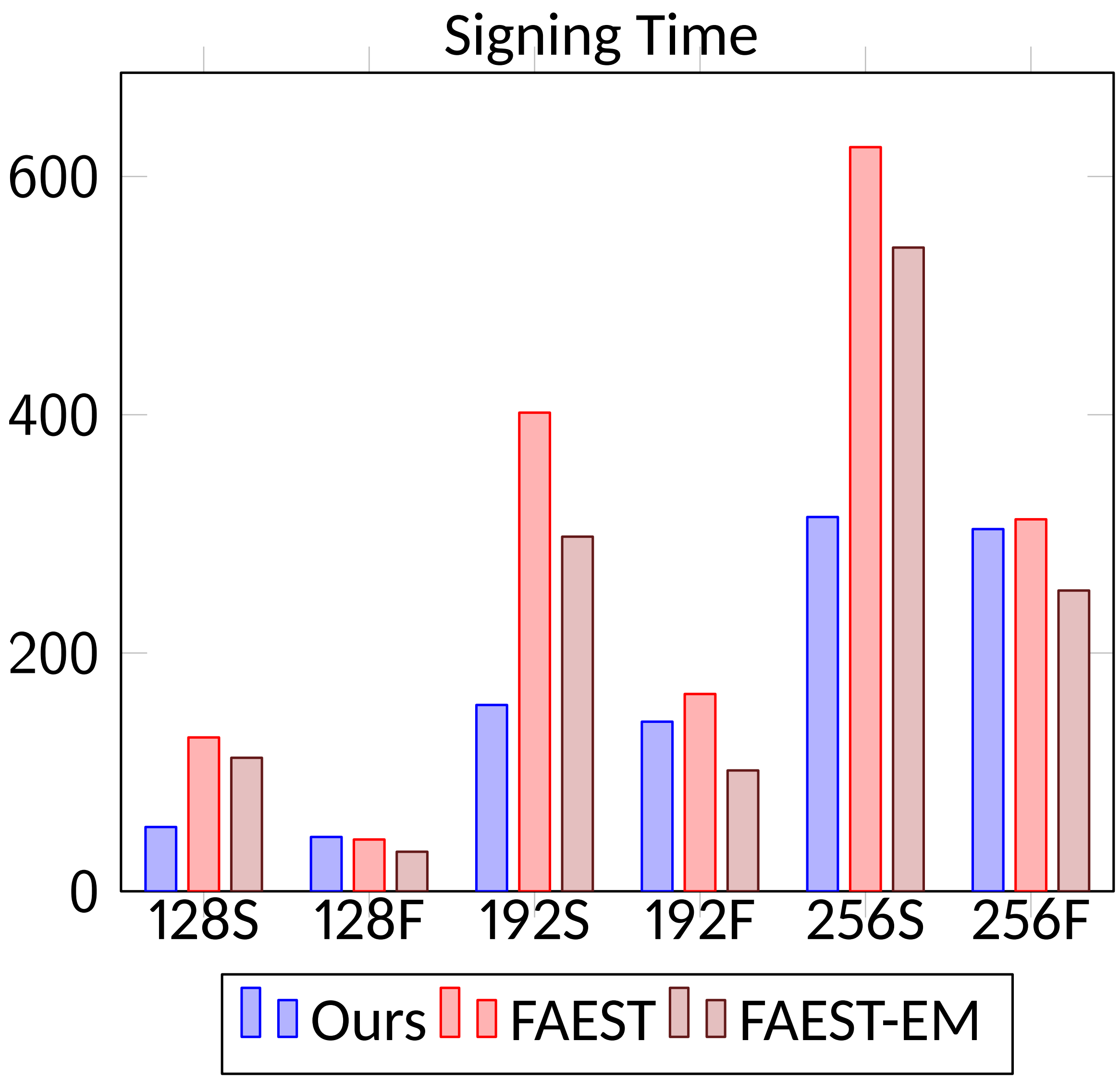
Security Level	m	k	w	m/d	Estimated Bit Security
NIST L1	1393	833	199	7	143.34
NIST L3	2190	1248	365	6	207.70
NIST L5	2928	1668	488	6	272.51

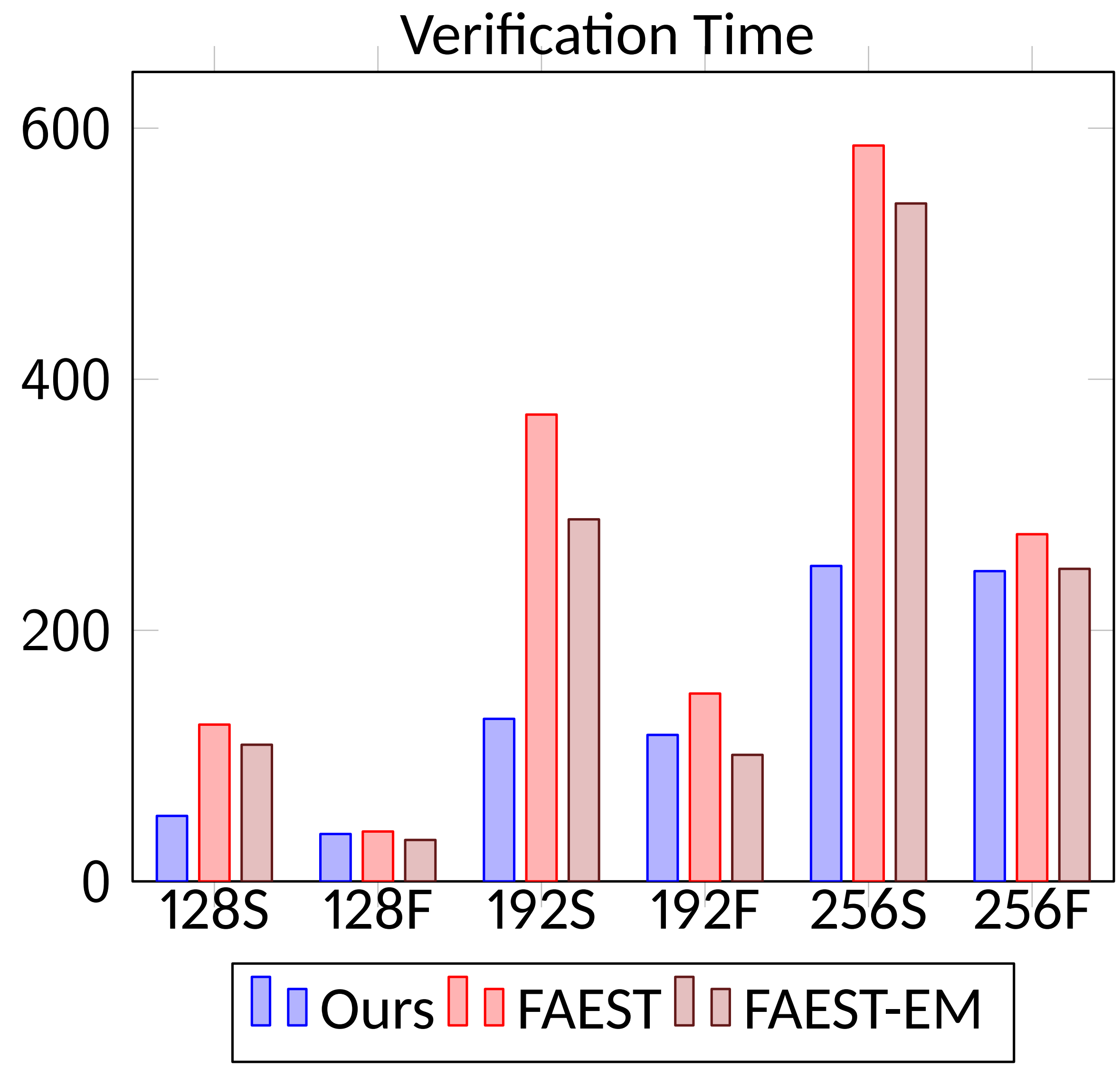
Performance

- "**S**mall but slow" variant is comparable
- "**F**ast but large" variant appears hard to tame

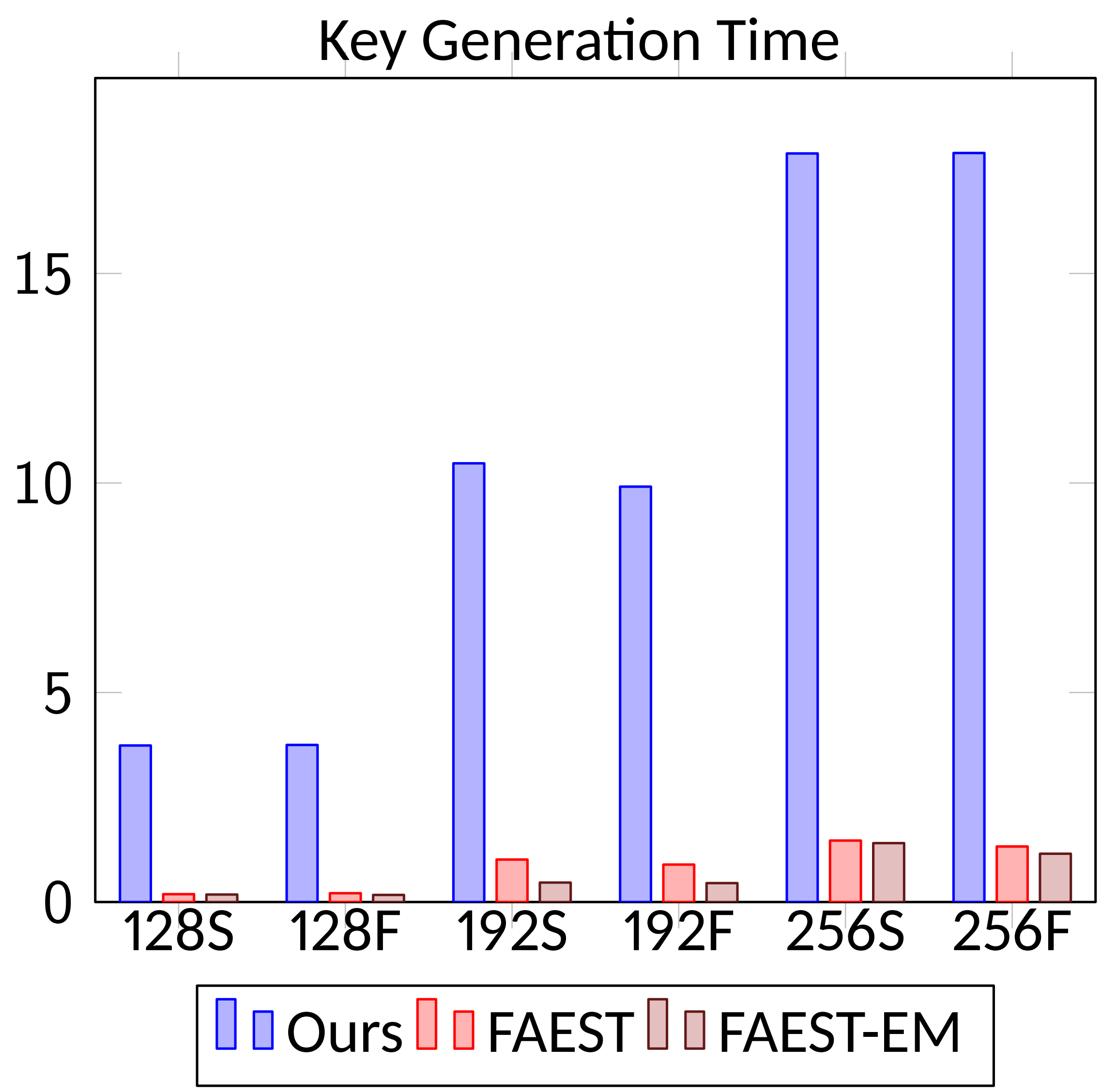
name	sig size	sk	pk	keygen	sign	verify
Ours_128S	4420	32	86	3.73557 ms	53.943 ms	52.1632 ms
Ours_128F	5512	32	86	3.74889 ms	45.5962 ms	37.7521 ms
Ours_192S	10584	48	142	10.4693 ms	156.395 ms	129.477 ms
Ours_192F	13776	48	142	9.91277 ms	142.364 ms	116.684 ms
Ours_256S	20720	64	190	17.8642 ms	314.161 ms	251.276 ms
Ours_256F	26496	64	190	17.8749 ms	304.009 ms	247.138 ms
FAEST_128S	5006	32	32	187.681 us	129.139 ms	124.891 ms
FAEST_128F	6336	32	32	210.627 us	43.4592 ms	39.7393 ms
FAEST_192S	12744	56	64	1.01353 ms	401.755 ms	371.869 ms
FAEST_192F	16792	56	64	893.735 us	165.605 ms	149.641 ms
FAEST_256S	22100	64	64	1.46707 ms	624.618 ms	586.189 ms
FAEST_256F	28400	64	64	1.32513 ms	312.229 ms	276.544 ms
FAEST_EM_128S	4566	32	32	178.814 us	112.059 ms	108.851 ms
FAEST_EM_128F	5696	32	32	169.509 us	33.1948 ms	33.0019 ms
FAEST_EM_192S	10824	48	48	464.477 us	297.662 ms	288.398 ms
FAEST_EM_192F	13912	48	48	451.677 us	101.44 ms	100.785 ms
FAEST_EM_256S	20956	64	64	1.40576 ms	540.346 ms	540.04 ms
FAEST_EM_256F	26736	64	64	1.15299 ms	252.45 ms	248.962 ms

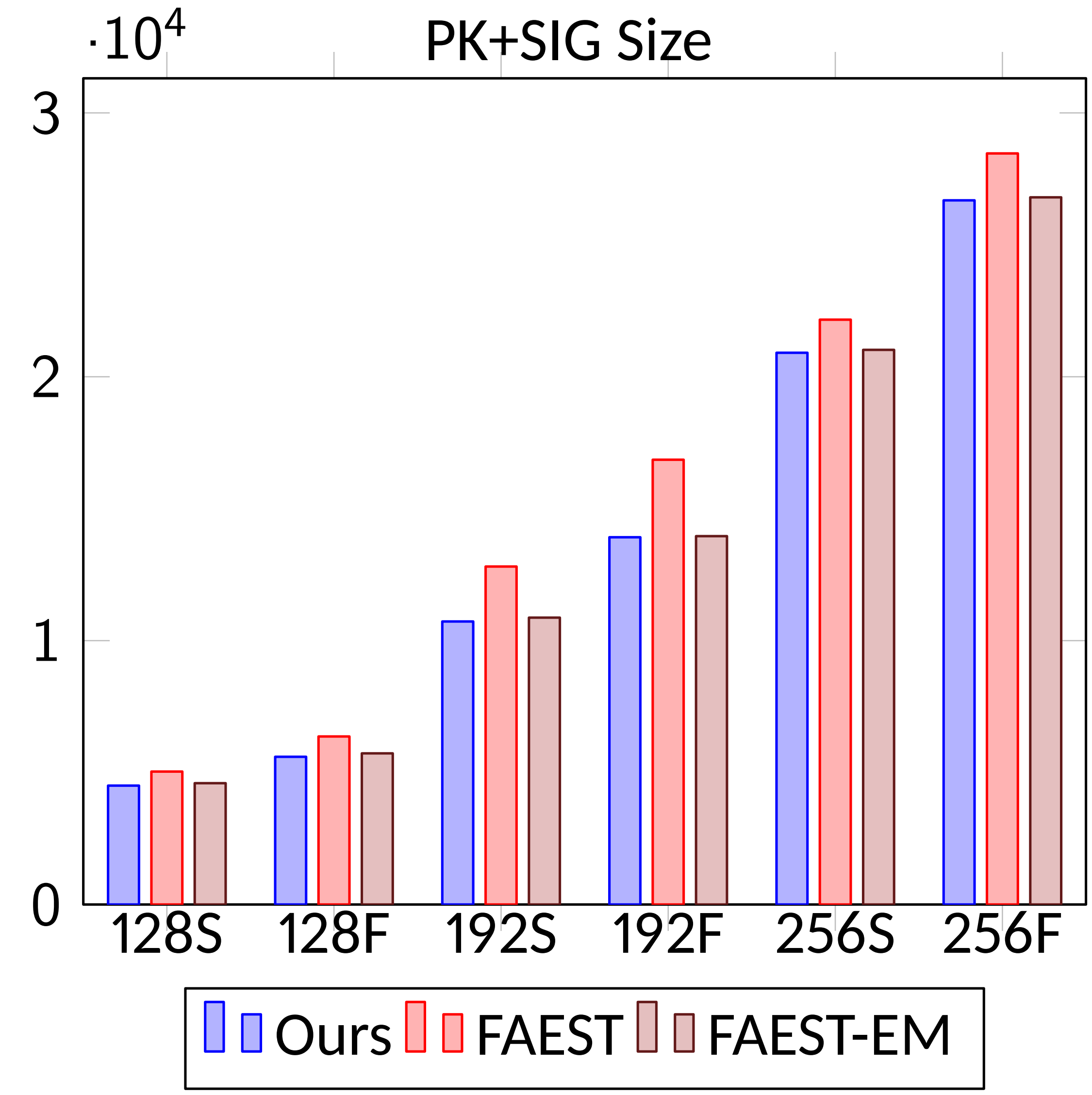
Performance





Performance

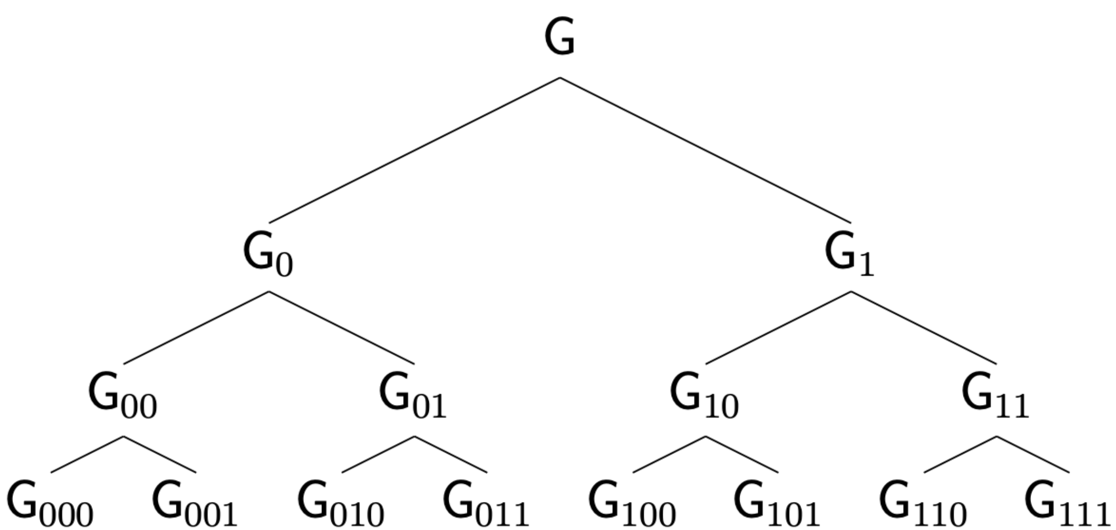




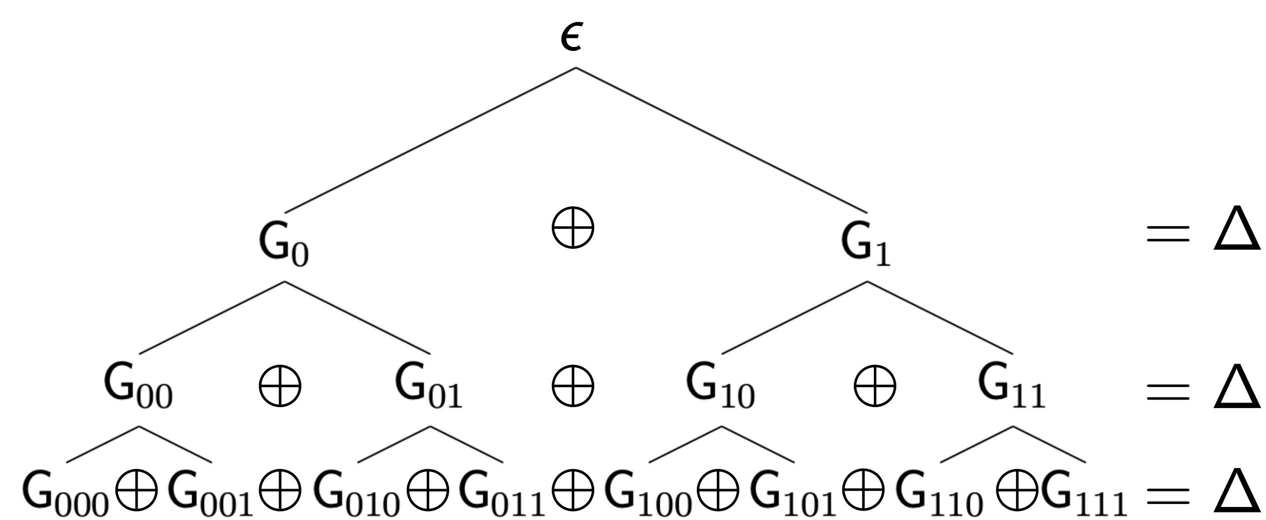
Half-tree Optimization

- Save computation/communication by introducing correlation at each level

GGM Tree



Correlated GGM Tree



- Expansion: $G_{00}||G_{01} = \mathrm{PRG}(G_0)$ $\quad$ $G_{00} = \mathrm{H}(G_0), G_{01} = G_0 \oplus G_{00}$
- Costs: $N\times$ RO or $2N\times$ RP $\quad$ $N\times$ RP
- Initial Setup: $G \leftarrow \mathbb{F}_2^\kappa$ $\quad$ $G_0 = k \leftarrow \mathbb{F}_2^\kappa \quad G_1 = \Delta - k$

Optimization?

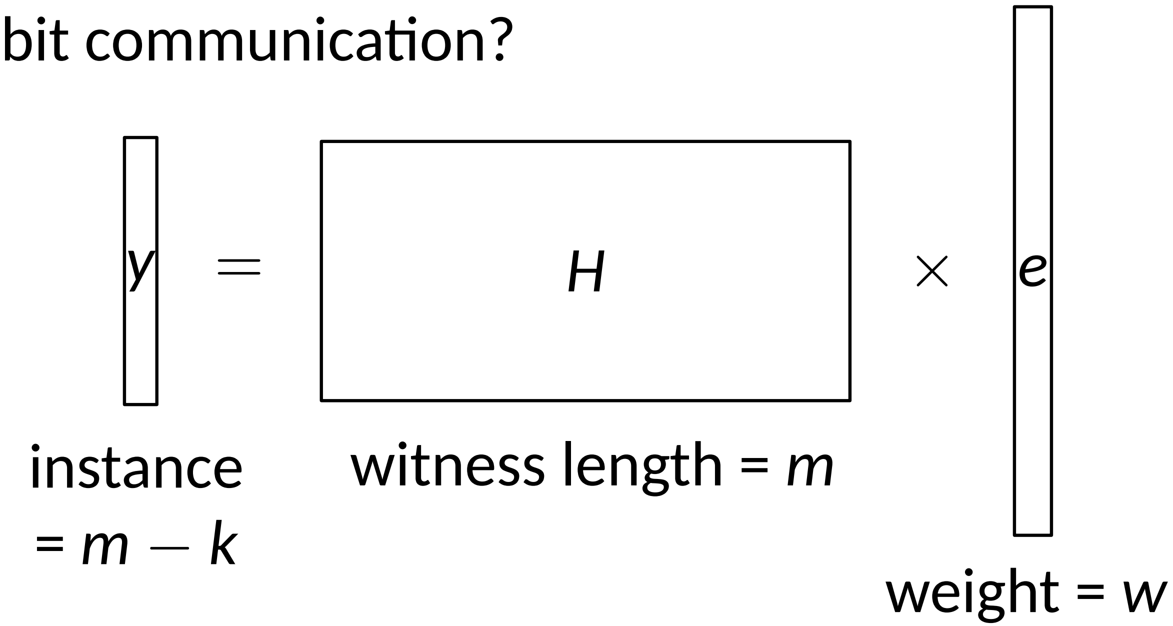
- We need $\ell := m + 2\kappa$ random bits for QuickSilver
- Half-tree gives κ bits
- How to expand it into ℓ bits with less than ℓ bit communication?

Scheme	SD Parameters					MPC Parameters			
	q	m	k	w	d	$\lvert\mathbb{F}_{\text{poly}}\rvert$	$\lvert\mathbb{F}_{\text{points}}\rvert$	t	p
Variant 1	2	1280	640	132	1	2^{11}	2^{22}	6	$\approx 2^{-69}$
Variant 2	2	1536	888	120	6	2^{8}	2^{24}	5	$\approx 2^{-79}$
Variant 3	2^{8}	256	128	80	1	2^{8}	2^{24}	5	$\approx 2^{-78}$

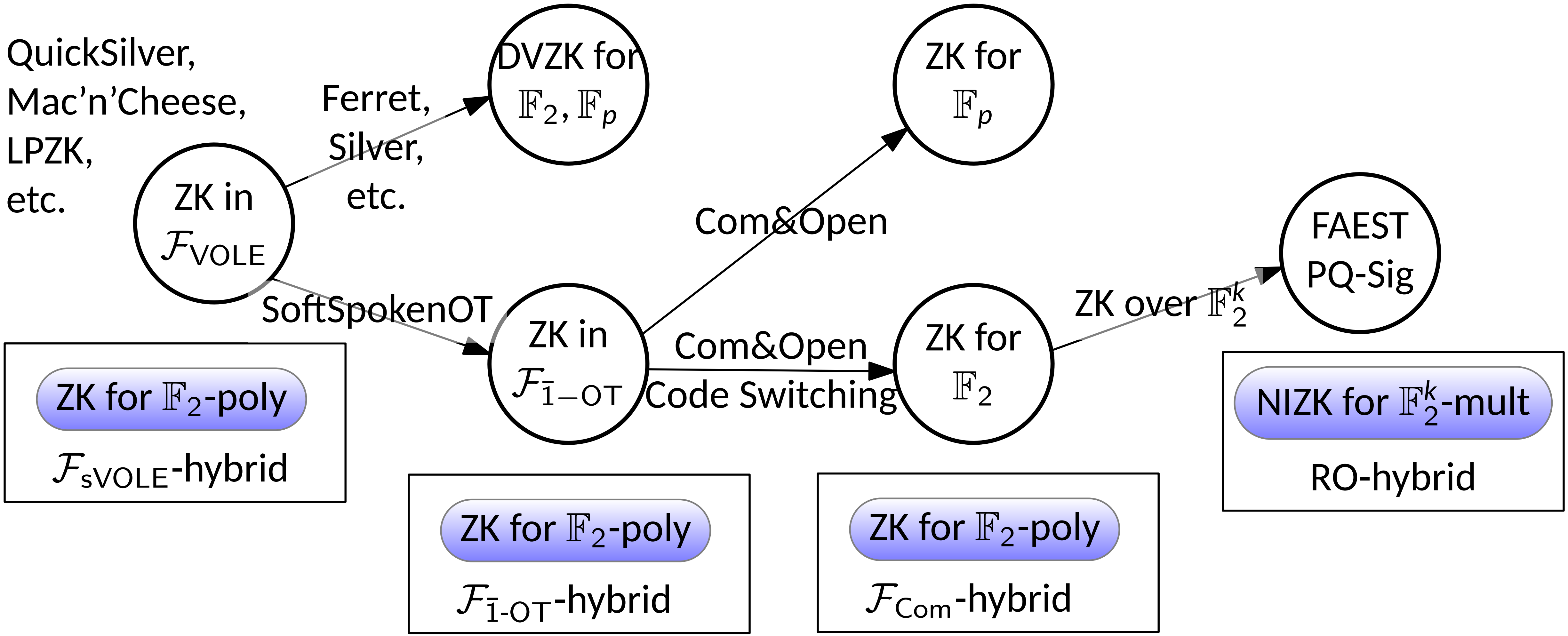
Table 3: SD and MPC parameters.

n	k	h	Best [34]	d_{conj} plain	(f, u)	d_{conj}	XL hybrid Sec. 4.2
2^{22}	64770	2735	104	2	$(0,0)$	2	<u>103</u>
2^{20}	32771	1419	99	3	$(1159,2)$	2	<u>98</u>
2^{18}	15336	760	95	3	$(657,7)$	2	104
2^{16}	7391	389	91	4	$(373,10)$	2	108
2^{14}	3482	198	86	6	$(197,11)$	2	106
2^{12}	1589	98	83	8	$(88,13)$	2	103
2^{10}	652	57	94	12	$(54,9)$	2	101

Table 2. Hybrid approach of Section 4.2 over $\mathbb{F}_2$ (Modeling 2).



ZK for Polynomial Constraints Over **Small** Fields



The 3-Round Protocol

Protocol $\Pi^t_{\text{2D-Rep}}$

PARAMETERS: Code $\mathcal{C}_{\mathsf{Rep}} = [\tau, 1, \tau]_p$ with $\mathbf{G}_{\mathcal{C}} = (1 \ldots 1) \in \mathbb{F}_p^{1\times\tau}$. VOLE size $q = p^r$.
INPUTS: Polynomials $f_i \in \mathbb{F}_{p^k}[X_1, \ldots, X_\ell]_{\leq 2}$, $i \in [t]$. The prover $\mathcal{P}$ also holds a witness $\mathbf{w} \in \mathbb{F}_p^\ell$ such that $f_i(\mathbf{w}) = 0$ for all $i \in [t]$.

Round 1. $\mathcal{P}$ does the following:

1. Call the functionality $\mathcal{F}_{\mathsf{sVOLE}}^{p,q,S_\Delta,\mathcal{C}_{\mathsf{Rep}},\ell+r\tau,\mathcal{L}}$ and receive $\mathbf{u} \in \mathbb{F}_p^{\ell+r\tau}, \mathbf{V} \in \mathbb{F}_q^{(\ell+r\tau)\times\tau}$.
   $\mathcal{V}$ receives done.
2. Compute $\mathbf{d} = \mathbf{w} - \mathbf{u}_{[1..\ell]} \in \mathbb{F}_p^\ell$ and send $\mathbf{d}$ to $\mathcal{V}$.
3. For $i \in [\ell+1..\ell+r\tau]$, embed $u_i \hookrightarrow \mathbb{F}_{q^\tau}$.
   For $i \in [\ell+r\tau]$, lift $\mathbf{v}_i \in \mathbb{F}_q^\tau$ into $v_i \in \mathbb{F}_{q^\tau}$.
   For $i \in [\ell]$, also embed $w_i \hookrightarrow \mathbb{F}_{q^\tau}$.

Round 2. $\mathcal{V}$ sends challenges $\chi_i \in \mathbb{F}_{q^\tau}$, $i \in [t]$.

Round 3. $\mathcal{P}$ does the following:

1. For each $i \in [t]$, compute $A_{i,0}, A_{i,1} \in \mathbb{F}_{q^\tau}$ such that
$$c_i(Y) = \bar{f}_i(w_1, \ldots, w_n) \cdot Y^2 + A_{i,1} \cdot Y + A_{i,0}.$$
2. Compute
$$u^* = \sum_{i\in[r\tau]} u_i X^{i-1} \qquad v^* = \sum_{i\in[r\tau]} v_i X^{i-1},$$
   where $\mathbb{F}_{q^\tau} \simeq \mathbb{F}_p[X]/F(X)$.
3. Compute $\tilde{b} = \sum_{i\in[t]} \chi_i \cdot A_{i,0} + v^* \in \mathbb{F}_{q^\tau}$ and $\tilde{a} = \sum_{i\in[t]} \chi_i \cdot A_{i,1} + u^* \in \mathbb{F}_{q^\tau}$ and send $(\tilde{a}, \tilde{b})$ to $\mathcal{V}$.

Verification. $\mathcal{V}$ runs the following check:

1. Call $\mathcal{F}_{\mathsf{sVOLE}}^{p,q,S_\Delta,\mathcal{C}_{\mathsf{Rep}},\ell+r\tau,\mathcal{L}}$ on input (get) and obtain $\boldsymbol{\Delta} \in \mathbb{F}_q^\tau$, $\mathbf{Q} \in \mathbb{F}_q^{(\ell+r\tau)\times\tau}$ such that $\mathbf{Q} = \mathbf{V} + \mathbf{u}^T\mathbf{G}_{\mathcal{C}}\mathsf{diag}(\boldsymbol{\Delta})$.
2. Compute $\mathbf{Q}' = \mathbf{Q}_{[1..\ell]} + \mathbf{d}^T\mathbf{G}_{\mathcal{C}}\mathsf{diag}(\boldsymbol{\Delta}) = \mathbf{V}_{[1..\ell]} + \mathbf{w}^T\mathbf{G}_{\mathcal{C}}\mathsf{diag}(\boldsymbol{\Delta})$.
3. Lift $\boldsymbol{\Delta}, \mathbf{q}'_1, \ldots, \mathbf{q}'_\ell, \mathbf{q}_{\ell+1}, \ldots, \mathbf{q}_{\ell+r\tau} \in \mathbb{F}_q^\tau$ into $\Delta, q'_1, \ldots, q'_\ell, q_{\ell+1}, \ldots, q_{\ell+r\tau} \in \mathbb{F}_{q^\tau}$.
4. For each $i \in [t]$, compute
$$c_i(\Delta) = \sum_{h\in[0,2]} \bar{f}_{i,h}(q'_1, \ldots, q'_\ell) \cdot \Delta^{2-h}$$
5. Compute $q^* = \sum_{i\in[r\tau]} q_{\ell+i} \cdot X^{i-1}$ such that $q^* = v^* + u^*\Delta$.
6. Compute $\tilde{c} = \sum_{i\in[t]} \chi_i \cdot c_i(\Delta) + q^*$.
7. Check that $\tilde{c} \stackrel{?}{=} \tilde{a} \cdot \Delta + \tilde{b}$.

Theorem 4. The Protocol $\Pi^t_{\text{2D-Rep}}$ is a ZKPoK with soundness error $\frac{3}{p^{r\tau}}$.

How to Handle Arbitrary $\mathcal{C}$?

- For subspace VOLE with general code $[n_{\mathcal{C}}, k_{\mathcal{C}}, d_{\mathcal{C}}]$ and witness $\mathbf{w} = \mathbb{F}_p^{\ell \times k_{\mathcal{C}}}$
- The committed witness is as follows

$$\ell\ \underset{n_{\mathcal{C}}}{\boxed{B}} = \boxed{V} - \underset{k_{\mathcal{C}}}{\boxed{A}} \times \underset{n_{\mathcal{C}}}{\boxed{G_{\mathcal{C}}}} \times \begin{bmatrix}\Delta_1 & & \\ & \ddots & \\ & & \Delta_{n_{\mathcal{C}}}\end{bmatrix}$$

Problem: Only row-wise linearity

In $Rep(\kappa)$, $k_{\mathcal{C}} = 1$

- Solution: Simulate VOLE in $\mathcal{P}$'s head once again

$\mathcal{P}$ $\qquad\qquad\qquad\qquad$ $\mathcal{V}$

Prepares $(2\ell + 2)$ rows in A'

$$A' = \begin{bmatrix} A \\ B \end{bmatrix}$$

$\xrightarrow{\text{DVZK } \pi \text{ as if } \Delta' \text{ is the key}}$

$\xleftarrow{\Delta'}$

$\xrightarrow{V = B + \Delta' \cdot A}$

$\mathcal{P}$ and $\mathcal{V}$ continue the subspace VOLE simulation

$\mathcal{V}$ accepts if

- π is valid under Δ'
- The opening of V is correct under $\mathrm{diag}(\vec{\Delta})$

The Code-Switching Technique

Protocol $\Pi^t_{\mathsf{2D-LC}}$

The protocol is parameterized by an $[n_\mathcal{C}, k_\mathcal{C}, d_\mathcal{C}]_p$ linear code $\mathcal{C}$, set $S_\Delta \subset \mathbb{F}_p^{n_\mathcal{C}}$ and a leakage space $\mathcal{L}$ (used in $\mathcal{F}_{\mathsf{sVOLE}}$).

INPUTS: Both parties hold a set of polynomials $f_i \in \mathbb{F}_p[X_1, \ldots, X_\ell]_{\leq 2}$, $i \in [t]$. $\mathcal{P}$ also holds a witness $\mathbf{w} \in \mathbb{F}_p^{k_\mathcal{C}\ell}$ such that $f_i(\mathbf{w}) = 0$, for all $i \in [t]$.

Round 1. $\mathcal{P}$ does as follows:

1. $\mathcal{P}$ and $\mathcal{V}$ call $\mathcal{F}_{\mathsf{sVOLE}}^{p,p,S_\Delta,\mathcal{C},2\ell+1,\mathcal{L}}$, $\mathcal{P}$ receives $\mathbf{U} \in \mathbb{F}_p^{(2\ell+2)\times k_\mathcal{C}}$, $\mathbf{V} \in \mathbb{F}_p^{(2\ell+2)\times n_\mathcal{C}}$, while $\mathcal{V}$ gets the message done.
2. $\mathcal{P}$ sets $\mathbf{V}_1 = \mathbf{V}_{[1..\ell+1]}$, $\mathbf{V}_2 = \mathbf{V}_{[\ell+2..2\ell+2]}$ and $\mathbf{R} = \mathbf{U}_{[\ell+2..2\ell+2]}$
3. $\mathcal{P}$ commits to its witness by sending $\mathbf{D} = \mathbf{W} - \mathbf{U}_{[1..\ell]}$.

Round 2. $\mathcal{V}$ samples $\boldsymbol{\chi} \leftarrow \mathbb{F}_p^t$ and sends it to $\mathcal{P}$.

Round 3. $\mathcal{P}$ proceeds as follows.

1. For each $i \in [t]$, compute
$$g_i(Y) := \sum\nolimits_{h\in[0,2]} f_{i,h}(\mathbf{r}_1 + \mathbf{w}_1 \cdot Y, \ldots, \mathbf{r}_\ell + \mathbf{w}_\ell \cdot Y) \cdot Y^{2-h} = \sum\nolimits_{h\in[0,1]} A_{i,h} \cdot Y^h$$
2. Compute $\widetilde{\mathbf{b}} = \sum_{i\in[t]} \chi_i \cdot A_{i,0} + \mathbf{r}_{\ell+1}$ and $\widetilde{\mathbf{a}} = \sum_{i\in[t]} \chi_i \cdot A_{i,1} + \mathbf{u}_{1,\ell+1}$, where $\mathbf{u}_{1,i}$ is the ith row of $\mathbf{U}$.
3. Send $(\widetilde{\mathbf{b}}, \widetilde{\mathbf{a}})$ to $\mathcal{V}$.

Round 4. $\mathcal{V}$ samples $\Delta' \leftarrow \mathbb{F}_p$ and sends it to the prover.

Round 5. $\mathcal{P}$ sends $\mathbf{S} = \mathbf{R} + \mathbf{U}_{[1..\ell+1]} \cdot \Delta' \in \mathbb{F}_p^{(\ell+1)\times n_\mathcal{C}}$ to $\mathcal{V}$

Round 6. $\mathcal{V}$ samples $\boldsymbol{\eta} \leftarrow \mathbb{F}_p^{\ell+1}$ and sends it to $\mathcal{P}$

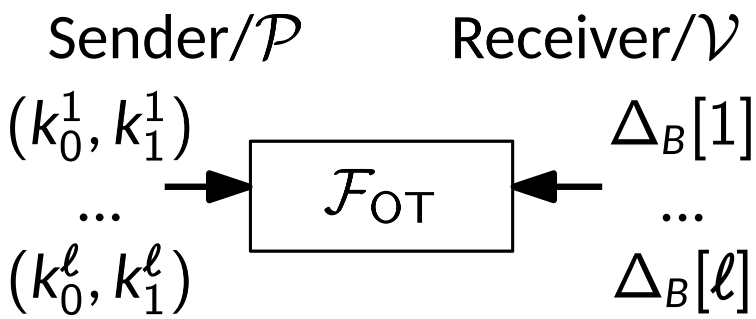
Round 7. $\mathcal{P}$ computes $\widetilde{\mathbf{v}} = \boldsymbol{\eta}^\top(\mathbf{V}_2 + \mathbf{V}_1 \cdot \Delta')$ and sends it to $\mathcal{V}$.

Verification. $\mathcal{V}$ runs the following checks.

1. *Check the constraints:*
   - Compute $\mathbf{S}' = \mathbf{S} + \begin{bmatrix}\mathbf{D}\\ 0\end{bmatrix} \cdot \Delta' = \mathbf{R} + \begin{bmatrix}\mathbf{W}\\ \mathbf{u}_{\ell+1}\end{bmatrix} \cdot \Delta'$.
   - For each $i \in [t]$, compute
   $$\mathbf{c}_i(Y) = \sum_{h\in[0,2]} f_{i,h}(\mathbf{s}'_1, \ldots, \mathbf{s}'_\ell) \cdot Y^{2-h}.$$
   - Let $\widetilde{\mathbf{s}} = \sum_{i\in[t]} \chi_i \cdot \mathbf{c}_i(\Delta') + \mathbf{s}'_{\ell+1}$.
   - Check that $\widetilde{\mathbf{s}} = \widetilde{\mathbf{b}} + \widetilde{\mathbf{a}} \cdot \Delta'$.
2. *Check the opening of* $\mathbf{S}$*:*
   - Call $\mathcal{F}_{\mathsf{sVOLE}}^{p,p,S_\Delta,\mathcal{C},2\ell+1,\mathcal{L}}$ on input (get) and obtain $\boldsymbol{\Delta} \in \mathbb{F}_p^{n_\mathcal{C}}$ and $\mathbf{Q} \in \mathbb{F}_p^{(2\ell+2)\times n_\mathcal{C}}$ such that $\mathbf{Q} = \mathbf{V} + \mathcal{C}(\mathbf{U}) \cdot \mathsf{diag}(\boldsymbol{\Delta})$
   - Set $\mathbf{Q}_1 = \mathbf{Q}_{[1..\ell+1]}$ and $\mathbf{Q}_2 = \mathbf{Q}_{[\ell+2..2\ell+2]}$.
   - Check that
   $$\boldsymbol{\eta}^\top(\mathbf{Q}_2 + \mathbf{Q}_1 \cdot \Delta') = \widetilde{\mathbf{v}} + \boldsymbol{\eta}^\top \cdot \mathcal{C}(\mathbf{S}) \cdot \mathsf{diag}(\boldsymbol{\Delta})$$

Theorem 3. The protocol $\Pi^t_{\text{2D-LC}}$ is a SHVZKPoK with soundness error $\frac{3}{p} + 2|S_\Delta|^{-d_\mathcal{C}}$ in the $\mathcal{F}_{\mathsf{sVOLE}}^{p,S_\Delta,\mathcal{C},2(\ell+2),\mathcal{L}}$-hybrid model

The Problem with LPN-based State-of-the-Art



The Problem with LPN-based State-of-the-Art

Sender/$\mathcal{P}$ Receiver/$\mathcal{V}$

(k_0^1, k_1^1) ... (k_0^ℓ, k_1^ℓ) $\rightarrow$ $\boxed{\mathcal{F}_{\mathsf{OT}}}$ $\leftarrow$ $\Delta_B[1]$ ... $\Delta_B[\ell]$

Extend (u)

$$m_1 := \mathsf{PRF}(k_0^1, j) + \mathsf{PRF}(k_1^1, j) + u$$

...

$$m_\ell := \mathsf{PRF}(k_0^\ell, j) + \mathsf{PRF}(k_1^\ell, j) + u$$

$\longrightarrow$

$$u\Delta_B[i] = \underbrace{\mathsf{PRF}(k_0^i, j)}_{\text{Sender}} + \underbrace{\mathsf{PRF}(k^i_{\Delta_B[i]}, j) + m_i\Delta_B[i]}_{\text{Receiver}}$$

Use LHL to remove selective failure leackage on Δ

The Problem with LPN-based State-of-the-Art

Sender/$\mathcal{P}$ Receiver/$\mathcal{V}$

$$\begin{array}{ccc} (k_0^1, k_1^1) & & \Delta_B[1] \\ \dots & \rightarrow \boxed{\mathcal{F}_{\mathsf{OT}}} \leftarrow & \dots \\ (k_0^\ell, k_1^\ell) & & \Delta_B[\ell] \end{array}$$

Extend (u)

$$m_1 := \mathsf{PRF}(k_0^1, j) + \mathsf{PRF}(k_1^1, j) + u$$

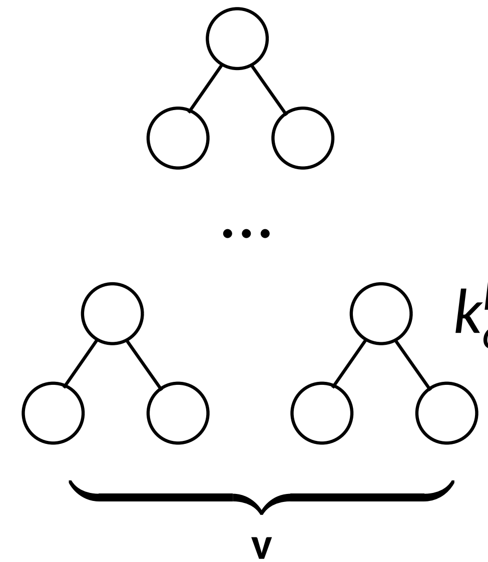
$$\dots$$

$$m_\ell := \mathsf{PRF}(k_0^\ell, j) + \mathsf{PRF}(k_1^\ell, j) + u$$

$$\longrightarrow$$

$$u\Delta_B[i] = \underbrace{\mathsf{PRF}(k_0^i, j)}_{\text{Sender}} + \underbrace{\mathsf{PRF}(k_{\Delta_B[i]}^i, j) + m_i\Delta_B[i]}_{\text{Receiver}}$$

Receiver/$\mathcal{V}$ Sender/$\mathcal{P}$



k_o^1, k_e^1

k_o^{h-1}, k_e^{h-1}

k_o^h, k_e^h

$$e \in \mathbb{F}_p^{2^h} \text{ s.t. } \begin{cases} e_\alpha = u \\ e_{\bar{\alpha}} = 0 \end{cases}$$

$\boxed{\mathcal{F}_{\mathsf{OT}}}$ $\alpha_1, \dots, \alpha_h$

$$m = \sum \mathbf{v}_i + v_u \longrightarrow$$

Use LHL to remove selective failure leackage on Δ

The Problem with LPN-based State-of-the-Art

Sender/$\mathcal{P}$ Receiver/$\mathcal{V}$

(k_0^1, k_1^1) $\rightarrow$ $\boxed{\mathcal{F}_{\mathsf{OT}}}$ $\leftarrow$ $\Delta_B[1]$

... ...

(k_0^ℓ, k_1^ℓ) $\Delta_B[\ell]$

Extend (u)

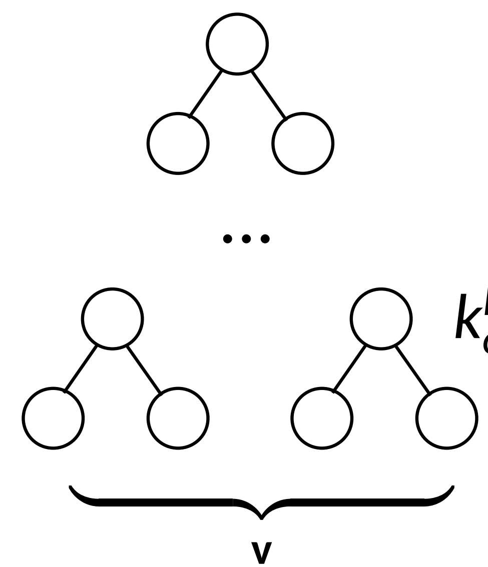
$$m_1 := \mathsf{PRF}(k_0^1, j) + \mathsf{PRF}(k_1^1, j) + u$$

...

$$m_\ell := \mathsf{PRF}(k_0^\ell, j) + \mathsf{PRF}(k_1^\ell, j) + u$$

$$u\Delta_B[i] = \underbrace{\mathsf{PRF}(k_0^i, j)}_{\text{Sender}} + \underbrace{\mathsf{PRF}(k^i_{\Delta_B[i]}, j) + m_i\Delta_B[i]}_{\text{Receiver}}$$

Receiver/$\mathcal{V}$ Sender/$\mathcal{P}$



k_o^1, k_e^1

k_o^{h-1}, k_e^{h-1}

k_o^h, k_e^h

$$e \in \mathbb{F}_p^{2^h} \text{ s.t. } \begin{cases} e_\alpha = u \\ e_{\bar{\alpha}} = 0 \end{cases}$$

$\boxed{\mathcal{F}_{\mathsf{OT}}}$ $\alpha_1, ..., \alpha_h$

$$m = \sum \mathbf{v}_i + v_u$$

Recover $\mathbf{b}_i, i \neq \alpha$

Recover $\mathbf{b}_\alpha$

Use LHL to remove selective failure leackage on Δ

The Problem with LPN-based State-of-the-Art

Sender/$\mathcal{P}$ Receiver/$\mathcal{V}$

(k_0^1, k_1^1) … (k_0^ℓ, k_1^ℓ) $\rightarrow$ $\boxed{\mathcal{F}_{\mathrm{OT}}}$ $\leftarrow$ $\Delta_B[1]$ … $\Delta_B[\ell]$

Extend (u)

$$m_1 := \mathrm{PRF}(k_0^1, j) + \mathrm{PRF}(k_1^1, j) + u$$

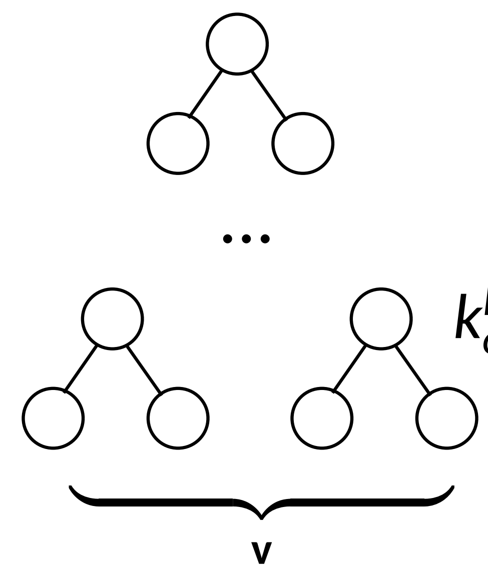
…

$$m_\ell := \mathrm{PRF}(k_0^\ell, j) + \mathrm{PRF}(k_1^\ell, j) + u$$

$$\longrightarrow$$

$$u\Delta_B[i] = \underbrace{\mathrm{PRF}(k_0^i, j)}_{\text{Sender}} + \underbrace{\mathrm{PRF}(k_{\Delta_B[i]}^i, j) + m_i\Delta_B[i]}_{\text{Receiver}}$$

Receiver/$\mathcal{V}$ Sender/$\mathcal{P}$



k_o^1, k_e^1 $\quad e \in \mathbb{F}_p^{2^h}$ s.t. $\begin{cases} e_\alpha = u \\ e_{\bar{\alpha}} = 0 \end{cases}$

$\boxed{\mathcal{F}_{\mathrm{OT}}}$ $\alpha_1, \ldots, \alpha_h$

k_o^{h-1}, k_e^{h-1}

k_o^h, k_e^h $\quad m = \sum \mathbf{v}_i + v_u \longrightarrow$

Recover $\mathbf{b}_i, i \neq \alpha$

Recover $\mathbf{b}_\alpha$

- Use Multiple $\mathcal{F}_{\mathrm{spVOLE}}$ to get sparse **e**
- Use LPN* to expand to pseudorandom **u**

Use LHL to remove selective failure leackage on Δ

Com&Open doesn't work when $\mathcal{P}$ is OT receiver

FAEST Signature

- Apply FS transform to $\Pi^t_{\text{2D-LC}}$ scheme
- $\mathsf{pk} = x, y \in \mathbb{F}_2^{128}$, $\mathsf{sk} = k \in \mathbb{F}_2^{128}$
- Relation: $y = \mathsf{Enc}_k(x)$
- For AES128, S-box is $\mathbb{F}_{2^8}$ inversion, so we can use 2D polynomial to express it

Theorem 5. *The Π_{FAEST} protocol, defined as*

$$\Pi_{\mathsf{FAEST}} = \mathsf{FS}^{H_{\mathsf{FS}}}[\mathsf{O2C}^{H_{\mathsf{O2C}}}[\Pi_{\textit{2D-Rep-OT}}]],$$

is a zero-knowledge non-interactive proof system in the CRS+RO model with knowledge error

$$2 \cdot (Q_{\mathsf{FS}} + Q_{\mathsf{Verify}}) \cdot \frac{2}{p^{r\tau}} + M \cdot (Q_{\mathsf{FS}} + Q_{\mathsf{Verify}}) \cdot \mathsf{AdvEB}^{\mathsf{VC}}_{\mathcal{A}'}[Q_{H_{\mathsf{O2C}}}] + \mathsf{AdvDist}^{\mathsf{VC.Setup},\mathsf{VC.TSetup}}_{\mathcal{D}},$$

where M is an upper bound on the number of VC *commitments sent during a run of* $\mathsf{O2C}[\Pi_{\textit{2D-Rep-OT}}]$.

Claimed Performance of FAEST

Scheme	$t_{\mathcal{P}}$ (ms)	$t_{\mathcal{V}}$ (ms)	|sign| (B)	Assumption
SDitH [FJR22b] (fast)	13.40	12.70	17 866	SD $\mathbb{F}_2$
SDitH [FJR22b] (short)	64.20	60.70	12 102	SD $\mathbb{F}_2$
SDitH [FJR22b] (fast)	6.40	5.90	12 115	SD $\mathbb{F}_{256}$
SDitH [FJR22b] (short)	29.50	27.10	8 481	SD $\mathbb{F}_{256}$
Rainier$_3$ [DKR+22]	2.96	2.92	6 176	RAIN$_3$
Rainier$_4$ [DKR+22]	3.47	3.42	6 816	RAIN$_4$
Limbo [dOT21] (fast)	2.61	2.25	23 264	Hash
Limbo [dOT21] (short)	24.51	21.82	13 316	Hash
SPHINCS+-SHA2 [HBD+22] (fast)	4.40	0.40	17 088	Hash
SPHINCS+-SHA2 [HBD+22] (short)	88.21	0.15	7 856	Hash
Falcon-512 [PFH+22]	0.11	0.02	666	Lattice
Dilithium2 [LDK+22]	0.07	0.03	2 420	Lattice
FAEST (**this work**, fast, $q = 2^8$)	2.28	2.11	6 583	Hash
FAEST (**this work**, short, $q = 2^{11}$)	11.05	10.18	5 559	Hash

Linear Combination Opening

- We can save the C-matrix communication if verifier only need to get a linear combination of the matrix B
- First P and V run Com/OT to get A, B', U'
- For a linear combination $\mathbf{r}$, P simply sends $\hat{c} := \mathbf{r}^T \cdot C \in \mathbb{F}_{2^\kappa}^{\tau}$ to the verifier
- Now the two parties can compute

$$\mathbf{r}^T \cdot B = \mathbf{r}^T \cdot A' + [0 \| \hat{c}] \cdot \mathrm{diag}(\Delta) + u \cdot [1\,1 \dots 1] \cdot \mathrm{diag}(\Delta)$$

- Perform consistency check as usual after sending $\hat{c}$

SD-in-the-Head

- An alternative approach towards Hamming weight checking
- Let S encodes the noise $S(\gamma_i) = \phi(e_i)$ for $i \in [m]$
- Let Q encodes the non-zero positions