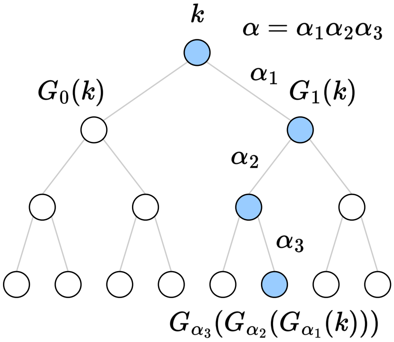
Half-Tree: Halving the Cost of Tree Expansion in COT and DPF

Xiaojie Guo[1,2] Kang Yang[1] Xiao Wang[3] Wenhao Zhang[3]
Xiang Xie[4,5] Jiang Zhang[1] Zheli Liu[2]

1 SKLC 2 NANKAI UNIVERSITY 1919 3 NORTHWESTERN UNIVERSITY QUAECUMQUE SUNT VERA 1851 4 上海期智研究院 SHANGHAI QI ZHI INSTITUTE 5 PADO

Motivation

- GGM tree is used to generate **correlated randomness** with communication **sublinear** in randomness length [SGRR19, BCG+19, BGI16, BCG+21, ...]
- However, GGM tree has no algebraic structure for efficiency improvement



Useful Correlated Randomness from GGM Tree

Correlated Randomness	Applications
Correlated OT (COT) / Subfield Vector-OLE (sVOLE)	Generic MPC [GMW87, ...], VOLE-based ZK [WYKW21, DIO21, BMRS21], PSI [GPR+21, RS21], ...
Distributed Point Function (DPF)	RAM-based MPC [Ds17], Two-server PIR [GI14, BGI16], Private heavy hitters [BBC+21], OLE extension [BCG+20], ...
Distributed Comparison Function (DCF)	Mixed-mode MPC [BGI19, BCG+21], Secure machine learning inference [GKCG22]

This Work

- More efficient COT / sVOLE / DPF / DCF protocols
- **Core idea**
  - Introducing extra correlation to GGM tree so that some nodes are summed to a **global offset**
  - If this global offset corresponds to the global key Δ of COT / sVOLE $\Rightarrow$ More efficient COT / sVOLE with global-key queries
  - If this global offset is only for internal nodes and not a part of output $\Rightarrow$ More efficient sVOLE / DPF / DCF
- Our settings
  - Semi-honest security in the UC framework [Can01]
  - Random permutation model (RPM) $\Rightarrow$ fixed-key AES
- Malicious security can be obtained by adding corresponding consistency check [YWL+20, WYKW21, BCG+20, BCG+21]

Our Results

Protocols	Asymptotic improvements		
	Computation	Communication	# Rounds
COT	$2\times$	$2\times$	–
sVOLE ver. 1	$2\times$	$1 \sim 2\times$	–
sVOLE ver. 2	$1.33\times$	$2\times$	–
DPF	$1.33\times$	$3\times$	$2\times$
DCF	$1.6\times$	$2 \sim 3\times$	$2\times$

- Computation is measured in **# AES calls** for tree expansion and does not count Learning Parity with Noise (LPN) encoding for COT / sVOLE
- This computation for tree expansion can be significant [Ds17, CRR21]

Comparison with Concurrent Work [BCG+22][1]

	Assump.	Corr.	Computation	Communication (bits)[b]	
				Sender → Receiver	Receiver → Sender
[BCG+22]	ROM	sVOLE	m RO calls	$2t(\log \frac{m}{t} - 1)\lambda + 3t \log \lvert\mathbb{K}\rvert$	$t \log \lvert\mathbb{F}\rvert$
	Ad-hoc[a]	sVOLE	m RP calls + $0.5m$ RO calls		
This work	RPM	COT	m RP calls	$t(\log \frac{m}{t} - 1)\lambda + \lambda$	–
		sVOLE	m RP calls	$t(\log \frac{m}{t} - 1) \log \lvert\mathbb{K}\rvert + \lambda$	$t(\log \frac{m}{t} + 1) \log \lvert\mathbb{F}\rvert$
		sVOLE	$1.5m$ RP calls	$t(\log \frac{m}{t} - 2)\lambda + 3t \log \lvert\mathbb{K}\rvert + \lambda$	$t \log \lvert\mathbb{F}\rvert$

[a] Based on the conjecture that the punctured result of the RPM-based UPF is unpredictable. This UPF uses GGM-style tree expansion $G(x) := \mathsf{H}_0(x) \,\|\, \mathsf{H}_1(x)$ for $\mathsf{H}_0(x) := \mathsf{H}(x) \oplus x$ and $\mathsf{H}_1(x) := \mathsf{H}(x) + x \mod 2^\lambda$.

[b] t: Hamming weight of regular LPN noise. m: Correlation length. $(\mathbb{F}, \mathbb{K})$: Base field and extension field of general sVOLE. Assume the two parties have access to random preprocessed COT / sVOLE tuples.

[1] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, Nicolas Resch, Peter Scholl: Correlated Pseudorandomness from Expand-Accumulate Codes. CRYPTO 2022.

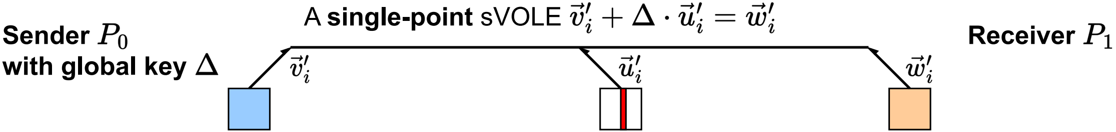
Revisiting COT / sVOLE [SGRR19, BCG+19, ...]

- sVOLE parameters: field $\mathbb{F}$ and its extension field $\mathbb{K}$
  - COT is a special case for $\mathbb{F} = \mathbb{F}_2$ and $\mathbb{K} = \mathbb{F}_{2^\lambda}$
- Correlation: $\vec{w} = \vec{v} + \vec{u} \cdot \Delta$ (with length $m > 0$)
  - Sender outputs $(\Delta, \vec{v}) \in \mathbb{K} \times \mathbb{K}^m$
  - Receiver outputs $(\vec{u}, \vec{w}) \in \mathbb{F}^m \times \mathbb{K}^m$
- Blueprint: $\underbrace{\textbf{single-point sVOLE}}_{\vec{u} \text{ has Hamming weight } 1}$ + LPN encoding = sVOLE

Revisiting COT / sVOLE [SGRR19, BCG+19, ...] (cont.)

- Example: regular LPN noise $\vec{e}$ + dual-LPN assumption [BCG+19]

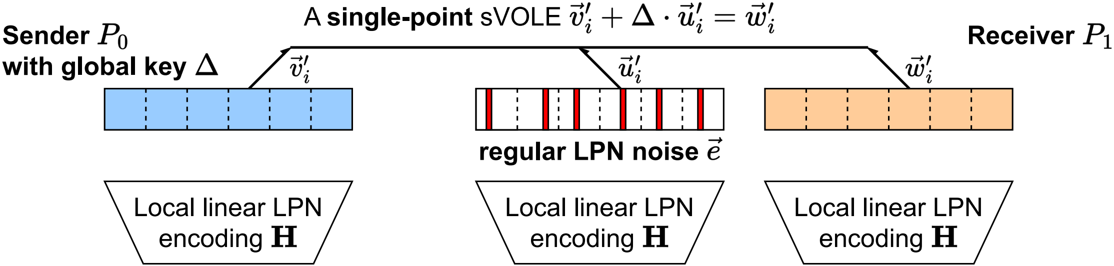
  (dual-LPN assumption: Given public $\mathbf{H}$, $\vec{e} \cdot \mathbf{H}$ is pseudorandom)



Revisiting COT / sVOLE [SGRR19, BCG⁺19, ...] (cont.)

- Example: regular LPN noise $\vec{e}$ + dual-LPN assumption [BCG⁺19]

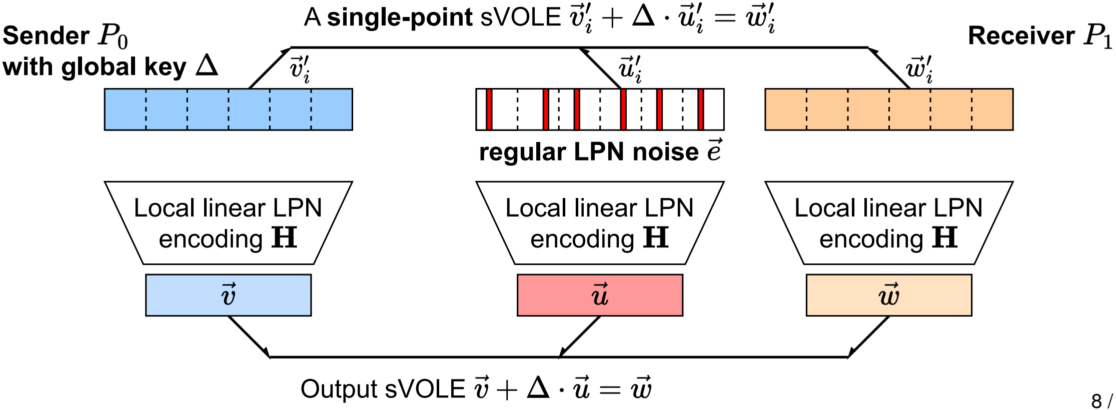
  Given public $\mathbf{H}$, $\vec{e} \cdot \mathbf{H}$ is pseudorandom



Revisiting COT / sVOLE [SGRR19, BCG+19, ...] (cont.)

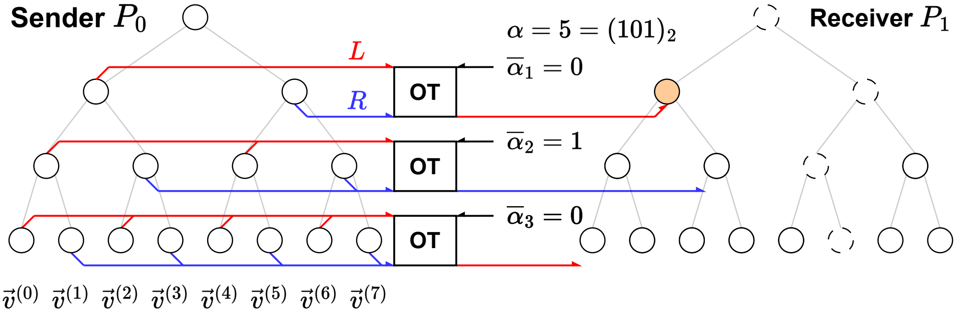
- Example: regular LPN noise $\vec{e}$ + dual-LPN assumption [BCG+19]

  Given public $\mathbf{H}$, $\vec{e} \cdot \mathbf{H}$ is pseudorandom



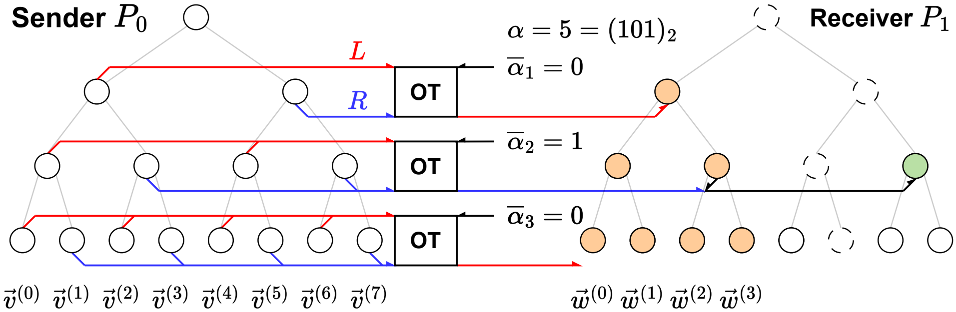
Revisiting COT / sVOLE [SGRR19, BCG+19, ...] (cont.)

- How to set up a **single-point** COT / sVOLE?



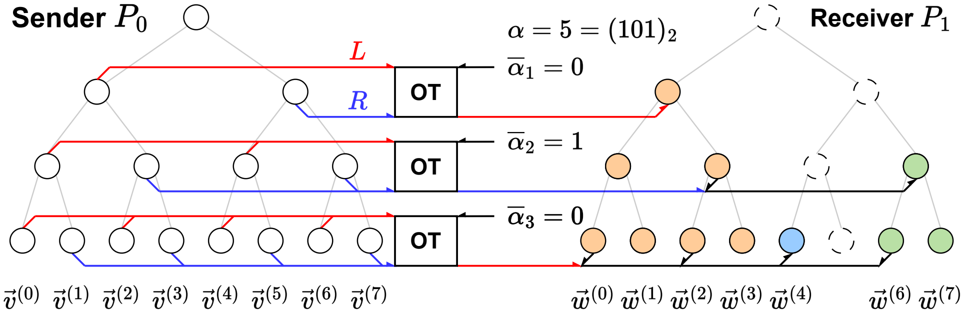
Revisiting COT / sVOLE [SGRR19, BCG+19, ...] (cont.)

- How to set up a **single-point** COT / sVOLE?



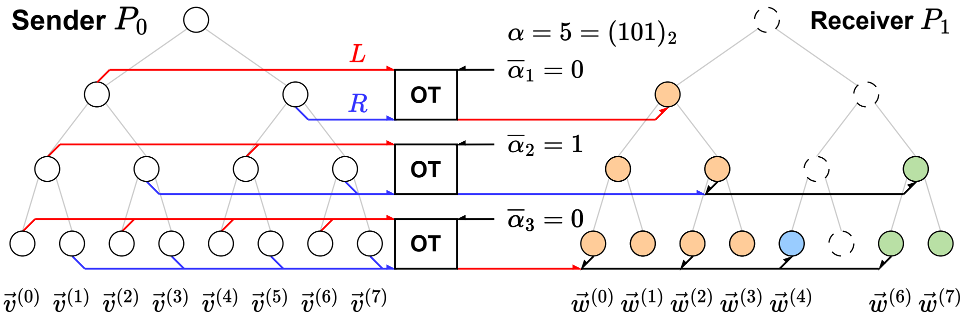
Revisiting COT / sVOLE [SGRR19, BCG+19, ...] (cont.)

- How to set up a **single-point** COT / sVOLE?



Revisiting COT / sVOLE [SGRR19, BCG+19, ...] (cont.)

- How to set up a **single-point** COT / sVOLE?

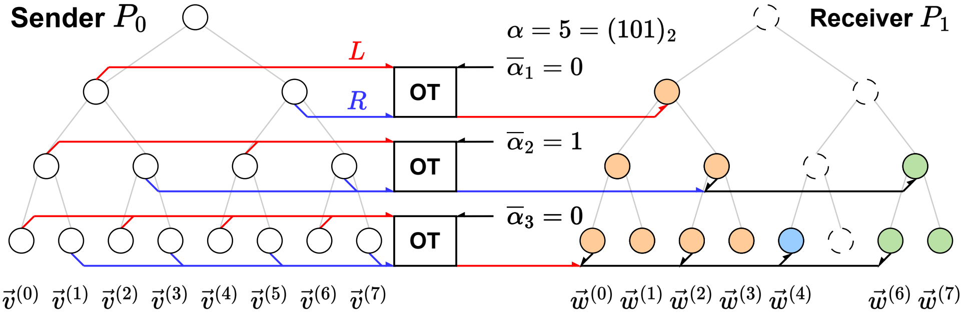


In single-point COT

$$\psi := \Delta \oplus (\oplus_{i\in[0,8)} \vec{v}^{(i)}) \longrightarrow \vec{w}^{(5)} := \psi \oplus (\oplus_{i\neq 5} \vec{w}^{(i)})$$

Revisiting COT / sVOLE [SGRR19, BCG+19, ...] (cont.)

- How to set up a **single-point** COT / sVOLE?

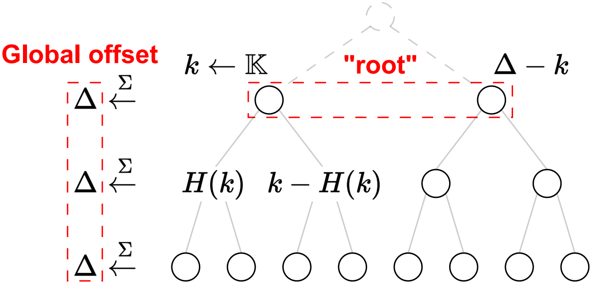


In single-point sVOLE

$\Delta, \mathsf{K}[\beta]$ ← One preprocessed (random) sVOLE $\mathsf{M}[\beta] = \mathsf{K}[\beta] + \beta \cdot \Delta$ → $\beta, \mathsf{M}[\beta]$

$\psi := -\mathsf{K}[\beta] + (\sum_{i \in [0,8)} \vec{v}^{(i)})$ ⟶ $\vec{w}^{(5)} := \mathsf{M}[\beta] + \psi - (\sum_{i \neq 5} \vec{w}^{(i)})$

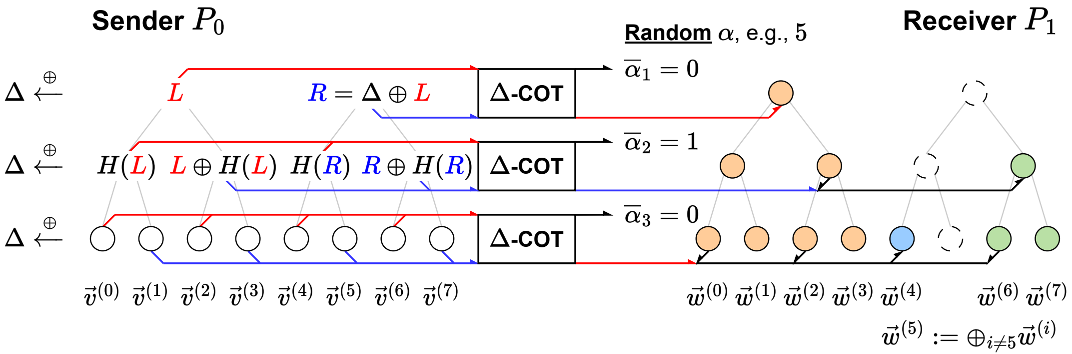
Correlated GGM (cGGM) Tree



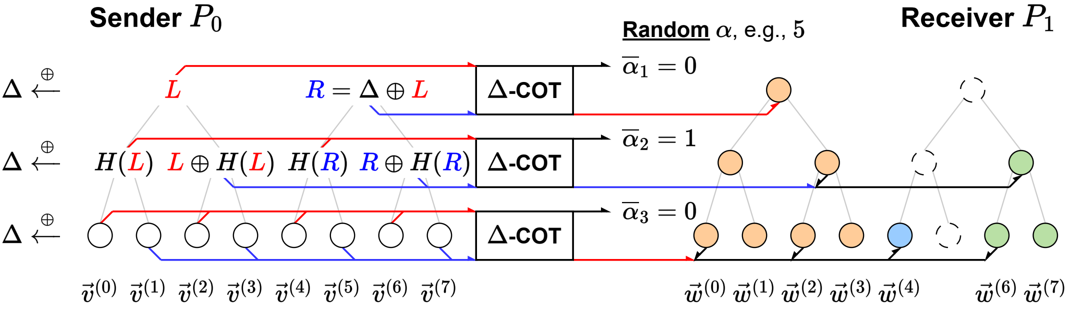
Hash function $H(x) := \pi(\sigma(x)) + \sigma(x)$ [GKWY20]

- $\pi : \mathbb{K} \to \mathbb{K}$ is modeled as random permutation
- $\sigma : \mathbb{K} \to \mathbb{K}$ is an efficiently computable **linear orthomorphism**
  - σ and $\sigma' : x \mapsto \sigma(x) - x$ are permutations, $\sigma(x + y) = \sigma(x) + \sigma(y)$
  - Candidates in [GKWY20]: ① $\sigma(x) := c \cdot x$, $c \in \mathbb{K} \setminus \{0, 1\}$, ② if $\mathbb{K} = \mathbb{F}_{2^{2n}}$, $\sigma(x) = \sigma(x_L \parallel x_R) := (x_L \oplus x_R) \parallel x_L$

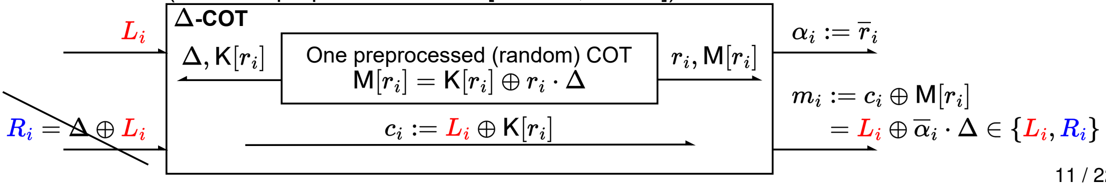
Single-point COT from cGGM Tree



Single-point COT from cGGM Tree



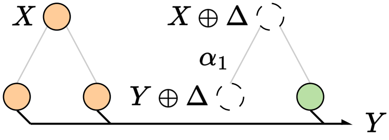
Instead of 2λ-bit OT messages
(also with preprocessed COT [IKNP03, Bea95])



Security of cGGM-based Single-point COT

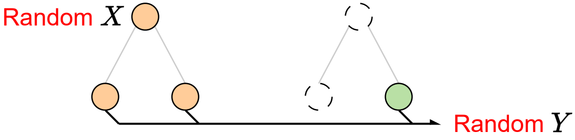
- Straightforward for corrupted sender
- Corrupted receiver: environment learns Δ from the honest sender's output
  - E.g., for the first two levels

Real world



$$H(X \oplus \Delta) \oplus \alpha_1 \cdot (X \oplus \Delta) = Y \oplus \Delta$$

Ideal world

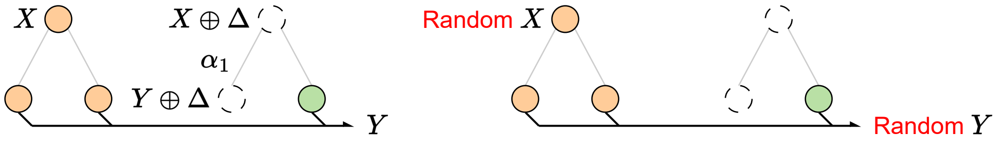


$$H(X \oplus \Delta) \oplus \alpha_1 \cdot (X \oplus \Delta) \neq Y \oplus \Delta$$

Security of cGGM-based Single-point COT

- Straightforward for corrupted sender
- Corrupted receiver: environment learns Δ from the honest sender's output
  - E.g., for the first two levels

Real world **Ideal world**



$$H(X \oplus \Delta) \oplus \alpha_1 \cdot (X \oplus \Delta) = Y \oplus \Delta$$

$$H(X \oplus \Delta) \oplus \alpha_1 \cdot (X \oplus \Delta) \neq Y \oplus \Delta$$

$\longrightarrow \pi(\sigma(X \oplus \Delta)) \oplus \sigma(X \oplus \Delta)$

① Relax single-point COT functionality to allow **guesses on Δ**
② Sim can extract every possible Δ from queries to $\pi^{\pm 1}$, and guess each extracted Δ
③ Sim programs $\pi^{\pm 1}$ on the correct Δ in the **ideal** world

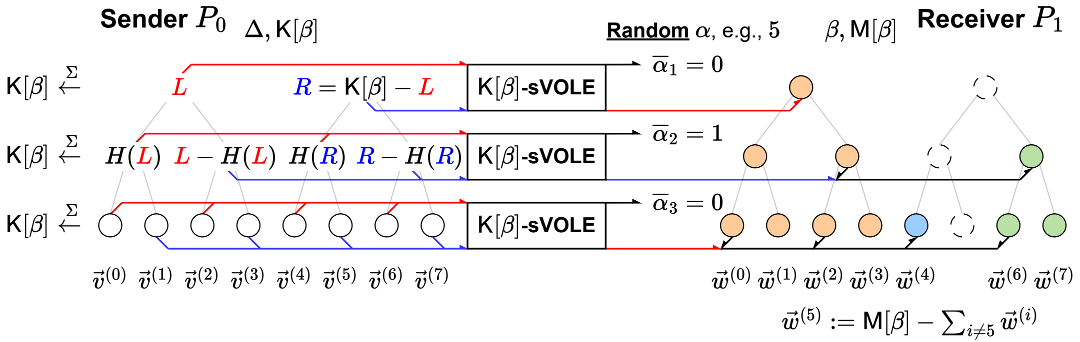
Extension: Single-point sVOLE from cGGM Tree

Sender P_0 **Receiver P_1**

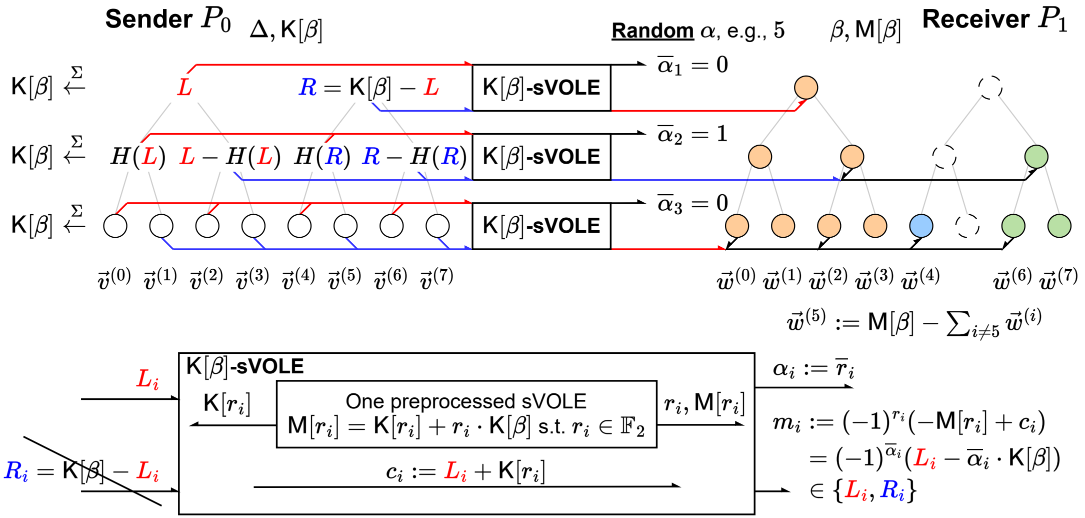
$\Delta, \mathsf{K}[\beta]$ ← One preprocessed (random) sVOLE → $\beta, \mathsf{M}[\beta]$

$\mathsf{M}[\beta] = \mathsf{K}[\beta] + \beta \cdot \Delta$

Extension: Single-point sVOLE from cGGM Tree



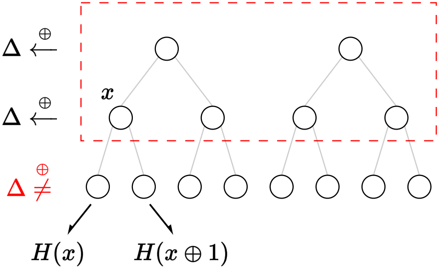
Extension: Single-point sVOLE from cGGM Tree



Single-point sVOLE from Pseudorandom cGGM (pcGGM) Tree

- Using single-point sVOLE blueprint [SGRR19, BCG+19, ...]
  - Pseudorandom off-path nodes & the punctured leaf are required

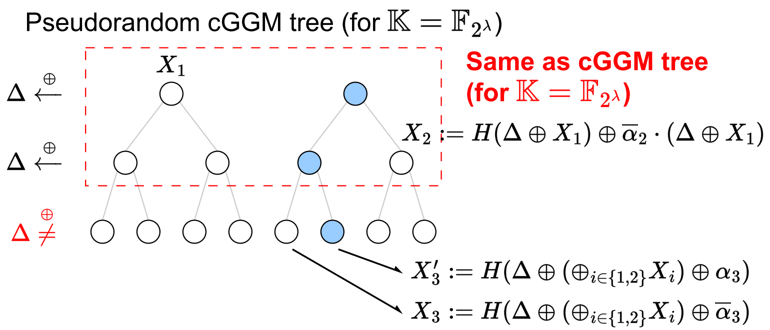
Pseudorandom cGGM tree (for $\mathbb{K} = \mathbb{F}_{2^\lambda}$)



Same as cGGM tree (for $\mathbb{K} = \mathbb{F}_{2^\lambda}$)

Single-point sVOLE from Pseudorandom cGGM (pcGGM) Tree

- Using single-point sVOLE blueprint [SGRR19, BCG+19, ...]
  - Pseudorandom off-path nodes & the punctured leaf are required



X_2, X_3, X_3' have **Circular Correlation Robustness** (CCR) form

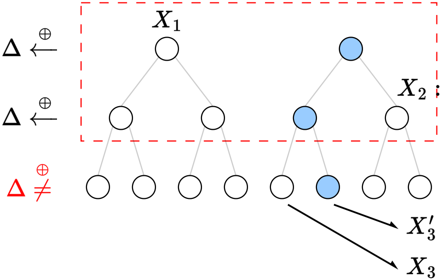
$H(\Delta \oplus x) \oplus b \cdot \Delta$ is pseudorandom

Single-point sVOLE from Pseudorandom cGGM (pcGGM) Tree

- Using single-point sVOLE blueprint [SGRR19, BCG+19, ...]
  - Pseudorandom off-path nodes & the punctured leaf are required

Pseudorandom cGGM tree (for $\mathbb{K} = \mathbb{F}_{2^\lambda}$)

Global offset, but **NOT** global key of single-point sVOLE (i.e., hidden from environment)



Same as cGGM tree (for $\mathbb{K} = \mathbb{F}_{2^\lambda}$)

$X_2 := H(\Delta \oplus X_1) \oplus \overline{\alpha}_2 \cdot (\Delta \oplus X_1)$

$X_3' := H(\Delta \oplus (\oplus_{i\in\{1,2\}} X_i) \oplus \alpha_3)$

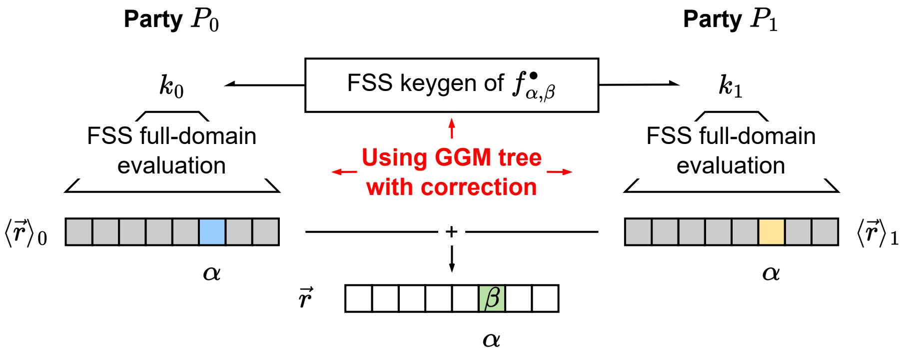
$X_3 := H(\Delta \oplus (\oplus_{i\in\{1,2\}} X_i) \oplus \overline{\alpha}_3)$

X_2, X_3, X_3' have **Circular Correlation Robustness** (CCR) form

$\underbrace{}_{H(\Delta \oplus x) \oplus b \cdot \Delta \text{ is pseudorandom}}$

Revisiting DPF & Its Protocol [BGI16, Ds17]

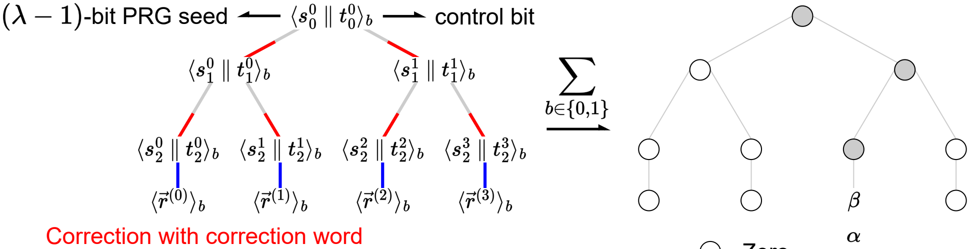
- Point function $f^{\bullet}_{\alpha,\beta}(x) := \begin{cases} \beta, & x = \alpha \\ 0, & x \neq \alpha \end{cases}$ with domain $\{0,1\}^n$ and range $\mathbb{G}$
- Distributed Point Function: Function Secret Sharing (FSS) of $f^{\bullet}_{\alpha,\beta}(x)$



- FSS keygen protocol is based on 2PC and the technique [Ds17]

Revisiting DPF & Its Protocol [BGI16, Ds17] (cont.)

E.g., $n = 2$, Party P_b ($b \in \{0,1\}$) with $k_b = (\langle s_0^0 \parallel t_0^0 \rangle_b, \mathsf{CW}_1, \ldots, \mathsf{CW}_{n+1})$



Zero

Pseudorandom conditioned on LSB = 1

Correction with correction word

$\mathsf{CW}_i = (\mathsf{HCW}_i, \mathsf{LCW}_i^0, \mathsf{LCW}_i^1)$ $(1 \leq i \leq n)$

$$\langle s_i^{2j} \parallel t_i^{2j} \rangle_b = G_0(\langle s_{i-1}^j \rangle_b) \oplus \langle t_{i-1}^j \rangle_b \cdot (\mathsf{HCW}_i \parallel \mathsf{LCW}_i^0)$$

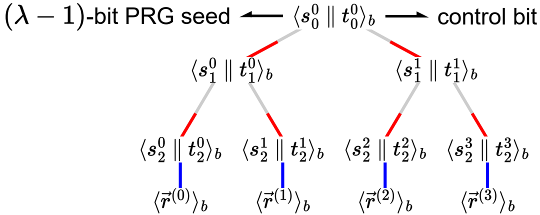
$$\langle s_i^{2j+1} \parallel t_i^{2j+1} \rangle_b = G_1(\langle s_{i-1}^j \rangle_b) \oplus \langle t_{i-1}^j \rangle_b \cdot (\mathsf{HCW}_i \parallel \mathsf{LCW}_i^1)$$

Output correction with correction word $\mathsf{CW}_{n+1} \in \mathbb{G}$

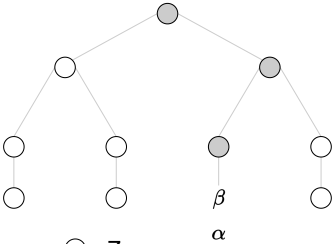
$$\langle \vec{r}^{(j)} \rangle_b = (-1)^b \cdot (\mathsf{Convert}_{\mathbb{G}}(\langle s_n^j \rangle_b) + \langle t_n^j \rangle_b \cdot \mathsf{CW}_{n+1})$$

Revisiting DPF & Its Protocol [BGI16, Ds17] (cont.)

E.g., $n = 2$, **Party** P_b $(b \in \{0,1\})$ **with** $k_b = (\langle s_0^0 \parallel t_0^0 \rangle_b, \mathsf{CW}_1, \ldots, \mathsf{CW}_{n+1})$



$\sum_{b\in\{0,1\}}$



Zero

Pseudorandom conditioned on LSB = 1

$\Downarrow$

Correction with correction word

$\mathsf{CW}_i = (\mathsf{HCW}_i, \mathsf{LCW}_i^0, \mathsf{LCW}_i^1)$ $(1 \leq i \leq n)$

$$\langle s_i^{2j} \parallel t_i^{2j} \rangle_b = G_0(\langle s_{i-1}^j \rangle_b) \oplus \langle t_{i-1}^j \rangle_b \cdot (\mathsf{HCW}_i \parallel \mathsf{LCW}_i^0)$$

$$\langle s_i^{2j+1} \parallel t_i^{2j+1} \rangle_b = G_1(\langle s_{i-1}^j \rangle_b) \oplus \langle t_{i-1}^j \rangle_b \cdot (\mathsf{HCW}_i \parallel \mathsf{LCW}_i^1)$$

HCW_i = upper $\lambda - 1$ bits of

$$G_{\overline{\alpha_i}}(\langle s_{i-1}^{\alpha_1 \ldots \alpha_{i-1}} \rangle_0) \oplus G_{\overline{\alpha_i}}(\langle s_{i-1}^{\alpha_1 \ldots \alpha_{i-1}} \rangle_1)$$

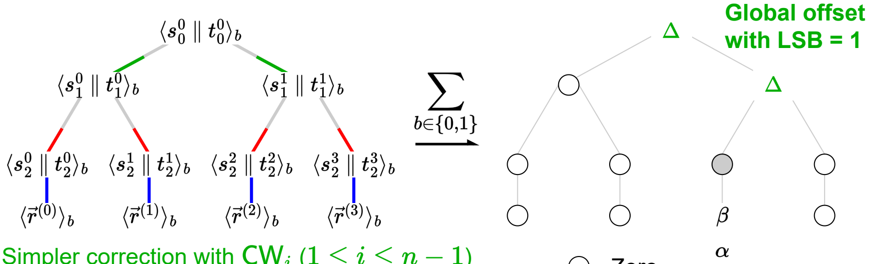
Output correction with correction word $\mathsf{CW}_{n+1} \in \mathbb{G}$

$$\langle \vec{r}^{(j)} \rangle_b = (-1)^b \cdot (\mathsf{Convert}_{\mathbb{G}}(\langle s_n^j \rangle_b) + \langle t_n^j \rangle_b \cdot \mathsf{CW}_{n+1})$$

Require **2-round** OT-based 2PC in distributed keygen protcol [Ds17] (where each α_i is XOR-shared)

Using pcGGM-style Technique in DPF & Its Protocol

E.g., $n = 2$, **Party** P_b $(b \in \{0,1\})$ **with** $k_b = (\langle s_0^0 \parallel t_0^0 \rangle_b, \mathsf{CW}_1, \ldots, \mathsf{CW}_{n+1})$



Simpler correction with CW_i $(1 \leq i \leq n-1)$

$$\langle s_i^{2j} \parallel t_i^{2j} \rangle_b = H(\langle s_{i-1}^j \parallel t_{i-1}^j \rangle_b) \oplus \langle t_{i-1}^j \rangle_b \cdot \mathsf{CW}_i$$

$$\langle s_i^{2j+1} \parallel t_i^{2j+1} \rangle_b = H(\langle s_{i-1}^j \parallel t_{i-1}^j \rangle_b) \oplus \langle s_{i-1}^j \parallel t_{i-1}^j \rangle_b \oplus \langle t_{i-1}^j \rangle_b \cdot \mathsf{CW}_i$$

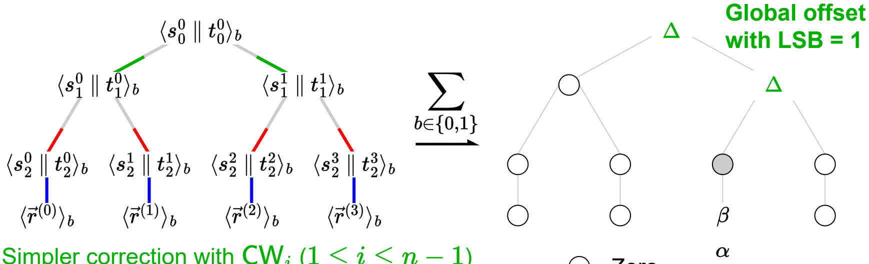
cGGM / pcGGM-style
tree expansion
for the first $n-1$ levels

○ Zero

● Pseudorandom conditioned on LSB = 1

Using pcGGM-style Technique in DPF & Its Protocol

E.g., $n = 2$, **Party** P_b $(b \in \{0,1\})$ **with** $k_b = (\langle s_0^0 \parallel t_0^0 \rangle_b, \mathsf{CW}_1, \ldots, \mathsf{CW}_{n+1})$



Simpler correction with CW_i $(1 \leq i \leq n-1)$

Zero

Pseudorandom conditioned on LSB = 1

$$\langle s_i^{2j} \parallel t_i^{2j} \rangle_b = H(\langle s_{i-1}^j \parallel t_{i-1}^j \rangle_b) \oplus \langle t_{i-1}^j \rangle_b \cdot \mathsf{CW}_i$$

$$\langle s_i^{2j+1} \parallel t_i^{2j+1} \rangle_b = H(\langle s_{i-1}^j \parallel t_{i-1}^j \rangle_b) \oplus \langle s_{i-1}^j \parallel t_{i-1}^j \rangle_b \oplus \langle t_{i-1}^j \rangle_b \cdot \mathsf{CW}_i$$

cGGM / pcGGM-style tree expansion for the first $n-1$ levels

Can be shared **in parallel** for all XOR-shared α_i's

$$\mathsf{CW}_i = H(\langle s_{i-1}^{\alpha_1 \ldots \alpha_{i-1}} \parallel t_{i-1}^{\alpha_1 \ldots \alpha_{i-1}} \rangle_0) \oplus H(\langle s_{i-1}^{\alpha_1 \ldots \alpha_{i-1}} \parallel t_{i-1}^{\alpha_1 \ldots \alpha_{i-1}} \rangle_1) \oplus \overline{\alpha_i} \cdot \Delta$$

Locally shared by summing all previous-level hashes [Ds17]

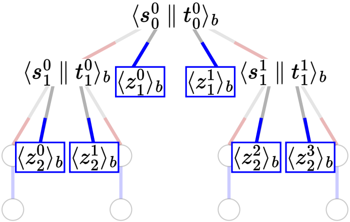
⇨ CW_i $(1 \leq i \leq n-1)$ can be computed in (amortized) **one round**, and has **CCR** form

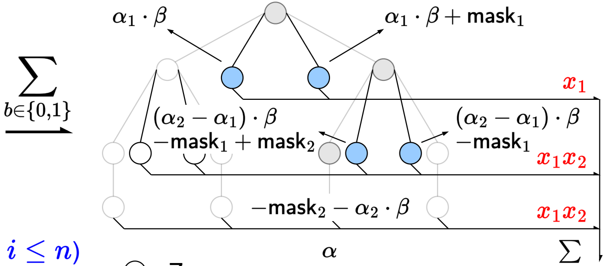
Revisiting DCF & Its Protocol [BCG+21]

- Comparison function $f^{<}_{\alpha,\beta}(x) := \begin{cases} \beta, x < \alpha \\ 0, x \geq \alpha \end{cases}$ with domain $\{0,1\}^n$ and range $\mathbb{G}$
- Distributed Comparison Function: Function Secret Sharing (FSS) of $f^{<}_{\alpha,\beta}(x)$
- $f^{<}_{\alpha,\beta}(x) = f^{\bullet}_{\alpha,-\alpha_n\cdot\beta}(x) + \alpha_{h+1}\cdot\beta$, where $h \in [0,n]$ corresponds to the **longest common prefix** $\alpha_1 \ldots \alpha_h = x_1 \ldots x_h$, and $\alpha_{n+1} := \alpha_n$
  - This common prefix is implicitly computed in DPF for $f^{\bullet}_{\alpha,-\alpha_n\cdot\beta}(x)$
  - $f^{\bullet}_{\alpha,-\alpha_n\cdot\beta}(x)$ and $\alpha_{h+1}\cdot\beta$ can be computed at the same time using extended tree structure

Revisiting DCF & Its Protocol [BCG+21] (cont.)

E.g., $n = 2$**, Party** P_b $(b \in \{0,1\})$ **with** $k_b = (\langle s_0^0 \parallel t_0^0 \rangle_b, \mathsf{CW}_1, \ldots, \mathsf{CW}_{n+1}, \mathsf{VCW}_1, \ldots, \mathsf{VCW}_n)$





Per-level correction with $\mathsf{VCW}_i \in \mathbb{G}$ $(1 \le i \le n)$

$$\langle z_i^{2j} \rangle_b = (-1)^b \cdot (G_0^{\mathbb{G}}(\langle s_{i-1}^j \rangle_b) + \langle t_{i-1}^j \rangle_b \cdot \mathsf{VCW}_i) \in \mathbb{G}$$

$$\langle z_i^{2j+1} \rangle_b = (-1)^b \cdot (G_1^{\mathbb{G}}(\langle s_{i-1}^j \rangle_b) + \langle t_{i-1}^j \rangle_b \cdot \mathsf{VCW}_i) \in \mathbb{G}$$

- Zero
- Non-zero
- Pseudorandom conditioned on LSB = 1

$f^{\bullet}_{\alpha, -\alpha_n \cdot \beta}(x) + \alpha_{h+1} \cdot \beta$

⇩

$\mathsf{VCW}_i \in \mathbb{G}$ depends on $G^{\mathbb{G}}_{\overline{\alpha_i}}(\langle s_{i-1}^{\alpha_1 \ldots \alpha_{i-1}} \rangle_1) - G^{\mathbb{G}}_{\overline{\alpha_i}}(\langle s_{i-1}^{\alpha_1 \ldots \alpha_{i-1}} \rangle_0)$

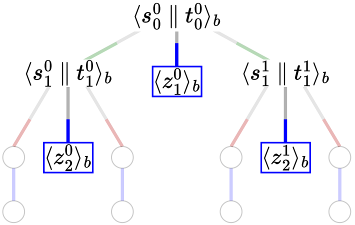
Also require **2-round** OT-based 2PC in distributed keygen protocol

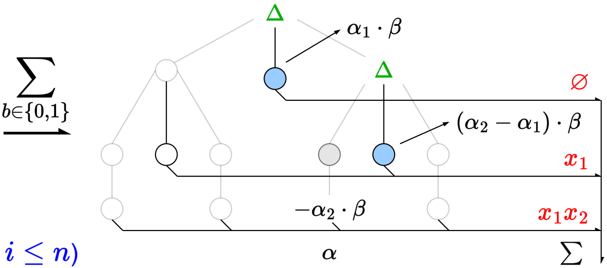
Optimized DCF & Its Protocol

- First optimization: using our optimized DPF & its protocol for DPF part
- Second optimization: simpler correction for DCF part

Optimized DCF & Its Protocol: Second Optimization

E.g., $n = 2$, Party P_b ($b \in \{0,1\}$) with $k_b = (\langle s_0^0 \parallel t_0^0 \rangle_b, \mathsf{CW}_1, \ldots, \mathsf{CW}_{n+1}, \mathsf{VCW}_1, \ldots, \mathsf{VCW}_n)$





Per-level correction with $\mathsf{VCW}_i \in \mathbb{G}$ $(1 \le i \le n)$

$$\langle z_i^j \rangle_b = (-1)^b \cdot (\mathsf{Convert}_{\mathbb{G}}(H(\langle s_{i-1}^j \parallel t_{i-1}^j \rangle_b \oplus 2)) + \langle t_{i-1}^j \rangle_b \cdot \mathsf{VCW}_i) \in \mathbb{G}$$

$f^{\bullet}_{\alpha, -\alpha_n \cdot \beta}(x) + \alpha_{h+1} \cdot \beta$

$\Downarrow$

Also in Δ CCR form

$\mathsf{VCW}_i \in \mathbb{G}$ $(1 \le i \le n)$ depends on

$$\mathsf{Convert}_{\mathbb{G}}(H(\langle s_{i-1}^{\alpha_1 \ldots \alpha_{i-1}} \parallel t_{i-1}^{\alpha_1 \ldots \alpha_{i-1}} \rangle_1 \oplus 2)) - \mathsf{Convert}_{\mathbb{G}}(H(\langle s_{i-1}^{\alpha_1 \ldots \alpha_{i-1}} \parallel t_{i-1}^{\alpha_1 \ldots \alpha_{i-1}} \rangle_0 \oplus 2))$$

Locally shared by summing all previous-level hashes [Ds17]

Thank You

Full version: eprint.iacr.org/2022/1431