SoftSpokenOT: Communication–Computation Tradeoffs in OT Extension

Crypto 2022 · `https://ia.cr/2022/192`

Lawrence Roy

July 3, 2022· Presented by Hongrui Cui

* Some acknowledgments?

Introduction

- Improving IKNP-style OTe
- Advantages: Minicrypt construction

$$\begin{tikzcd}
\text{Base OT} \arrow[r, bend left, "\substack{\text{[IKNP,KOS]}\\ \text{[KK,OOS]}\\ \textbf{This Work}}"] & \text{OT Ext} \arrow[r, bend left, "\text{Ferret, Silver}"] & \text{OT PCG}
\end{tikzcd}$$

# of OTs	small (~ 100)	medium ($\sim 1,000$)	large ($\sim 1,000,000$)
Assumptions	LWE/DDH	PRG/CRHF	LPN*
Advantages	None	Light Comp Heavy Comm	Heavy Comp Light Comm

Performance

Protocol	Semi-honest Security Communication KB	bits/OT	Time (ms) localhost	LAN	WAN	Malicious Security Time (ms) localhost	LAN	WAN
IKNP [IKNP03] / KOS [KOS15]	160010	128	391	1725	15525	443	1802	15662
SoftSpoken ($k = 1$)	160009	128	243	1590	15420	<u>298</u>	1637	15648
SoftSpoken ($k = 2$)	80009	64	**210**	815	7730	**255**	893	7985
SoftSpoken ($k = 3$)	53759	43	<u>223</u>	568	5208	322	677	5419
SoftSpoken ($k = 4$)	40008	32	261	<u>433</u>	3995	311	<u>530</u>	4114
SoftSpoken ($k = 5$)	32510	26	337	**348**	3271	454	**465**	3447
SoftSpoken ($k = 6$)	27509	22	471	488	2811	588	613	2985
SoftSpoken ($k = 7$)	23760	19	777	843	2380	899	966	<u>2554</u>
SoftSpoken ($k = 8$)	20008	16	1259	1314	<u>1916</u>	1293	1322	**2130**
SoftSpoken ($k = 9$)	18759	15	2302	2338	2439	2460	2457	2590
SoftSpoken ($k = 10$)	16259	13	3984	3983	4097	4126	4132	4223
Ferret [YWL+20]	2976	2.38	2156	2160	2825	2240	2242	3108
Silent (Quasi-cyclic) [BCG+19a]	**127**	**0.10**	7735	7736	8049			
Silent (Silver, weight 5) [CRR21]	<u>127</u>	<u>0.10</u>	613	613	**746**			

Table 1: Time and communication required to generate 10^7 OTs, averaged over 50 runs. The best entry in each column is **bolded**, and the second best is <u>underlined</u>. Communication costs for maliciously secure versions are within 10 KB of the semi-honest ones. The setup costs are included.

Main Techniques

- Revisiting IKNP

P_A | P_B

$\mathbf{m}_0, \mathbf{m}_1 \longleftarrow \boxed{\mathcal{F}_{\mathrm{OT}}} \longrightarrow b \in \{0,1\}, \mathbf{m}_b$

$\mathbf{u} := \mathbf{m}_0 \oplus \mathbf{m}_1$ | $\Delta := 1 \oplus b$

$\mathbf{v} := \mathbf{m}_1$ | $\mathbf{w} := \mathbf{m}_b$

$$\boxed{\mathbf{v}} = \boxed{\mathbf{w}} - \boxed{\mathbf{u}} \times \Delta$$

Repeat for κ times

$$\boxed{\mathrm{V}} = \boxed{\mathrm{W'}} - \boxed{\mathrm{U'}} \times \boxed{\begin{matrix}\Delta_1 & & \\ & \ddots & \\ & & \Delta_\kappa\end{matrix}}$$

κ-bit entropy

$$\xrightarrow{C := U \oplus U'}$$

Outputs U, V $\quad W := W' \oplus C \cdot \mathrm{diag}(\Delta)$

- Main overhead: sending C

$|C| = \#\mathrm{OT} * \kappa$

Revisiting IKNP

- Hash Correlated-OT to Random-OT

$$\boxed{V} = \boxed{W} - \boxed{\mathbf{u}} \times \boxed{\mathrm{Rep}(\kappa)} \times \begin{bmatrix} \Delta_1 & & \\ & \ddots & \\ & & \Delta_\kappa \end{bmatrix}$$

$$y_0 := H(W - 0 \cdot \vec{\Delta})$$
$$y_1 := H(W - 1 \cdot \vec{\Delta})$$

$$y_u := H(W - u \cdot \vec{\Delta})$$

Recall $\mathbf{u} := \mathbf{m}_0 \oplus \mathbf{m}_1$

- Sender's Security: H-preimage κ-hamming distance

- Receiver's Security: PRG and Base-OT security

Reducing Derandomization Complexity using PPRF

P_A $\qquad\qquad$ P_B

$F : \mathbb{F}_{2^k} \to \{0,1\}^\kappa \longleftarrow \boxed{\Pi_{\mathsf{PPRF}}} \longrightarrow x^* \in \mathbb{F}_{2^k}, F^* : F \setminus \{x^*\}$

$PRG : \mathbb{F}_2^\kappa \to \mathbb{F}_2^\ell$

$\mathbf{u} := \sum_x PRG(F(x))$

$\mathbf{v} := \sum_x -x \cdot PRG(F(x))$

$\Delta := x^*$

$\mathbf{w} := \sum_x (\Delta - x) PRG(F(x))$

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

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

$$\boxed{\mathrm{V}} = \boxed{\mathrm{W'}} - \boxed{\mathrm{U'}} \times \begin{bmatrix} \Delta_1 & & \\ & \ddots & \\ & & \Delta_{\kappa/k} \end{bmatrix}$$

κ-bit entropy

Step 1: From Base-OT to $\mathbb{F}_{2^k}$-VOLE

- Use punctured PRF to get $(2^k - 1)$-out-of-2^k OT

P_A $\quad x^* := 010 \quad$ P_B

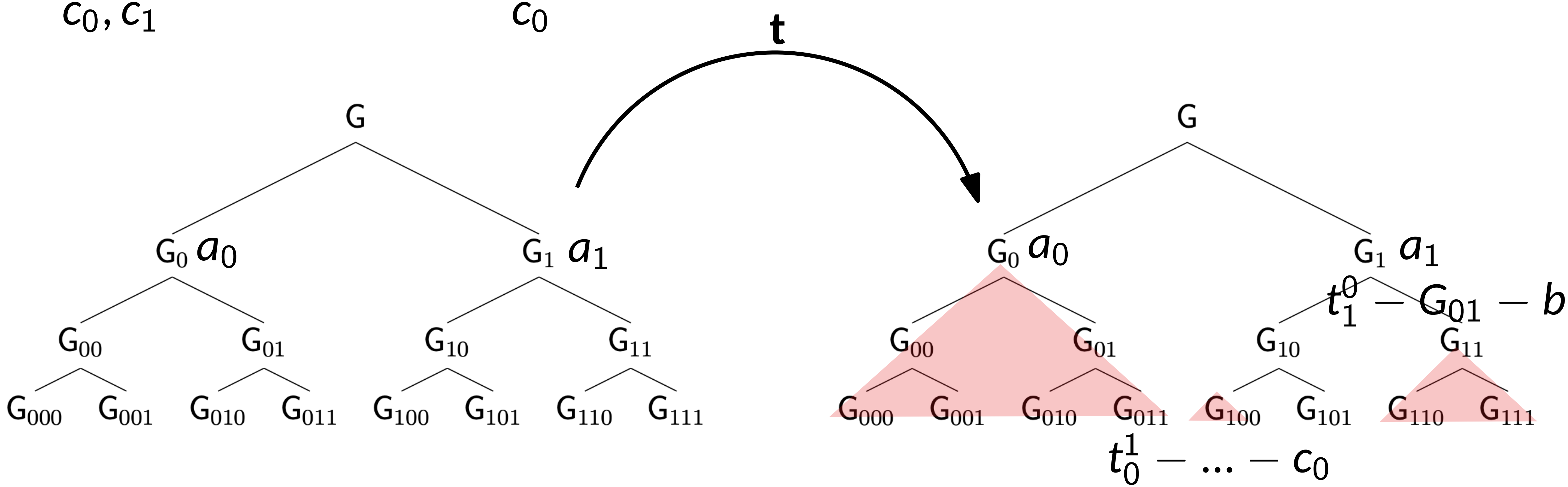
a_0, a_1 $\quad$ $3 \times \mathcal{F}_{\mathrm{OT}}$ $\quad$ a_0

b_0, b_1 $\quad\quad$ b_1

c_0, c_1 $\quad\quad$ c_0

$$t_0^0 := G_{00} + G_{10} + b_0, t_1^0 := G_{01} + G_{11} + b_1$$

$$t_0^1 := G_{000} + \ldots + G_{110} + c_0$$



$F : \mathbb{F}_{2^3} \to \mathbb{F}_2^\kappa$ $\quad\quad$ $F^* : \mathbb{F}_{2^3} \setminus x^* \to \mathbb{F}_2^\kappa$

Step 1: Consistency Checks

- Secure against malicious P_B
- Malicious P_A may launch selective failure attack.

$F, L \in \mathcal{L}$

$\mathcal{F}_{\mathrm{PPRF}}$

$x^* \leftarrow \mathbb{F}_{2^k}$

abort if $x^* \notin L$

$F^* := F \setminus \{x^*\}$

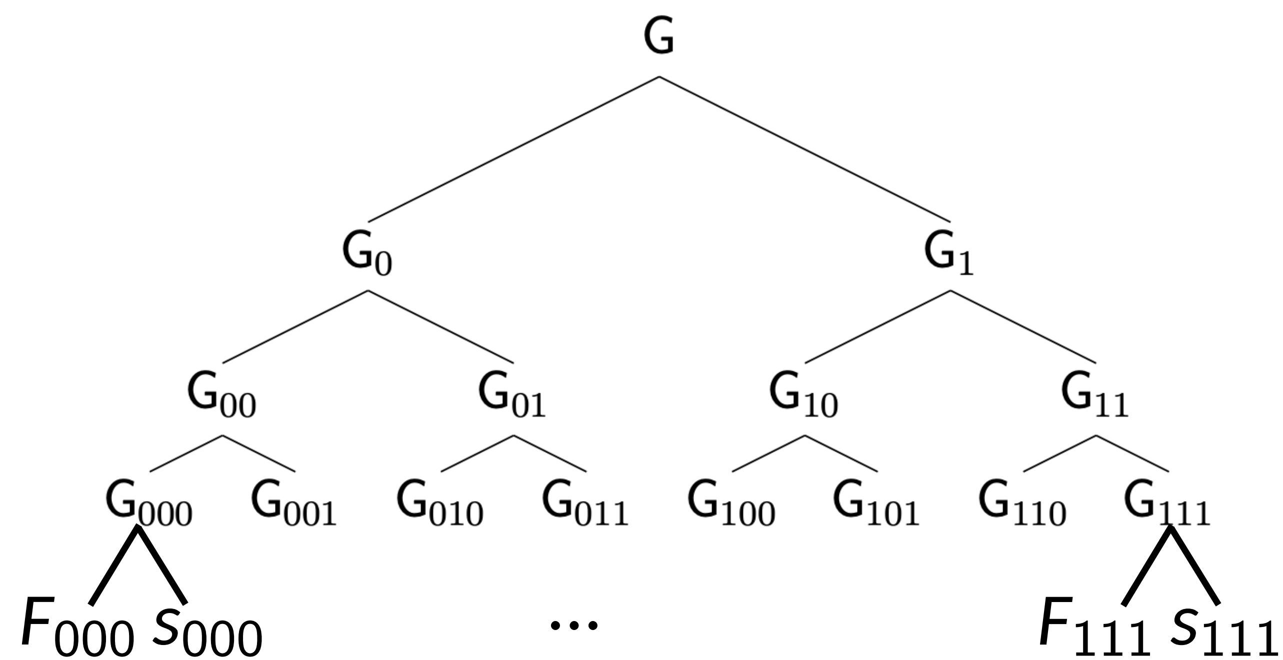
P_A $\qquad$ P_B

$s := \sum_x s_x$

$\tau := H(F_{000}, ..., F_{111})$

Recovers s_{x^*} from s

Checks $\tau := H(F^*_{000}, ..., F^*_{111})$



Simulator can extract F, L from $\mathbf{t}, \tau, s$

Step 1: Building Small Field VOLE

P_A $\quad\quad$ P_B

$$F : \mathbb{F}_{2^k} \to \{0,1\}^{\kappa} \longleftarrow \boxed{\Pi_{\mathsf{PPRF}}} \longrightarrow x^* \in \mathbb{F}_{2^k}, F^* : F \setminus \{x^*\}$$

$$PRG : \mathbb{F}_2^{\kappa} \to \mathbb{F}_2^{\ell}$$

$$\mathbf{u} := \sum_x PRG(F(x))$$
$$\mathbf{v} := \sum_x -x \cdot PRG(F(x))$$

$$\Delta := x^*$$
$$\mathbf{w} := \sum_x (\Delta - x) PRG(F(x))$$

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

- PRG ensures privacy of $\mathbf{u}$
- Notice $PRG(F(x^*))$ is cancelled out in $\mathbf{w}$

Step 2: From $\mathbb{F}_{2^k}$-VOLE to Subspace VOLE

- Goal:

$$\underbrace{V}_{n_{\mathcal{C}}} = \underbrace{W}_{n_{\mathcal{C}}} - \underbrace{U}_{k_{\mathcal{C}}} \times \underbrace{G_{\mathcal{C}}}_{n_{\mathcal{C}}} \times \begin{pmatrix} \Delta_1 & & \\ & \ddots & \\ & & \Delta_{n_{\mathcal{C}}} \end{pmatrix} \quad (\ell \text{ rows})$$

- Define

$$T_{\mathcal{C}} := \begin{bmatrix} G_{\mathcal{C}} \\ H_{\mathcal{C}} \end{bmatrix}$$

- Starting Point: $n_{\mathcal{C}} \times \mathbb{F}_{2^k}$-VOLE

$$V = W' - U' \times \begin{pmatrix} \Delta_1 & & \\ & \ddots & \\ & & \Delta_{n_{\mathcal{C}}} \end{pmatrix}$$

$T_{\mathcal{C}}$-decompose

$$U' \xrightarrow{T_{\mathcal{C}}\text{-decompose}} U \times G_{\mathcal{C}} + C \times H_{\mathcal{C}}$$

Step 2-1: Sending Syndrome for Correction

$P_A \xrightarrow{C} P_B$

Sets $W := W' - C \cdot H_{\mathcal{C}}$

$W' - C \cdot H_{\mathcal{C}} \cdot \Delta$

$U' \cdot \Delta - C \cdot H_{\mathcal{C}} \cdot \Delta$

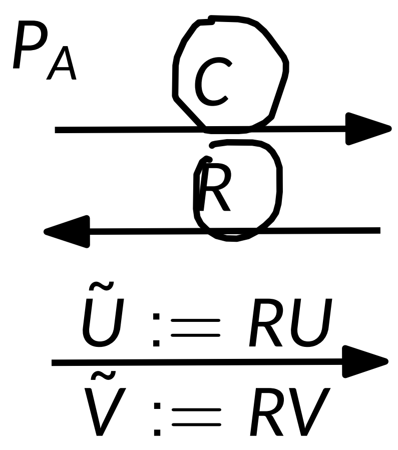
$(U' - C \cdot H_{\mathcal{C}})\Delta$

$$V = W' - \left(U \times G_{\mathcal{C}} + C \times H_{\mathcal{C}} \right) \times \begin{bmatrix} \Delta_1 & & \\ & \ddots & \\ & & \Delta_{n_{\mathcal{C}}} \end{bmatrix}$$

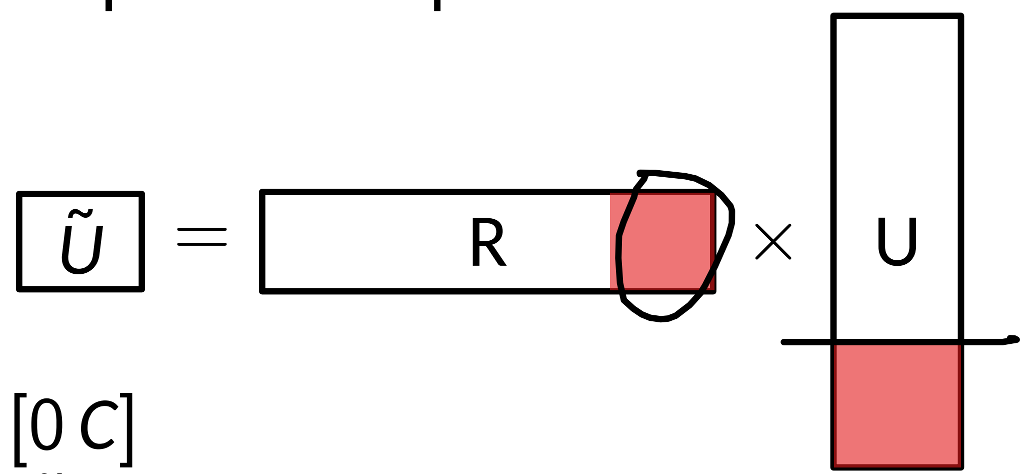
$$= W - U \times G_{\mathcal{C}} \times \begin{bmatrix} \Delta_1 & & \\ & \ddots & \\ & & \Delta_{n_{\mathcal{C}}} \end{bmatrix}$$

Step 2-2: Consistency Checking

- P_A may send incorrect C, so P_B samples $R : \mathbb{F}_{2^k}^{\ell} \to \mathbb{F}_{2^k}^{m}$ for checking



- P_B **checks** $RW = \tilde{V} + \tilde{U} \cdot G_{\mathcal{C}} \cdot \mathrm{diag}(\vec{\Delta})$
- Both parties output the first h rows



- Define $[U\,\bar{C}] := U' T_{\mathcal{C}}^{-1} - [0\,C]$
- $\bar{U} := RU - \tilde{U}$, $\bar{V} := RV - \tilde{V}$
- **check** $\iff$ $\bar{V} + [\bar{U} \;\; R\bar{C}]\mathrm{diag}(\vec{\Delta}) = 0$
- Let $\|[\bar{U} \;\; R\bar{C}]\|_0 = t$, P_B aborts with probability $2^{-k \cdot t}$
- We only consider small t
- The value of $\bar{U}$ is limited to a small set $\mathcal{W}_{pre}$

Step 2-2: Defining Functionality

- Pre-commitment witness $\bar{U}$

$$\mathcal{F}_{\text{VOLE}}^{p,q,\mathcal{C},\ell,\mathcal{L}}$$

if P_S is corrupted:
  recv. $U \in \mathbb{F}_p^{\ell \times k_{\mathcal{C}}}, V \in \mathbb{F}_q^{\ell \times n_{\mathcal{C}}}$ from $\mathcal{A}$
else:
  $U \xleftarrow{\$} \mathbb{F}_p^{\ell \times k_{\mathcal{C}}}, V \xleftarrow{\$} \mathbb{F}_q^{\ell \times n_{\mathcal{C}}}$
if P_R is corrupted:
  recv. $\overleftarrow{\Delta} \in \mathbb{F}_q^{n_{\mathcal{C}}}, W \in \mathbb{F}_q^{\ell \times n_{\mathcal{C}}}$ from $\mathcal{A}$
  $V := -U G_{\mathcal{C}} \operatorname{diag}(\overleftarrow{\Delta}) + W$
else:
  $\overleftarrow{\Delta} \xleftarrow{\$} \mathbb{F}_q^{n_{\mathcal{C}}}$
  $W := U G_{\mathcal{C}} \operatorname{diag}(\overleftarrow{\Delta}) + V$
send U, V to P_S
$\mathsf{Send/Abort}(\overleftarrow{\Delta}, W, \mathcal{L})$

$$\mathcal{F}_{\text{VOLE-pre}}^{p,q,\mathcal{C},\ell,\mathcal{L},M}$$

if P_S is malicious:
  recv. $\mathcal{W}_{\text{pre}} \subseteq \{0,1\}^*$ from $\mathcal{A}$
  recv. $U_{\text{pre}} \colon \mathcal{W}_{\text{pre}} \to \mathbb{F}_p^{\ell \times k_{\mathcal{C}}}$ from $\mathcal{A}$
  recv. $V_{\text{pre}} \colon \mathcal{W}_{\text{pre}} \times \mathbb{F}_q^{n_{\mathcal{C}}} \to \mathbb{F}_q^{\ell \times n_{\mathcal{C}}}$ from $\mathcal{A}$
  recv. $L_{\text{pre}} \colon \mathcal{W}_{\text{pre}} \to \mathcal{L}$ from $\mathcal{A}$
send "commit" to P_R
run $\mathcal{F}_{\text{VOLE}}^{p,q,\mathcal{C},\ell,\mathcal{L}}$
instead of $\mathsf{Send/Abort}$:
  if P_S is malicious:
    recv. $w_{\text{pre}} \in \mathcal{W}_{\text{pre}}, \overleftarrow{L}_{\text{off}} \in \mathbb{F}_q^{n_{\mathcal{C}}}$ from $\mathcal{A}$
    if $U \neq U_{\text{pre}}(w_{\text{pre}}) \vee \underline{V} \neq V_{\text{pre}}(w_{\text{pre}}, \overleftarrow{\Delta}) \vee \overleftarrow{\Delta} + \overleftarrow{L}_{\text{off}} \notin L_{\text{pre}}(w_{\text{pre}})$
      send "check failed" to P_R
      abort
  send $\overleftarrow{\Delta}, W$ to P_R

M

Step 2-2: The Simulator

$\mathcal{S}^{p,q,\mathcal{C},\ell}_{\text{sub-VOLE-mal-R}}$

recv. $\overleftarrow{\Delta} \in \mathbb{F}_q^{n_\mathcal{C}}, W' \in \mathbb{F}_q^{\ell \times n_\mathcal{C}}$ from $\mathcal{A}$
send $\overleftarrow{\Delta}, W'$ to P_R
$C \xleftarrow{\$} \mathbb{F}_p^{\ell \times (n_\mathcal{C} - k_\mathcal{C})}$
send C to P_R
$W := W' - [0\ C] T_\mathcal{C} \operatorname{diag}(\overleftarrow{\Delta})$
send $\overleftarrow{\Delta}, W_{[h]\cdot}$ to $\mathcal{F}^{p,q,\mathcal{C},h,\mathcal{L},M}_{\text{VOLE-pre}}$
recv. $R \in \mathcal{R}$ from P_R
$U_\$ \xleftarrow{\$} \mathbb{F}_q^{\ell \times k_\mathcal{C}}$
$\tilde{U} := R U_\$$
$\tilde{V} := R W - \tilde{U} G_\mathcal{C} \operatorname{diag}(\overleftarrow{\Delta})$
send $\tilde{U}, \tilde{V}$ to P_R

$\mathsf{Precom}(\bar{C}, R, R^{-1})$:
$\mathcal{W}_{\text{pre}} := \{\bar{U} \in \mathbb{F}_q^{m \times k_\mathcal{C}} \mid t \geq \| [\bar{U}\ \ R\bar{C}] T_\mathcal{C} \|_0 \}$
$U^\star_{\text{pre}}(\bar{U}) := U - R^{-1} \bar{U}$
$V^\star_{\text{pre}}(\bar{U}, \overleftarrow{\Delta}) := V + R^{-1} [\bar{U}\ \ R\bar{C}] T_\mathcal{C} \operatorname{diag}(\overleftarrow{\Delta})$
$L'_0 := L' - \overleftarrow{\Delta}_0$ for some $\overleftarrow{\Delta}_0 \in L'$
$L_{\text{pre}}(\bar{U}) := L'_0 \cap \{\overleftarrow{\Delta} \mid 0 = [\bar{U}\ \ R\bar{C}] T_\mathcal{C} \operatorname{diag}(\overleftarrow{\Delta})\}$
return $\mathcal{W}_{\text{pre}}, U^\star_{\text{pre}}, V^\star_{\text{pre}}, L_{\text{pre}}$

$\mathcal{S}^{p,q,\mathcal{C},\ell}_{\text{sub-VOLE-mal-S}}$

recv. $U' \in \mathbb{F}_p^{\ell \times n_\mathcal{C}}, V \in \mathbb{F}_q^{\ell \times n_\mathcal{C}}$ from $\mathcal{A}$
send U', V to P_S
recv. $L' \in \mathcal{L}$ from P_S:
recv. $C \in \mathbb{F}_p^{\ell \times (n_\mathcal{C} - k_\mathcal{C})}$ from P_S
$[U\ \bar{C}] := U' T_\mathcal{C}^{-1} - [0\ C]$

$R \xleftarrow{\$} \mathcal{R}$
abort if $\operatorname{rank}(R\bar{C}) < \operatorname{rank}(\bar{C})$
find $R^{-1} \in \mathbb{F}_q^{\ell \times m}$ s.t. $R^{-1} R \bar{C} = \bar{C}$
$\mathcal{W}_{\text{pre}}, U^\star_{\text{pre}}, V^\star_{\text{pre}}, L_{\text{pre}} := \mathsf{Precom}(\bar{C}, R, R^{-1})$
send $\mathcal{W}_{\text{pre}}, U^\star_{\text{pre}}, V^\star_{\text{pre}}, L_{\text{pre}}$ to $\mathcal{F}^{p,q,\mathcal{C},h,\mathcal{L},M}_{\text{VOLE-pre}}$
send R to P_S

recv. $\tilde{U} \in \mathbb{F}_q^{m \times k_\mathcal{C}}, \tilde{V} \in \mathbb{F}_q^{m \times n_\mathcal{C}}$ from P_S
$\bar{U} := R U - \tilde{U}; \quad U^\star := U^\star_{\text{pre}}(\bar{U})$
$\bar{V} := R V - \tilde{V}; \quad V^\star := V - R^{-1} \bar{V}$
send ${U^\star}_{[h]\cdot}, {V^\star}_{[h]\cdot}$ to $\mathcal{F}^{p,q,\mathcal{C},h,\mathcal{L},M}_{\text{VOLE-pre}}$
find $\overleftarrow{L}_{\text{off}} \in -L'$ s.t. $\bar{V} = [\bar{U}\ \ R\bar{C}] T_\mathcal{C} \operatorname{diag}(\overleftarrow{L}_{\text{off}})$
abort if none exist
send $\bar{U}, \overleftarrow{L}_{\text{off}}$ to $\mathcal{F}^{p,q,\mathcal{C},h,\mathcal{L},M}_{\text{VOLE-pre}}$

Step 3: From Subspace VOLE to Random OT

- Idea 1: Use the Leakage-resilience and Pseudorandomness of TCR

$\mathsf{TCR\text{-}real}^{H,p,q,\mathcal{C},\mathcal{L}}$	$\mathsf{TCR\text{-}ideal}^{H,p,q,\mathcal{C},\mathcal{L}}$
$\overleftarrow{\Delta} \xleftarrow{\$} \mathbb{F}_q^{nc}$	$\overleftarrow{\Delta} \xleftarrow{\$} \mathbb{F}_q^{nc}$
$\underline{\text{QUERY}(\overleftarrow{x} \in \mathbb{F}_p^{kc} \setminus \{0\}, \overleftarrow{y} \in \mathbb{F}_q^{nc}, \tau \in \mathcal{T})\text{:}}$	$\underline{\text{QUERY}(\overleftarrow{x} \in \mathbb{F}_p^{kc} \setminus \{0\}, \overleftarrow{y} \in \mathbb{F}_q^{nc}, \tau \in \mathcal{T})\text{:}}$
return $H(\overleftarrow{x} G_{\mathcal{C}} \odot \overleftarrow{\Delta} + \overleftarrow{y}, \tau)$	$z \xleftarrow{\$} \{0,1\}^\lambda$
	return z
$\underline{\text{LEAK}(L \in \mathcal{L})\text{:}}$	$\underline{\text{LEAK}(L \in \mathcal{L})\text{:}}$
abort if $\overleftarrow{\Delta} \notin L$	abort if $\overleftarrow{\Delta} \notin L$
(a) Real world.	(b) Ideal world.

Figure 6: Oracles for TCR definition. $\boxed{\text{Calls to QUERY must not be repeated on the same input.}}$

Idea 2: Use Uniform-hash to ensure input-uniqueness

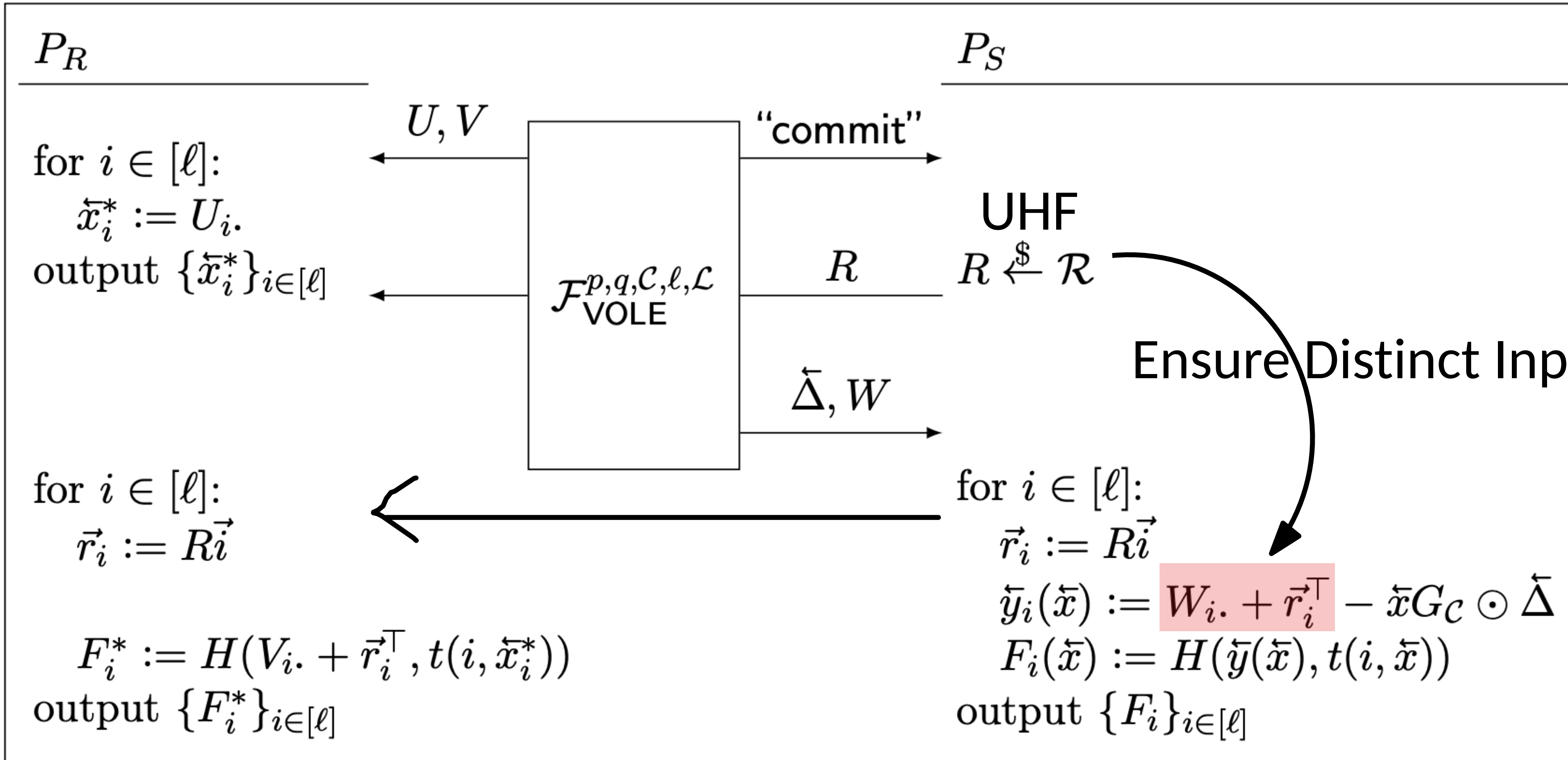


Figure 11: $\binom{p^{k_{\mathcal{C}}}}{1}$-OT extension protocol. Note that the parties for the base VOLE are swapped, with P_S (instead of P_R) getting $\overleftarrow{\Delta}$. If P_S receives “check failed” from the VOLE then the protocol is aborted immediately. For semi-honest security, the “commit” and R steps are skipped, and $\vec{r}_i := 0$.

Summary

- Improving IKNP using PPRF: κ-bit per OT $\rightarrow \kappa/k$-bit per OT
- Rectified security proof: fixing KOS, PSS, OOS errors
- The security proof seems a bit involved, albeit correct in general

- Mysterious claim: $\log N \times \binom{2}{1}$-OT $\equiv 1 \times \binom{N}{1}$-OT

Finally, we hash the subspace VOLE using a correlation robust (CR) hash to build random $\binom{N}{1}$, a correlation (x, m_x) and $(m_0, \ldots, m_{N-1})$ where the m_y are all random. These may used directly, or to encode lookup tables representing multiple small-secret $\binom{2}{1}$-OTs [KK13].

- From [KK13]:

We evaluate performance improvements of Construction 1, and corresponding two- and multi-party SFE improvements. Recall that in the semi-honest model, a single instance of 1-out-of-n OT may be used to generate $\log n$ instances of 1-out-of-2 OT over slightly shorter strings with no additional cost. More precisely, the cost of OT_ℓ^m is exactly equal to the cost of $\binom{n}{1}$-$\mathrm{OT}_{\ell \log n}^{m/\log n}$. This observation will allow us to leverage our efficient construction of $\binom{n}{1}$-OT_ℓ^m to obtain improved efficiency for 1-out-of-2 OT, and consequently for secure computation.