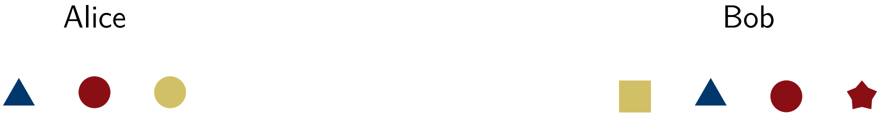
VOLE-PSI: Fast OPRF and Circuit-PSI from Vector-OLE

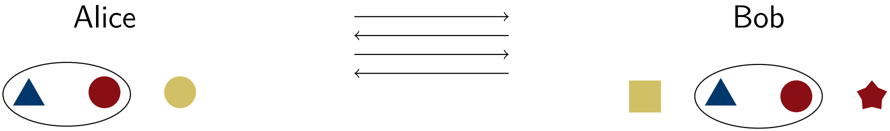
Peter Rindal, **Phillipp Schoppmann**

Eurocrypt 2021

Private Set Intersection (PSI)



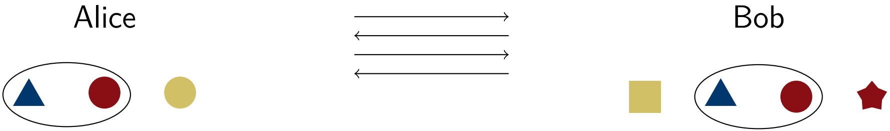
Private Set Intersection (PSI)



Alice

Bob

Private Set Intersection (PSI)



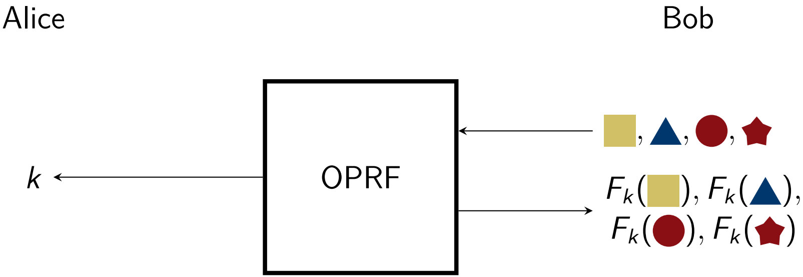
Alice

Bob

Variants:

- One or both parties get the output,
- Associated values,
- Output is secret-shared,
- ...

PSI from OPRF



Alice

Bob

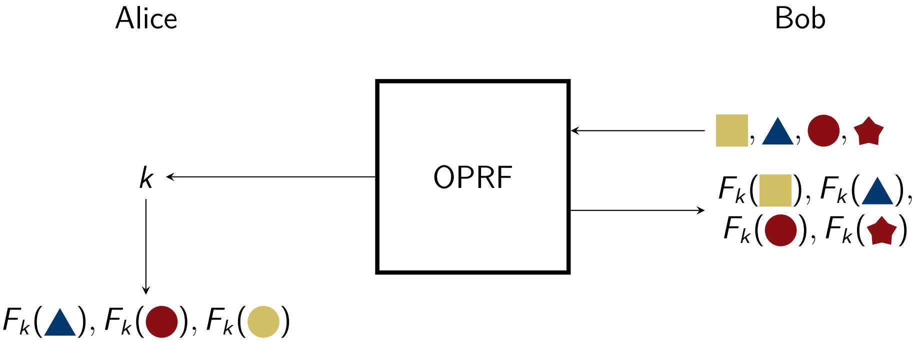
OPRF

k

■, ▲, ●, ★

F_k(■), F_k(▲),
F_k(●), F_k(★)

PSI from OPRF



Alice

Bob

OPRF

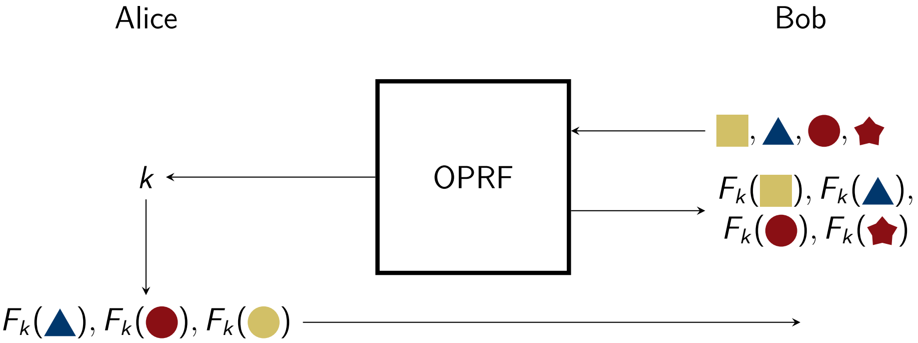
■, ▲, ●, ★

k

$F_k($■$)$, $F_k($▲$)$,
$F_k($●$)$, $F_k($★$)$

$F_k($▲$)$, $F_k($●$)$, $F_k($●$)$

PSI from OPRF



Alice

Bob

■, ▲, ●, ★

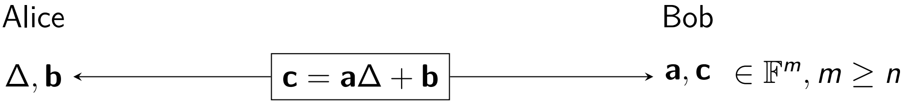
k

OPRF

F_k(■), F_k(▲),
F_k(●), F_k(★)

F_k(▲), F_k(●), F_k(●)

Our Full OPRF Protocol



Alice

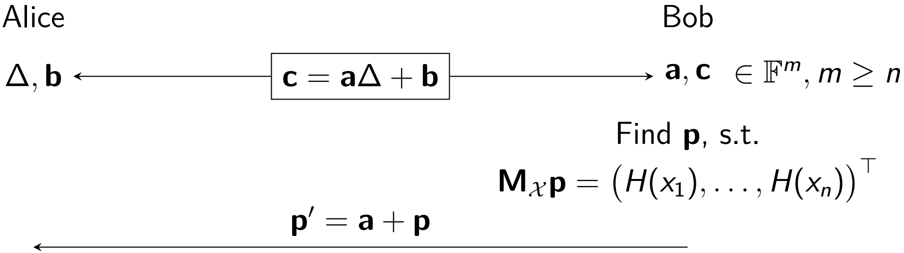
Bob

$\Delta, \mathbf{b}$

$\boxed{\mathbf{c} = \mathbf{a}\Delta + \mathbf{b}}$

$\mathbf{a}, \mathbf{c} \in \mathbb{F}^m, m \geq n$

Our Full OPRF Protocol



Alice

Bob

$\Delta, \mathbf{b}$

$\boxed{\mathbf{c} = \mathbf{a}\Delta + \mathbf{b}}$

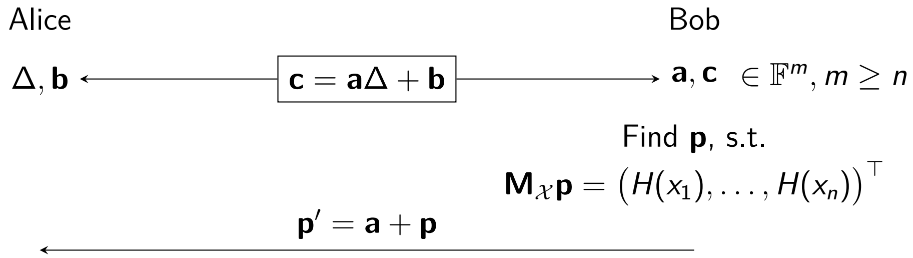
$\mathbf{a}, \mathbf{c} \in \mathbb{F}^m, m \geq n$

Find $\mathbf{p}$, s.t.

$$\mathbf{M}_{\mathcal{X}}\mathbf{p} = \big(H(x_1), \ldots, H(x_n)\big)^\top$$

$\mathbf{p}' = \mathbf{a} + \mathbf{p}$

Our Full OPRF Protocol



Alice

Bob

$\Delta, \mathbf{b}$

$\boxed{\mathbf{c} = \mathbf{a}\Delta + \mathbf{b}}$

$\mathbf{a}, \mathbf{c} \ \in \mathbb{F}^m, m \geq n$

Find $\mathbf{p}$, s.t.

$\mathbf{M}_{\mathcal{X}}\mathbf{p} = \big(H(x_1), \ldots, H(x_n)\big)^\top$

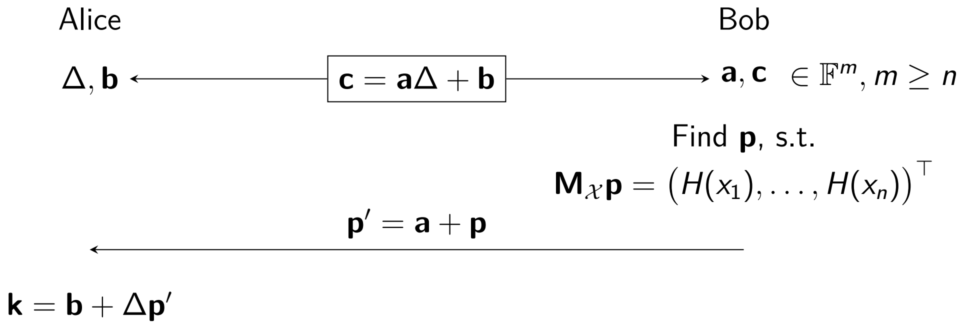
$\mathbf{p}' = \mathbf{a} + \mathbf{p}$

$$\mathbf{k} = \mathbf{b} + \Delta\mathbf{p}'$$

$F(y) = H(\mathbf{M}_y\mathbf{k} - \Delta H(y))$

$F(x) = H(\mathbf{M}_x\mathbf{c})$

Our Full OPRF Protocol



Alice

Bob

$\Delta, \mathbf{b}$

$\boxed{\mathbf{c} = \mathbf{a}\Delta + \mathbf{b}}$

$\mathbf{a}, \mathbf{c} \in \mathbb{F}^m, m \geq n$

Find $\mathbf{p}$, s.t.

$\mathbf{M}_{\mathcal{X}}\mathbf{p} = \big(H(x_1), \dots, H(x_n)\big)^\top$

$\mathbf{p}' = \mathbf{a} + \mathbf{p}$

$\mathbf{k} = \mathbf{b} + \Delta\mathbf{p}'$

$F(y) = H(\mathbf{M}_y\mathbf{k} - \Delta H(y))$

$F(x) = H(\mathbf{M}_x\mathbf{c})$

Observation:

$$\mathbf{M}_y\mathbf{k} - \Delta H(y) \overset{\text{def. } \mathbf{k}}{=} \mathbf{M}_y(\mathbf{b} + \Delta(\mathbf{a} + \mathbf{p})) - \Delta H(y) \overset{y=x}{=} \mathbf{M}_x(\mathbf{b} + \Delta\mathbf{a}) \overset{\text{def. VOLE}}{=} \mathbf{M}_x\mathbf{c}.$$

The Vandermonde Solver

Let $\mathbf{M}_x = (1, x, x^2, \ldots, x^{n-1})$. Then $\mathbf{M}_x\mathbf{v}$ corresponds to evaluating the degree-$(n-1)$ polynomial with coefficients $v_1, \ldots, v_n$ at x.

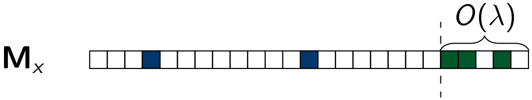
- $m = n$.
- Solving $\mathbf{M}_{\mathcal{X}}\mathbf{p} = \mathbf{h}_{\mathcal{X}}$ corresponds to polynomial interpolation.
- Computing $\mathbf{M}_{\mathcal{X}}\mathbf{v}$ corresponds to multi-point polynomial evaluation.

Both can be done in $O(n \log^2 n)$.

The PaXoS Solver

Introduced by [PRTY20]: Allows for linear-time encoding and decoding, as well as $m = O(n)$!

Idea inspired by cuckoo hashing: Choose exactly two random 1-bits per row. Additionally, add $O(\lambda)$ uniform bits.



Solving $\mathbf{M}_{\mathcal{X}}\mathbf{p} = \mathbf{h}_{\mathcal{X}}$ can be done in $O(n + \lambda^3)$ time!

Evaluation: PSI (Semihonest)

n	Protocol	Communication (MB)			Total running time (s)			
		P_1	P_2	Total	LAN	100 Mbps	10 Mbps	1 Mbps
2^{16}	[KKRT16]	–	–	7.730	0.1160	0.7250	6.884	68.82
	[CM20]	0.5790	4.764	5.343	0.5853	0.6437	4.870	47.49
	[PRTY20]	12.62	0.5898	13.21	0.6460	1.682	11.86	112.8
	Ours	0.9965	2.702	3.699	0.1720	0.4510	3.277	31.18
	Ours (w/setup)	1.171	3.062	4.232	0.5030	1.067	6.742	63.33

🟦 Best (with one-time setup) ⬜ Best (without one-time setup)

Evaluation: PSI (Semihonest)

n	Protocol	Communication (MB)			Total running time (s)			
		P_1	P_2	Total	LAN	100 Mbps	10 Mbps	1 Mbps
2^{20}	[KKRT16]	–	–	128.5	2.441	11.93	114.8	1143
	[CM20]	10.03	77.63	87.66	8.148	9.071	78.38	780.0
	[PRTY20]	214.0	10.49	224.5	5.885	24.09	195.6	1910
	Ours	12.06	40.55	52.61	4.398	8.496	48.69	449.7
	Ours (w/setup)	12.62	40.93	53.55	5.396	9.850	53.35	487.7

🟦 Best (with one-time setup) ⬜ Best (without one-time setup)

Evaluation: PSI (Semihonest)

n	Protocol	Communication (MB)			Total running time (s)			
		P_1	P_2	Total	LAN	100 Mbps	10 Mbps	1 Mbps
2^{24}	[KKRT16]	–	–	2137	43.90	199.1	1910	–
	[CM20]	176.3	1266	1442	189.6	198.1	1289	12 860
	[PRTY20]	3364	184.5	3548	101.7	392.0	–	–
	Ours	204.2	645.7	849.9	90.74	156.4	814.2	7296
	Ours (w/setup)	204.7	646.1	850.9	92.81	158.7	819.9	7335

🟦 Best (with one-time setup) ⬜ Best (without one-time setup)

Evaluation: PSI (Malicious)

n	Protocol	Communication (MB)			Total running time (s)			
		P_1	P_2	Total	LAN	100 Mbps	10 Mbps	1 Mbps
2^{16}	[PRTY20]	12.62	2.097	14.71	0.6510	1.808	13.13	125.5
	Ours	1.390	2.702	4.092	0.2250	0.5260	3.627	34.77
	Ours (w/setup)	1.564	3.062	4.626	0.5560	1.147	7.109	66.72
2^{20}	[PRTY20]	214.0	33.55	247.6	6.119	26.12	215.2	2410
	Ours	17.31	40.55	57.86	5.150	9.599	54.09	495.0
	Ours (w/setup)	17.86	40.93	58.79	6.157	10.94	58.76	532.6
2^{24}	[PRTY20]	3364	536.9	3901	102.8	422.1	–	–
	Ours	271.3	645.7	917.0	104.0	174.5	881.0	7876
	Ours (w/setup)	271.9	646.1	918.0	106.0	176.8	886.7	7914

🟦 Best (with one-time setup) ▨ Best (without one-time setup)

Related Work

- Silver [CRR21]: Faster VOLE generator than the one we used [SGRR19]. Can be used to reduce both communication and computation time of our protocol.
- Oblivious Key-Value Stores [GPRTY21]: More efficient variant of PaXoS. Can significantly reduce communication overhead of our protocol.

Questions?