A post-quantum Distributed OPRF from the Legendre PRF

Novak Kaluderovic
Nan Cheng
Katerina Mitrokotsa

University of St.Gallen

Institute of Computer Science

Overview (#1)

Oblivious Pseudo-random Function (OPRF)

$$F_k : (k, x) \rightarrow F_k(x)$$

- Private Set Intersection
- Password-Authenticated Key Exchange

Research problem

How to construct an **efficient** OPRF with **strong security guarantees**?

Research problem

post-quantum malicious adversary; many corruptions;

How to construct an **efficient** OPRF with **strong security guarantees**?

practical computation/comm.

Our Results

Π_{dOPRF}**:** A **Verifiable** Oblivious Legendre PRF via MPC in **2 Rounds**

Our Results

Protocol	λ	Primitive	Comm.[KB] Offline	Comm.[KB] Online	Rounds	Runtime [ms] Offline	Runtime [ms] Online
PESTO [10]	128	Pairing, ECDSA	-	-	2	-	260*
GC-OPRF [25]		GC, AES-128	-	4634	4	-	1842
Legendre-OPRF [11]		OT, ZK	-	911	9		1049*
Our Π_{dOPRF}		RSS-(1,4)	432	144	2	204	215
Our Π_{dOPRF}		RSS-(2,7)	93900	6303	2	1076	815
GC-OPRF [25]	256	GC, AES-256	-	6628	4	-	2320
Our Π_{dOPRF}		RSS-(1,4)	1728	579	2	208	637

Least rounds + practical efficiency + post-quantum guarantee

[PESTO] Baum, Carsten, et al. "PESTO: proactively secure distributed single sign-on, or how to trust a hacked server." 2020 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, 2020.
[GC-OPRF] Sebastian Faller, Astrid Ottenhues, and Johannes Ottenhues. Composable oblivious pseudo-random functions via garbled circuits. Cryptology ePrint Archive, Paper 2023/1176, 2023
[Legendre-OPRF] Ward Beullens, Lucas Dodgson, Sebastian Faller, and Julia Hesse. The 2hash oprf framework and efficient post-quantum instantiations. Cryptology ePrint Archive, Paper 2024/450, 2024.

Background (#2)

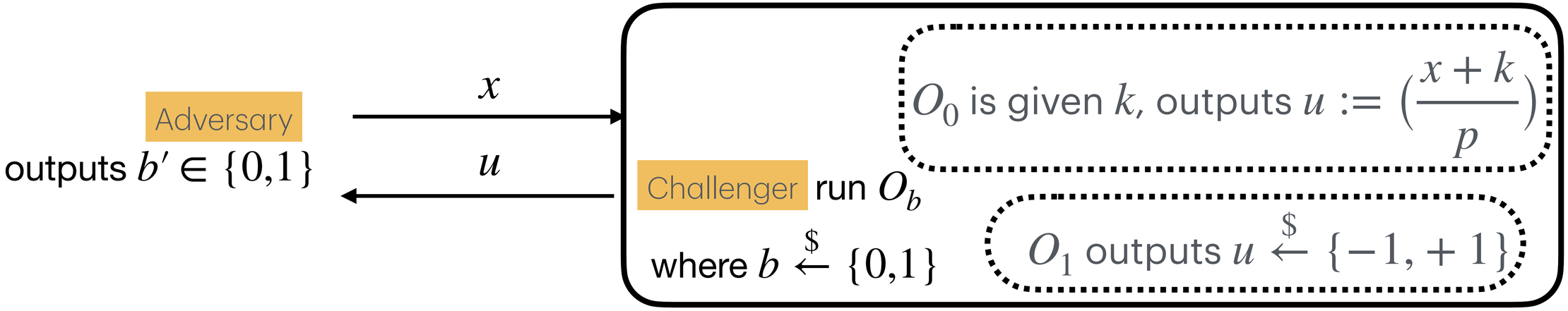
Legendre PRF

Definition

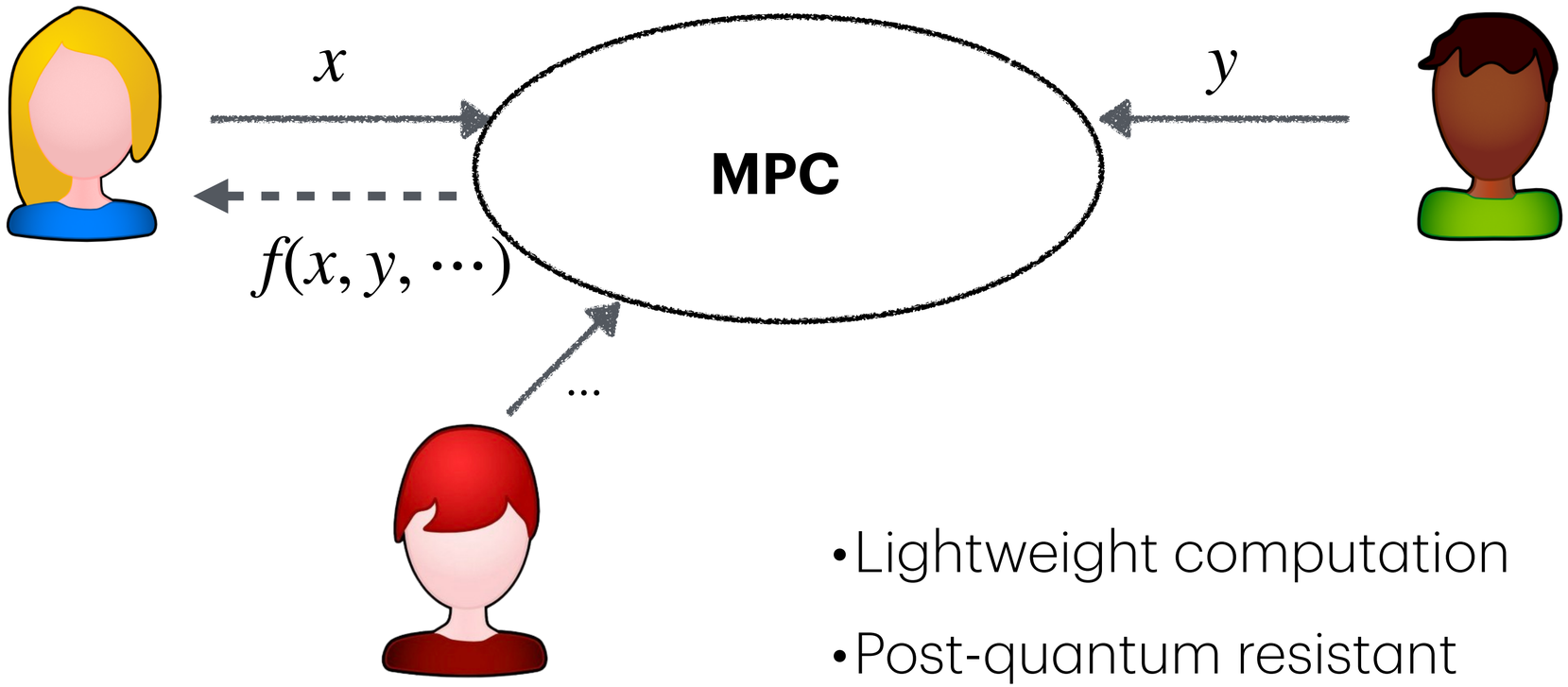
The Legendre PRF is the family of functions $\{F_k\}_{k \in \mathbb{K}}$ with key space being $\mathbb{K} = \mathbb{F}_p$ that is parameterised as $F_k : \mathbb{F}_p \to \{-1, +1\}$ and defined as:

$$F_k(x) := \left(\frac{x+k}{p}\right)$$

Decisional Shifted Legendre Symbol Problem (DSLSP)



Multiparty Secure Computation (MPC)



- Lightweight computation
- Post-quantum resistant
- **Heavy communication**

Replicated Secret Sharing (RSS-(t,n))

				
$[a]$	(a_0, a_1)	(a_1, a_2)	(a_2, a_0)	$a := \sum_{i=0}^{2} a_i$
$[b]$	(b_0, b_1)	(b_1, b_2)	(b_2, b_0)	$b := \sum_{i=0}^{2} b_i$

Naive Framework (#3)

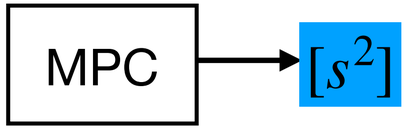
MPC Framework for Legendre PRF

Client

C

Servers

$S_1, \cdots, S_n$



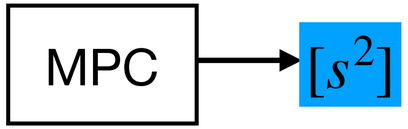
MPC Framework for Legendre PRF

Client

Servers

C

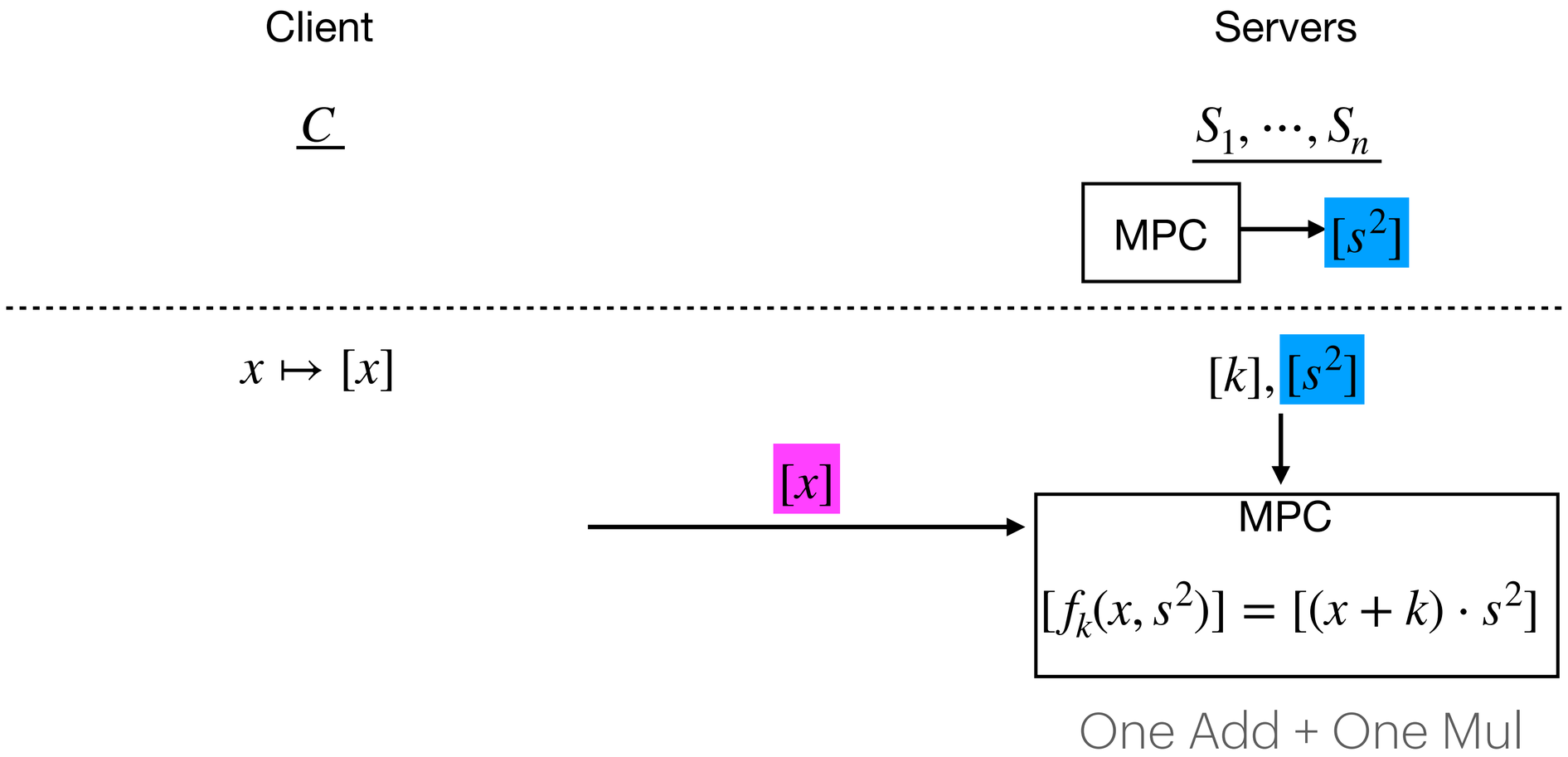
$S_1, \cdots, S_n$



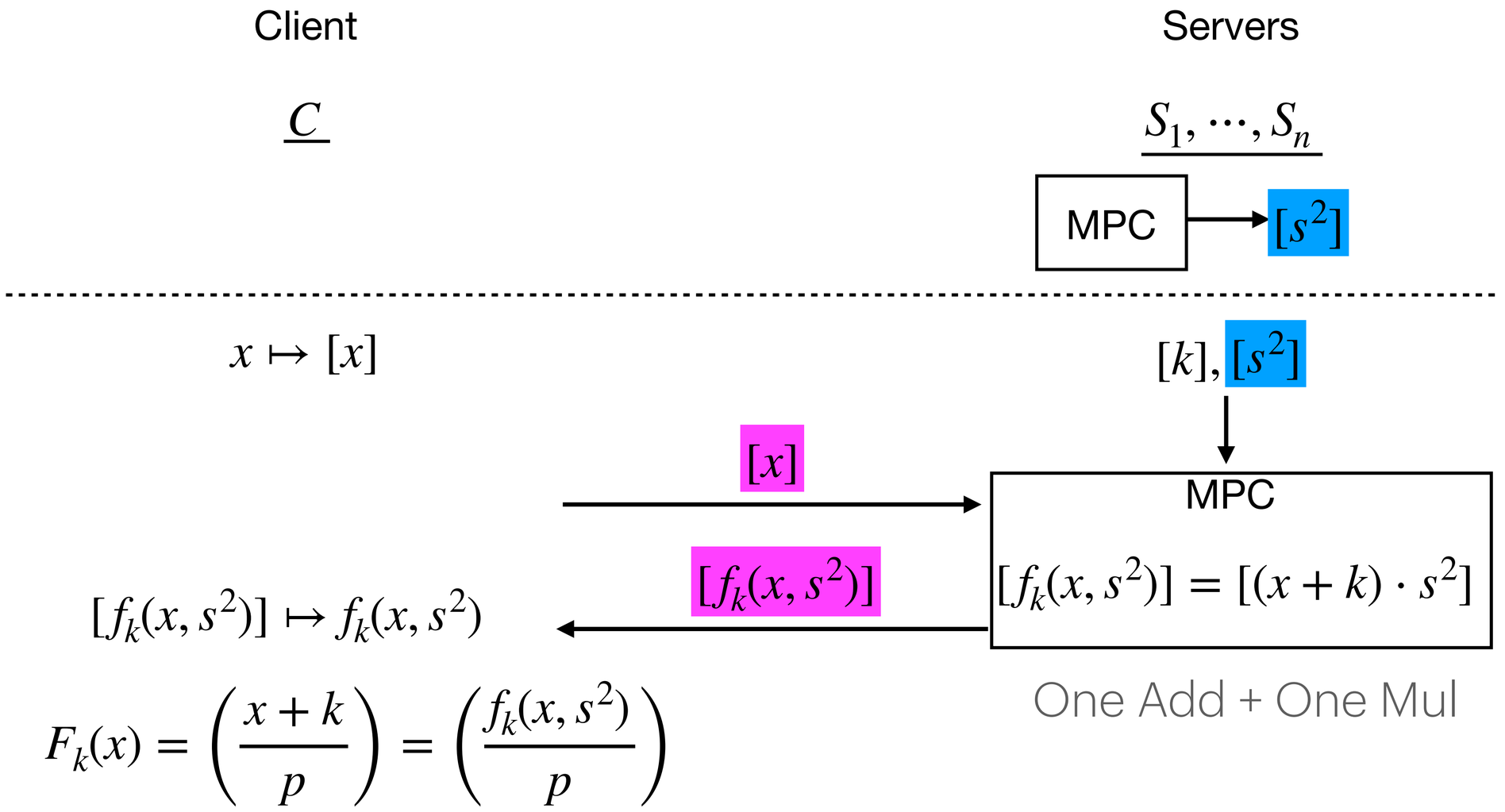
$x \mapsto [x]$

$[k], [s^2]$

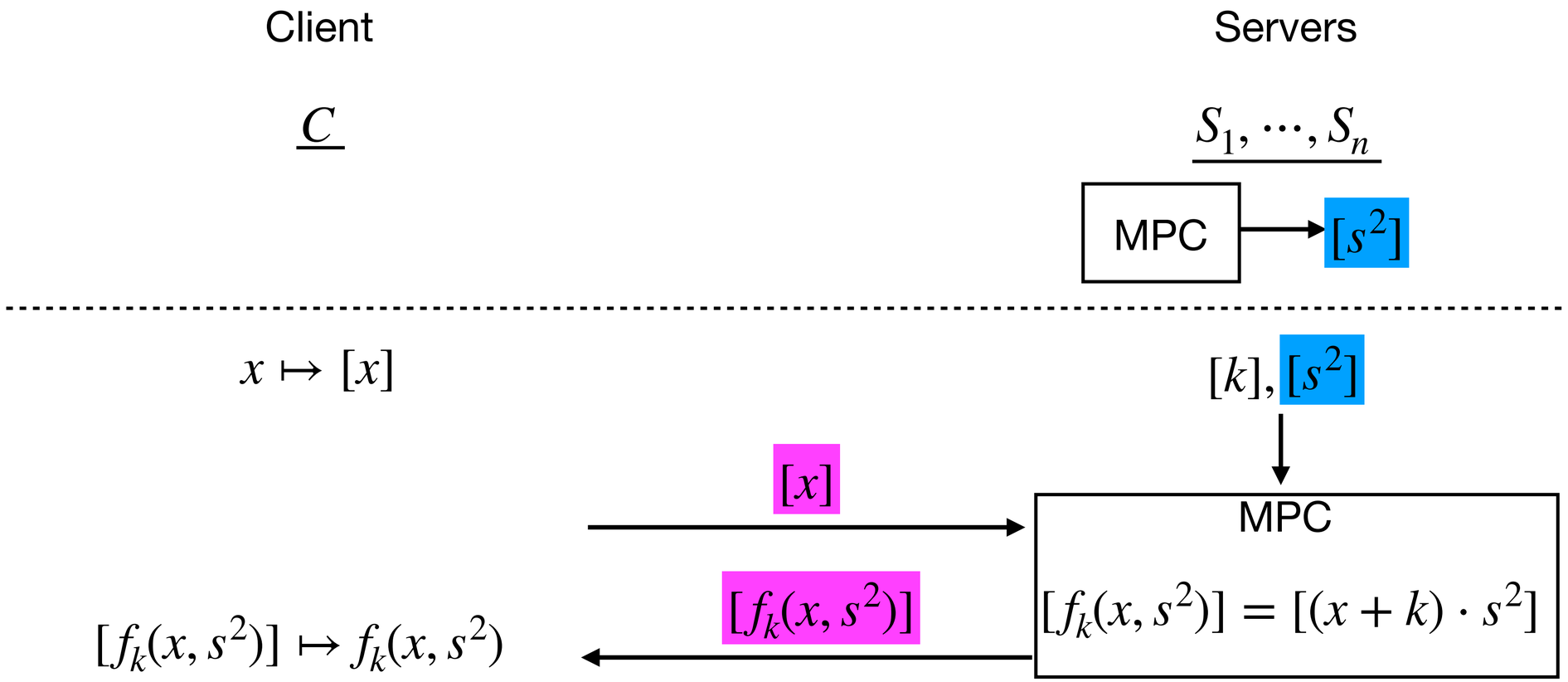
MPC Framework for Legendre PRF



MPC Framework for Legendre PRF



MPC Framework for Legendre PRF



$$F_k(x) = \left(\frac{x+k}{p}\right) = \left(\frac{f_k(x, s^2)}{p}\right)$$

Can we design a round-optimized protocol?

Secure Multiplication over RSS-(t,n)

$\Pi_{\text{MulAndReveal}}$: Aggregating all local products ($t < n/2$)

				
$[a]$	(a_0, a_1)	(a_1, a_2)	(a_2, a_0)	$a := \sum_{i=0}^{2} a_i$
$[b]$	(b_0, b_1)	(b_1, b_2)	(b_2, b_0)	$b := \sum_{i=0}^{2} b_i$
ab?	(a_0b_0, a_0b_1, a_1b_0)	(a_1b_1, a_1b_2, a_2b_1)	(a_2b_2, a_0b_2, a_2b_0)	

Q1. How to verify local products correctness?

Our Insight (#4)

Overview - observation #1

Theorem I

In the RSS-(t,n) scheme, given two RSS $[a], [b]$, **assuming up to t corrupted parties**, for any $T_1, T_2 \in \mathbb{T}^2$ which indicates share indices, then

Claim-I: When $t < n/3$, there exists at least one honest party that holds both a_{T_1} and b_{T_2}.

Overview - observation #1

Theorem I

In the RSS-(t,n) scheme, given two RSS $[a], [b]$, **assuming up to t corrupted parties**, for any $T_1, T_2 \in \mathbb{T}^2$ which indicates share indices, then

Claim-I: When $t < n/3$, there exists at least one honest party that holds both a_{T_1} and b_{T_2}.

A1. 1. Return all local products —> 2. Identify inconsistency —> 3. Accept/Abort

🤦 *However, individual local products leak information of a or b!!*

Overview - observation #2

Definition (t, n)-Doubly replicated secret sharing (DRSS)

The (t, n)-Doubly replicated secret sharing (DRSS) scheme among n parties is defined similarly to RSS, with the exception that the set $\mathbb{T}^2$ is used instead of $\mathbb{T}$ as the share indices. Here,

- there are in total $N = \binom{n}{t}^2$ DRSS shares;
- each party holds $A = \binom{n-1}{t}^2$ DRSS shares.

💡 *Our solution:*

- *mask RSS multiplication results with a random DRSS of zero*
- *mask every DRSS share with a random ASS of zero*

Our Protocol (#5)

Our robust $\Pi_{\text{MulAndReveal}}$

➡ Step 1. Local computation: RSS x RSS -> DRSS

➡ Step 2. DRSS + DRSS(0) -> DRSS; Re-randomized every DRSS share

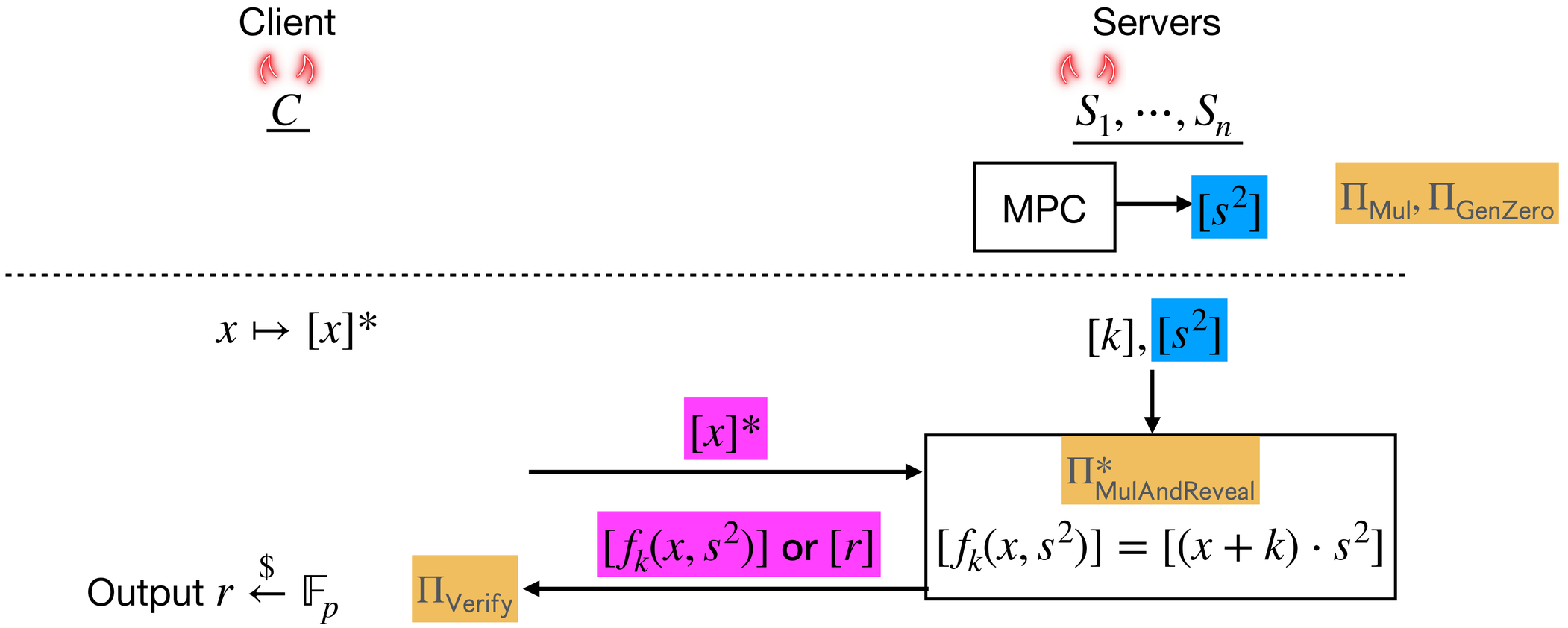
➡Step 3. Every party returns all DRSS local shares and one hash tag

💡*Optimization over step 3: Specify pre-defined t+1 servers return each DRSS share*

$$n \cdot \binom{n-1}{t}^2 \longrightarrow (t+1) \cdot \binom{n}{t}^2$$

$11\,\%, 16\,\%$ commu. reduction for RSS-(1,4) and RSS-(2,7) respectively

Our MPC Framework for Legendre PRF



Our Final Result

Note: assuming malicious client and $t < n/3$ servers

➡A collusion of client and t servers learns nothing about k.

➡An honest client is able to verify result from up to t malicious servers

A **Verifiable** Oblivious Legendre PRF via MPC **in 2 Rounds**

No inter-server communication
in online execution

Comparison & Discussion (#5)

Comparison to SOTA MPC

Ref	Scheme	Comm.	Non-interactive	#Rounds
Our Π_{dOPRF}	RSS-(1,4)	52	✅	**2**
Dalskov19	RSS-(1,4)	36	❌	3
Chida23	RSS-(1,3)	201	❌	9
Our Π_{dOPRF}	RSS-(2,7)	1687	✅	**2**
Chida23	RSS-(2,5)	745	❌	9

[Dalskov19] Anders P. K. Dalskov, Daniel Escudero, and Marcel Keller. “Fantastic Four: Honest-Majority Four-Party Secure Computation With Malicious Security”, USENIX Security 2021
[Chida23] Koji Chida, Koki Hamada, Dai Ikarashi, Ryo Kikuchi, Daniel Genkin, Yehuda Lindell, and Ariel Nof. “Fast Large-Scale Honest-Majority MPC for Malicious Adversaries”. Journal of Cryptology 36.3 (2023)

Discussions

- $\Pi_{\text{MulAndReveal}}$ can easily be adapted to $\Pi_{\text{InnerProdAndReveal}}$ (biometric authentication)
- **Efficiency bottleneck:**

  ➡Offline: generation cost of Π_{Mul}, Π_{GenZero} (*Special batch protocol or PCG*)

  ➡Online: volume of DRSS shares

- On-going work: Higher throughput dOPRF over Gold PRF (multiple bits output)