CERIAS

The Center for Education and Research in Information Assurance and Security

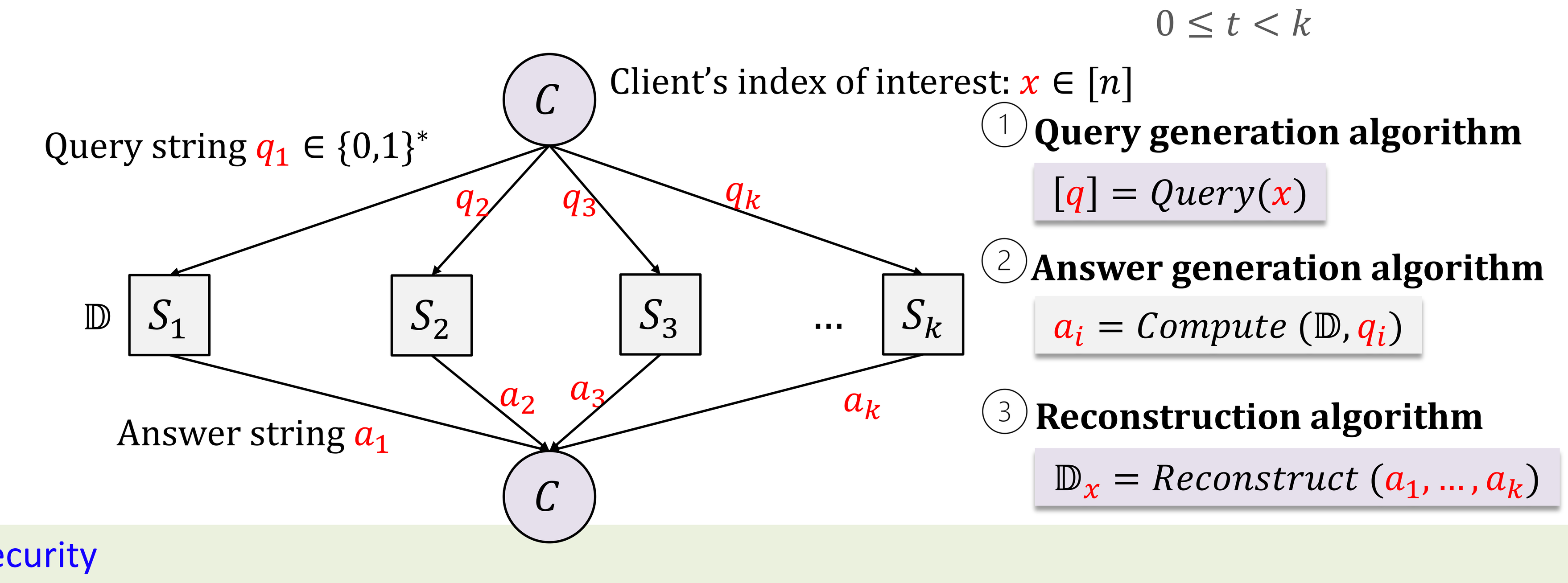
More is Merrier: Relax the Non-Collusion Assumption in Multi-Server PIR

Tiantian Gong, Ryan Henry, Alexandros Psomas, Aniket Kate

What is Private Information Retrieval (PIR) [CKGS95]

A (k,t)-PIR scheme for a database $\mathbb{D}$ with n entries:

▷ k-out-of-k, t-private
$0 \le t < k$



Security

- **Correctness** – C can reconstruct $\mathbb{D}_x$: $H(X|a_1 = Compute(\mathbb{D}, q_1), \ldots, a_k = Compute(\mathbb{D}, q_k)) = 0$
- **Privacy** (IT, computational) – less than $t+1$ parties learn no extra info: $H\left(X|\{q_j\}_{j\in S, |S| \le t}\right) = H(X)$

$H(\cdot)$ computes the entropy of a random variable; X is the random variable for x

1. Multi-Server 1-Private PIR is Efficient

Our model: $(\ell, k, 1)$**-PIR** where $\ell \ge k \ge 2$, database size $N = n \times b$
▷ ℓ total servers, k-out-of-k, 1-private

Communication complexity	IT privacy	Computational privacy
Single-server	N	polylogarithmic
2-Server	$n^{O(\sqrt{\frac{\log\log n}{\log n}})}$ [DG16]	polylogarithmic
3-Server	$2^{O(\sqrt{\log n \log\log n})}$ [Efremenko09]	-
k-Server	1-private: $n^{O(\frac{\log\log k}{k \log k})}$ [BIKR02] t-private: $O(\frac{k^2}{t}\log(k\, n^{1/\lfloor\frac{2k-1}{t}\rfloor}))$ [WY05]	1-private: $(\lambda+2)\log k \left\lfloor \log\frac{n}{\lambda}\right\rfloor + \frac{\ell}{\ell-1}b$ [HH19] (e.g., $\lambda = 128$, then $130 \log k (\lceil \log n \rceil - 7) + \ell/(\ell-1)b$)
$O(\log n)$-Server	$O(\log^2 n \log\log n)$ [CKGS95]	-

Computation complexity (cPIR): polylogarithmic [LMW23] VS $\frac{N}{\ell-1}$ bit operations per server [HH19]

But $(\ell, k, 1)$**-PIR** assumes **no pairwise collusion**

yet collusion is **easy** to implement over unobserved communication channels and **impossible** to detect

Anonymous channels

Encrypted communication channels

Side channels

Can we relax the non-collusion assumption?

2. What we do and How

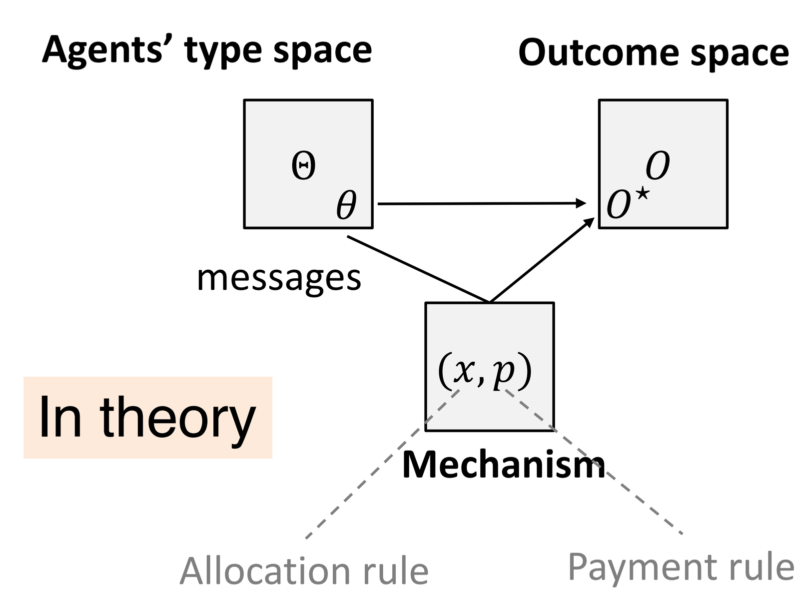
What : Relax the non-collusion assumption to rationality assumption, i.e., servers are rational or malicious

How: Given two premises / our facts

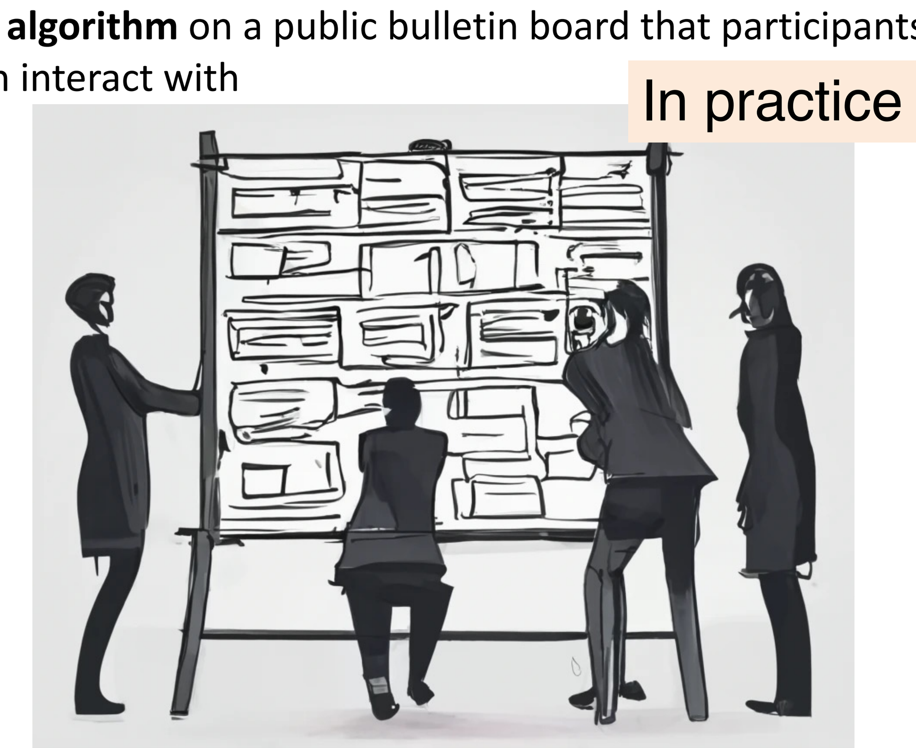
1. We cannot directly detect collusion and collusion can be realized with any protocol, e.g., MPC, TEE, magical clouds, etc.
2. After successful collusion, at least some colluding parties have learned something nontrivial about the index x or entry $\mathbb{D}_x$ --- denote as $f(x)$

we **design a mechanism** such that
(a) it induces a ***game*** where *exactly* one of the servers can take advantage of the ***information gain*** $f(x)$ to maximize its utility at the expense of others,
(b) resulting in some party ***unwilling*** to collude to give others such an advantage

What is a mechanism



An **algorithm** on a public bulletin board that participants can interact with



3. Mechanism overview

Mechanism M

Unknown: secrets $f(x)$, $f(x^r)$ Known: server set $\{S_1, \ldots, S_\ell\}$

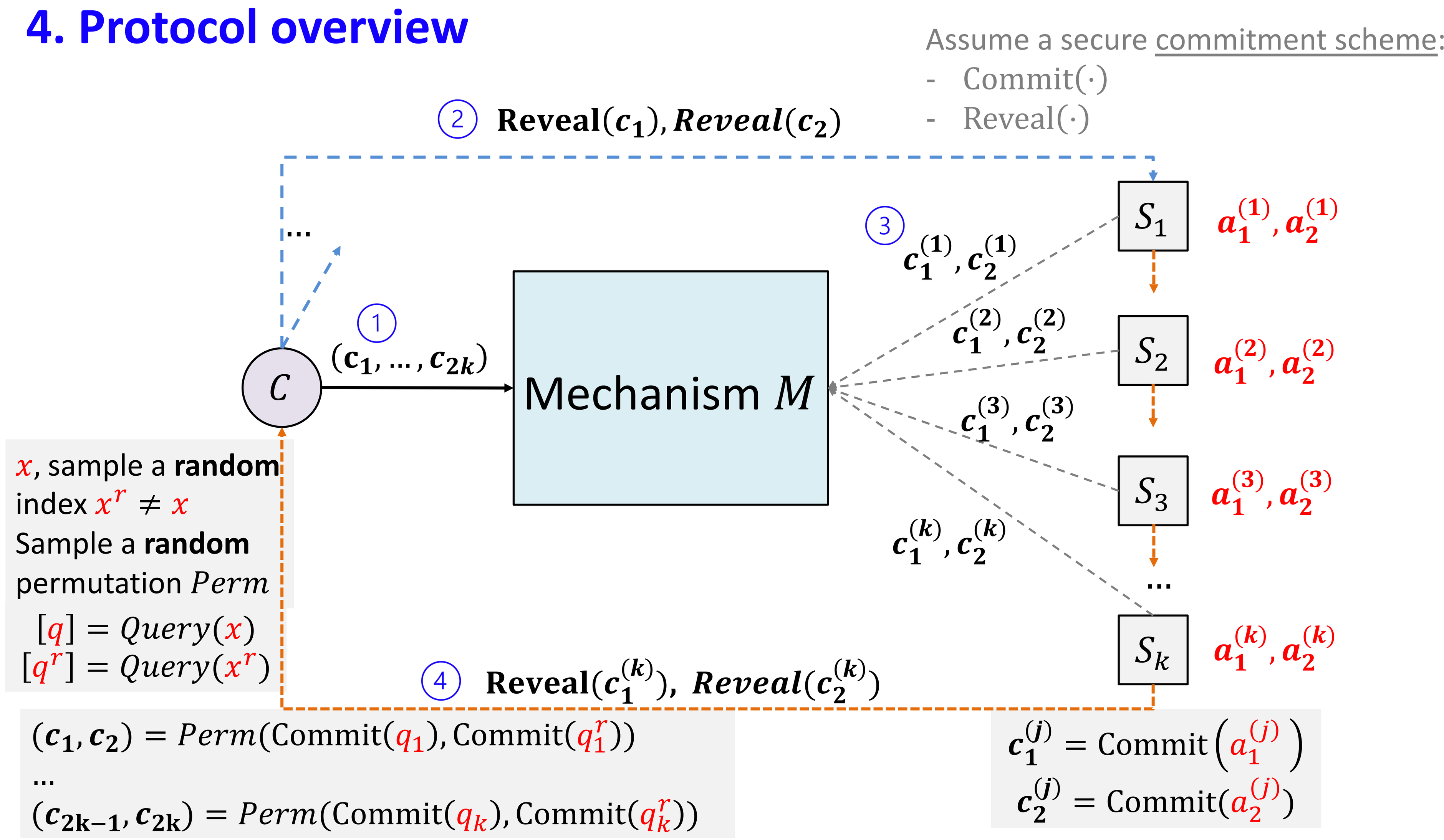
▷ **Winner selection rule W**

If any server S_i tells M the correct secret first along with its input and a *zero-knowledge proof* [Groth16, GWC19, OB22] of inequality is not provided by other servers S_{-i} by time Δ, select S_i *as winner* and mark all other servers as *colluders*

▷ **Payment rule P**

(1) Reward the *winner* a proper amount;
(2) Penalize each marked *colluder* a proper amount;
(3) Penalize S_1 a proper amount if it tells a wrong secret;
(4) Charge proper amount of service fees for each queried server from the client and transfer to servers if there is no collusion after a privacy protection window

4. Protocol overview



5. Communication and computation overhead

On paper One additional commitment per message – instantiated with SHA-3

Implementation as a smart contract on Ethereum

CheckCircuits(·) checks if the function is trivial with oracle services

Table 1. Gas costs summary

Normal service	Gas	Dollars	Collusion resolution	Gas	Dollars
Contract deployment	4697299	\$8.63	Accuse(·)	223766	\$0.41
Deposit(·)	105436	\$0.19	CheckCircuits(·)	66991+	\$0.12+
PostRequests(·)	405657	\$0.74	VerifyExchange(·)	61822	\$0.11
SubmitResponse(·)	97400	\$0.18	VerifyGeneralFunc(·)	275279	\$0.51
ClaimServiceFee(·)	33103	\$0.06	zkVerify(·)	2286423	\$4.20

PURDUE UNIVERSITY

CERIAS