VeriSimplePIR

Verifiability in SimplePIR with No Online Overhead for Honest Servers

Leo de Castro
MIT

Keewoo Lee
UC Berkeley

ia.cr/2024/341

(to appear in USENIX '24)

This Talk

Maliciously Secure PIR

The server sends an initial commitment to the database.

All query-response pairs are verified to be consistent with this commitment.

Optimized for Honest Servers

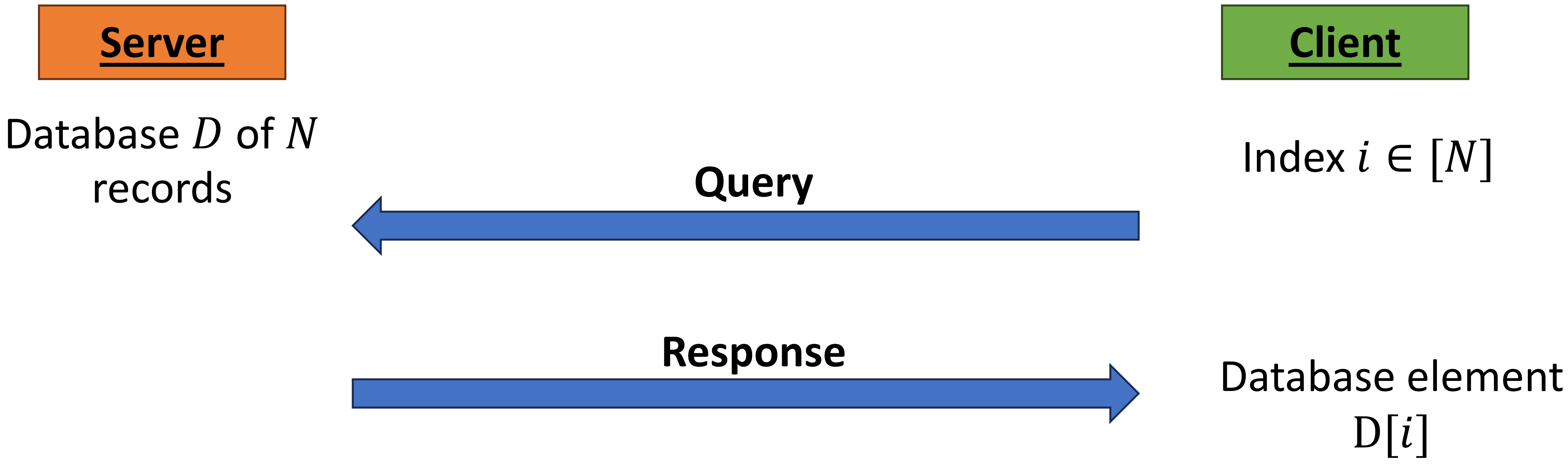
Novel **reusable** proof of consistency that remains secure across many queries.

Fast Online Performance

Online performance is essentially the same as SimplePIR.

Malicious PIR at the rate of the memory throughput.

Private Information Retrieval (PIR)

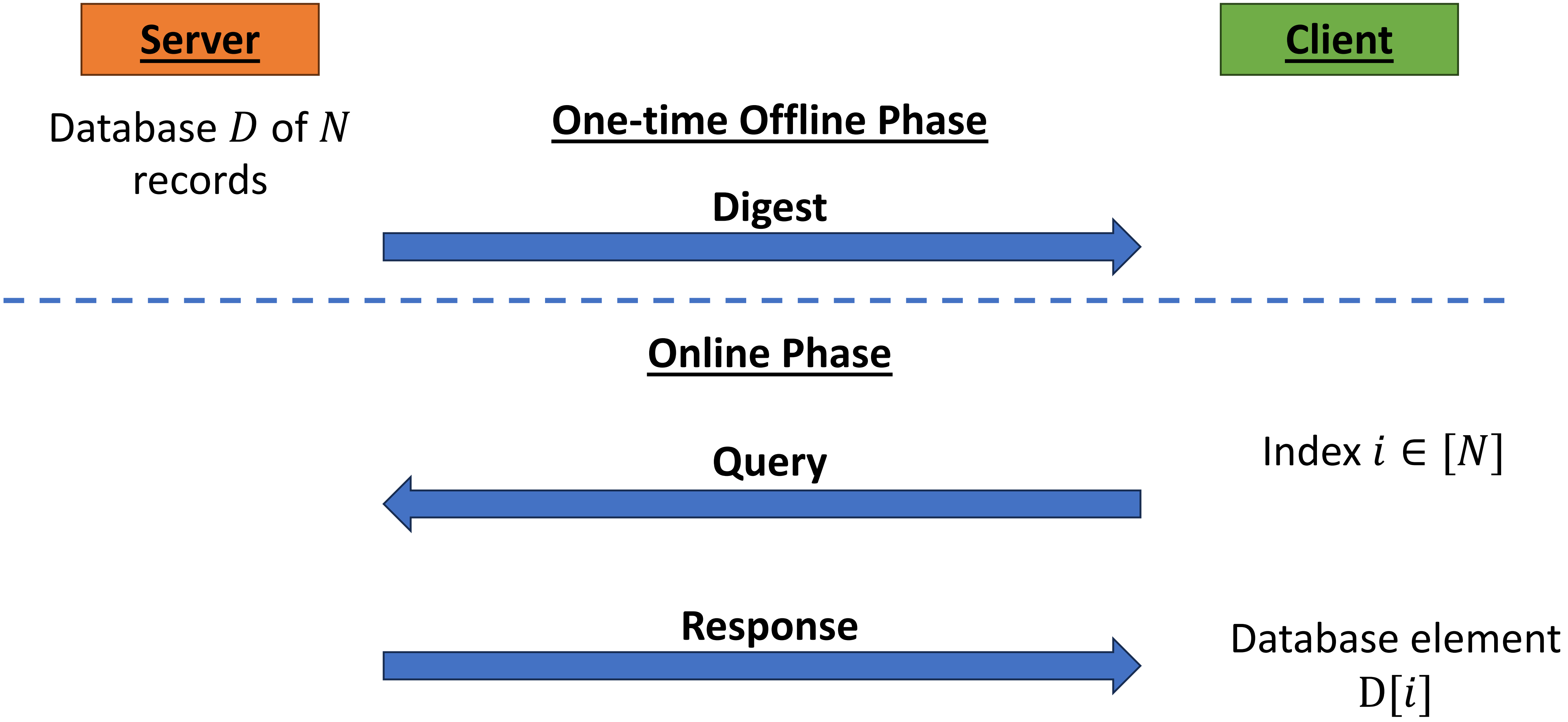


Privacy Requirement

Queries for two indices $i, j \in [N]$ should be indistinguishable.

(We are not concerned with database hiding in this work.)

PIR with Preprocessing



Selective Failure Attack

Server

Database D

Index	Record
1	☑
2	☑
3	😈
4	☑
….	….
N-1	☑
N	☑

Query

Response

A malicious server that can observe the client's failure will be able to identify a query for a corrupted index.

Client

Index $i \in [N]$

If record is, ☑ recover $\mathrm{D}[i]$

If record is, 😈 abort.

Verifiable PIR Definition

Server

Client

Database D of N records

One-time Offline Phase

Commitment to D →

Online Phase

Index $i \in [N]$

← **Query**

Response, Proof →

Verify that the response is consistent with the committed database.

If verification passes, the client should recover the correct database element $D[i]$.

Verification failure should not leak anything about the query index.

Background: Regev Additively Homomorphic Encryption

- **KeyGen():** Output a secret key $s \in \mathbb{Z}_q^n$.
- **Encrypt($s \in \mathbb{Z}_q^n, \mu \in \mathbb{Z}_p^m$):** Encrypt μ in the ciphertext $(A, u) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$.
- **Decrypt($s \in \mathbb{Z}_q^n,\ H \in \mathbb{Z}_q^{\ell \times n},\ v \in \mathbb{Z}_q^\ell$):**
  Output the message in $\mathbb{Z}_q^\ell$ encrypted by (H, v).
- **Eval($A \in \mathbb{Z}_q^{m \times n},\ u \in \mathbb{Z}_q^m, D \in \mathbb{Z}_p^{\ell \times m}$):**
  Output $H = D \cdot A, v = D \cdot u$ as the new ciphertext $(H, v) \in \mathbb{Z}_q^{\ell \times n} \times \mathbb{Z}_q^\ell$.

Eval is a linear function of the ciphertext.
The matrix H is independent of the secret, the message, and the error.

Background: SimplePIR

Server | $A \in \mathbb{Z}_q^{m \times n}$ | **Client**

Database
$D \in \mathbb{Z}_p^{\ell \times m}$

One-time Offline Phase

$H = D \cdot A$

Online Phase

Index $i \in [N]$
Sample secret key s.
Encrypt index in (A, u).

$u \in \mathbb{Z}_q^m$

$v = D \cdot u$

$v \in \mathbb{Z}_q^\ell$

Run Decrypt(s, H, v)
to recover D$[i]$.

Background: SimplePIR (succinct visual)

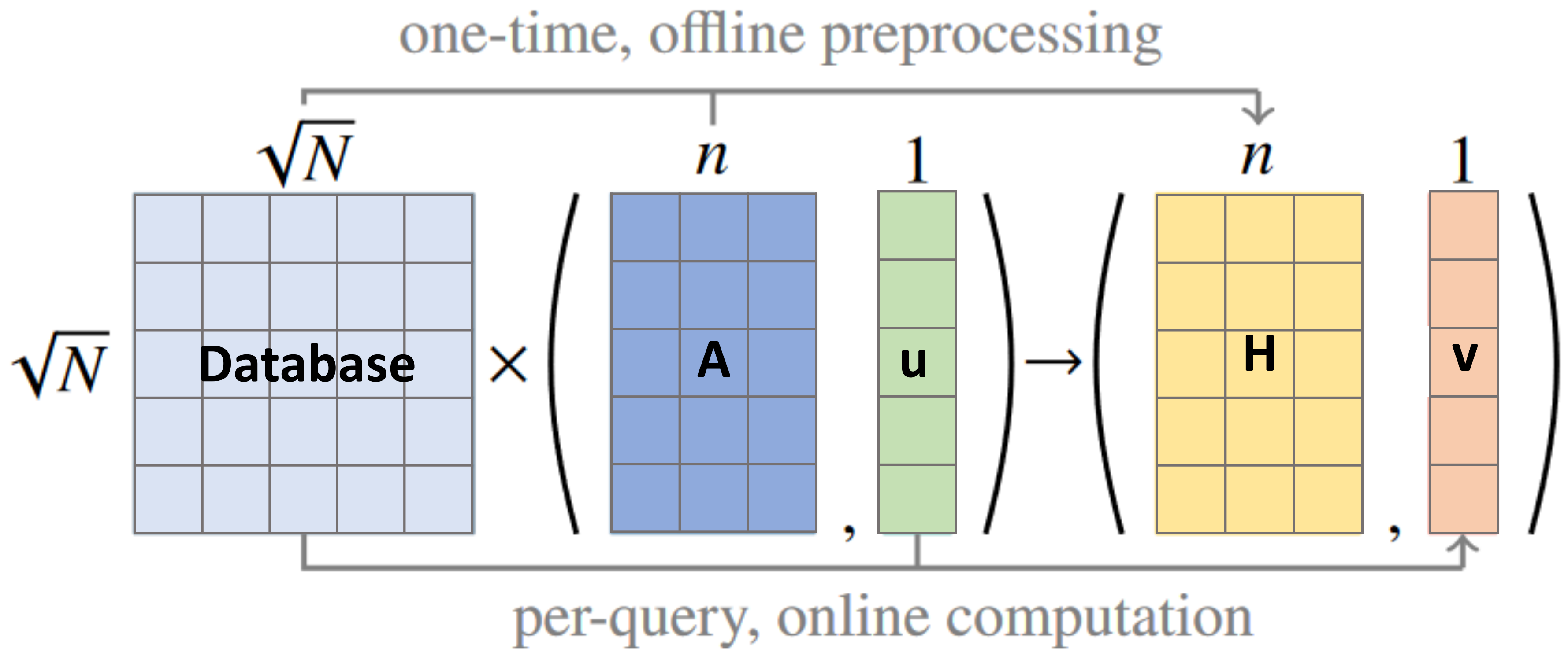


Figure 3 in SimplePIR: [ia.cr/2022/949](ia.cr/2022/949)

Background: Short Integer Solutions (SIS)

The $\mathbf{SIS}_{\boldsymbol{n,q,m,\beta}}$ problem in the ℓ_∞ norm

Given a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, find a non-zero integral vector $\mathbf{z} = (\mathbf{x} \in \mathbb{Z}^m, \mathbf{e} \in \mathbb{Z}^n)$ such that $\mathbf{zA} = \mathbf{e} \pmod{q}$ and $||\mathbf{z}||_\infty \leq \beta$.

The $\mathrm{LWE} \Rightarrow \mathrm{SIS}$ reduction means we don't have to increase SimplePIR parameters.

Background: Extractible SIS Proofs

Prover

$A \in \mathbb{Z}_q^{m\times n}$

Verifier

$D \in \mathbb{Z}_p^{\ell\times m}$

$H = D \cdot A$

H (commitment to D) $\longrightarrow$

$\longleftarrow$ C

Sample $C \leftarrow \{0,1\}^{\lambda\times\ell}$

$Z = C \cdot D$

Z $\longrightarrow$

Check $||Z||$ is small,

$Z \cdot A = C \cdot H$

$\exists$ an efficient extractor that can extract a short matrix D' such that $H = D' \cdot A$.

Two different solutions $H = D \cdot A = D' \cdot A$ give $0 = (D - D') \cdot A$ where $||D - D'||$ is short.
SIS $\Longrightarrow$ This commitment is computationally binding.

Extending SIS Proofs

SimplePIR digest is a commitment to the database.
How to prove consistency with a query?

For a query u, we expect the response $v = D \cdot u$.

Idea: Use the extractable proof for the following commitment.

$$D \cdot [A \;\; u] = [H \;\; v]$$

The proof $Z = C \cdot D$ is ***identical*** to the proof for $H = D \cdot A$.

Verifiable PIR from Extractable SIS Proofs

Server

$A \in \mathbb{Z}_q^{m \times n}$

Client

$D \in \mathbb{Z}_p^{\ell \times m}$ $H = D \cdot A$

$C = \mathrm{Hash}(A, H) \in \{0,1\}^{\lambda \times \ell}$

$Z = C \cdot D$

One-time Offline Phase

H, Z →

Check $\|Z\|$ is small,

$Z \cdot A = C \cdot H$

Online Phase

Index $i \in [N]$

Query ciphertext u with key s.

← $u \in \mathbb{Z}_q^m$

$v = D \cdot u$

$C' = \mathrm{Hash}(A, H, u, v)$

$C' \in \{0,1\}^{\lambda \times \ell}$

$Z' = C' \cdot D$

$v \in \mathbb{Z}_q^\ell, Z' \in \mathbb{Z}^{\lambda \times m}$ →

Check $\|Z'\|$ is small,

$Z' \cdot [A \;\; u] = C' \cdot [H \;\; v]$

Run $\mathrm{Decrypt}(s, H, v)$ to recover D$[i]$.

Verifiable Linearly Homomorphic Encryption

We can verify general computations of the form
$$D \cdot [\mu_1, \mu_2, \dots, \mu_k] = [\gamma_1, \gamma_2, \dots, \gamma_k]$$
for the linear function $D \in \mathbb{Z}_p^{\ell \times m}$.

Encrypt $[\mu_1, \mu_2, \dots, \mu_k]$ into ciphertexts $[u_1, u_2, \dots, u_k] = U \in \mathbb{Z}_q^{m \times k}$.
Output ciphertexts are $[v_1, v_2, \dots, v_k] = V \in \mathbb{Z}_q^{\ell \times k}$.

The proof is always $Z = C \cdot D \in \mathbb{Z}^{\lambda \times m}$.
Verification checks that $\|Z\|$ is small and $Z \cdot [A \;\; U] = C \cdot [H \;\; V]$.
All ciphertexts must use the same A matrix and different secrets.

Verifiable Linearly Homomorphic Encryption

Server

$D \in \mathbb{Z}_p^{\ell \times m}$ $\quad H = D \cdot A$

$C = \mathrm{Hash}(A, H) \in \{0,1\}^{\lambda \times \ell}$

$Z = C \cdot D$

$A \in \mathbb{Z}_q^{m \times n}$

Function Commitment

H, Z →

Client

Check $\|Z\|$ is small

$Z \cdot A = C \cdot H$

Homomorphic Evaluation

Input is $\vec{\mu} \in \mathbb{Z}_p^{m \times k}$.

Sample key $S \in \mathbb{Z}_q^{n \times k}$

Encrypt $\vec{\mu}$ into ciphertext (A, U).

← $U \in \mathbb{Z}_q^{m \times k}$

$V = D \cdot U$

$C' = \mathrm{Hash}(A, H, U, V)$

$C' \in \{0,1\}^{\lambda \times \ell}$

$Z' = C' \cdot D$

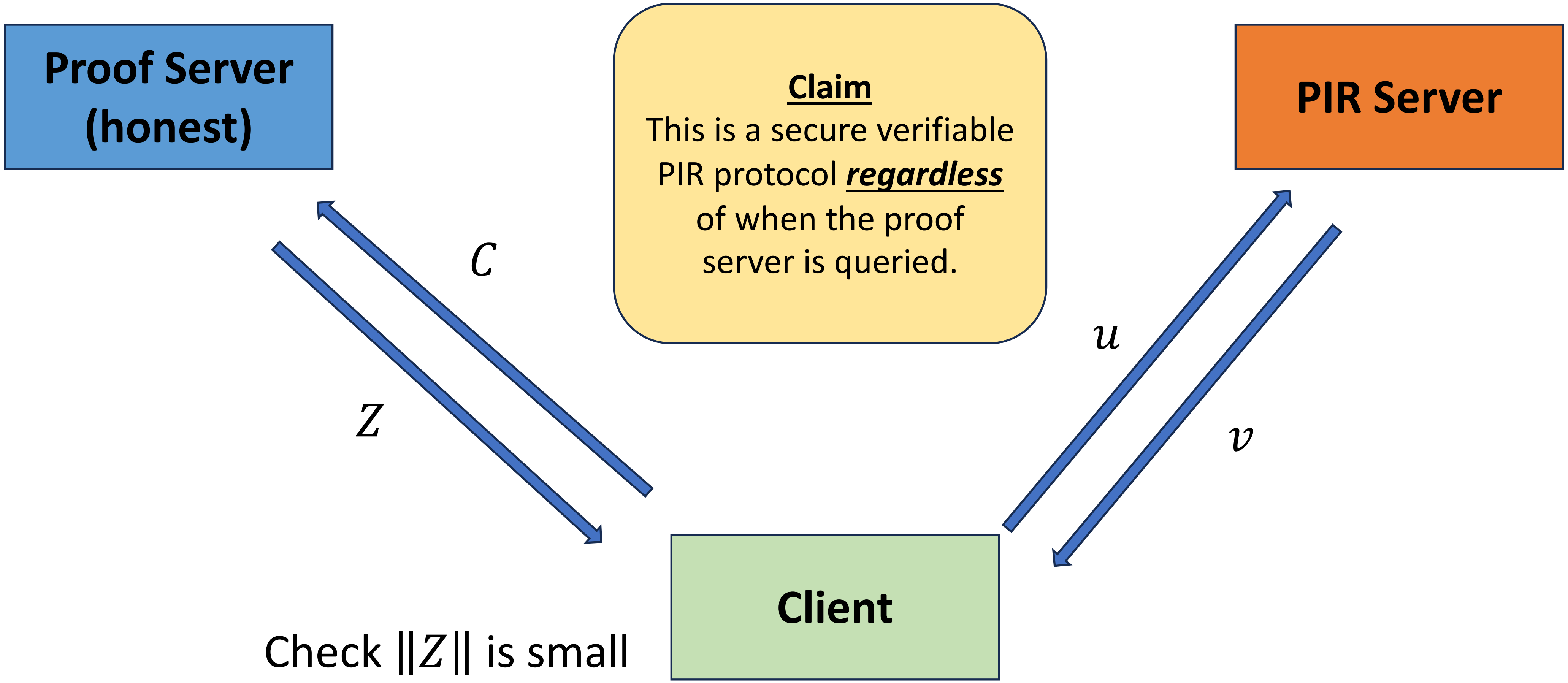
$V \in \mathbb{Z}_q^{\ell \times k}, Z' \in \mathbb{Z}^{\lambda \times m}$ →

Check $\|Z'\|$ is small,

$Z' \cdot [A \;\; U] = C' \cdot [H \;\; U]$

Run $\mathrm{Decrypt}(S\,H, V) \in \mathbb{Z}_p^{\ell \times k}$

Thought Experiment: A Second Server



Check $||Z||$ is small
$Z \cdot [A \;\; u] = C \cdot [H \;\; v]$

Run Decrypt(s, H, v) to recover D$[i]$.

No Leakage if Verification Passes

Security holds as long as the PIR Server has no information about C.

Claim: If Z is honestly computed and verification passes,
w.h.p. there is no leakage on C.

The only response v that passes verification $Z \cdot [A \;\; u] = C \cdot [H \;\; v]$
is the ***exact*** value $v = D \cdot u$ for the D fixed by the initial commitment.

1. If the PIR server behaves honestly, the perfect completeness means that passing is perfectly simulatable.
2. If the PIR server behaves maliciously, the negligible soundness error will likely catch them.

Reusable Proof with no Verification Failure

Proof Server (honest)		Client		PIR Server
	$\xleftarrow{C}$ $\xrightarrow[Z]{}$	Check $\|Z\|$ is small $Z \cdot A = C \cdot H$		
		Check $Z \cdot u_1 = C \cdot v_1$	$\xrightarrow{u_1}$ $\xleftarrow[v_1]{}$	
		Check $Z \cdot u_2 = C \cdot v_2$	$\xrightarrow{u_2}$ $\xleftarrow[v_2]{}$	
		….		
		Check $Z \cdot u_k = C \cdot v_k$	$\xrightarrow{u_k}$ $\xleftarrow[v_k]{}$	

Claim

This is a secure verifiable PIR protocol as long as verification passes for each previous query-response pair.

Intuition

By the previous slide, there's no leakage on the proof randomness with each verification, so the PIR server always has no information about C.

Proof Server from Verifiable LHE

The proof $Z = C \cdot D = (D^T \cdot C^T)^T$ is a linear function of D^T.
We can ***verifiably*** compute Z using our verifiable LHE construction.

Proof Server

Client

$A' \in \mathbb{Z}_q^{\ell \times n}$

Proof Server:

$D^T \in \mathbb{Z}_p^{m \times \ell}$
$H' = D^T \cdot A'$

Proof Server → Client: H' (commitment to D^T)

Client:

$C \leftarrow \{0,1\}^{\lambda \times \ell}$
Encrypt rows of C into U with secret S.

Client → Proof Server: $U \in \mathbb{Z}_q^{\ell \times \lambda}$

Proof Server:

$V = D^T \cdot U$
$C' = \text{Hash}(A', H', U, V)$
$Z' = C' \cdot D^T$

Proof Server → Client: $Z' \in \mathbb{Z}^{\lambda \times \ell}$

Client:

Check $\|Z'\|$ is small
$Z' \cdot [A' \; U] = C' \cdot [H' \;\; V]$

Run Decrypt(S, H', V) to recover Z.

Consistency Check for Precomputed Proof

Proof Server

$A_1 \in \mathbb{Z}_q^{m \times n}, A_2 \in \mathbb{Z}_q^{\ell \times n}$

Client

$D \in \mathbb{Z}_p^{\ell \times m}$
$H_1 = D \cdot A_1, H_2 = D^T \cdot A_2$

H_1, H_2, Z_1, Z_2 →

Check Z_i is short and
$Z_i A_i = C_i H_i$ for $i = 1,2$.

← $U \in \mathbb{Z}_q^{\ell \times \lambda}$

$C \leftarrow \{0,1\}^{\lambda \times \ell}$
Encrypt rows of C into U with secret S.

$V = D^T \cdot U$
$C' = \text{Hash}(A_2, H_2, U, V)$
$Z' = C' \cdot D^T$

$Z' \in \mathbb{Z}^{\lambda \times \ell}$ →

Check $||Z'||$ is small
$Z' \cdot [A_2\ U] = C' \cdot [H_2\ V]$
Run Decrypt(S, H_2, V) to recover Z.
Check that $||Z||$ is small and $ZA_1 = CH_1$.

Claim: $Z = C \cdot D$
Use VLHE soundness to show that $Z = C \cdot D'$ for some short D'. Use SIS to show that $D' = D$.

VeriSimplePIR

Server

$A_1 \in \mathbb{Z}_q^{m \times n}, A_2 \in \mathbb{Z}_q^{\ell \times n}$

Client

$D \in \mathbb{Z}_p^{\ell \times m}$

$H_1 = D \cdot A_1, H_2 = D^T \cdot A_2$

H_1, H_2, Z_1, Z_2

Proof Server Protocol

C, Z

Online Phase

Index $i \in [N]$

Query ciphertext u with key s.

$u \in \mathbb{Z}_q^m$

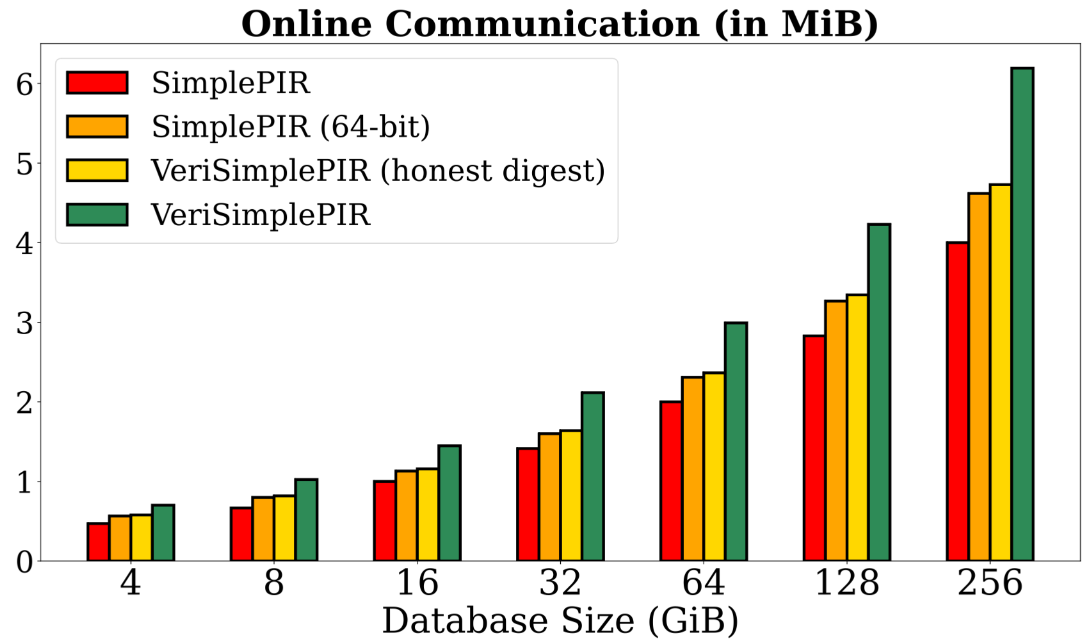
If the check $Z \cdot u = C \cdot v$ fails, rerun the proof server protocol.

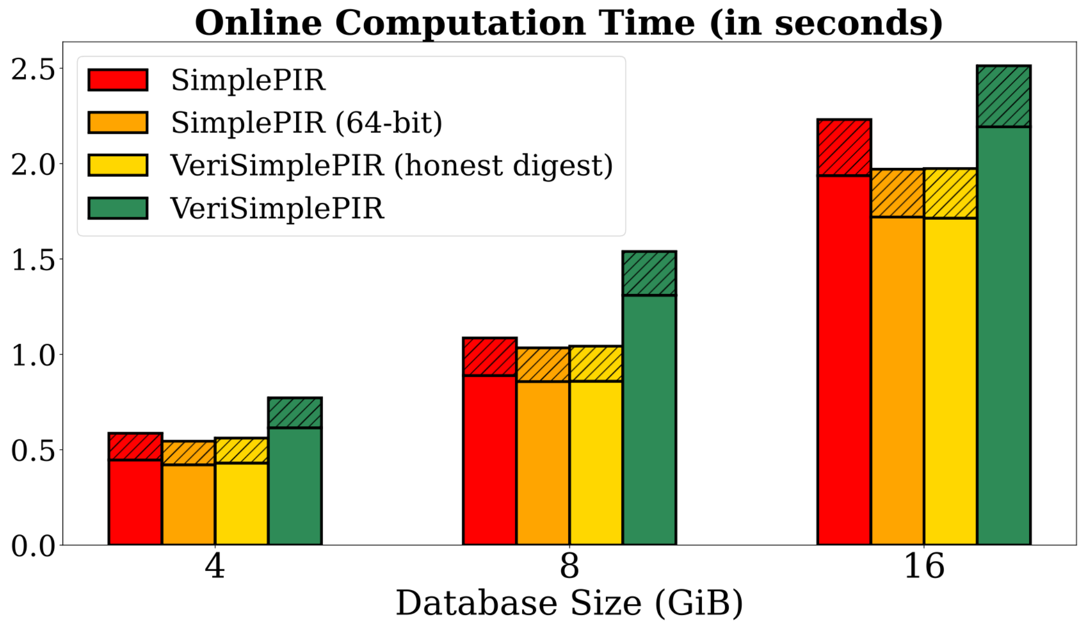
$v \in \mathbb{Z}_q^\ell$

Check $Z \cdot u = C \cdot v$

Run Decrypt(s, H_1, v) to recover D$[i]$.

Performance





Machine-word modulus supports a huge variety of optimizations, including massive parallelism and GPU acceleration.

Future Directions

- Can we reduce the size of the download?
  - Can we verify DoublePIR?
  - The DoublePIR hint is a computationally binding commitment, but the opening proof is very large.
- Can we efficiently update the database commitment?
- Can we preprocess other proofs in this way?

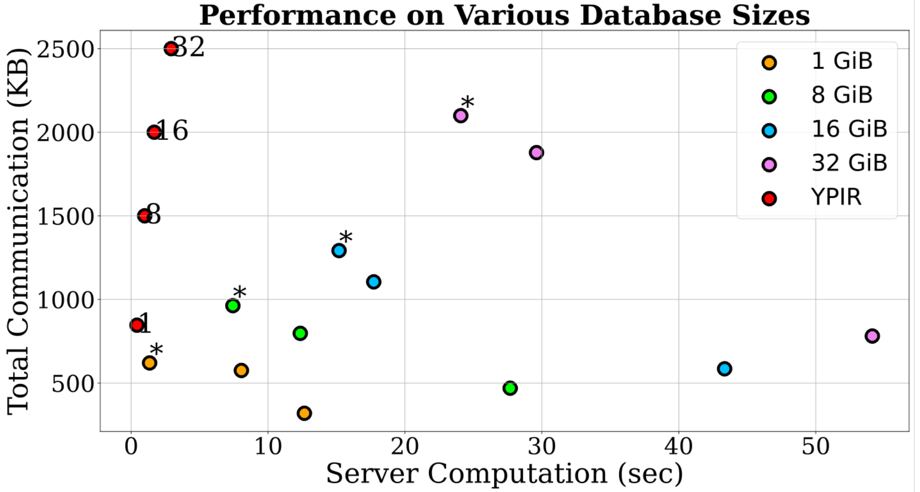
Stateless PIR with Low Communication

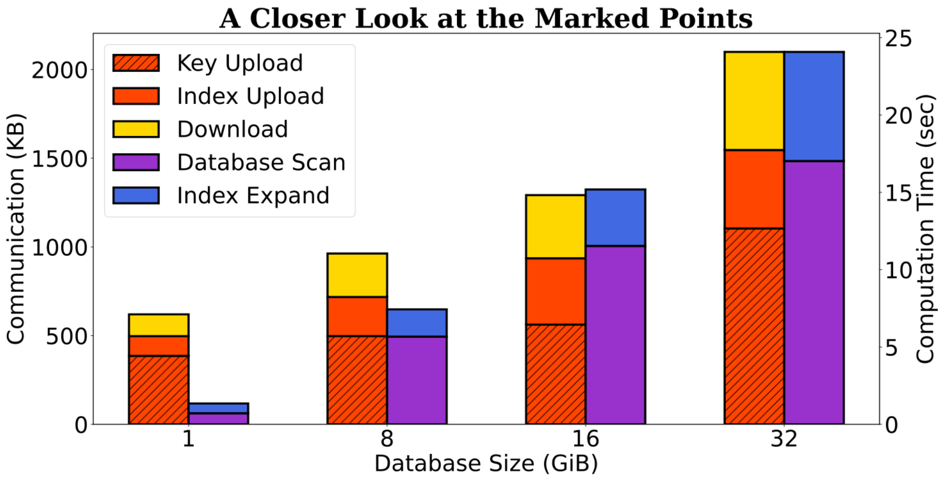
Stateless PIR

- One-time download is very large.
- **Question**: What is the best PIR protocol in a stateless setting (no offline phase)?

WhisPIR: Stateless PIR

- Only upload is one ciphertext + one rotation key.
- New analysis of SEALPIR expansion routine optimized for one rotation key.
- Spiral-style database scan easily supports large entries (many kilobytes).





Thank You!

ia.cr/2024/341