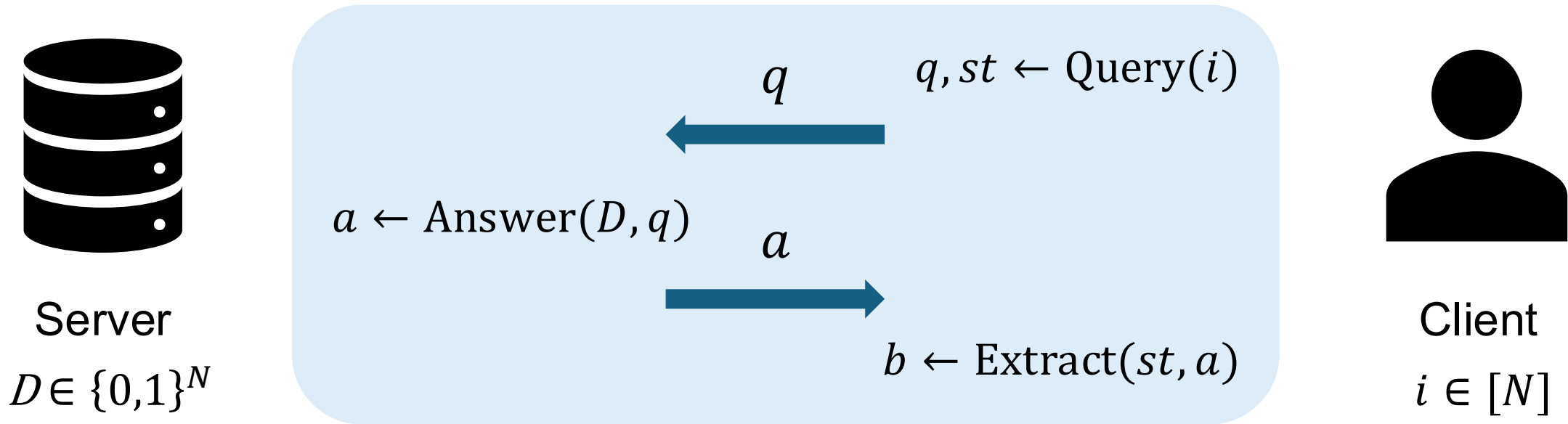


Barely Doubly-Efficient SimplePIR

Keewoo Lee

Ethereum Foundation

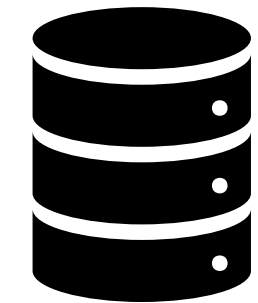
Private Information Retrieval (PIR) [CGKS95, KO97]



- **Correctness:** $b = D[i]$
- **Client Privacy:** Query q hides Index i
- **Client Efficiency:** $o(N)$ Client Work

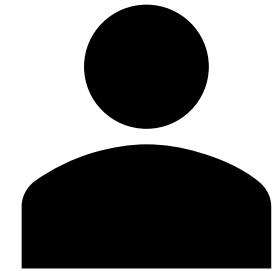
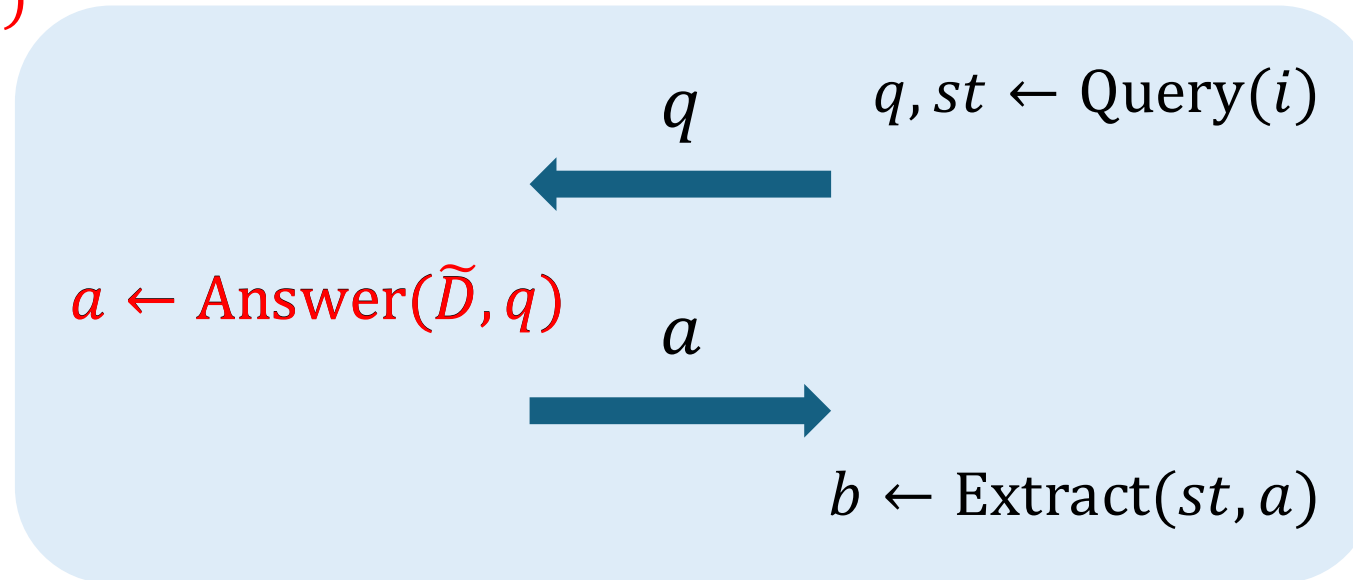
Private Information Retrieval (PIR) [CGKS95, KO97]

$\tilde{D} \leftarrow \text{Preprocess}(D)$



Server

$D \in \{0,1\}^N$



Client

$i \in [N]$

- Correctness: $b = D[i]$
- Client Privacy: Query q hides Index i
- Client Efficiency: $o(N)$ Client Work

A Lower Bound [BIM00]

If Server doesn't touch an entry during Answer, Server learns that Client didn't query that entry. Thus, Server's work should be $\Omega(N)$. **Way Out?**

Doubly-Efficient PIR (DEPIR)

Q. If DB preprocessing is allowed, could we hope for a PIR scheme that is

$o(N)$ online work efficient for both client and server, i.e., doubly-efficient?

- First Multi-Server Scheme: Beimel-Ishai-Malkin (Crypto 2000)
 - Information-theoretic security
 - Open Problem: Single-Server DEPIR
- First Single-Server Scheme: Lin-Mook-Wichs (STOC 2023, Best Paper Award)
 - $O(N^{1+\epsilon})$ preprocess & $\text{polylog}(N)$ online work
 - Based on Ring-LWE
 - Open Problem: DEPIR based on plain LWE

This Work!

Our Contribution

- First DEPIR Scheme from plain LWE

- [LMW23] is based on Ring-LWE and heavily relies on its ring structure
- Closes the gap between what we know how to build from RLWE and from LWE (c.f. FHE)
- “Black-box Crypto is Useless for DEPIR” [LMW25]
 - ✓ This work explores minimal structure sufficient for DEPIR

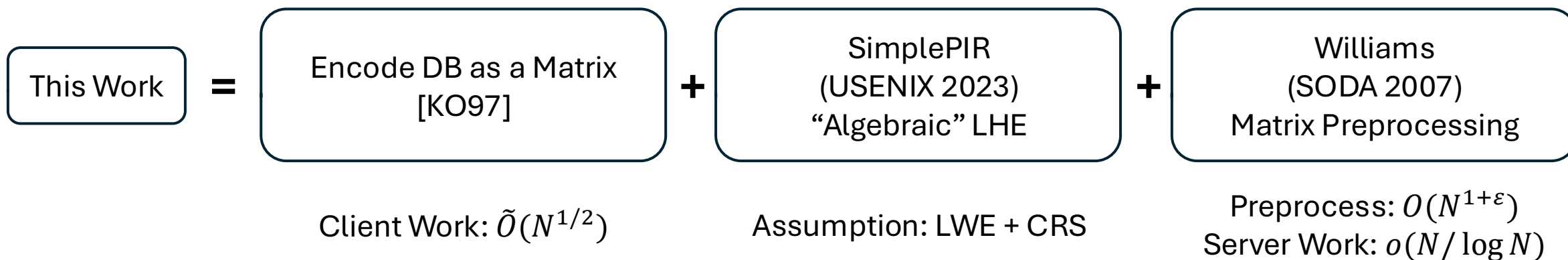
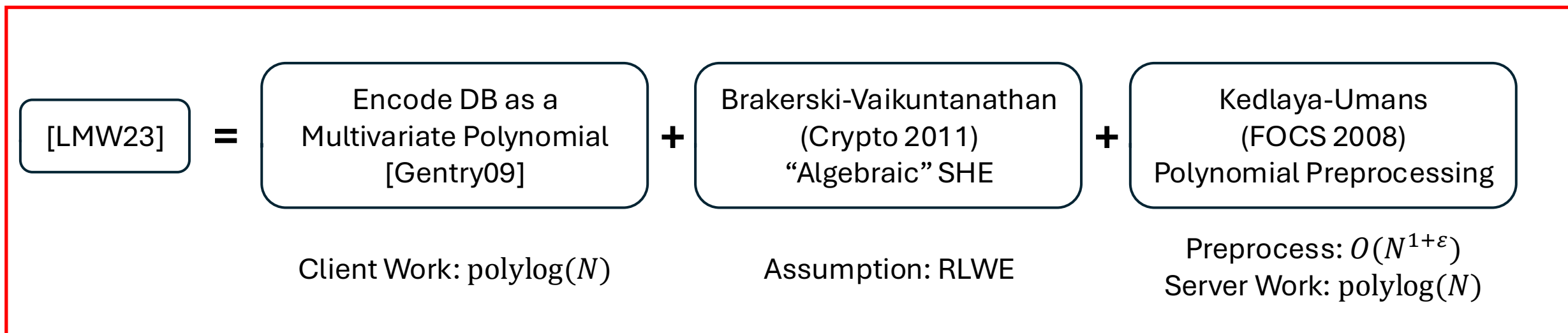
- Simpler DEPIR Construction

- Shares a high-level template with [LMW23], but uses arguably simpler machinery
 - ✓ SimplePIR (USENIX 2023) + Williams’ Matrix Preprocessing (SODA 2007)

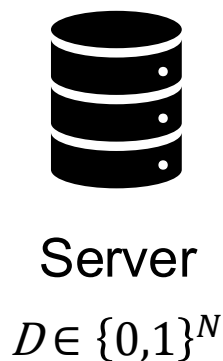
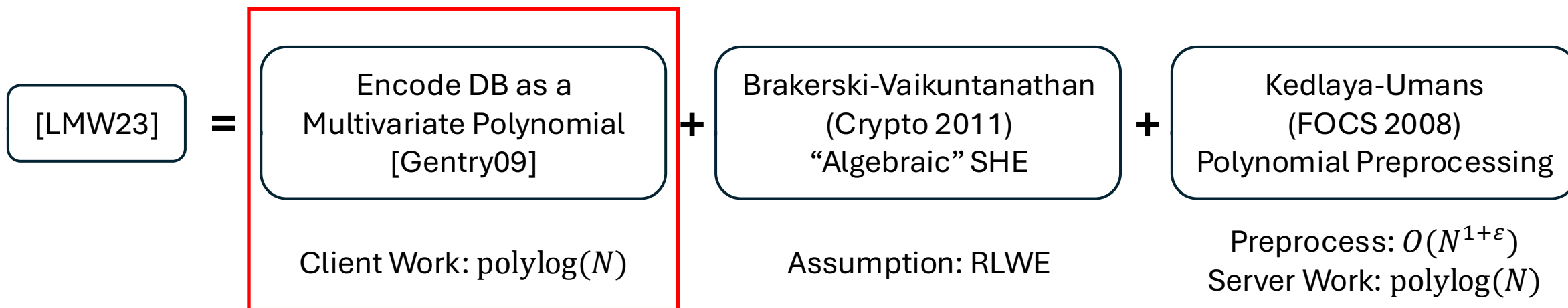
- Limitations

- Only “Barely” Doubly-Efficient: $\mathcal{o}(N/\log N)$ online work (vs. $\text{polylog}(N)$ in [LMW23])
- Require CRS for technical reasons

Construction Overview

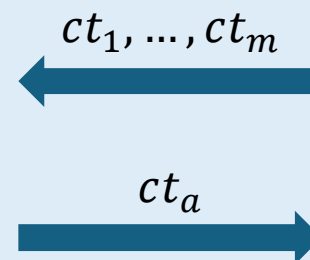


Construction Overview: [LMW23]

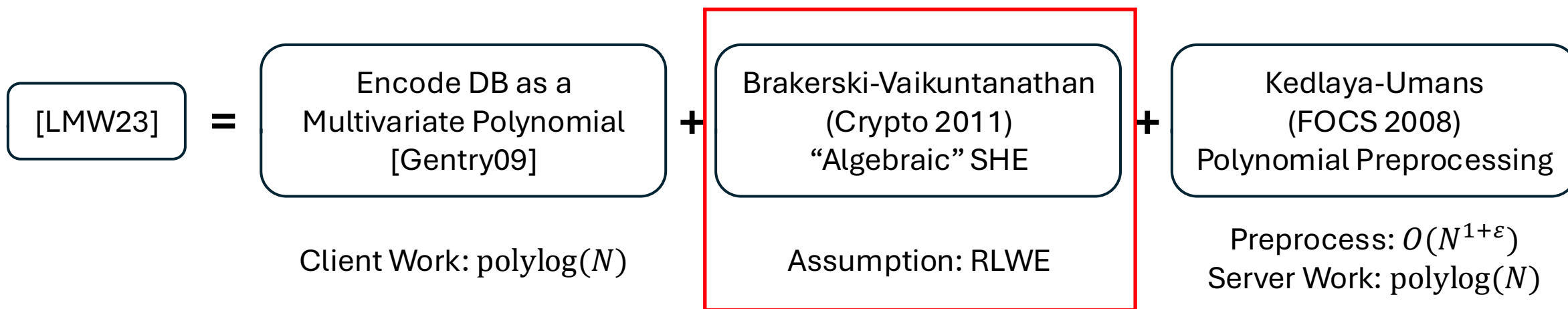


$$f_D \in \mathbb{Z}_p[X_1, \dots, X_m] \quad (p: \text{prime}, p^m \geq N)$$

$$ct_a \leftarrow \text{SHE.Eval}(f_D; ct_1, \dots, ct_m)$$

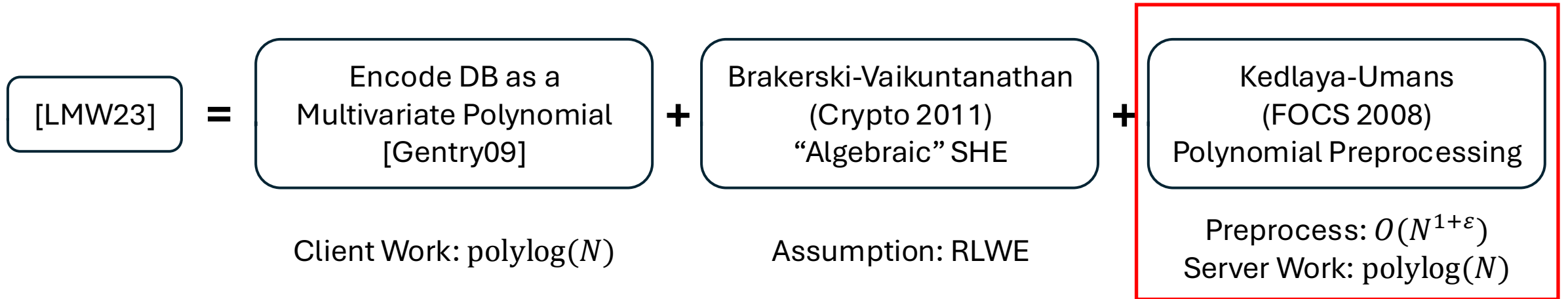


Construction Overview: [LMW23]



- Algebraic SHE: Homomorphic PolyEval is itself a PolyEval over a “Nice” Ciphertext Ring
- [BV11]: BGV but w/o ModSwitch & Relinearization
 - Algebraic SHE with Ciphertext Ring of form $R = \mathbb{Z}_q[X, Y] / \langle X^n + 1, Y^D \rangle$

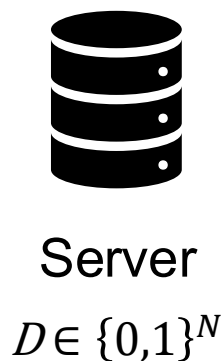
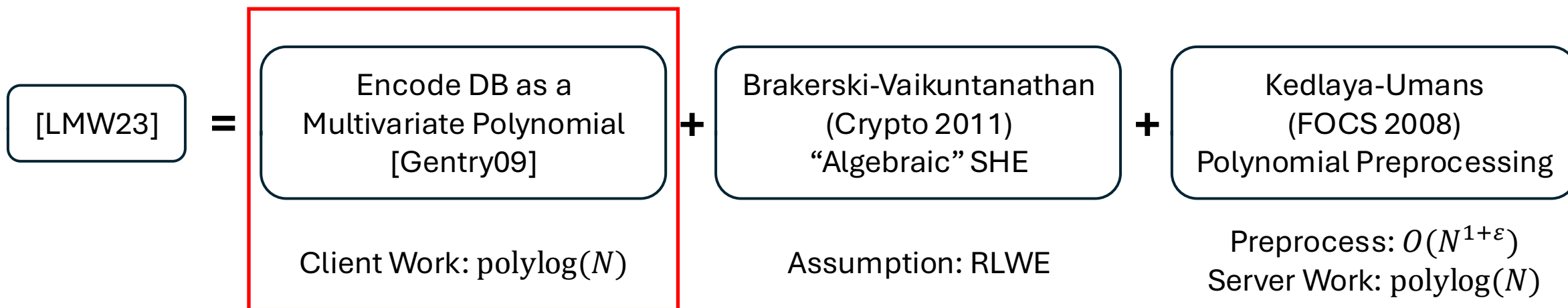
Construction Overview: [LMW23]



Theorem ([KU08,LMW23]). Let $R = \mathbb{Z}_q[X, Y]/\langle X^n + 1, Y^D \rangle$. For any $f \in R[X_1, X_2, \dots, X_m]$, with individual $\deg < d$, there is a preprocessing scheme for f -evaluation with running time:

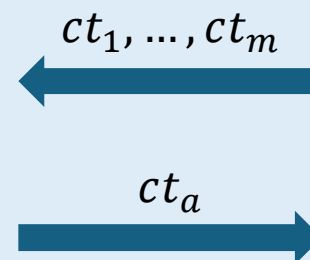
- Preprocess: $d^m \cdot \text{poly}(d, m, \log|R|) \cdot O(m \cdot (\log m + \log d + \log \log|R|))^m$
- f -Evaluation: $\text{poly}(d, m, \log|R|)$ (vs. Naïve $d^m \cdot \text{polylog}|R|$)

Construction Overview: [LMW23]

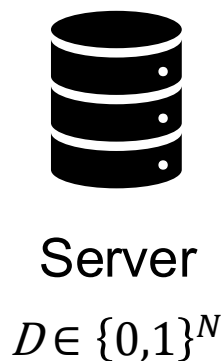
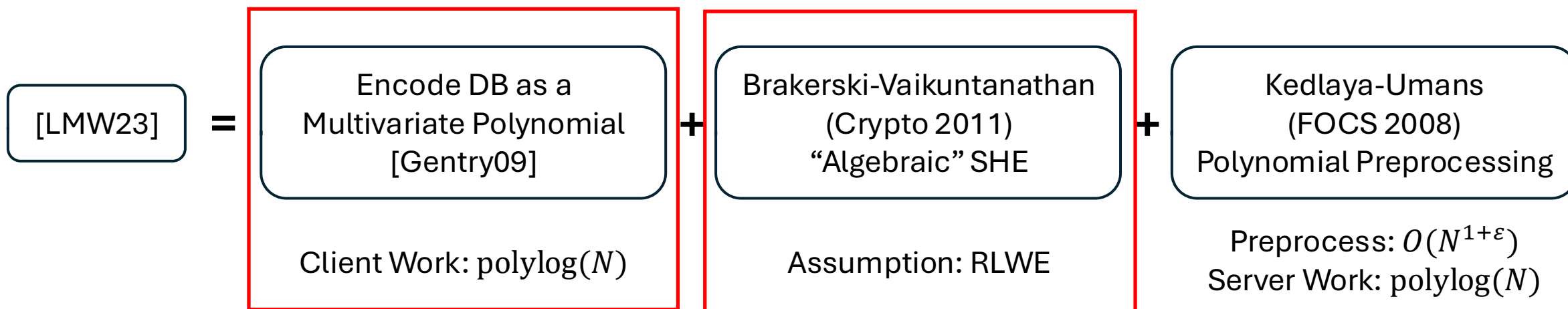


$$f_D \in \mathbb{Z}_p[X_1, \dots, X_m] \quad (p: \text{prime}, p^m \geq N)$$

$$ct_a \leftarrow \text{SHE.Eval}(f_D; ct_1, \dots, ct_m)$$

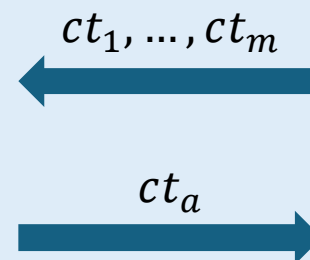


Construction Overview: [LMW23]

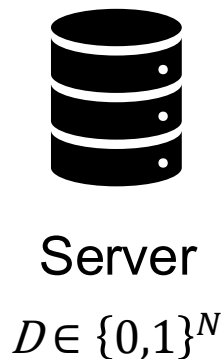
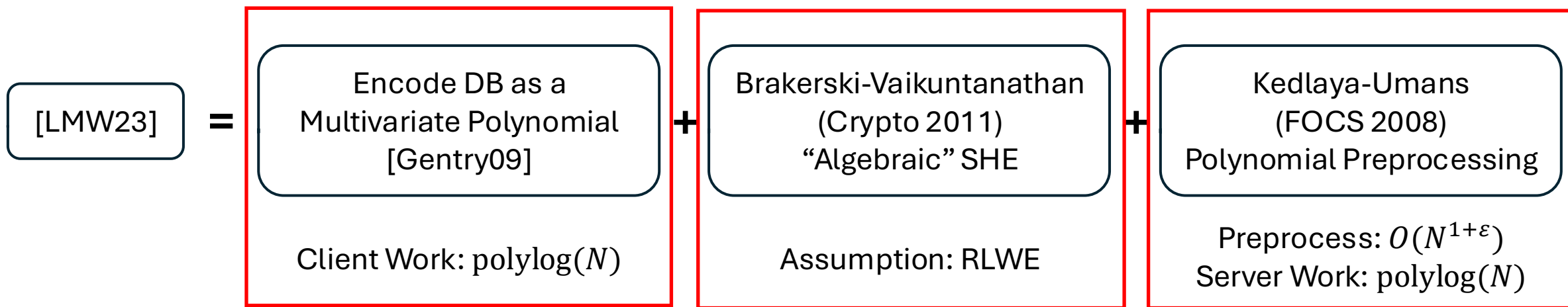


$$f_D \in \mathbb{Z}_p[X_1, \dots, X_m] \quad (p: \text{prime}, p^m \geq N)$$

$$ct_a \leftarrow \hat{f}_D(ct_1, \dots, ct_m)$$



Construction Overview: [LMW23]



$$f_D \in \mathbb{Z}_p[X_1, \dots, X_m] \quad (p: \text{prime}, p^m \geq N)$$

$$\tilde{f}_D \leftarrow \text{Preprocess}(\hat{f}_D)$$

$$ct_a \leftarrow \hat{f}_D(ct_1, \dots, ct_m) \text{ using } \tilde{f}_D$$

$$ct_1, \dots, ct_m$$

$$ct_a$$

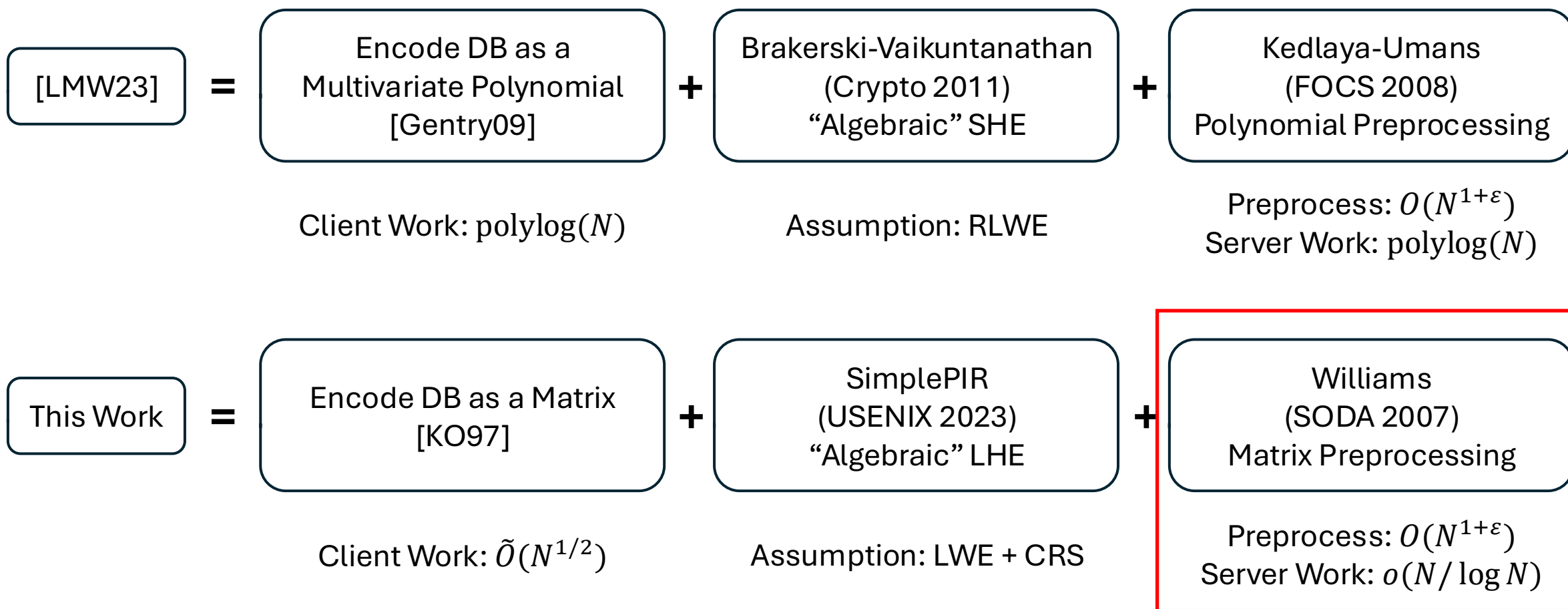
Why doesn't [LMW23] generalize to LWE?

Theorem ([KU08,LMW23]). Let $R = \mathbb{Z}_q[X, Y]/\langle X^n + 1, Y^D \rangle$. For any $f \in R[X_1, X_2, \dots, X_m]$, with individual $\deg < d$, there is a preprocessing scheme for f -evaluation with running time:

- Preprocess: $d^m \cdot \text{poly}(d, m, \log|R|) \cdot O(m \cdot (\log m + \log d + \log \log|R|))^m$
- f -Evaluation: $\text{poly}(d, m, \log|R|)$

- [KU08] requires the number of variables m to be small.
 - An LWE ciphertext, however, is a vector/matrix of $\text{poly}(\lambda)$ many $\mathbb{Z}_q$ elements.
 - c.f. Scheme based on Module-LWE with “constant” rank is already given in [LMW23].

Construction Overview



Fast MatVec Multiplication with Preprocessing

Theorem ([Wil07]). Let R be a finite ring. For any $\varepsilon \in (0,1)$ and matrix $D \in R^{m \times m}$, there is a preprocessing scheme for D -MatVecMult with running time:

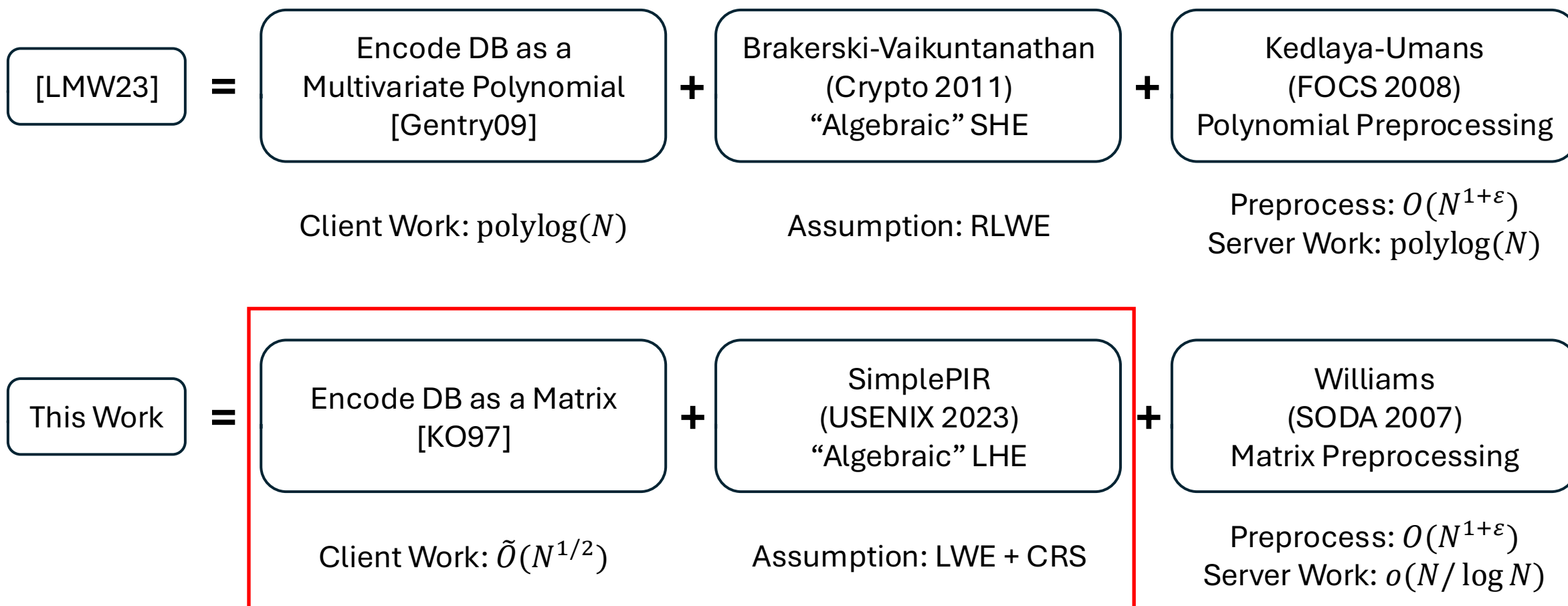
- Preprocess: $O(m^{2+\varepsilon \cdot \log |R|})$ R -ops
- D -MatVecMult: $O(m^2 / (\varepsilon \log m)^2)$ R -ops

- Main idea: Partition the matrix into blocks of size $(\varepsilon \log m) \times (\varepsilon \log m)$
- Preprocess: Per block, precompute MatVecMult for all possible inputs

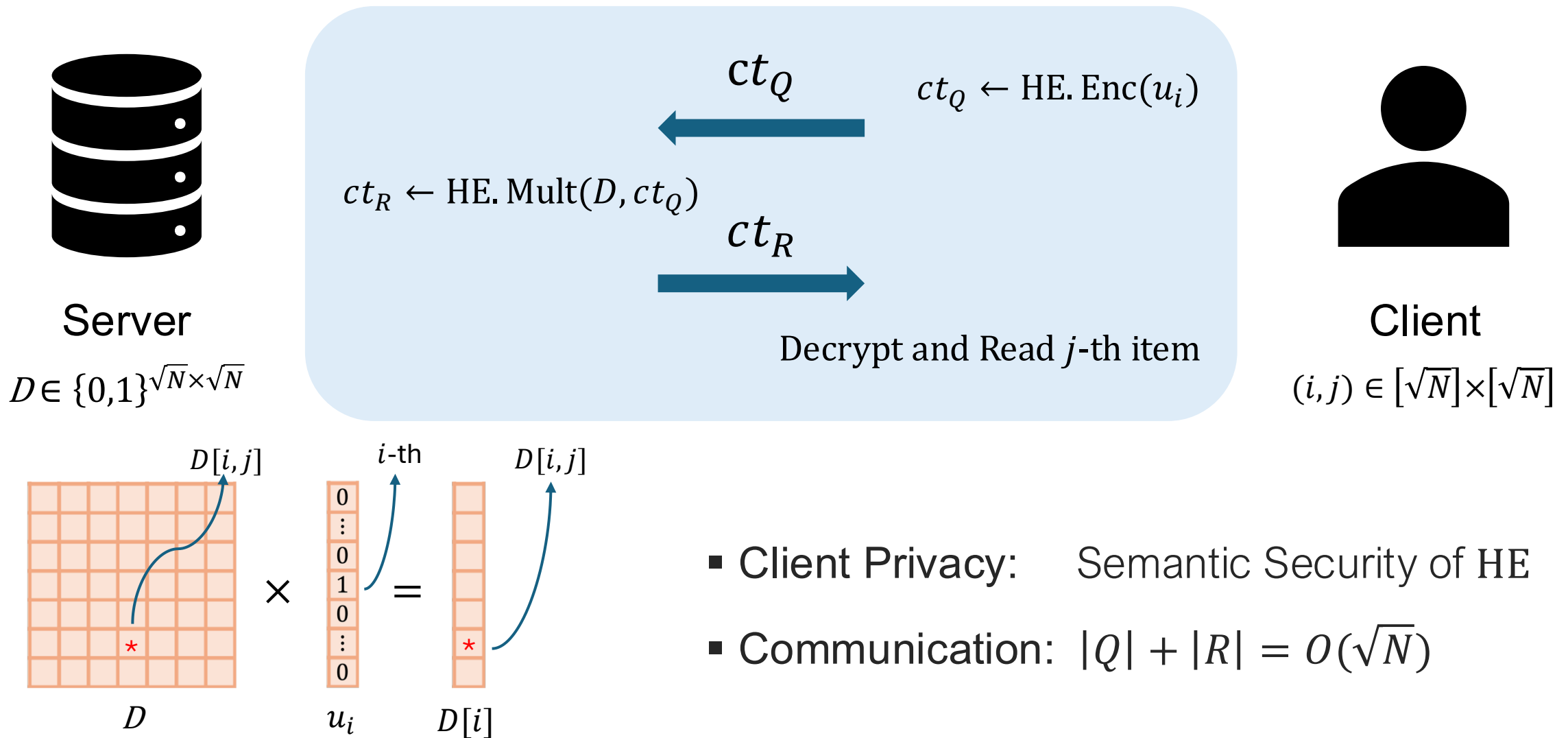
$$\checkmark \#(\text{blocks}) \times \#(\text{possible inputs}) \times \text{MatMult} = \frac{m^2}{(\varepsilon \log m)^2} \times |R|^{\varepsilon \log m} \times O(\varepsilon \log m)^2 = O(m^{2+\varepsilon \cdot \log |R|})$$

- Evaluate: $\frac{m^2}{(\varepsilon \log m)^2}$ Lookups (+ Careful summation)

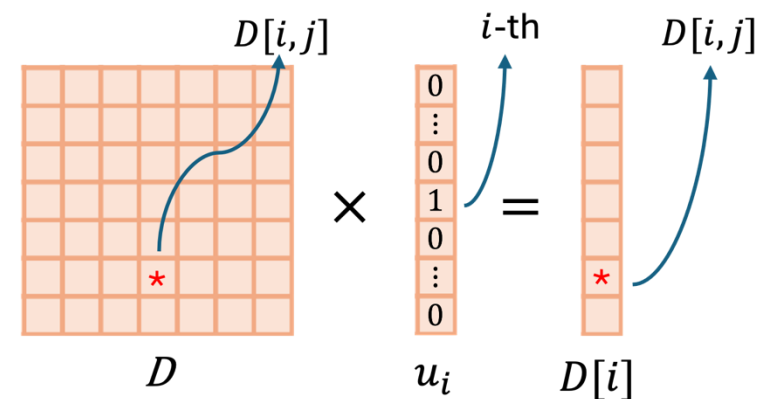
Construction Overview



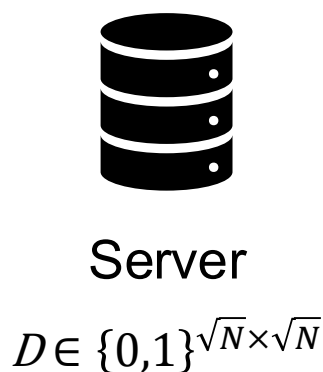
PIR from Linearly Homomorphic Encryption



PIR from LWE Encryption

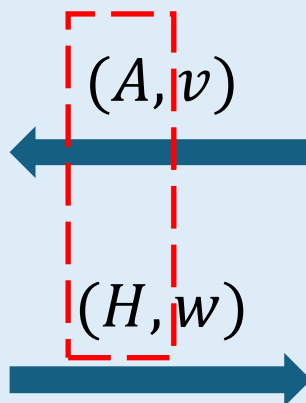


Do not depend on (i, j) !



$$\boxed{H \leftarrow DA}$$

$$\boxed{w \leftarrow Dv}$$



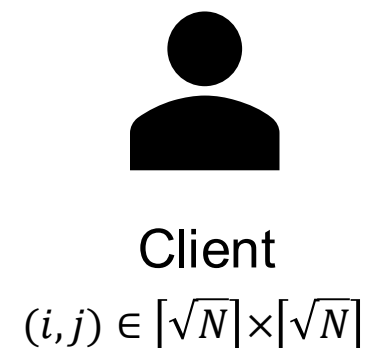
$$s \leftarrow \mathbb{Z}_q^n \quad A \leftarrow \mathbb{Z}_q^{\sqrt{N} \times n}$$

$$v \leftarrow As + \left\lfloor \frac{q}{2} \right\rfloor \cdot u_i + e$$

small

$$w - Hs = \left\lfloor \frac{q}{2} \right\rfloor \cdot Du_i + De$$

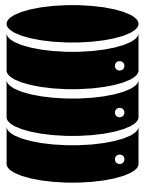
small



Bottleneck?!

SimplePIR [Henzinger-Hong-Corrigan-Gibbs-Meiklejohn-Vaikuntanathan; USENIX'23]

Do not depend on (i, j) !

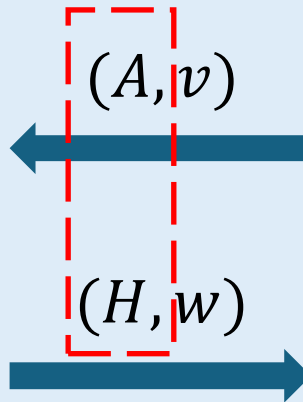


Server

$$D \in \{0,1\}^{\sqrt{N} \times \sqrt{N}}$$

$$\text{CRS } A \in \mathbb{Z}_q^{\sqrt{N} \times n}$$

$$\begin{aligned} H &\leftarrow DA \\ w &\leftarrow Dv \end{aligned}$$



$$\begin{aligned} s &\leftarrow \mathbb{Z}_q^n & A &\leftarrow \mathbb{Z}_q^{\sqrt{N} \times n} \\ v &\leftarrow As + \left\lfloor \frac{q}{2} \right\rfloor \cdot u_i + e \end{aligned}$$

$$w - Hs = \left\lfloor \frac{q}{2} \right\rfloor \cdot Du_i + De$$

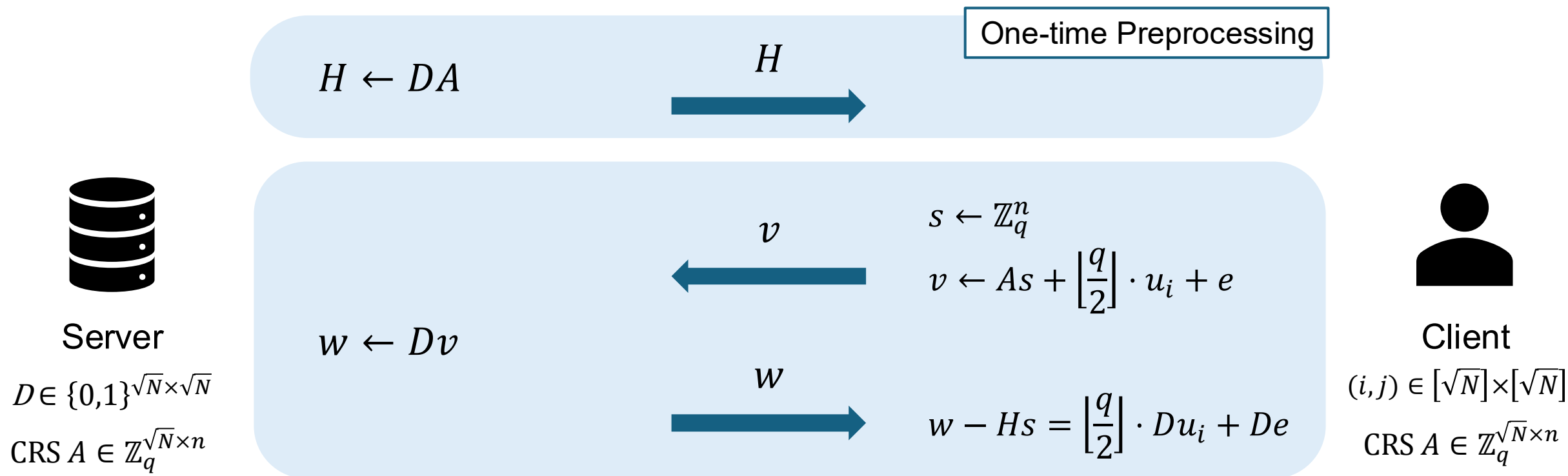


Client

$$(i, j) \in [\sqrt{N}] \times [\sqrt{N}]$$

$$\text{CRS } A \in \mathbb{Z}_q^{\sqrt{N} \times n}$$

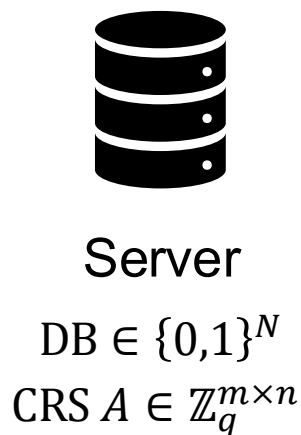
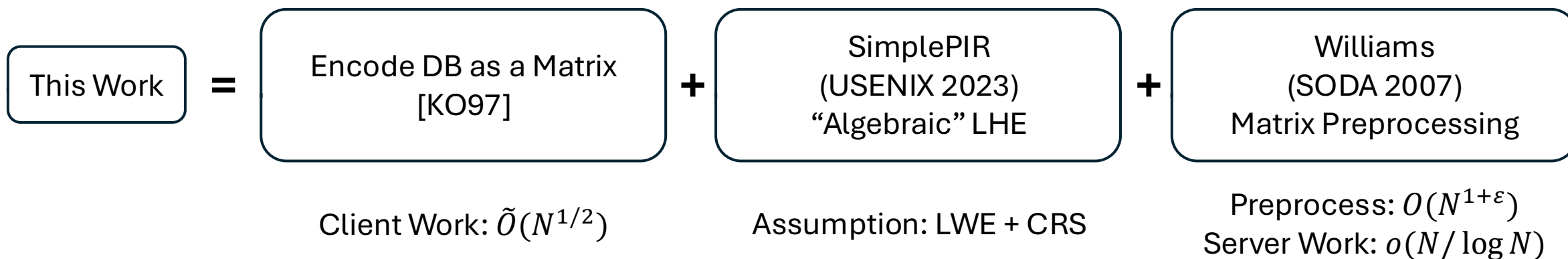
SimplePIR [Henzinger-Hong-Corrigan-Gibbs-Meiklejohn-Vaikuntanathan; USENIX'23]



- Algebraic “LHE”: Homomorphic MatVecMult is itself a MatVecMult over a “Nice” Ciphertext Ring
- SimplePIR: Algebraic LHE with Ciphertext Ring $R = \mathbb{Z}_q$

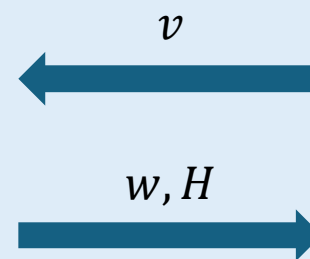
Our Construction

- ❖ Simple construction, but still requires careful analysis:
 - n and q depend on N to support homomorphism.
 - ✓ CRS is required to remove dependency on n
 - ✓ need a CRT layer to suppress dependency on q
 - ✓ plaintext modulus p should be set with care



$D \in \mathbb{Z}_p^{m \times m}$ ($m^2 \cdot \log p \geq N$)
 $\tilde{D} \leftarrow \text{Preprocess}(D)$, $H \leftarrow DA$

$w \leftarrow Dv$ using $\tilde{D}$



Summary

- First DEPIR Scheme from plain LWE (vs. Ring-LWE in [LMW23])
 - Simple construction from SimplePIR & Williams' Matrix Preprocessing
- Limitations
 - Only “Barely” Doubly-Efficient: $o(N/\log N)$ online work (vs. $\text{polylog}(N)$ in [LMW23])
 - Require CRS for technical reasons
- Open problems
 - Better efficiency, e.g. $O(N^{1-\varepsilon})$
 - Removing CRS assumption
 - RAM-FHE from plain LWE

Thank You!