KU LEUVEN

Pirouette: Query Efficient Single-Server PIR

Jiayi Kang, Leonard Schild
COSIC, KU Leuven

28 May 2026 @ NTNU

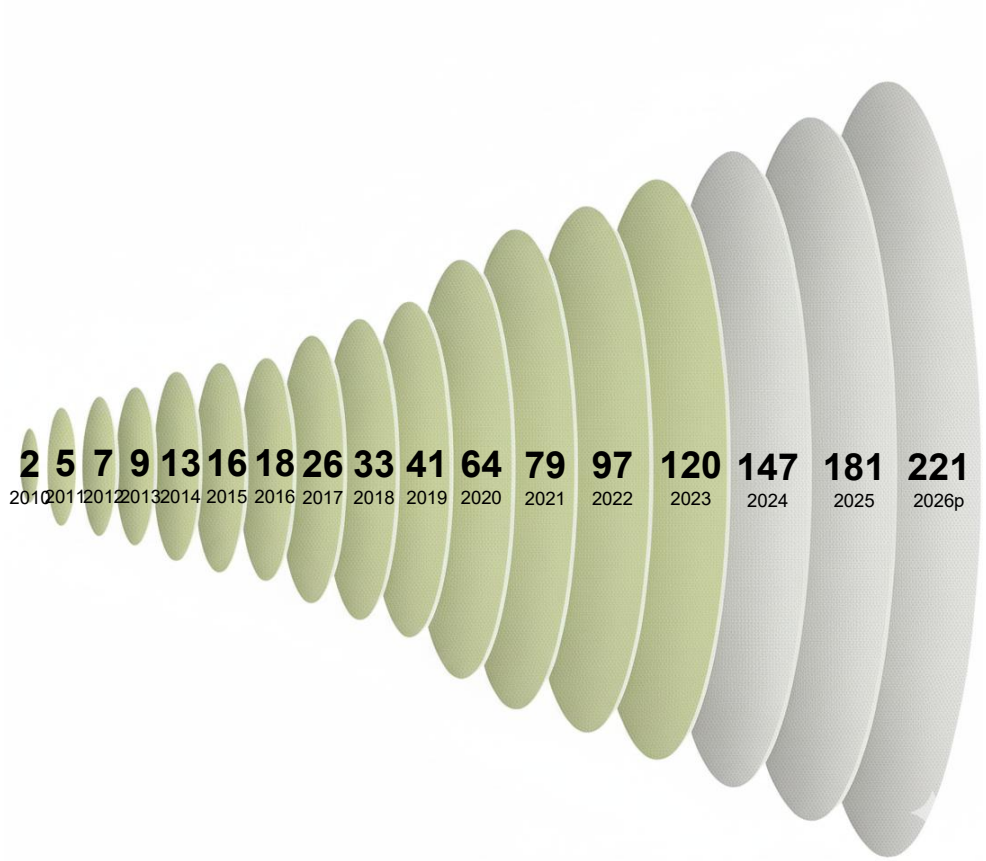
About me

- Postdoc at COSIC, KU Leuven (also where I accomplished my PhD)
- Starting July 2026: Postdoc at Monash University

Research Interests:

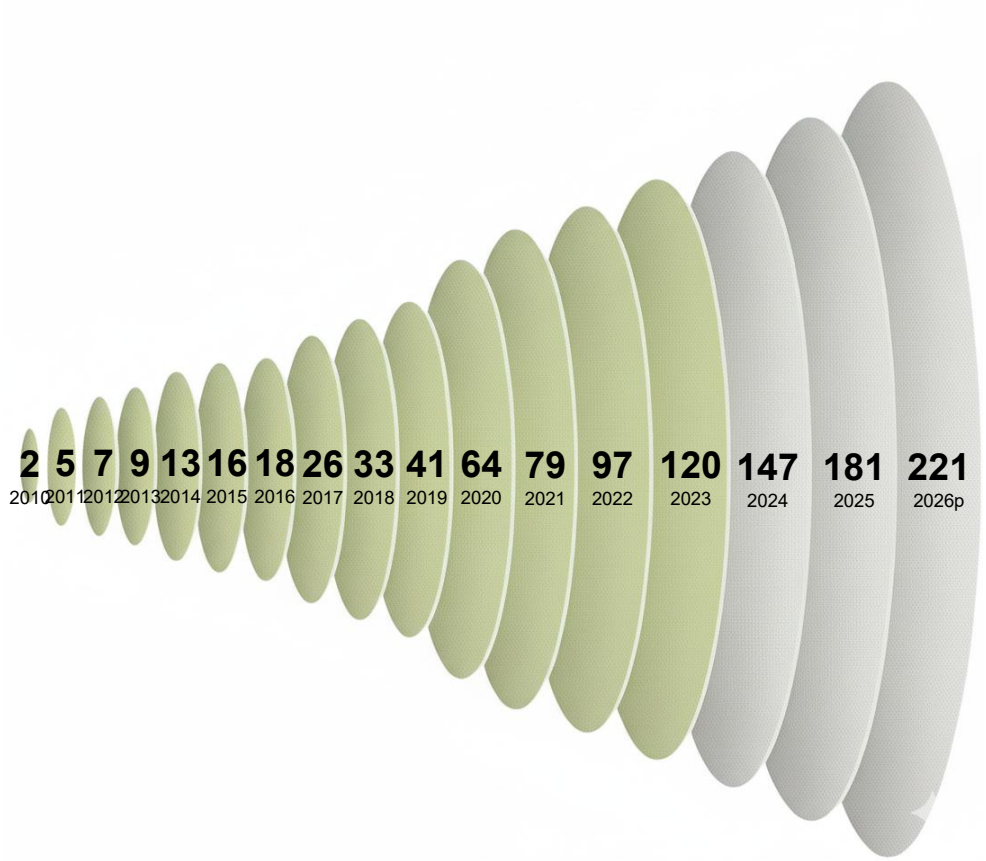
- Fully Homomorphic Encryption (FHE)
  - theoretical foundations (e.g. bootstrapping)
  - applications (e.g. secure machine learning, private information retrieval)
- Zero-knowledge proofs (ZKP), in particular lattice-based ZKP
- Intersection of FHE and ZKP
  - cryptographic protocols that provide *both* privacy and integrity

Big data vs. small local devices

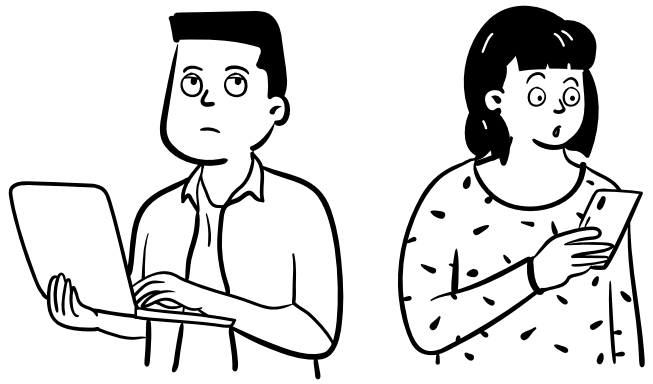


Prediction from Hinrich foundation,
Units in ZB (= 10^9 TB)

Big data vs. small local devices



Prediction from Hinrich foundation,
Units in ZB (= 10^9 TB)

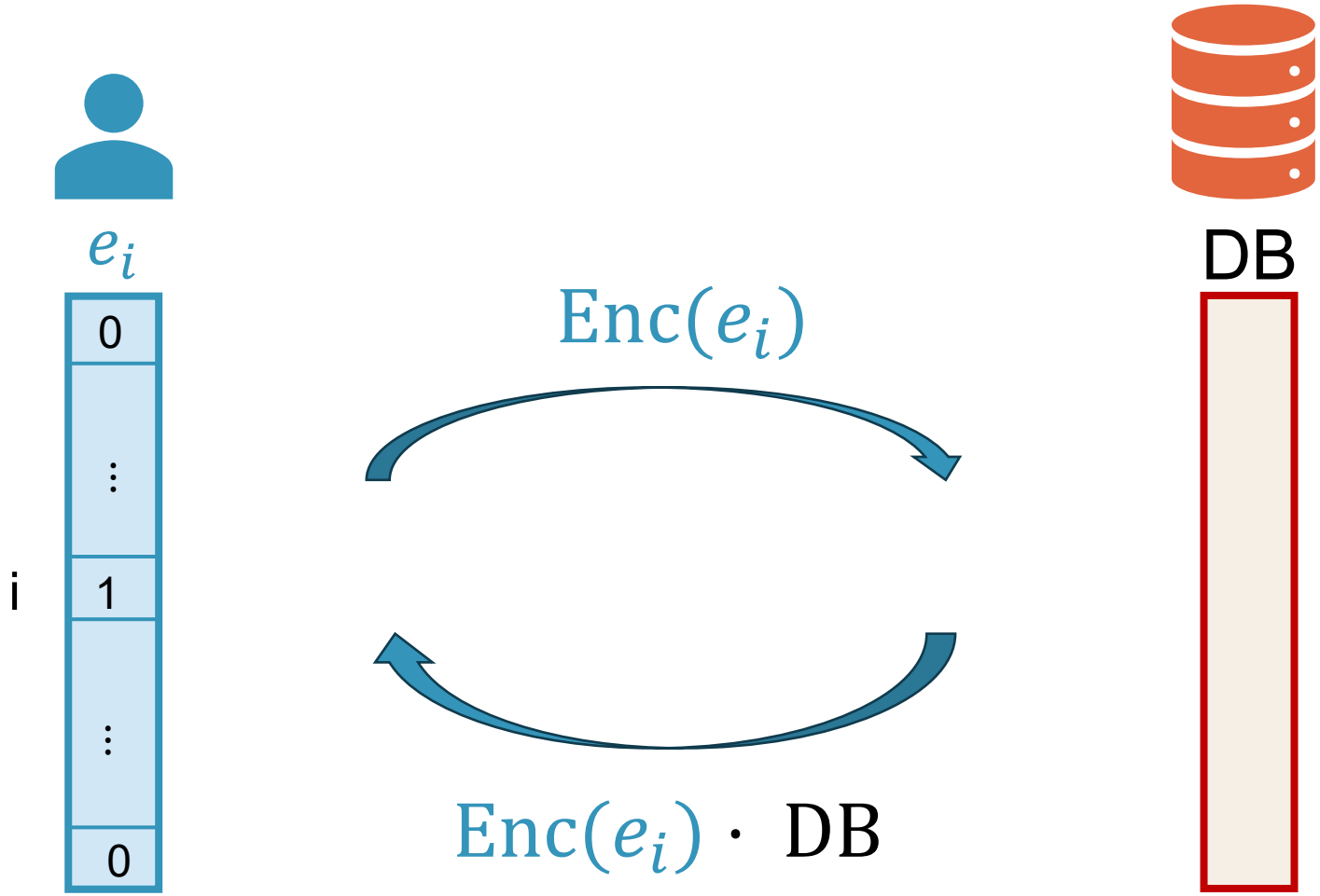


Private information retrieval

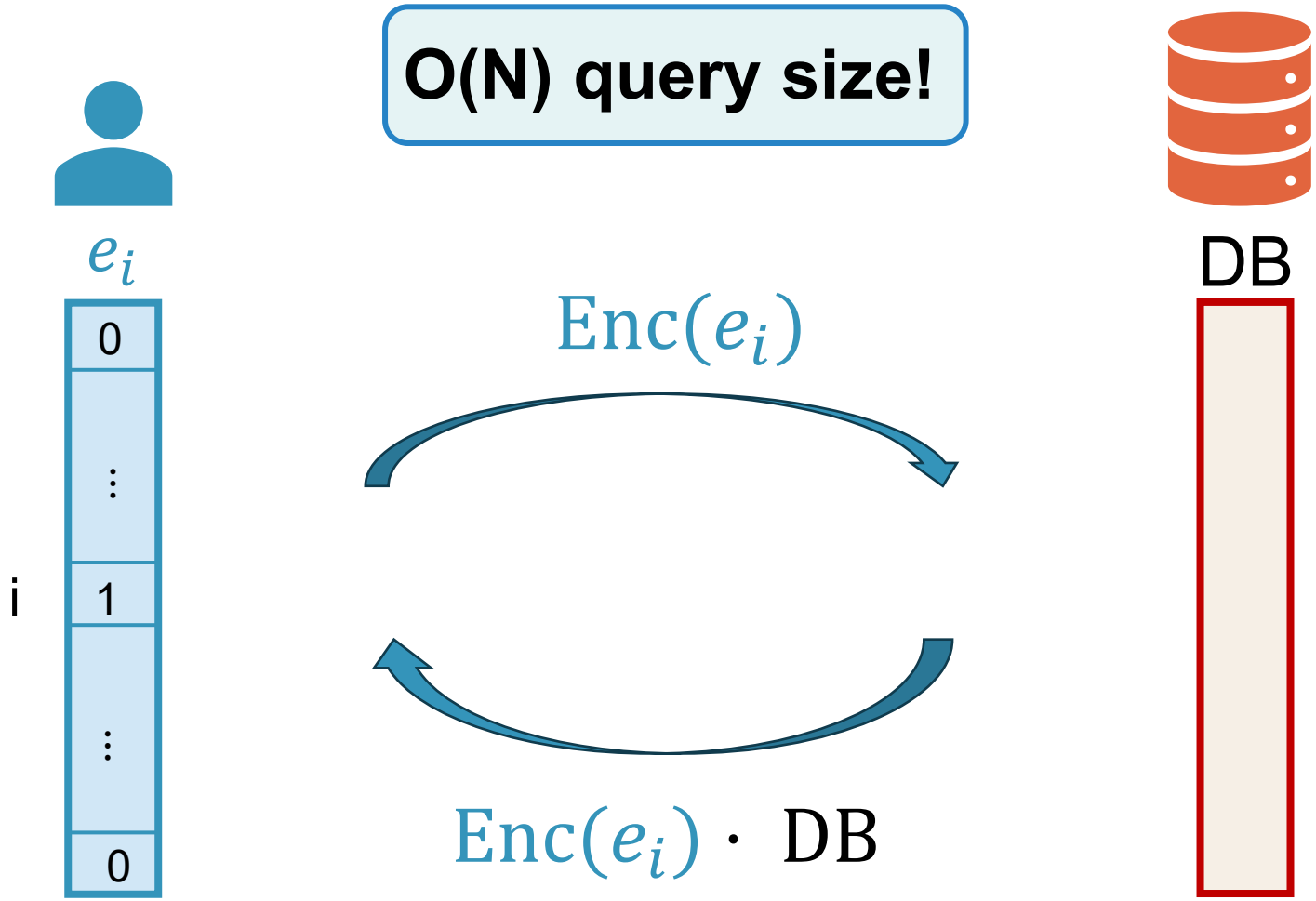
- holds query index i $\in \{1, \ldots, N\}$
- learns DB[i]

- holds database DB $\in \mathbb{F}^N$
- learns nothing about the index i

Naïve PIR from homomorphic encryption



Naïve PIR from homomorphic encryption



Modern PIR database structures [KPS25]

- Hypercube structure [ACLS18, MCR21,…]
  - k-dimensional DB of size $N^{1/k} \times \cdots \times N^{1/k}$
  - query size $O(\mathrm{k} \cdot N^{1/k})$

Modern PIR database structures [KPS25]

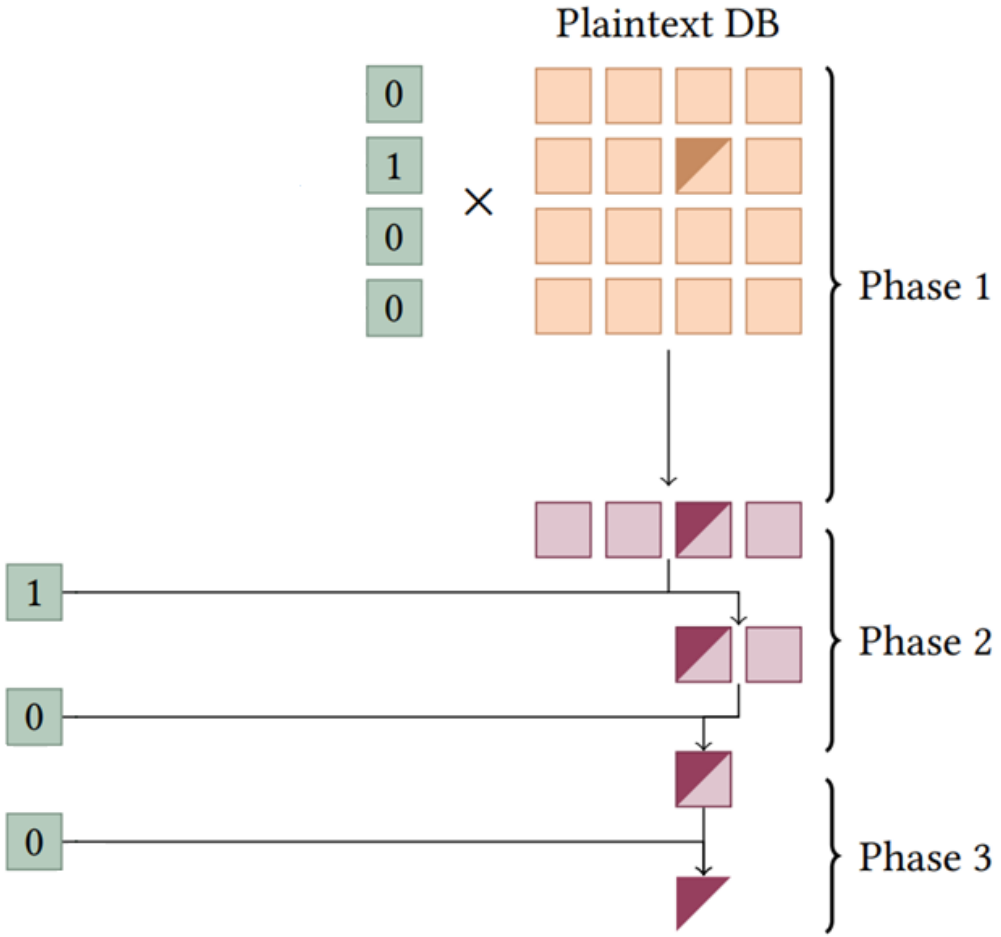
- Hypercube structure [ACLS18, MCR21,…]
  - k-dimensional DB of size $N^{1/k} \times \cdots \times N^{1/k}$
  - query size $O(\mathrm{k} \cdot N^{1/k})$
- Tree structure [PT20, …]
  - log(N)-dimensional DB of size 2 in each dimension
  - query size $O(logN)$

Modern PIR database structures [KPS25]

- Hypercube structure [ACLS18, MCR21,…]
  - k-dimensional DB of size $N^{1/k} \times \cdots \times N^{1/k}$
  - query size $O(\mathrm{k} \cdot N^{1/k})$
- Tree structure [PT20, …]
  - log(N)-dimensional DB of size 2 in each dimension
  - query size $O(logN)$
- Combined structure in Respire [BMW24]
  - DB of size $N = 2^{\nu_1+\nu_2+\nu_3}$ in a $(1+\nu_2+\nu_3)$-dimensional hypercube
  - query index $idx \in \mathbb{Z}_{2^{\nu_1}} \times \mathbb{Z}_2^{\nu_2} \times \mathbb{Z}_2^{\nu_3}$

Example DB structure in Respire

- Example: $\nu_1 = 2, \nu_2 = 2, \nu_3 = 1$.
- Query index $idx \in \mathbb{Z}_{2^{\nu_1}} \times \mathbb{Z}_2^{\nu_2} \times \mathbb{Z}_2^{\nu_3}$
  - the first component is represented as a unit vector
- DB entries encoded as (part of) polynomials
- Query complexity: $O(2^{\nu_1} + \nu_2 + \nu_3)$

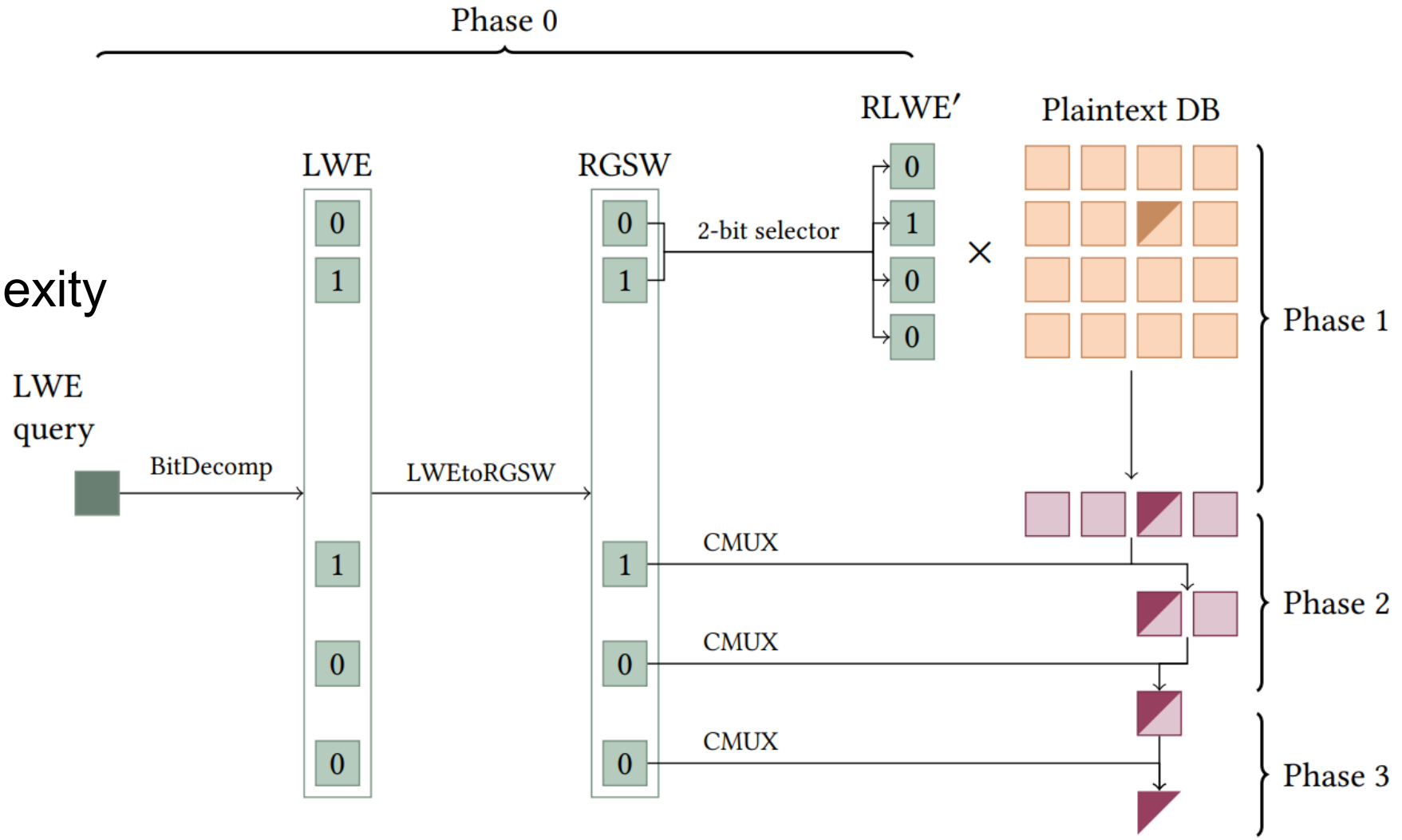


Overview of Pirouette protocol

- Query: LWE(i),

where i ∈ {1, … , N}

- $O(1)$ query complexity



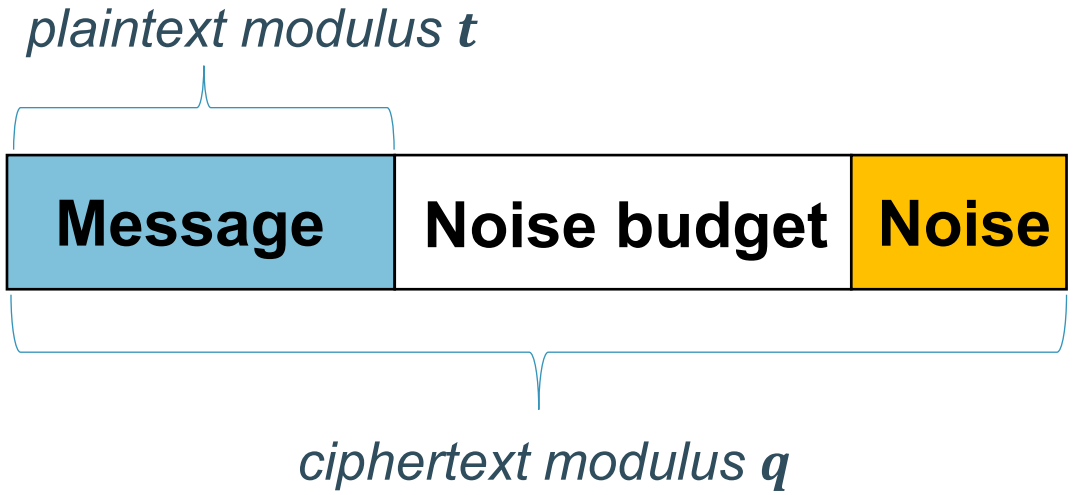
KU LEUVEN

LWE encryption

- LWE encryption

$$\mathrm{LWE}(m) = (\mathbf{a}, b) \in \mathbb{Z}_q^{n+1}$$
$$b = \mathbf{a} \cdot \mathbf{s} + \Delta \cdot m + e \mod q$$

where $\mathbf{a} \leftarrow \mathbb{Z}_q^n$ is uniformly sampled, $\Delta = \frac{q}{t}$



LWE encryption

- LWE encryption

$$\mathrm{LWE}(m) = (\mathbf{a}, b) \in \mathbb{Z}_q^{n+1}$$
$$b = \mathbf{a} \cdot \mathbf{s} + \Delta \cdot m + e \mod q$$

where $\mathbf{a} \leftarrow \mathbb{Z}_q^n$ is uniformly sampled, $\Delta = \frac{q}{t}$

- Programmable bootstrapping (PBS)
  - evaluate an arbitrary function while refreshing noise
    - e.g. bit decomposition
  - key component: blind rotation
  - typical plaintext moduli: t=4 or 5 bits

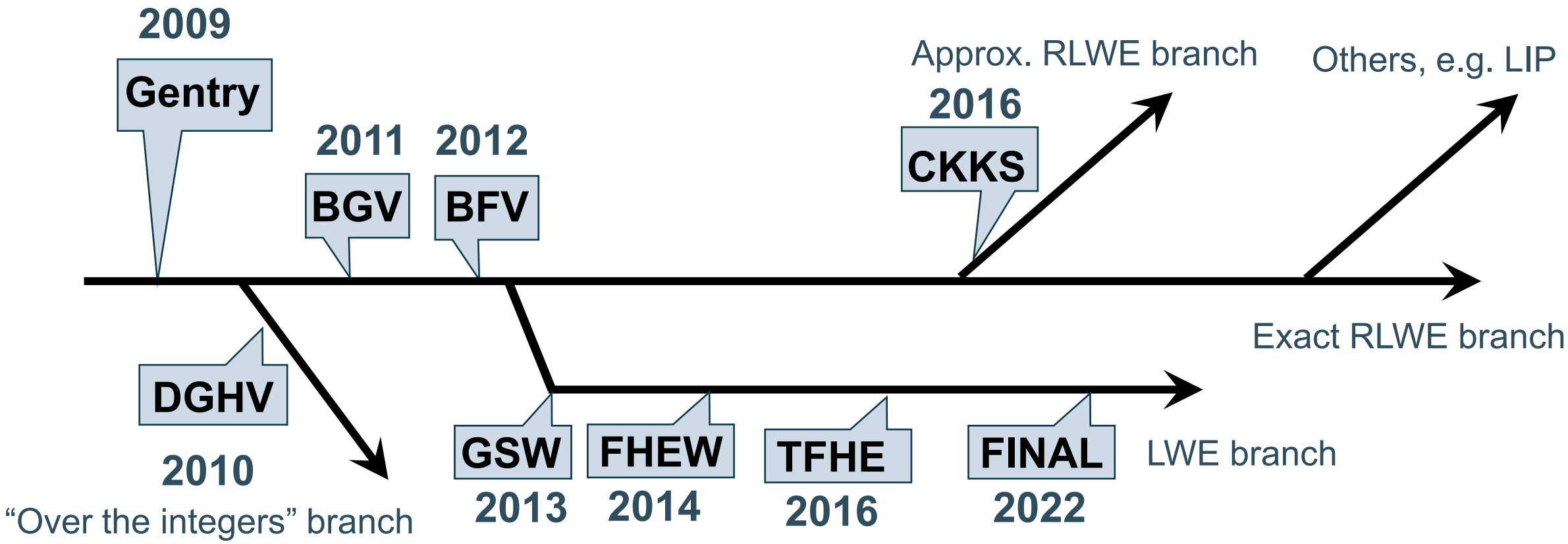
High-precision LWE

- Consider LWE encryption with large plaintext moduli
  - e.g. t=25 bits
- High-precision bit decomposition, decomposing 25 bits requires
  - 50 blind rotations in [LMP22]

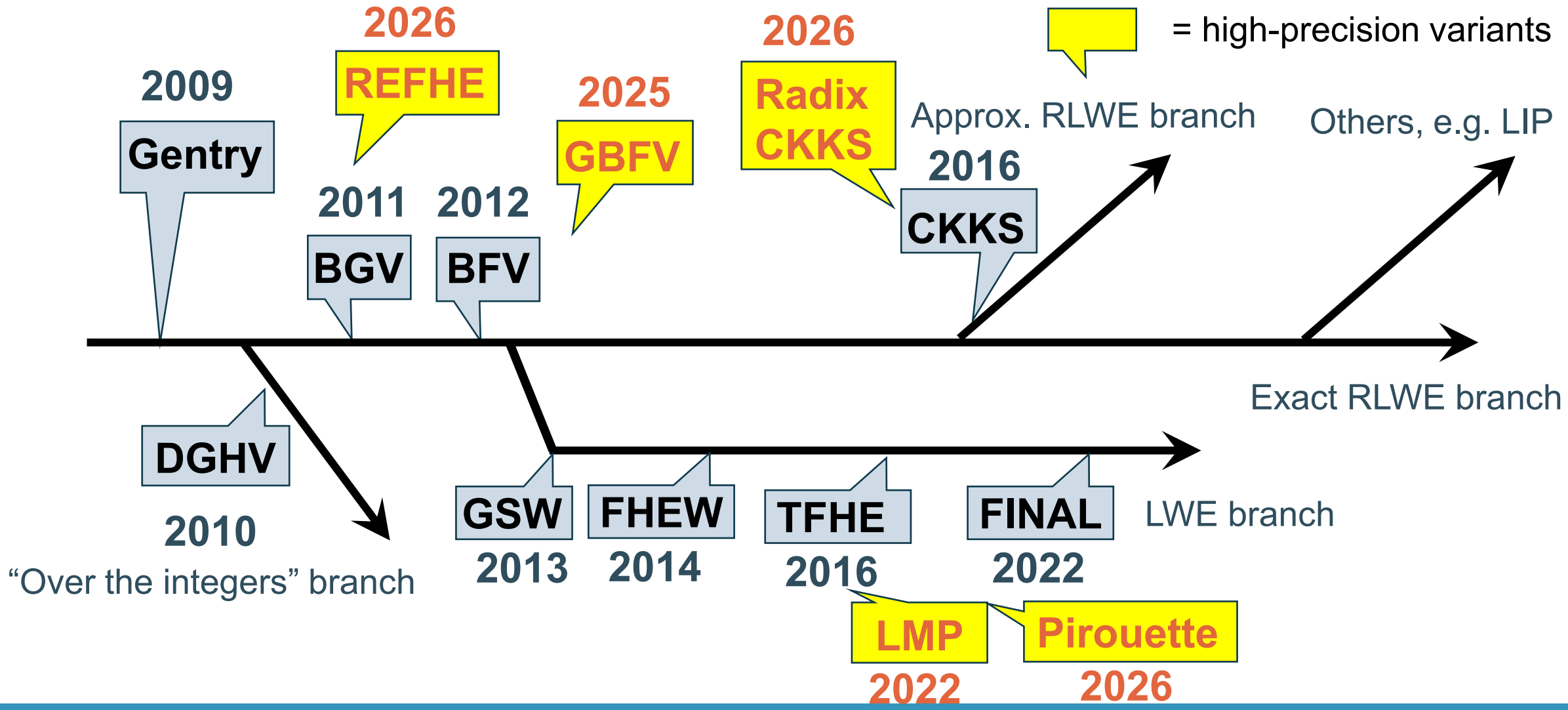
High-precision LWE

- Consider LWE encryption with large plaintext moduli
  - e.g. t=25 bits
- High-precision bit decomposition, decomposing 25 bits requires
  - 50 blind rotations in [LMP22]
  - 15 blind rotations in our work
- Key idea:
  1. decompose into five LWE of 5 bits (10 blind rotations)
  2. Multi-value PBS, decomposing each LWE into bits (5 blind rotations)

Overview of FHE schemes



Overview of high-precision FHE schemes



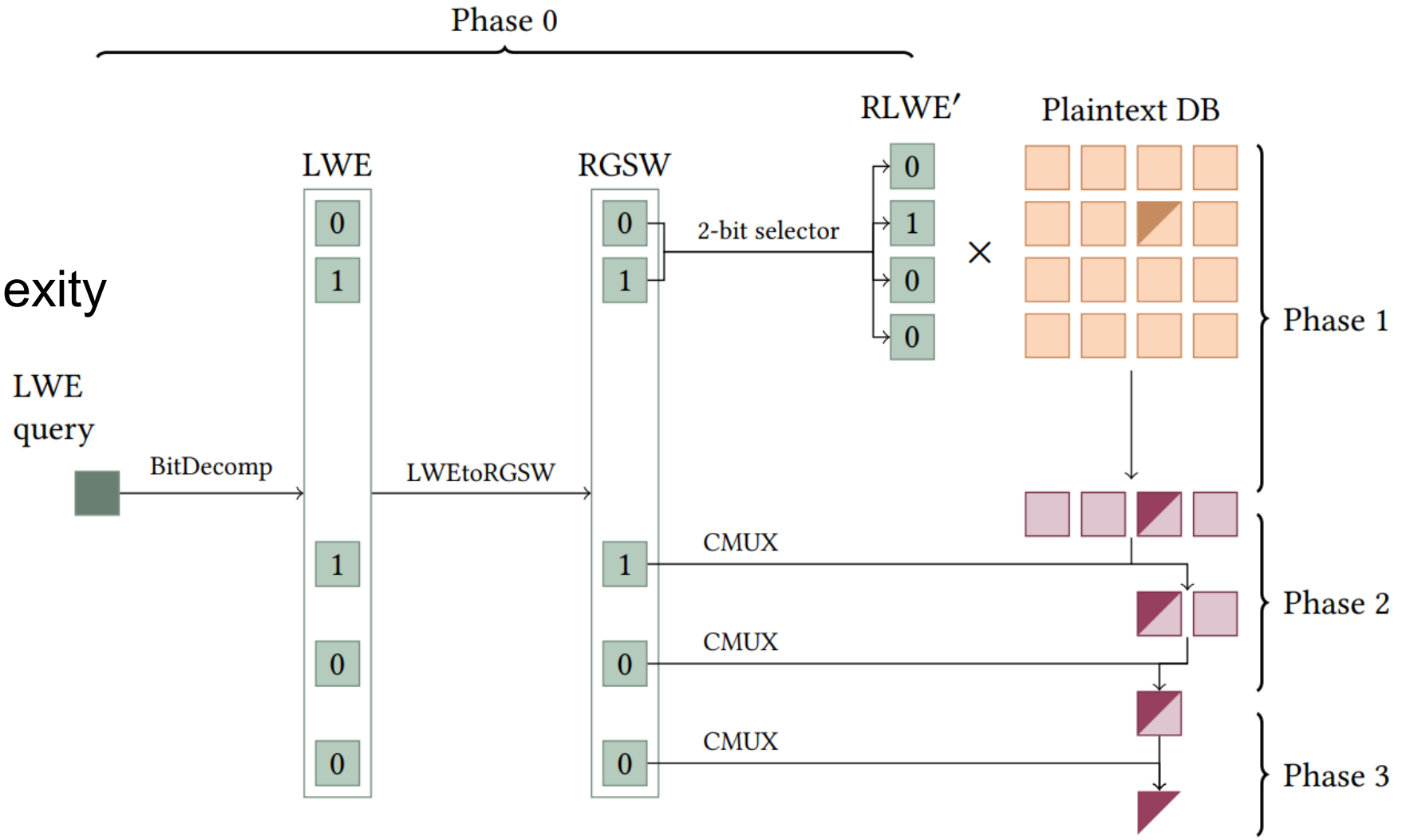
KU LEUVEN

Overview of Pirouette protocol

- Query: LWE(i),

where i ∈ {1, … , N}

- $O(1)$ query complexity



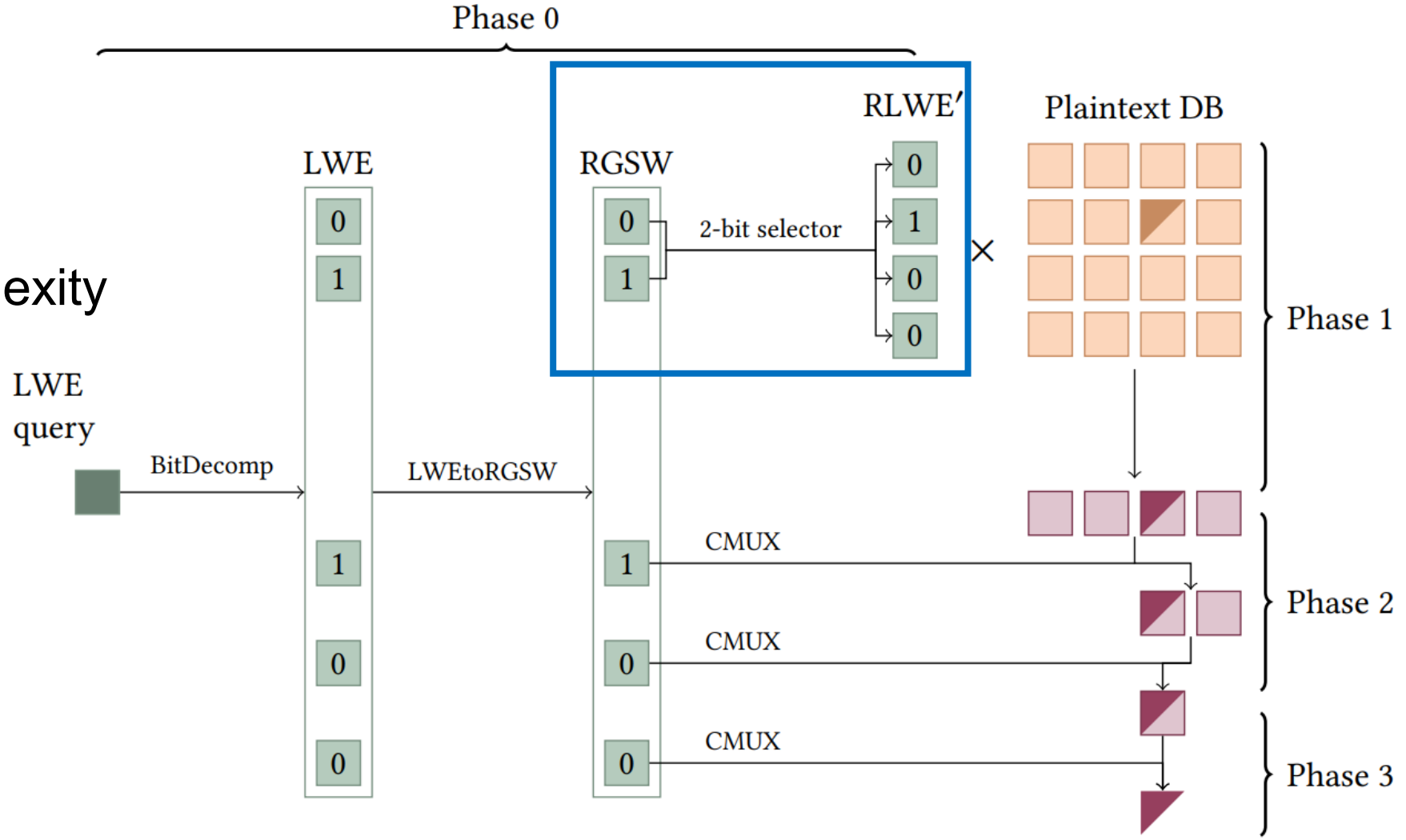
KU LEUVEN

Overview of Pirouette protocol

- Query: LWE(i),

where i ∈ {1, … , N}

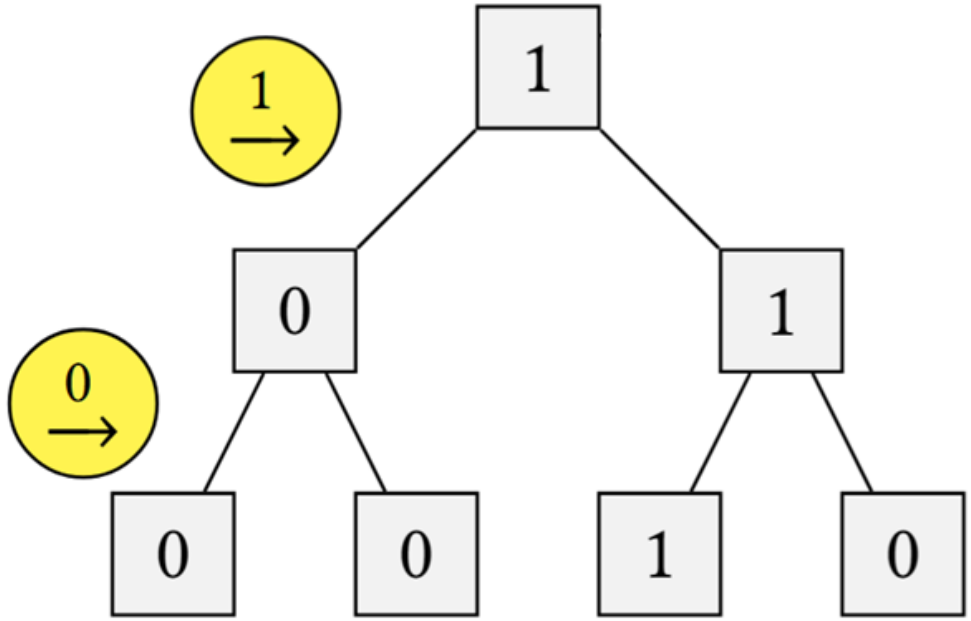
- $O(1)$ query complexity



KU LEUVEN

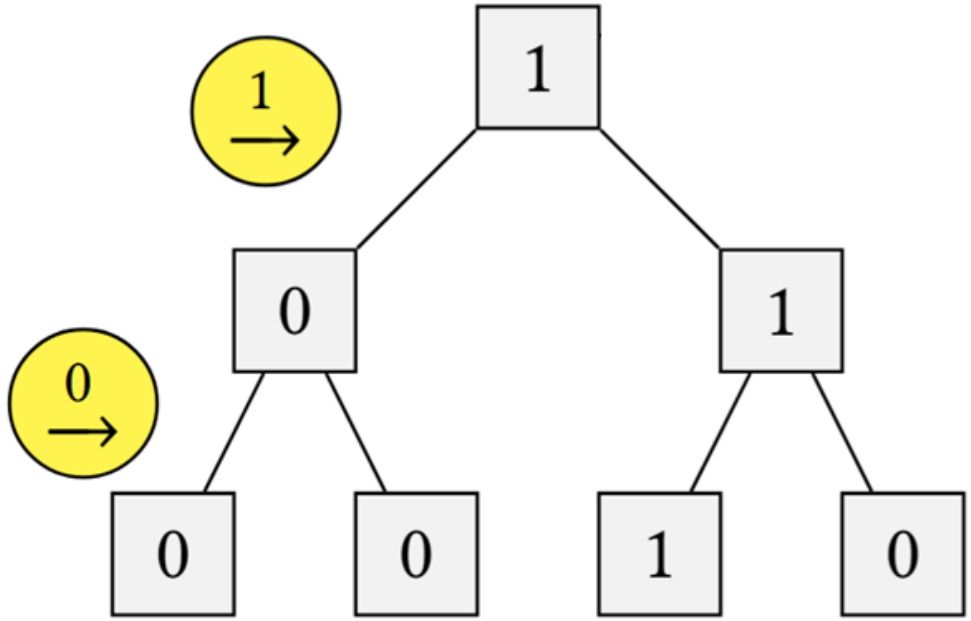
Construction of the unit vector

- Input: ν_1 RGSW ciphertexts
  - yellow circles
- Output: 2^{ν_1} RLWE' ciphertexts of a unit vector
  - grey circles
- Algorithm: tree traversal algorithm in [CDPP22] instantiated using RLWE'



Construction of the unit vector

- Input: ν_1 RGSW ciphertexts
  - yellow circles
- Output: 2^{ν_1} RLWE' ciphertexts of a unit vector
  - grey circles
- Algorithm: tree traversal algorithm in [CDPP22] instantiated using RLWE'
- Concurrent work: similar construction in VIA [LWZ26] (DMUX gates)



Comparison: query size

- Pirouette query: LWE(i)= $(\mathbf{a}, b) \in \mathbb{Z}_q^{n+1}$
  - for a database of 2^{25} records, query size is just 36 B
  - i.e. seed (32B) to derive $\mathbf{a}$ and the b component (4B)
  - $\Rightarrow$ 4B (32 bits) if the query seed is set once by the server, giving an expansion factor of 32/25 = 1.28
- T-Respire: combining transciphering [BPR24] with Respire
  - 336 B for the same DB of 2^{25} records

Holistic performance

Database	Metric	Respire	T-Respire	PIROUETTE
$2^{25} \times 256$ B	Offline Comm.	4 MB	91 MB	1.2 GB
(8 GB)	Query Size	14.8 KB	336 B	36 B
	Response Size		2.0 KB	
	Computation	30 s	486 s	60 s

Pirouette is the most applicable in scenarios where:

- online bandwidth, particularly C2S, is severely constrained
- offline bandwidth is widely available
  - e.g. wired offline & wireless online
- multi-core parallelization or HW accelerators
- building block for general cryptographic protocols

Conclusion

- Pirouette, a PIR protocol with small query size
  - high-precision bit decomposition for LWE ciphertexts
  - novel approach to reduce C2S communication

To appear in PET 2026
ia.cr/2025/680

jiayi.kang2@outlook.com
leonard.schild@esat.kuleuven.be