UQÀM

DEEL

DEpendable & Explainable Learning

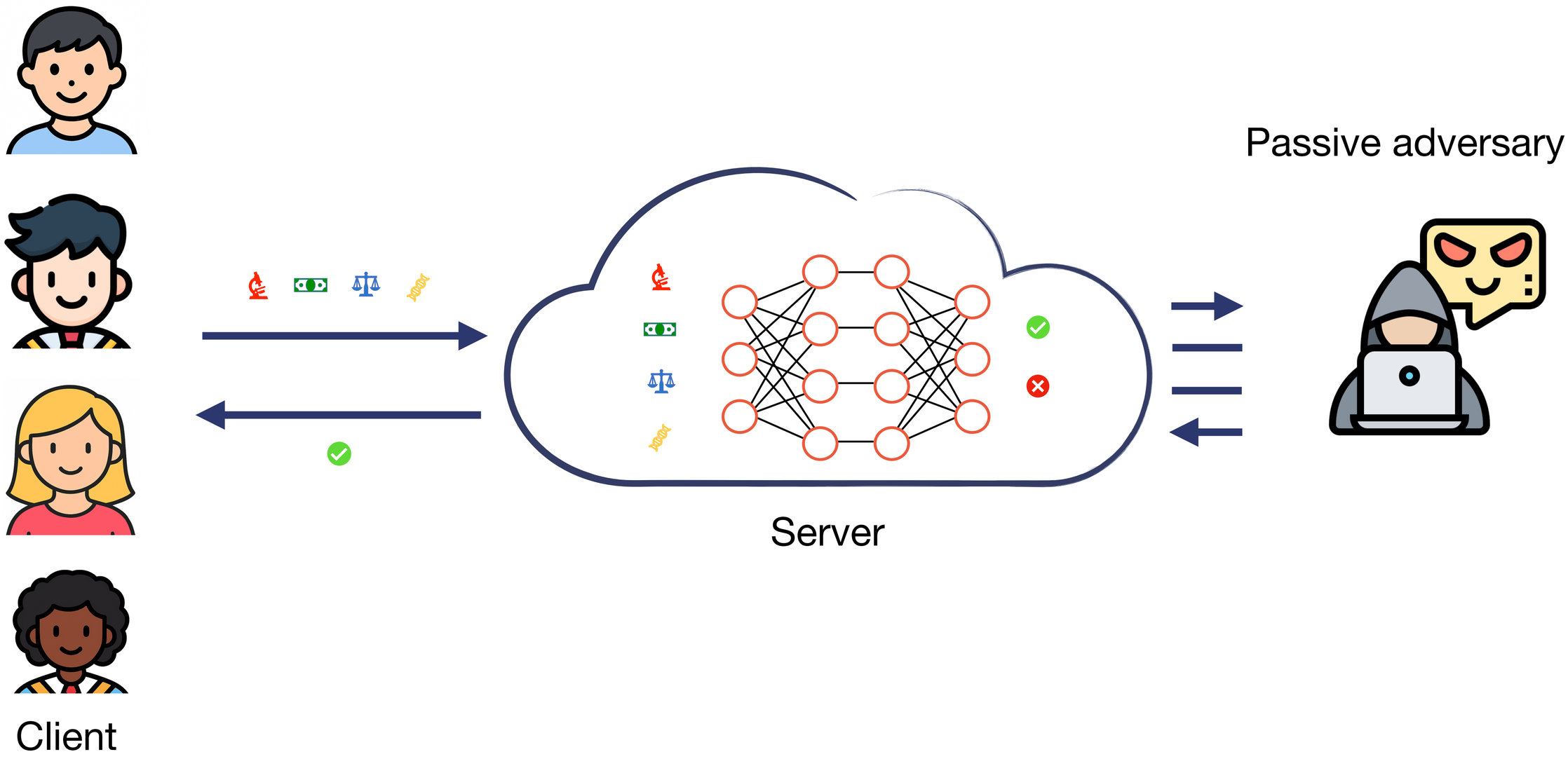
Oblivious Exact (Un)Learning of Extremely Randomized Trees

Sofiane Azogagh, Zelma Aubin Birba, Sébastien Gambs and Marc-Olivier Killijian

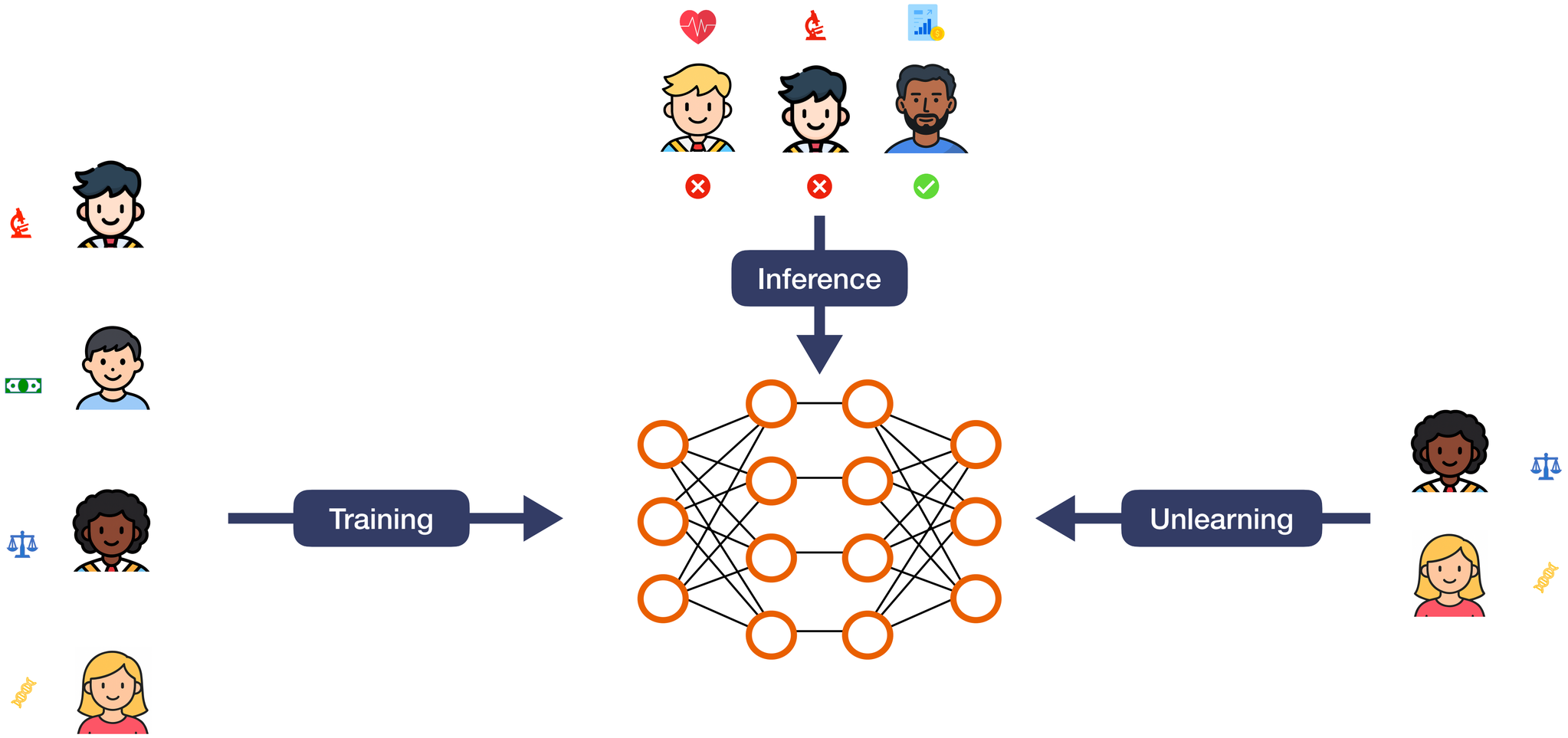
Slides

SATML'26 - March 24th 2026

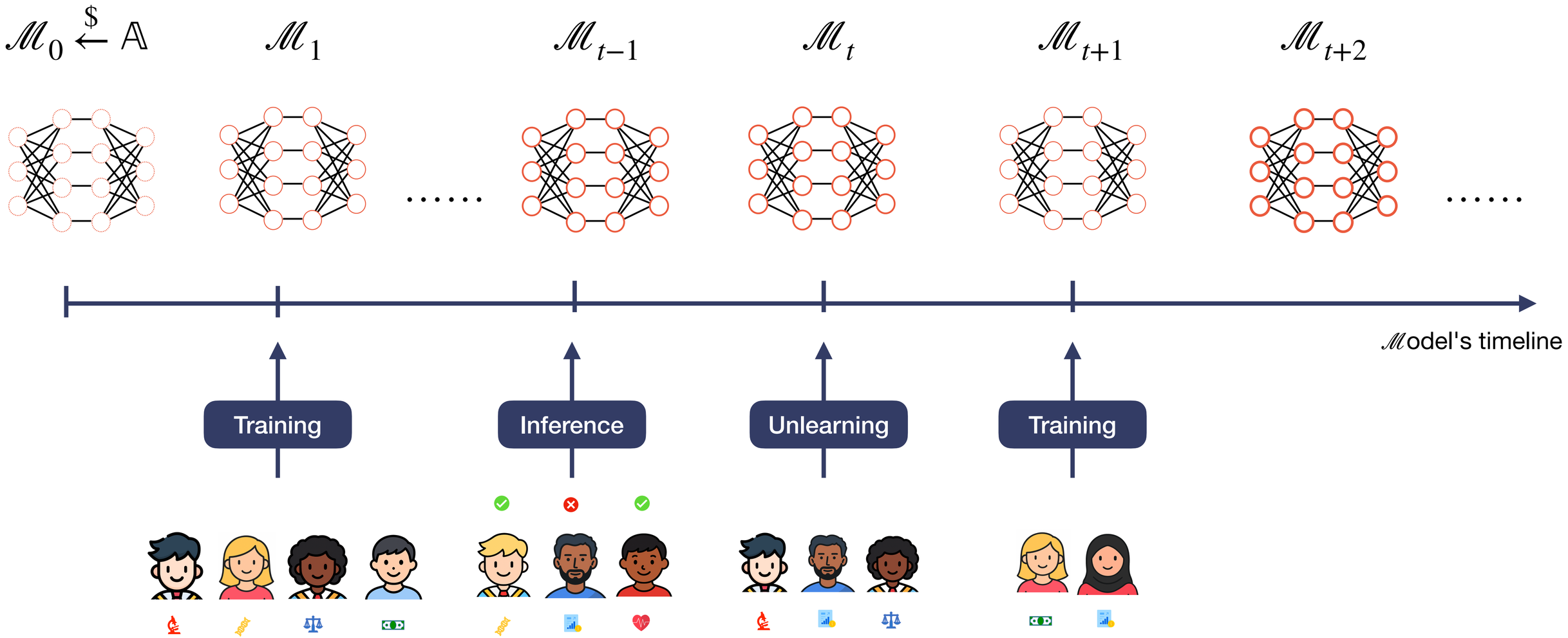
Outsourcing the computation



Different phases in ML

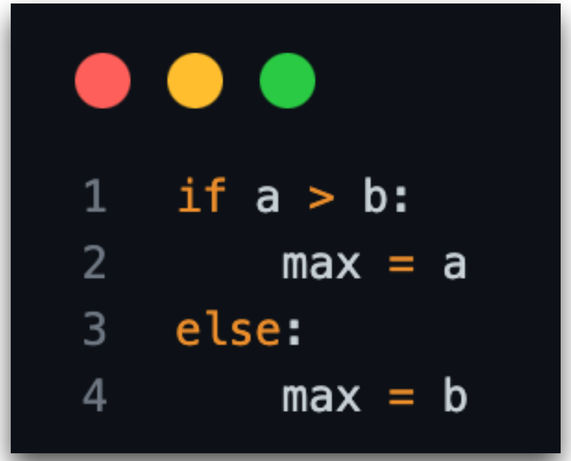


Online Learning



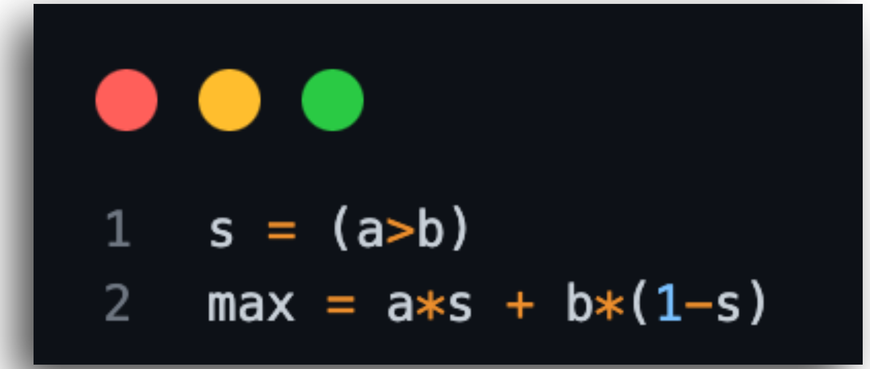
Oblivious computation

Just a definition



```
1   if a > b:
2       max = a
3   else:
4       max = b
```

Non oblivious



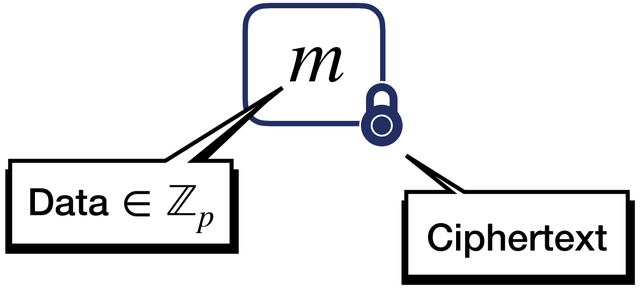
```
1   s = (a>b)
2   max = a*s + b*(1-s)
```

Oblivious

Oblivious algorithms are algorithms whose **access patterns** (e.g., which memory addresses they touch) and **control flow** (e.g., which branch they take) are **independent of the input data values**.

Oblivious computation

Fully Homomorphic Encryption (FHE)



TFHE

Addition

Multiplication

$x + y = x + y$

$x \times y = x \times y$

Arithmetic operations

Function evaluation

(Functional Bootstrapping)

$f(x) = f(x)$

$f(0)\ \ f(1)\ \ f(2)\ \ \ldots\ldots\ \ f(p-1)$

LUT : Look-Up-Table

Non arithmetic operations

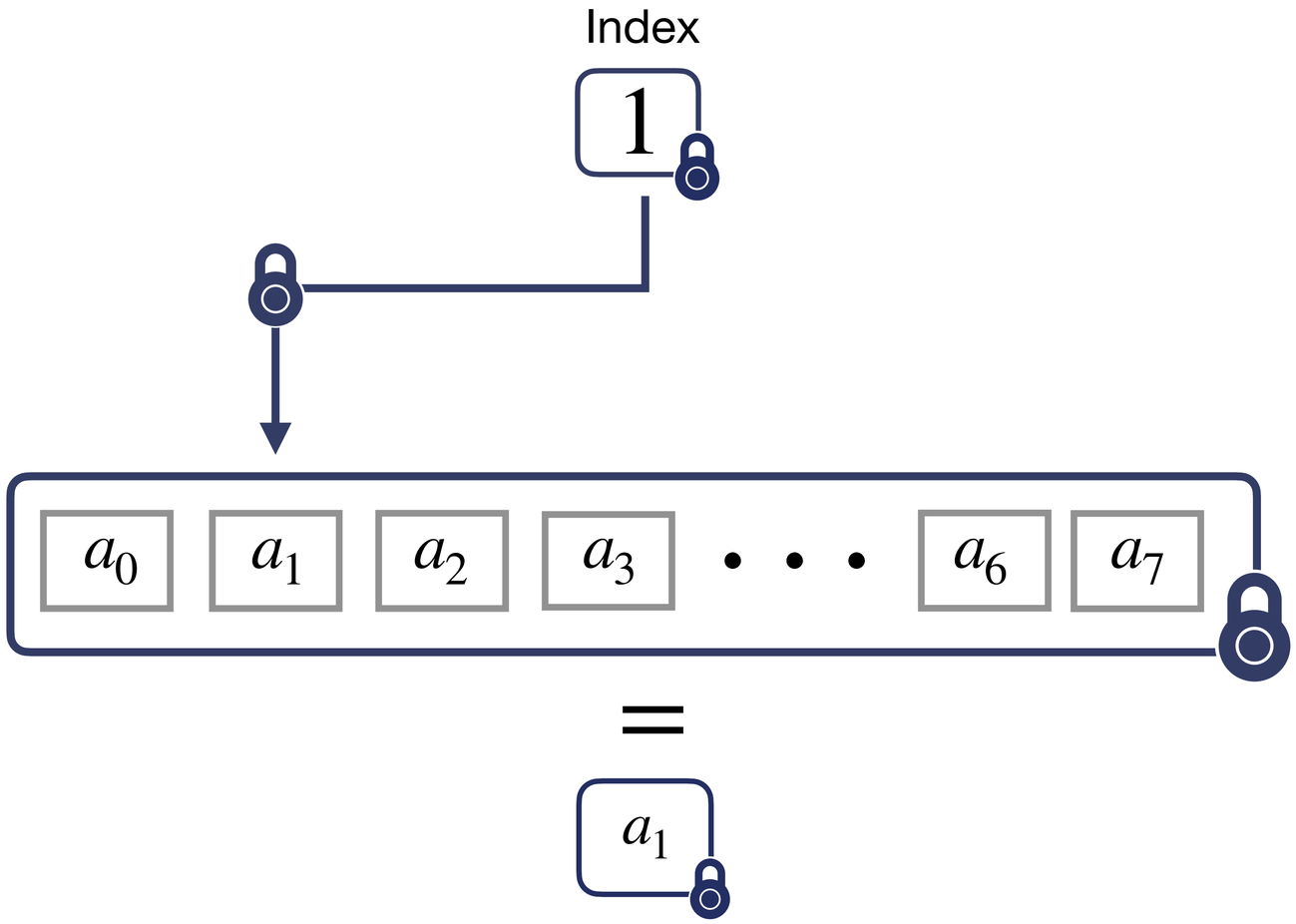
Chillotti et al. : TFHE: Fast Fully Homomorphic Encryption Over the Torus. J Cryptol 33, 34–91 (2020)

Oblivious computation

RevoLUT library

sofianeazogagh/revoLUT

Blind Read



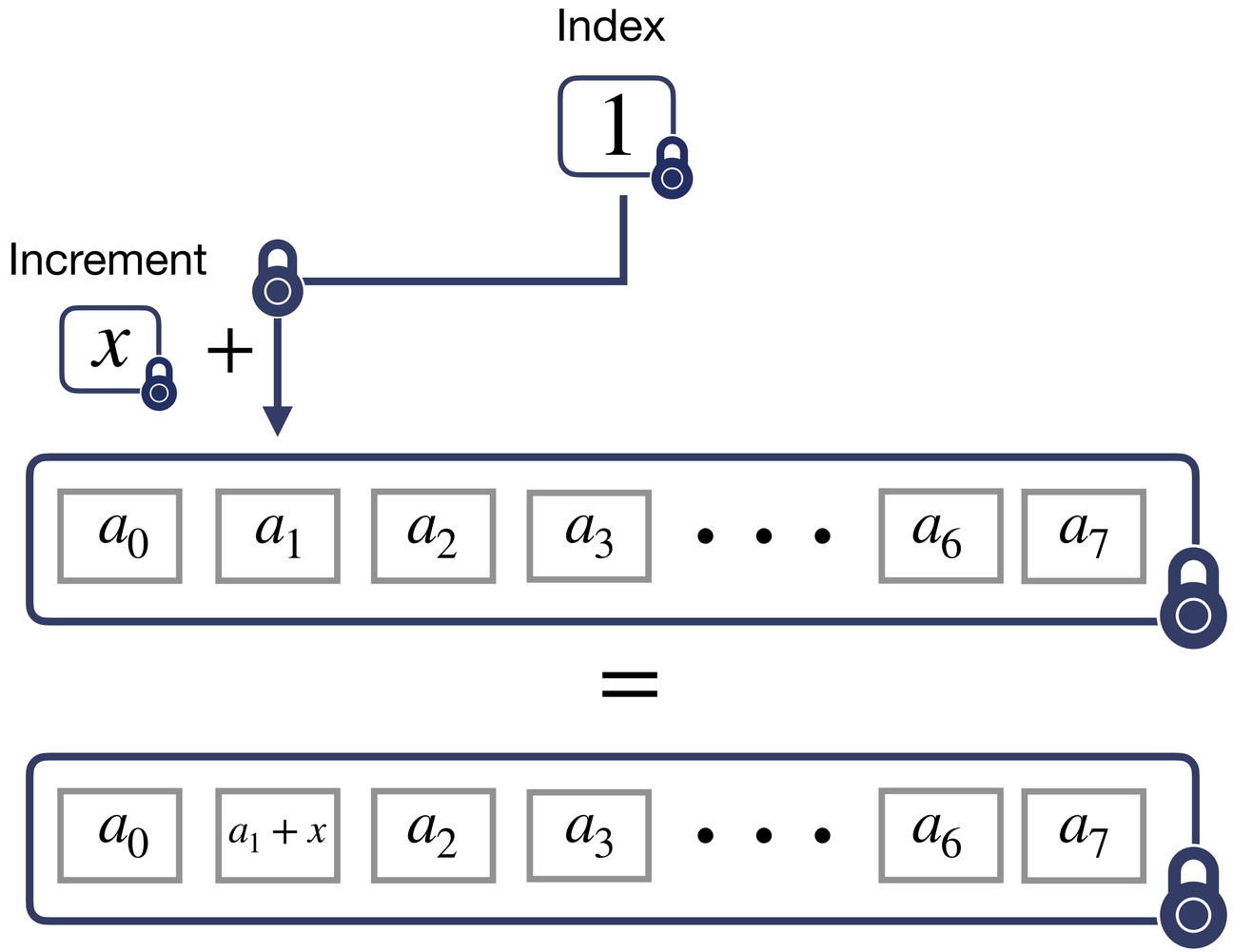
Oblivious computation

RevoLUT library

sofianeazogagh/revoLUT

Blind Read

Blind Write

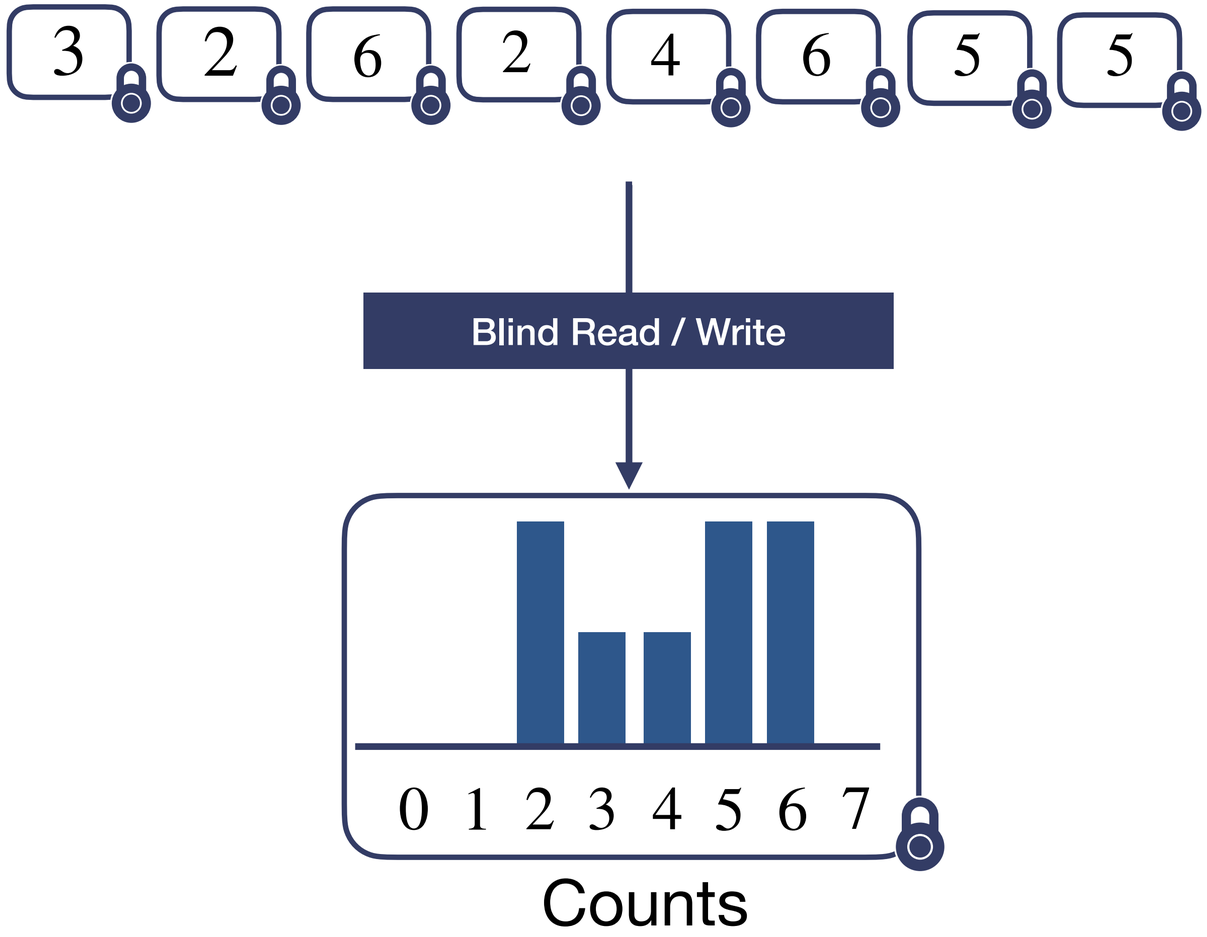


Oblivious computation

RevoLUT library

sofianeazogagh/revoLUT

- Blind Read
- Blind Write
- Blind Count

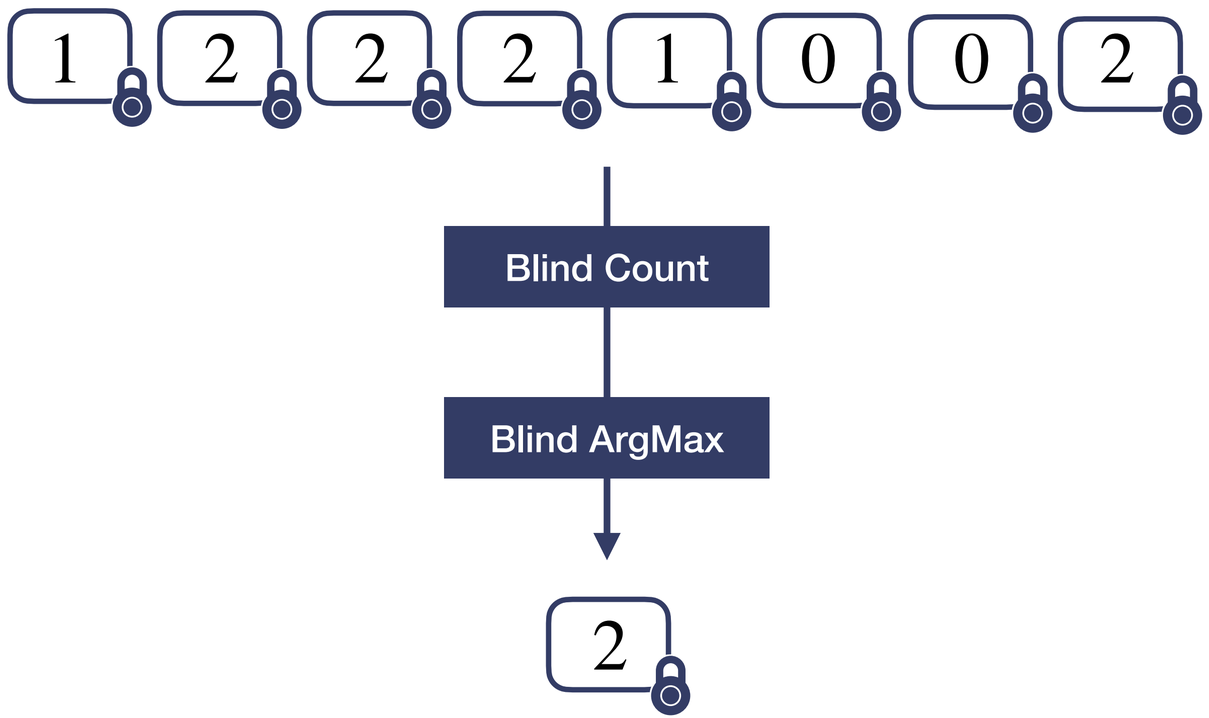


Oblivious computation

RevoLUT library

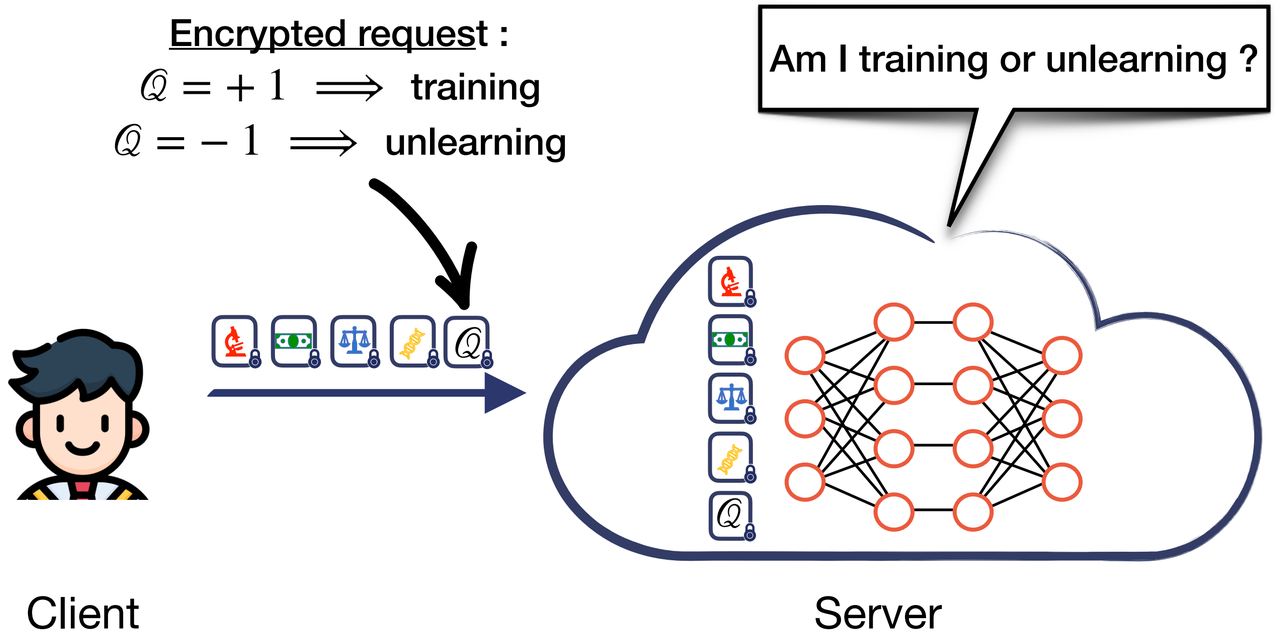
sofianeazogagh/revoLUT

- Blind Read
- Blind Write
- Blind Count
- Blind Majority



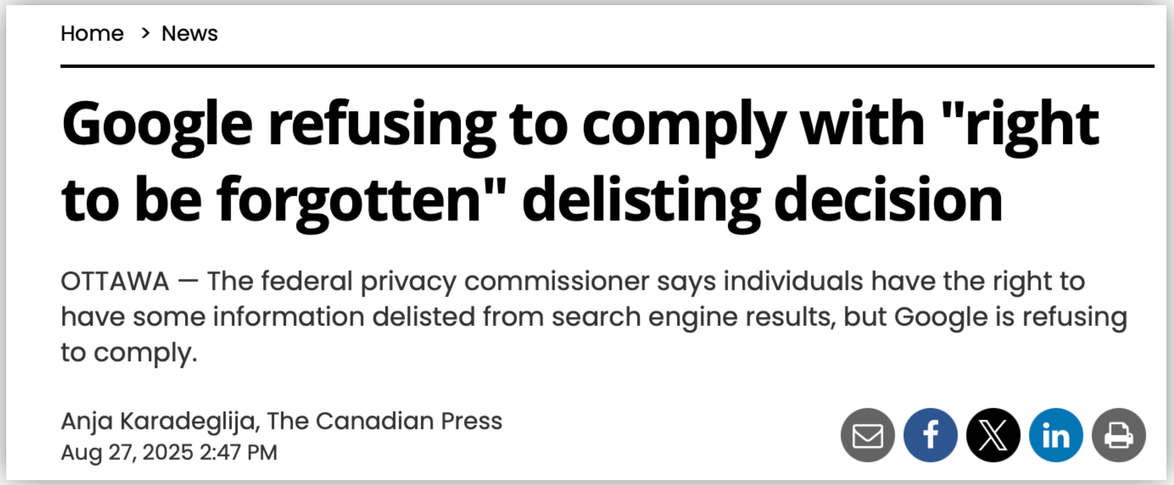
Oblivious unlearning

Goal : **Hiding unlearning requests** from the server



Why ?

- To protect **users privacy**
- To enforce the **right to be forgotten**



Home > News

Google refusing to comply with "right to be forgotten" delisting decision

OTTAWA — The federal privacy commissioner says individuals have the right to have some information delisted from search engine results, but Google is refusing to comply.

Anja Karadeglija, The Canadian Press
Aug 27, 2025 2:47 PM

Extremely Randomized Tree (ERT)

θ : Threshold

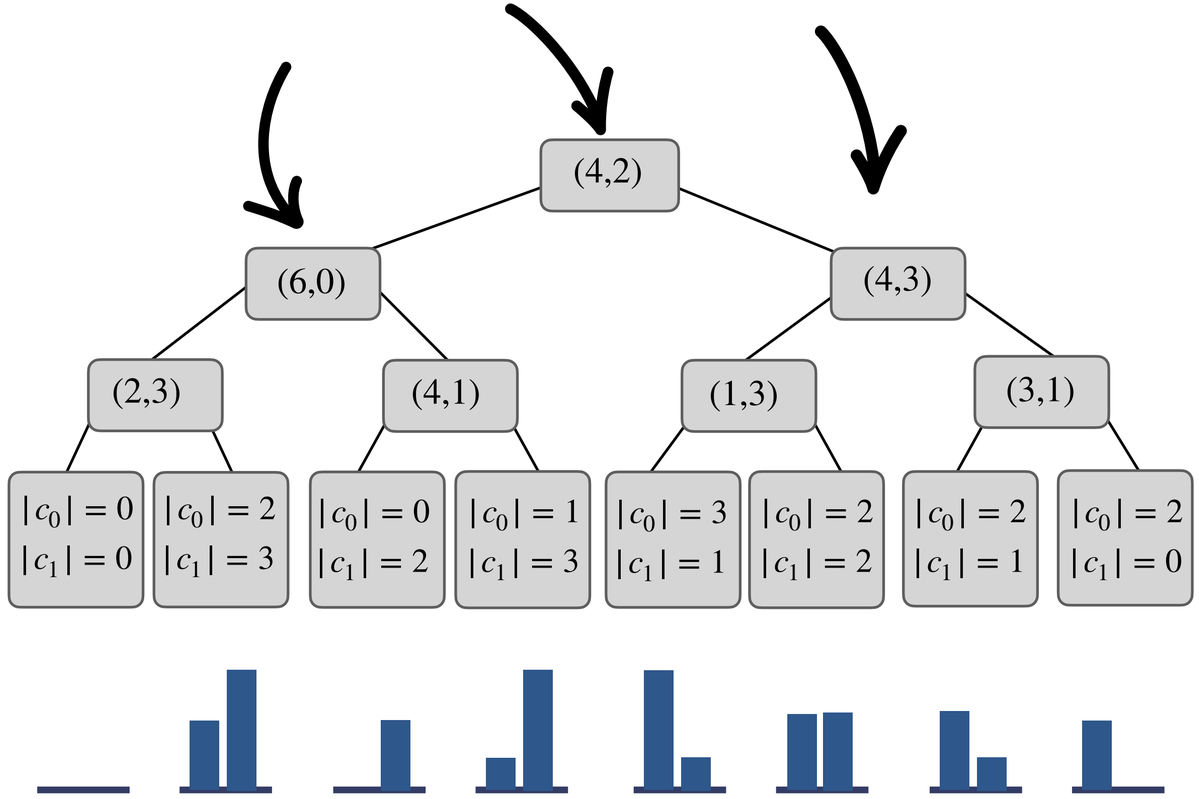
I : Feature index

Let $S = \{(X_j, y_j)\}_{j=1}^{n}$ be the training set where $X_j \in \mathbb{R}^l$ and $y_j \in \{c_0, \ldots, c_{\ell-1}\}$.

Two steps to build an ERT :

1. Build the internal nodes : randomized splits
2. Count class occurrences : accumulating class counts in leaves

Splits (θ, I) are selected among k random generated candidates



Each leaves stores the number of samples from each class c_i that reach it

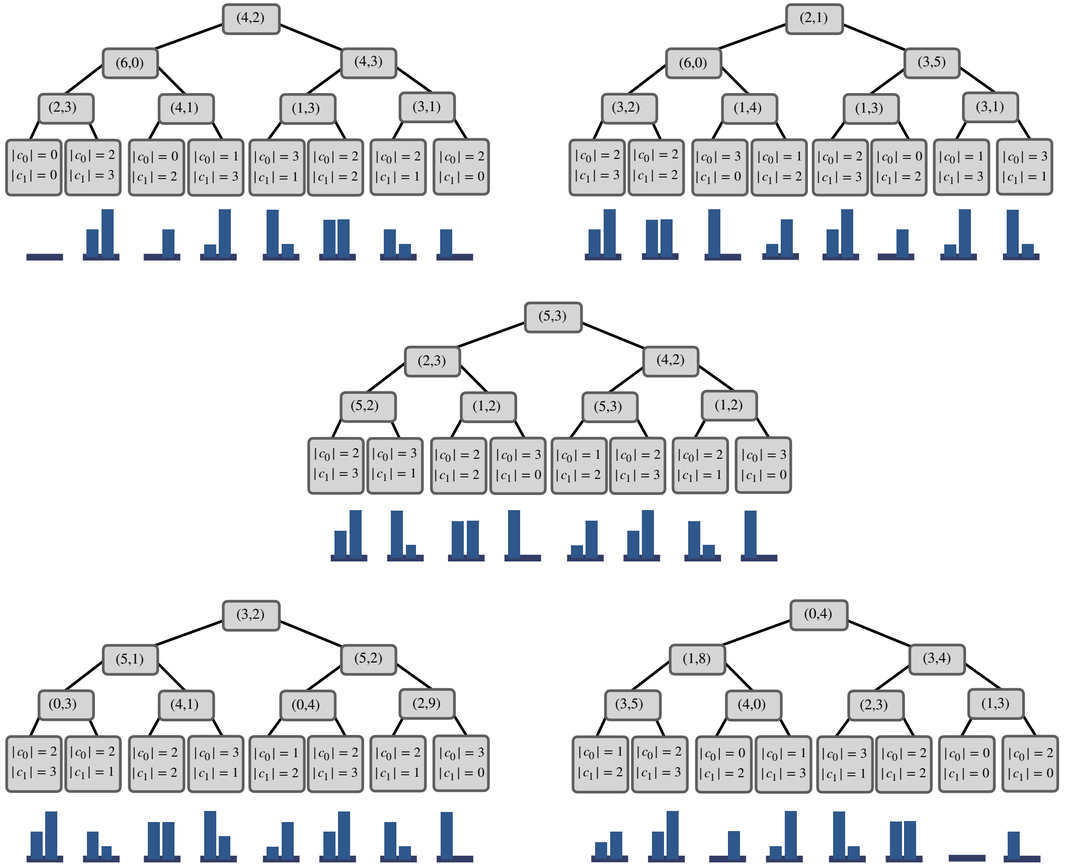
Geurts, P., Ernst, D. & Wehenkel, L. Extremely randomized trees. Mach Learn 63, 3–42 (2006)

Extremely Randomized Tree (ERT)

θ : Threshold

I : Feature index

Let $S = \{(X_j, y_j)\}_{j=1}^{n}$ be the training set where $X_j \in \mathbb{R}^l$ and $y_j \in \{c_0, \ldots, c_{\ell-1}\}$.



Two steps to build an ERT :

1. Build the internal nodes : randomized splits
2. Count class occurrences : accumulating class counts in leaves

Geurts, P., Ernst, D. & Wehenkel, L. Extremely randomized trees. Mach Learn 63, 3–42 (2006)

How to run ERTs on encrypted data ?

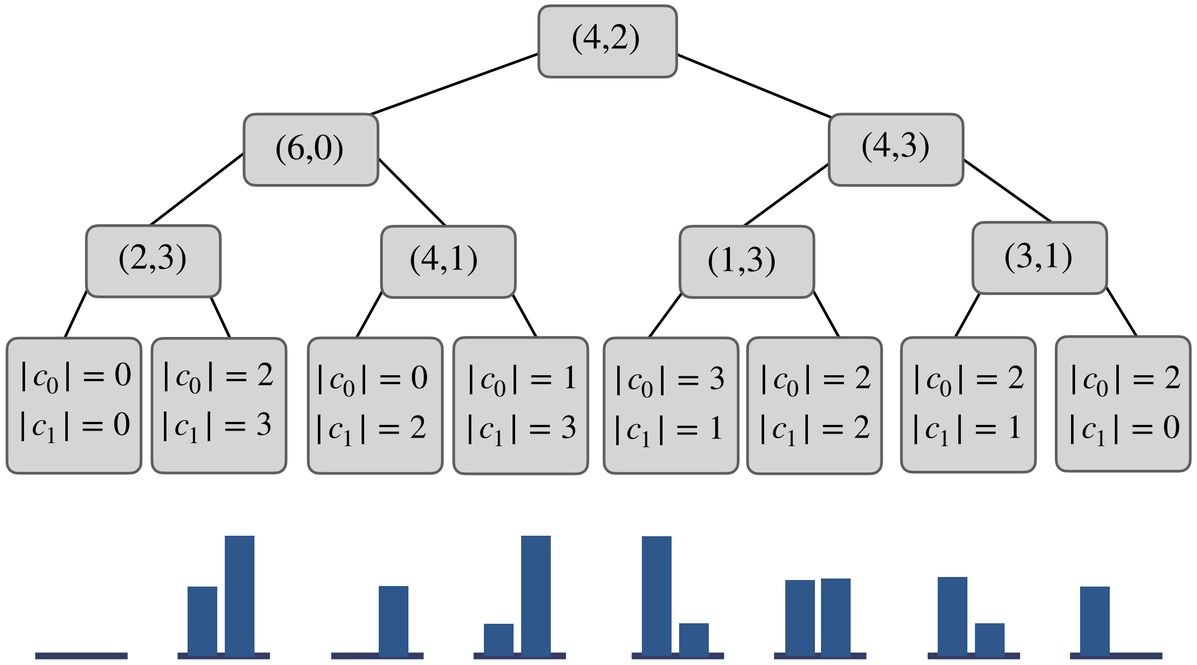
N : TFHE parameter

Client

1. Features quantization :

$$x \leftarrow \left\lfloor \frac{x - x_{\min}}{x_{\max} - x_{\min}} \cdot (N-1) \right\rfloor \in \mathbb{Z}_{\mathbb{N}}$$

2. One hot encoding classes



How to run ERTs on encrypted data ?

N : TFHE parameter

Client

1. Features quantization :

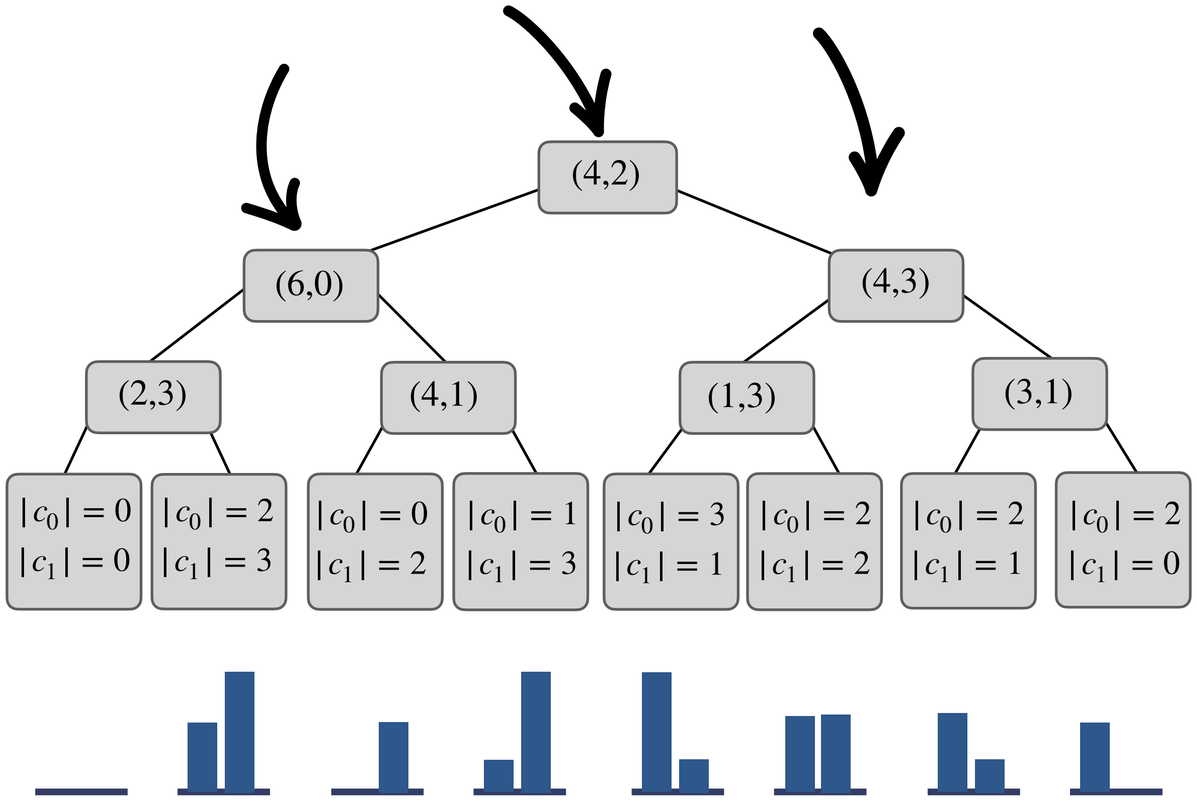
$$x \leftarrow \left\lfloor \frac{x - x_{\min}}{x_{\max} - x_{\min}} \cdot (N-1) \right\rfloor \in \mathbb{Z}_{\mathbb{N}}$$

2. One hot encoding classes

Server

3. Fix $k = 1$ to avoid non FHE-friendly criterion (e.g Gini Index, Information Gain etc..)

Splits (θ, I) are sampled randomly from $\mathbb{Z}_N \times \mathbb{Z}_N$



How to run ERTs on encrypted data ?

N : TFHE parameter

Client

1. Features quantization :

$$x \leftarrow \left\lfloor \frac{x - x_{\min}}{x_{\max} - x_{\min}} \cdot (N-1) \right\rfloor \in \mathbb{Z}_{\mathbb{N}}$$

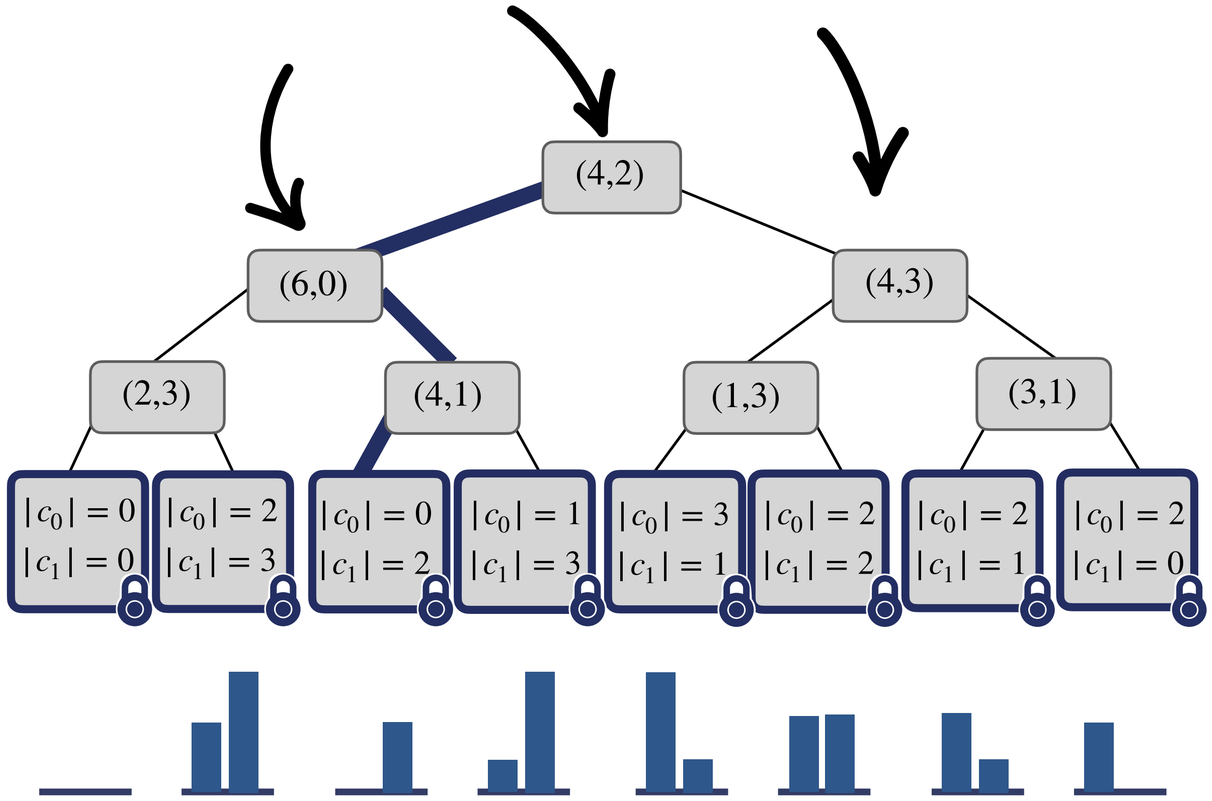
2. One hot encoding classes

Server

3. Fix $k = 1$ to avoid non FHE-friendly criterion (e.g Gini Index, Information Gain etc..)

4. The **traversal process** and the **leaves updates** are **oblivious**

Splits (θ, I) are sampled randomly from $\mathbb{Z}_N \times \mathbb{Z}_N$



How to run ERTs on encrypted data ?

N : TFHE parameter

Client

1. Features quantization :

$$x \leftarrow \left\lfloor \frac{x - x_{\min}}{x_{\max} - x_{\min}} \cdot (N-1) \right\rfloor \in \mathbb{Z}_{\mathbb{N}}$$

2. One hot encoding classes

Server

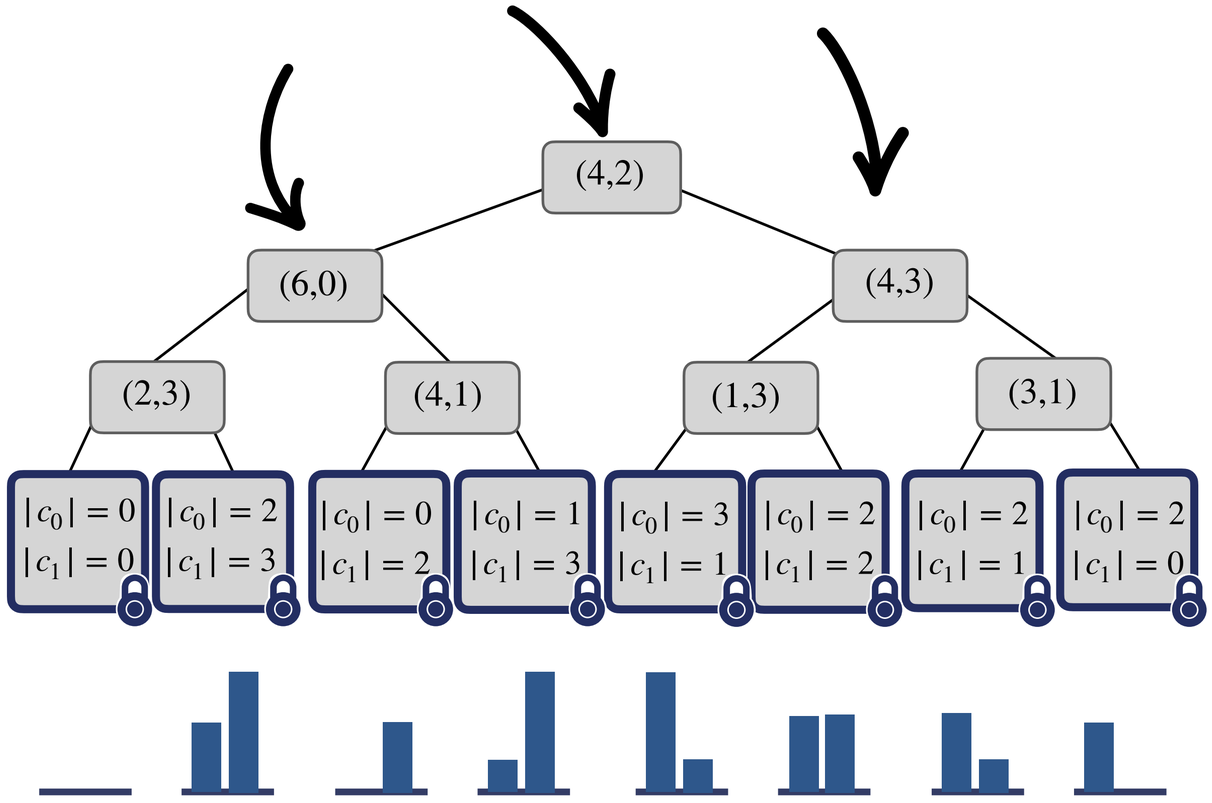
3. Fix $k = 1$ to avoid non FHE-friendly criterion (e.g Gini Index, Information Gain etc..)

4. The **traversal process** and the **leaves updates** are **oblivious**

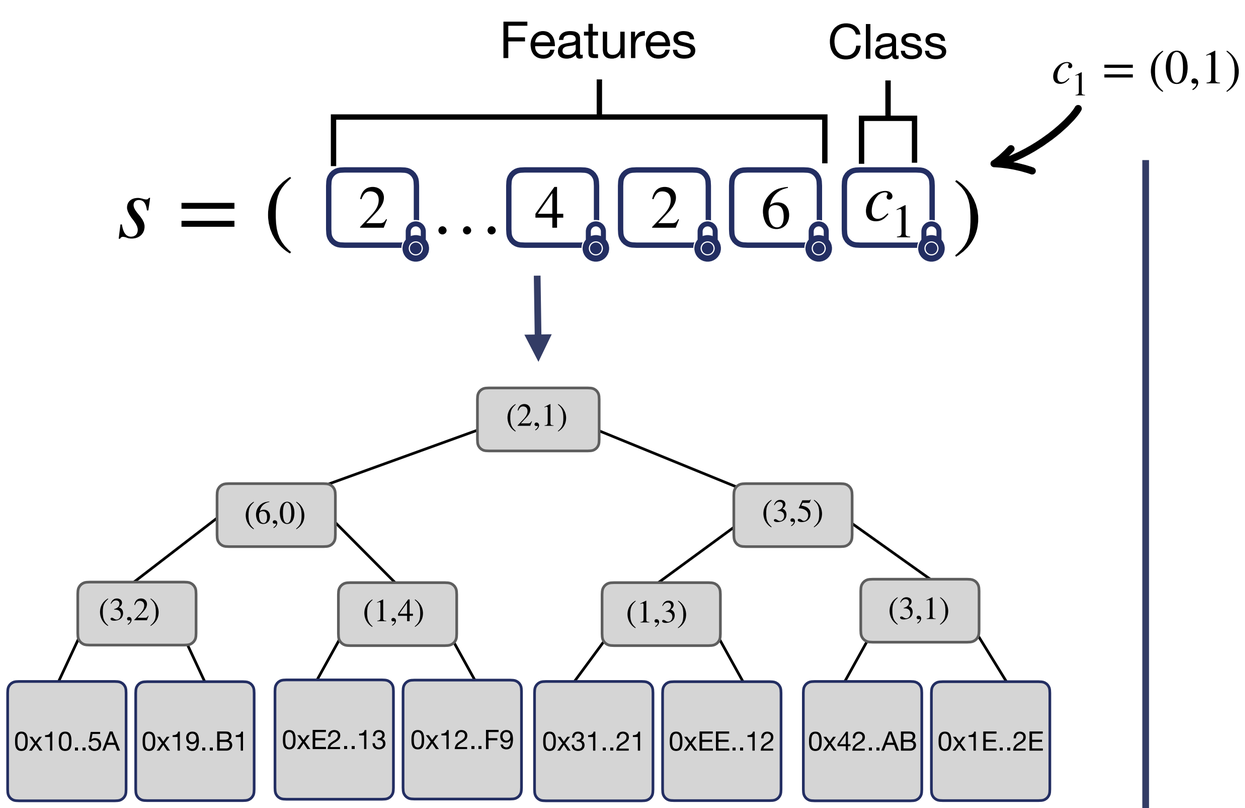
5. Internal nodes remains **public**

   Leaves become **private**

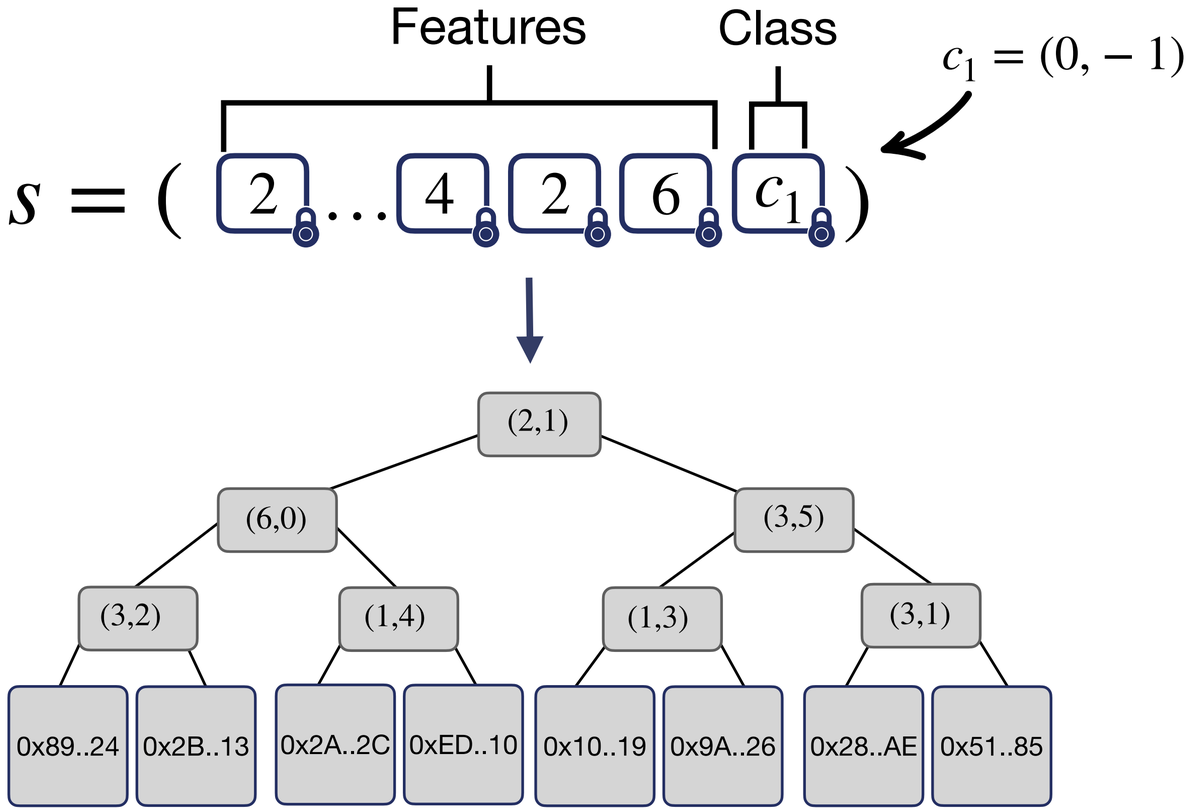
Splits (θ, I) are sampled randomly from $\mathbb{Z}_N \times \mathbb{Z}_N$



Server's View



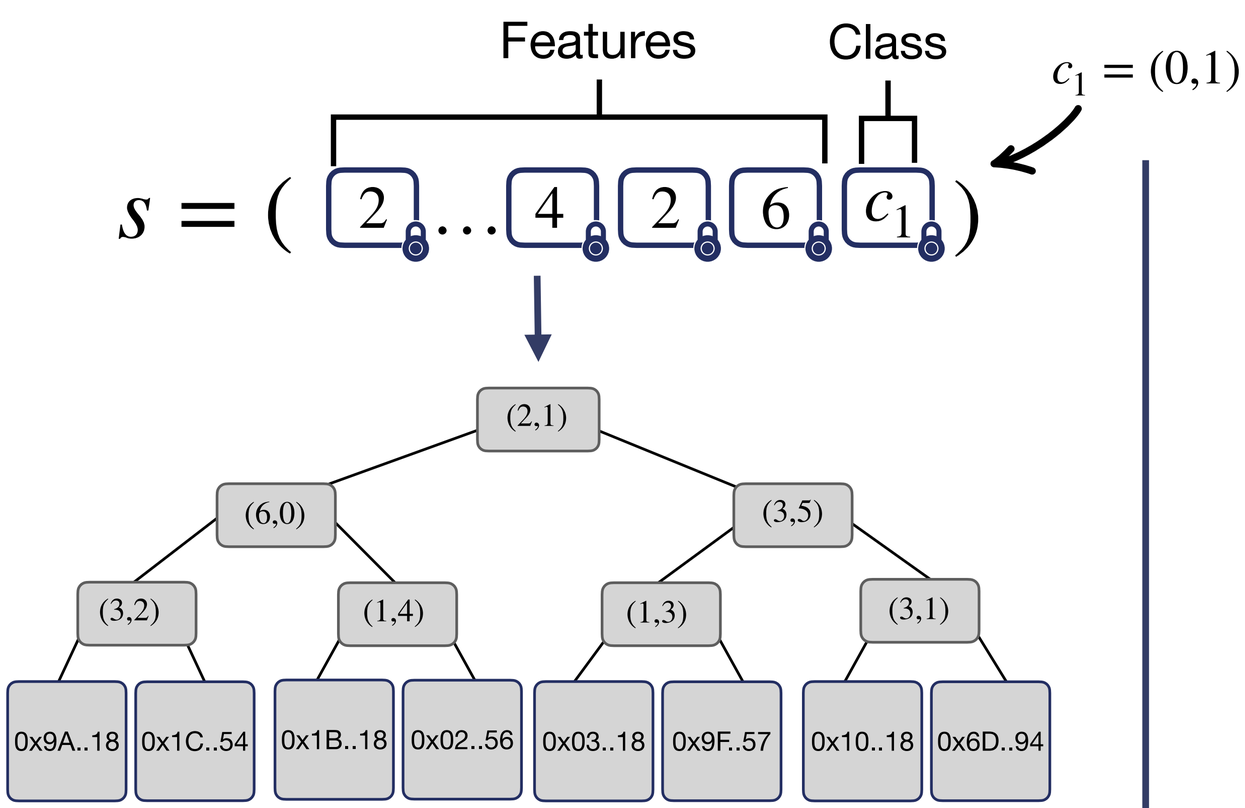
Training



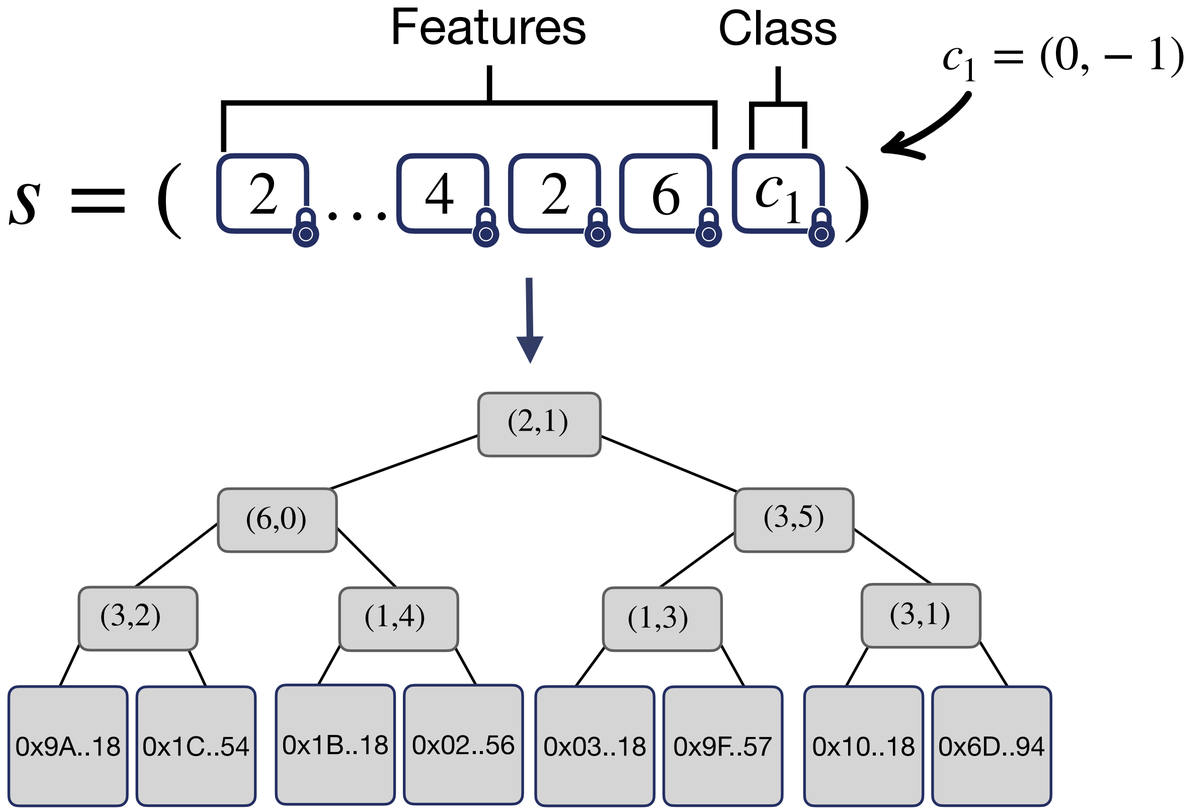
Unlearning

Server's View

Obliviousness



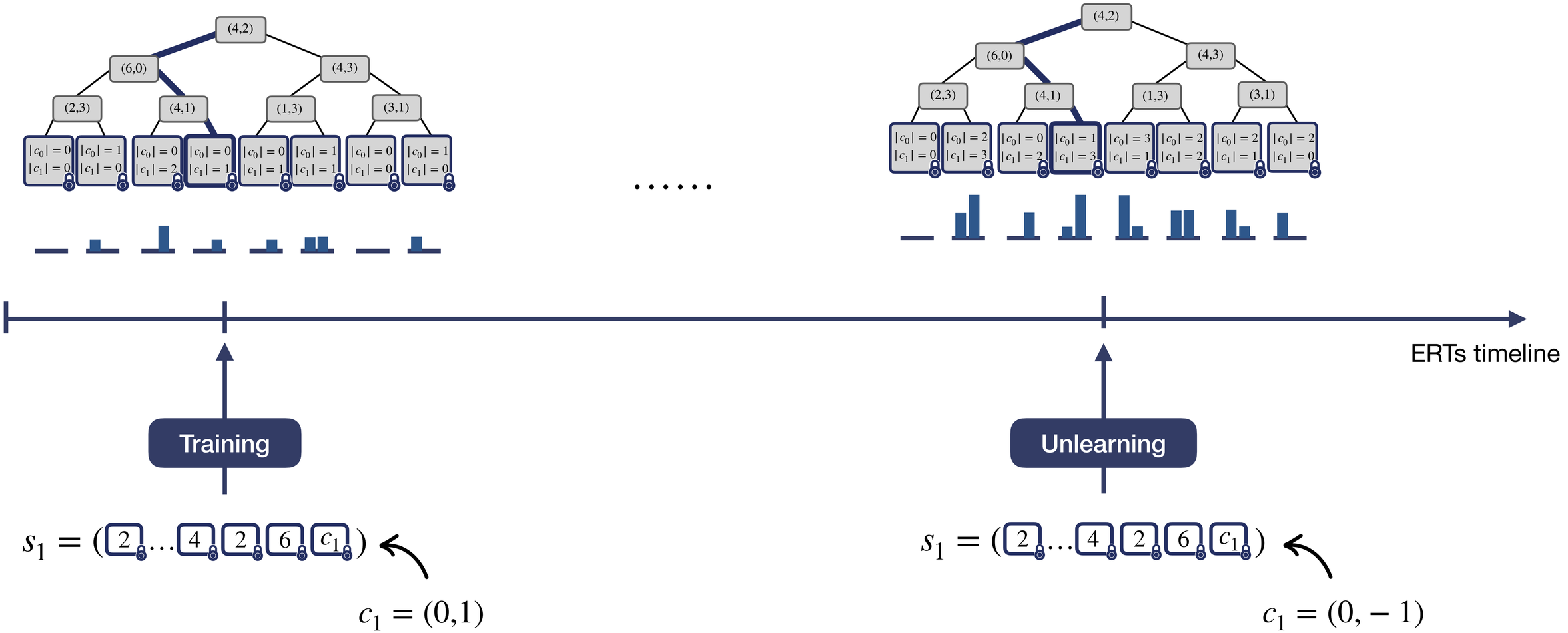
Training



Unlearning

Exact Unlearning

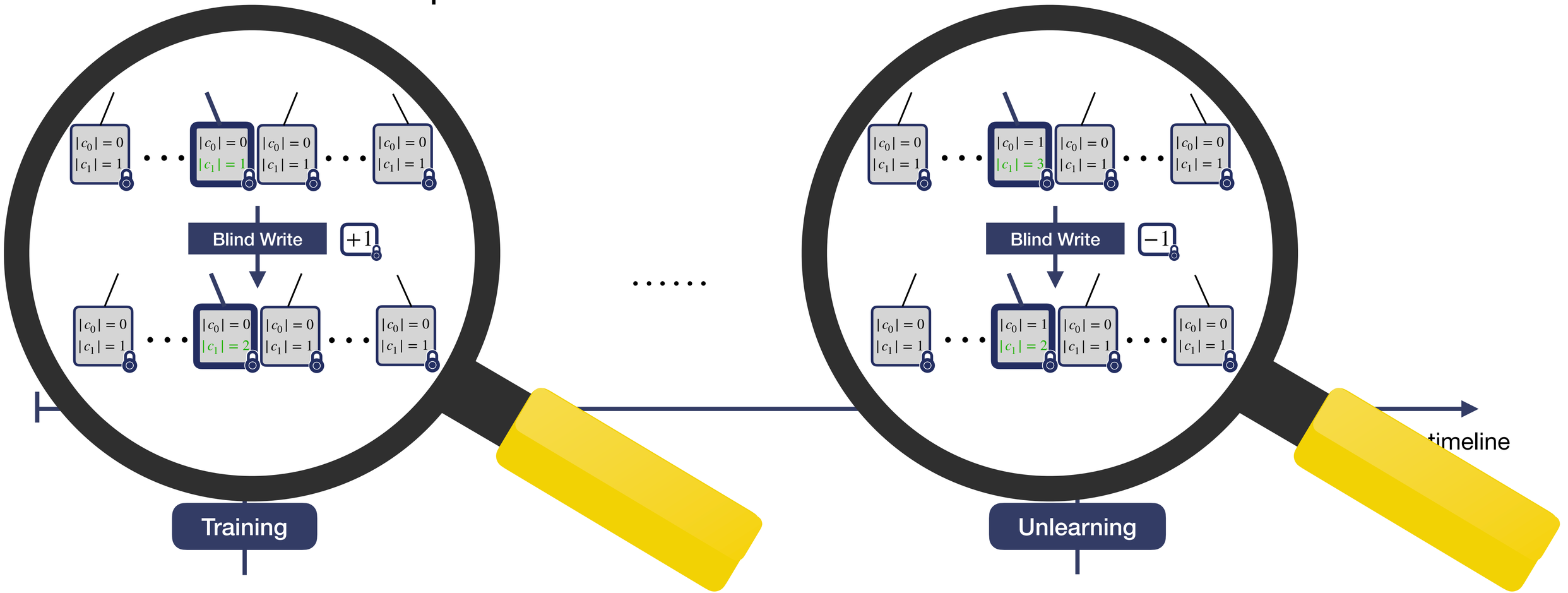
Because the traversal process is **deterministic** :



Exact Unlearning

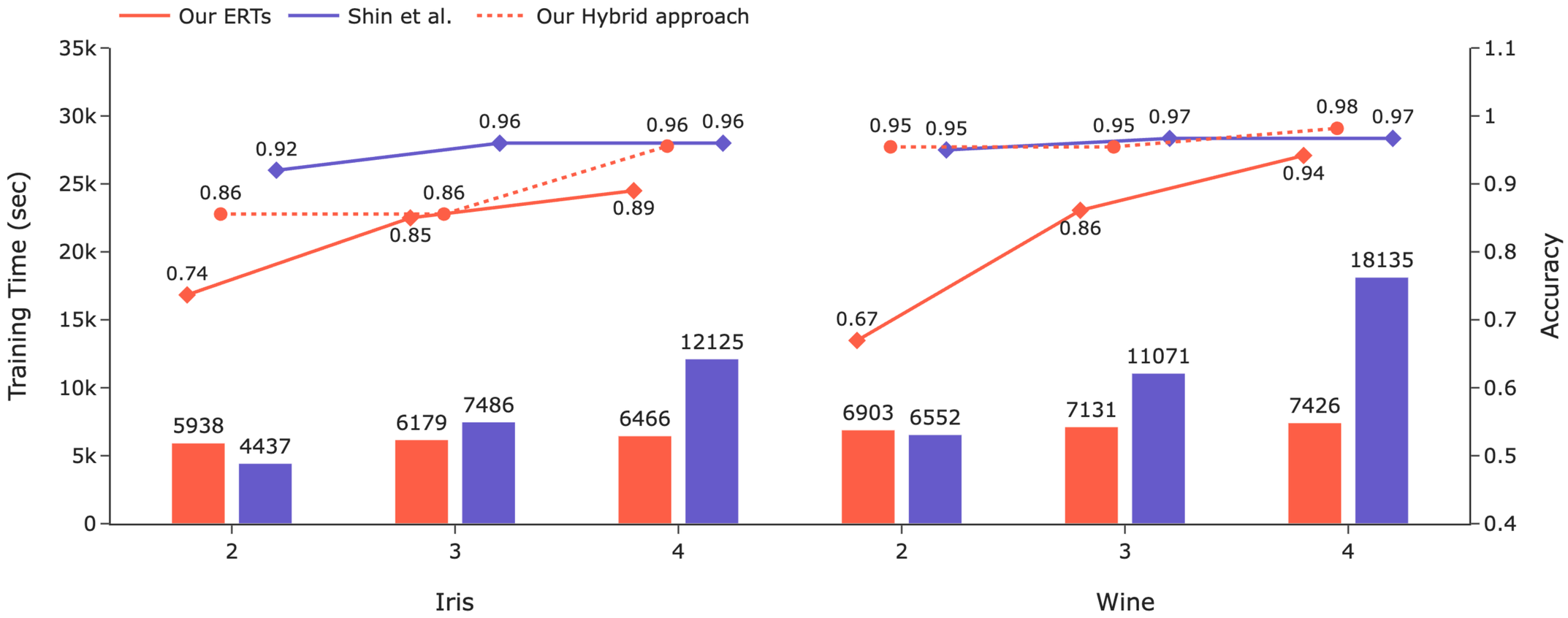
Correctness

Because the traversal process is **deterministic** :



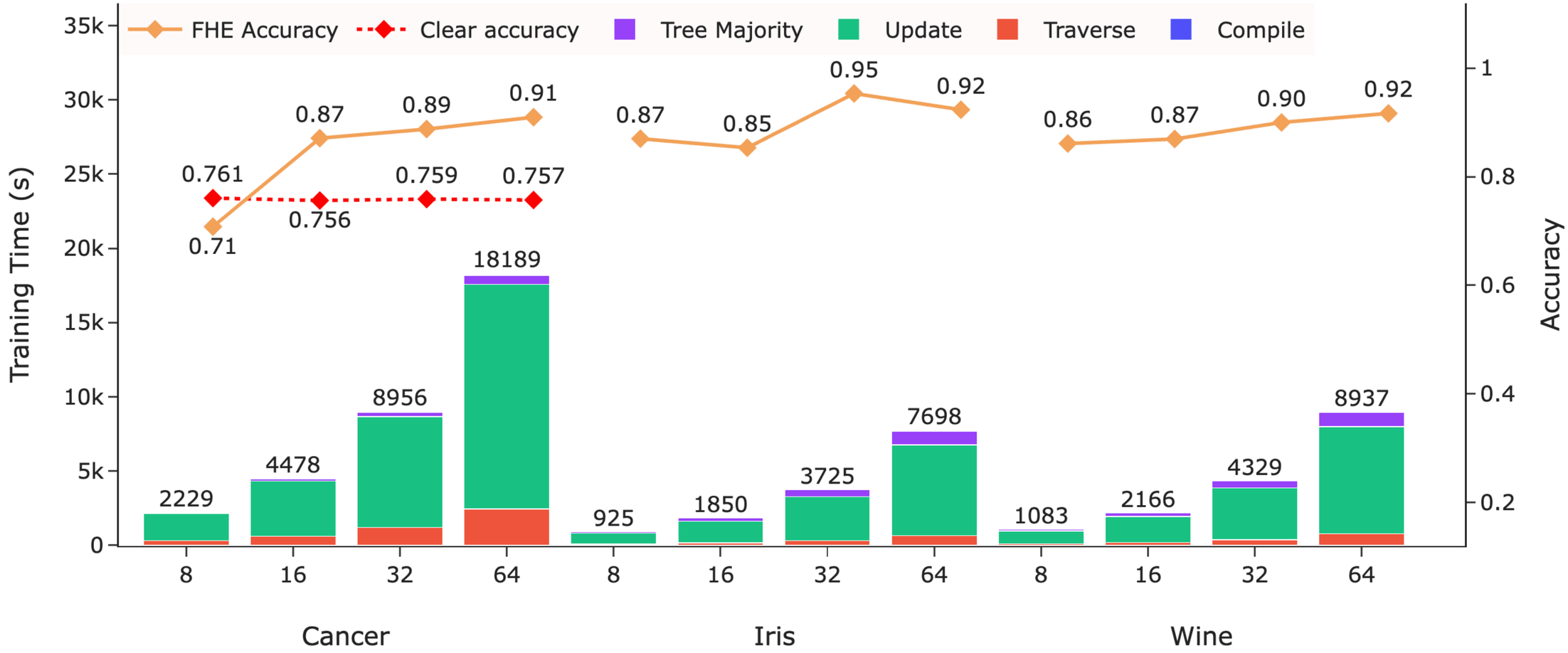
Experimental results

Comparison with **Random Forest** using **FHE** :



Shin et al. : Fully homomorphic training and inference on binary decision tree and Random Forest, Computer Security — ESORICS'24

Experimental results



Key Takeaways

- **Oblivious computation** enables compliance
- Extremely Randomized Trees (ERTs) are **simple** and **FHE-friendly**
- Unlearning is **indistinguishable** from training
- No accuracy loss on encrypted data

Full paper : ia.cr/2025/1409

sofianeazogagh/oblivious_unlearning

Thank you !

See you at the poster session

azogagh.sofiane@uqam.ca

sofianeazogagh.github.io