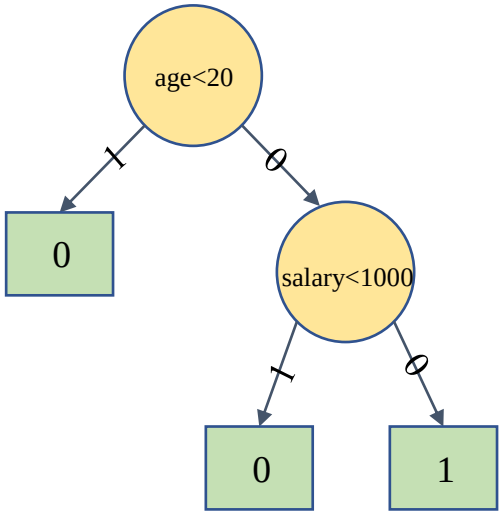
ACSAC 2023
December 4–8, 2023 • Austin, Texas, USA

Mostree: Malicious Secure Private Decision Tree Evaluation with Sublinear Communication

Jianli Bai (University of Auckland)
Xiangfu Song (National University of Singapore)
Xiaowu Zhang (CloudWalk Technology)
Qifan Wang (University of Auckland)
Shujie Cui (Monash University)
Ee-Chien Chang (National University of Singapore)
Giovanni Russello (University of Auckland)

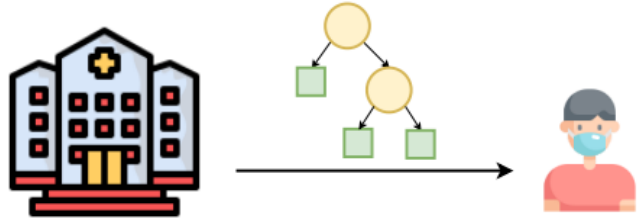
Decision Trees

- Tree structure model
  - Simple and interpretable
- Applications
  - Healthcare system
  - Spam detection
  - Credit-risk assessment
  - …



Privacy Challenges

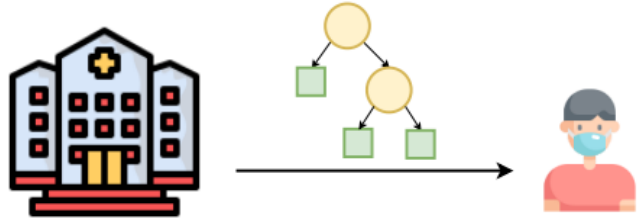
- The decision tree model is valuable



- The feature vector is sensitive

Privacy Challenges

- The decision tree model is valuable



- The feature vector is sensitive

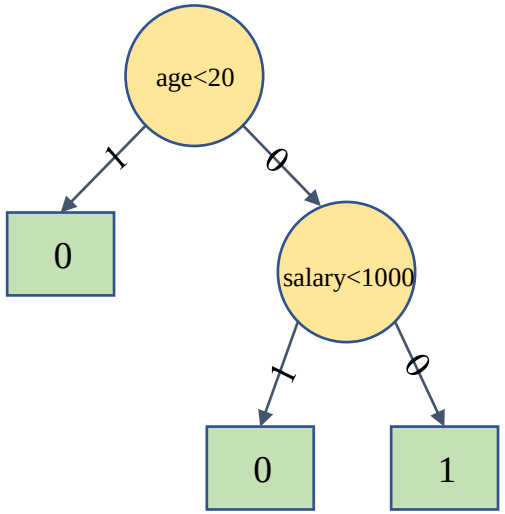
- How to perform private decision tree evaluation (PDTE)?

Sensitive Feature/Tree Information

- Feature example:

age	gender	salary
35	male	3000

- Tree example:



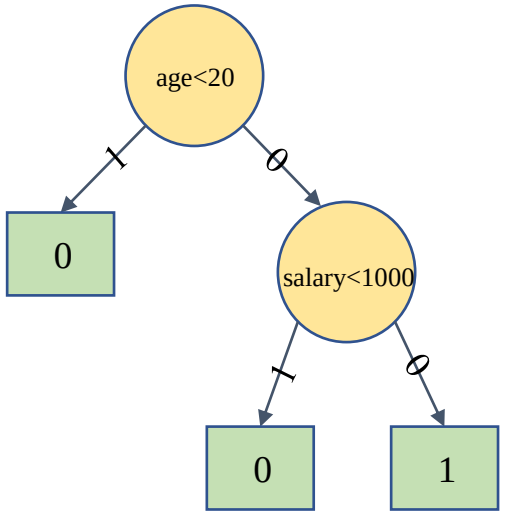
Sensitive Feature/Tree Information

- Feature example:

age	gender	salary
35	male	3000

- Sensitive information
  - Attribute values (e.g., age=35, gender=male,…)

- Tree example:

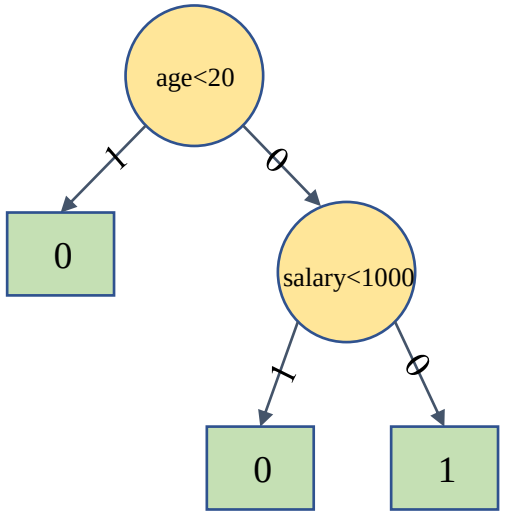


Sensitive Feature/Tree Information

- Feature example:

age	gender	salary
35	male	3000

- Tree example:



- Sensitive information
  - Attribute values (e.g., age=35, gender=male,…)
- Public information
  - Feature dimension (e.g., 3)
  - Attribute names and their organization (e.g., age→gender→salary)

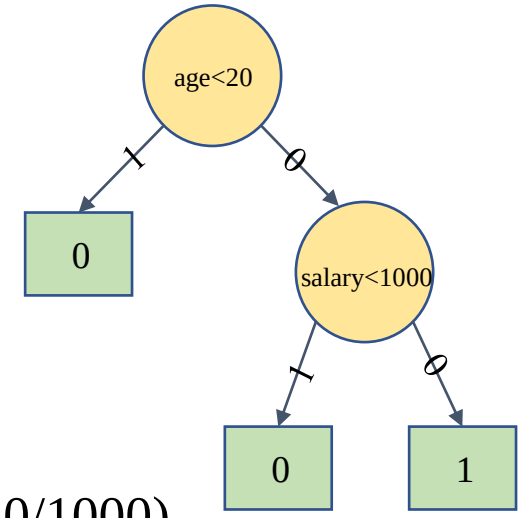
Sensitive Feature/Tree Information

- Feature example:

age	gender	salary
35	male	3000

- Sensitive information
  - Attribute values (e.g., age=35, gender=male,…)
- Public information
  - Feature dimension (e.g., 3)
  - Attribute names and their organization (e.g., age→gender→salary)

- Tree example:



- Sensitive information
  - Threshold (e.g., 20/1000)
  - Attributes of interest (e.g., age/salary)
  - Depth of leaf nodes (e.g., 1/2)

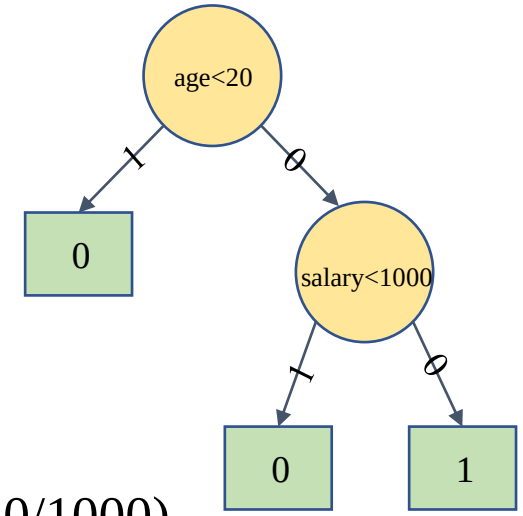
Sensitive Feature/Tree Information

- Feature example:

age	gender	salary
35	male	3000

- Sensitive information
  - Attribute values (e.g., age=35, gender=male,…)
- Public information
  - Feature dimension (e.g., 3)
  - Attribute names and their organization (e.g., age→gender→salary)

- Tree example:



- Sensitive information
  - Threshold (e.g., 20/1000)
  - Attributes of interest (e.g., age/salary)
  - Depth of leaf nodes (e.g., 1/2)
- Public information
  - The number of nodes (e.g., 5)
  - Maximal depth of the tree (e.g., 2)

System Model/Threat Model

CP

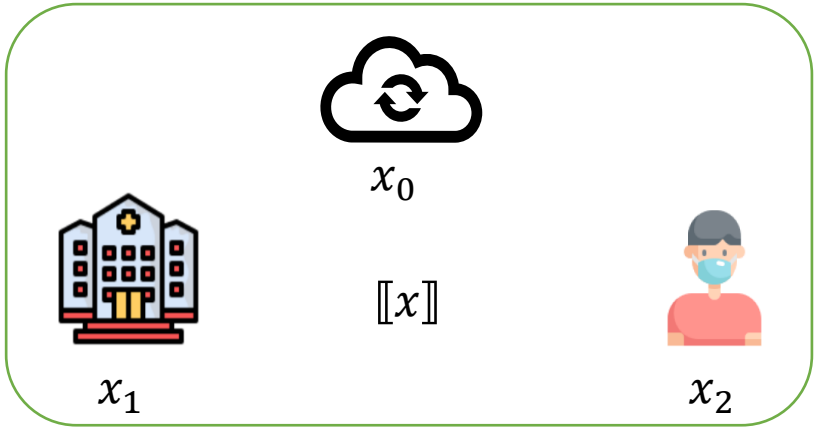
MO

FO

- System model:
  - Model owner (MO)
  - Feature owner (FO)
  - Computing party (CP)
- Threat model:
  - **Honest-majority setting**: At most one party can be **malicious**.
  - **Security goals**: Achieve *privacy* in any case, and *correctness* if the protocol completes (i.e., malicious security with abort).

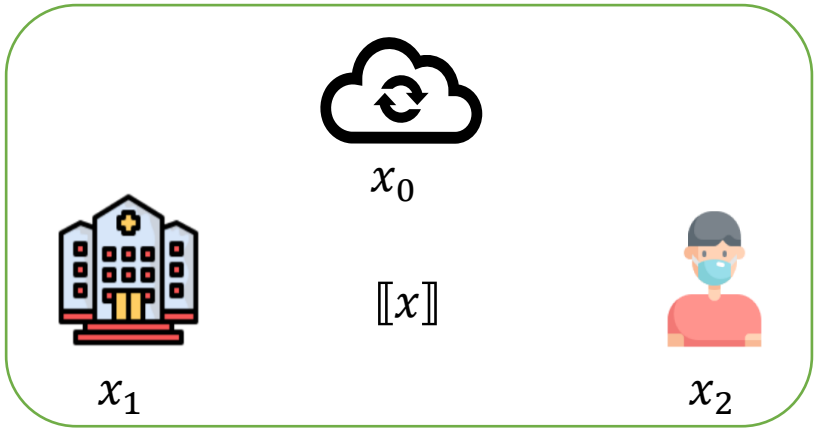
Cryptographic Primitives

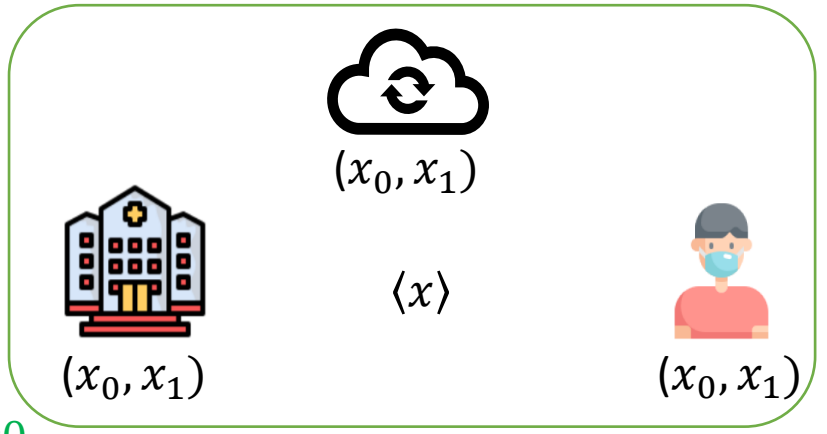
- $\binom{n}{n}$-secret sharing $[\![x]\!]$
  - $x = x_0 + x_1 + \cdots + x_{n-1} \bmod L$
  - $n = 2$ or 3 in this talk
  - if $n = 3$, P_0 has (x_0), P_1has (x_1), and P_2 has(x_2)



Cryptographic Primitives

- $\binom{n}{n}$-secret sharing $[\![x]\!]$
  - $x = x_0 + x_1 + \cdots + x_{n-1} \bmod L$
  - $n = 2$ or 3 in this talk
  - if $n = 3$, P_0 has (x_0), P_1has (x_1), and P_2 has(x_2)
- $\binom{3}{2}$-secret sharing $\langle x \rangle$
  - $x = x_0 + x_1 + x_2 \bmod L$
  - P_0 has (x_0, x_2), P_1has (x_1, x_0), and P_2 has(x_2, x_1)
  - Also known as replicated secret-sharing (RSS)
  - Multiplication: $\langle z \rangle \leftarrow \langle x \rangle \cdot \langle y \rangle$
    1) P_i computes $z_i \leftarrow x_i y_i + x_i y_{i-1} + x_{i-1} y_i + r_i$, where $r_0 + r_1 + r_2 = 0$.
    2) P_i sends z_i to P_{i+1} and receives z_i from P_i .
    3) P_i outputs (z_i, z_{i-1})

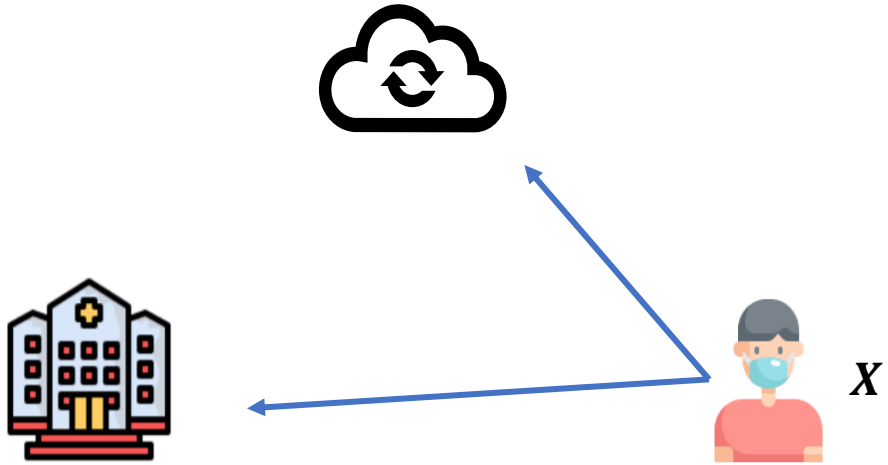




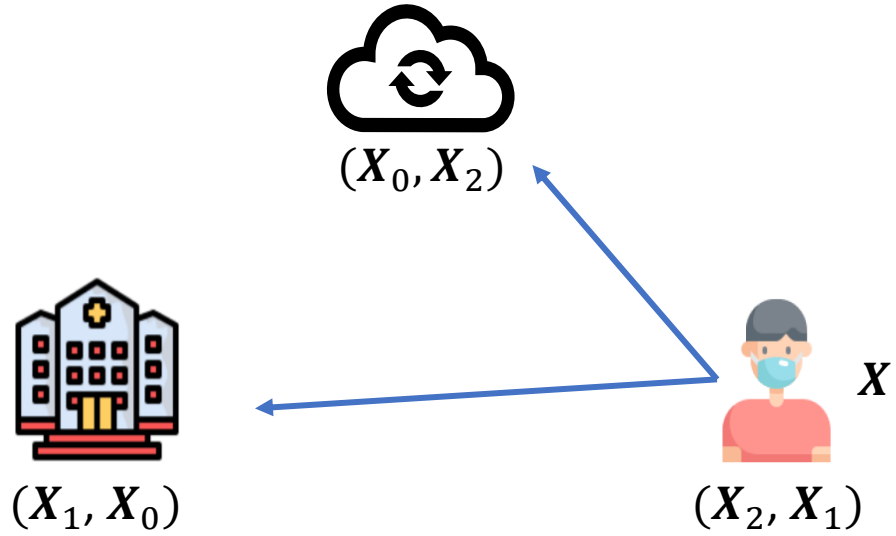
Mostree: High-level Description

X

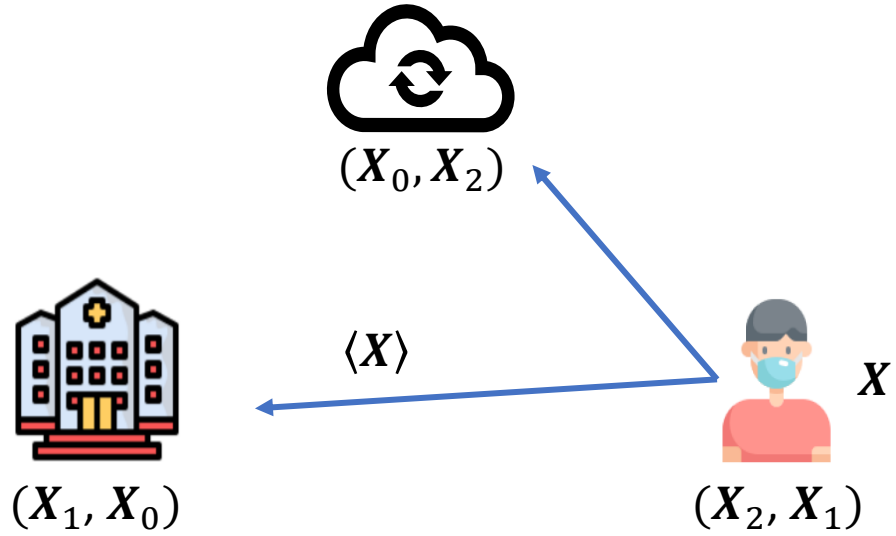
Mostree: High-level Description



Mostree: High-level Description



Mostree: High-level Description



Mostree: High-level Description

(X_0, X_2)

T

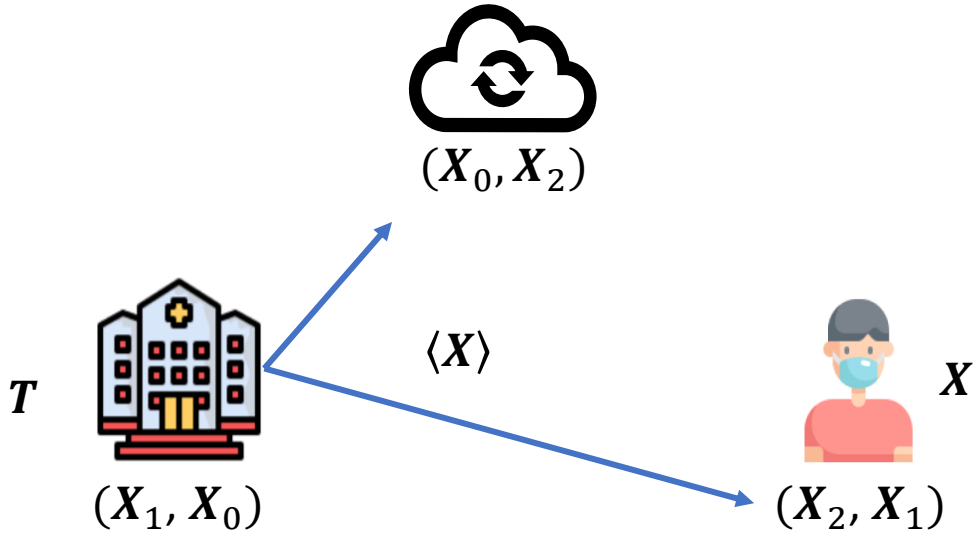
(X_1, X_0)

$\langle X \rangle$

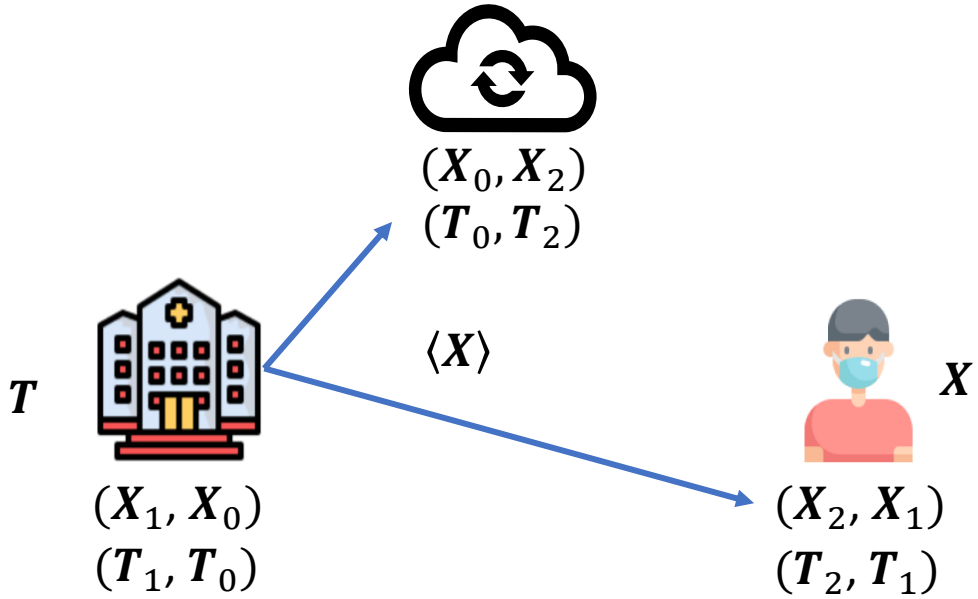
X

(X_2, X_1)

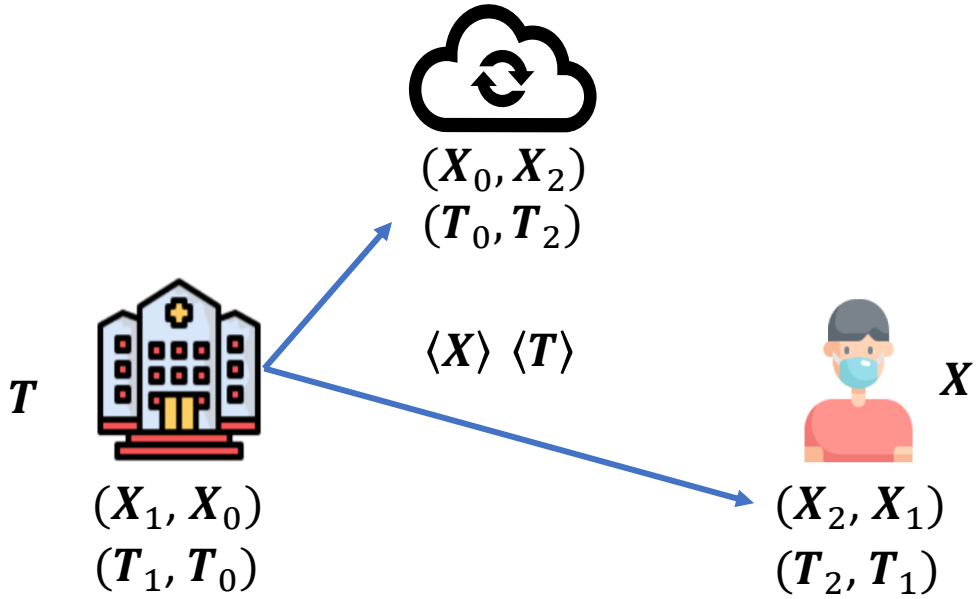
Mostree: High-level Description



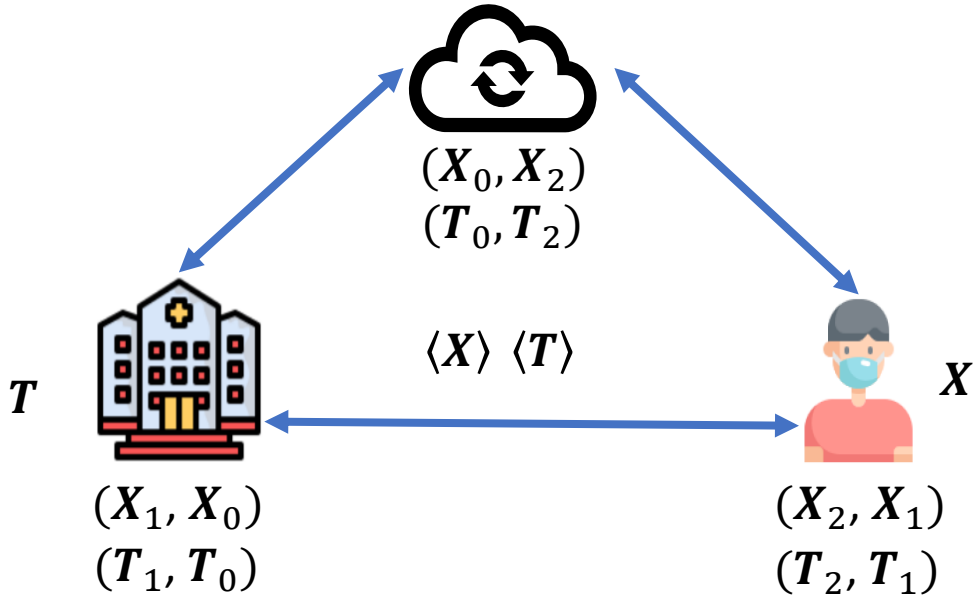
Mostree: High-level Description



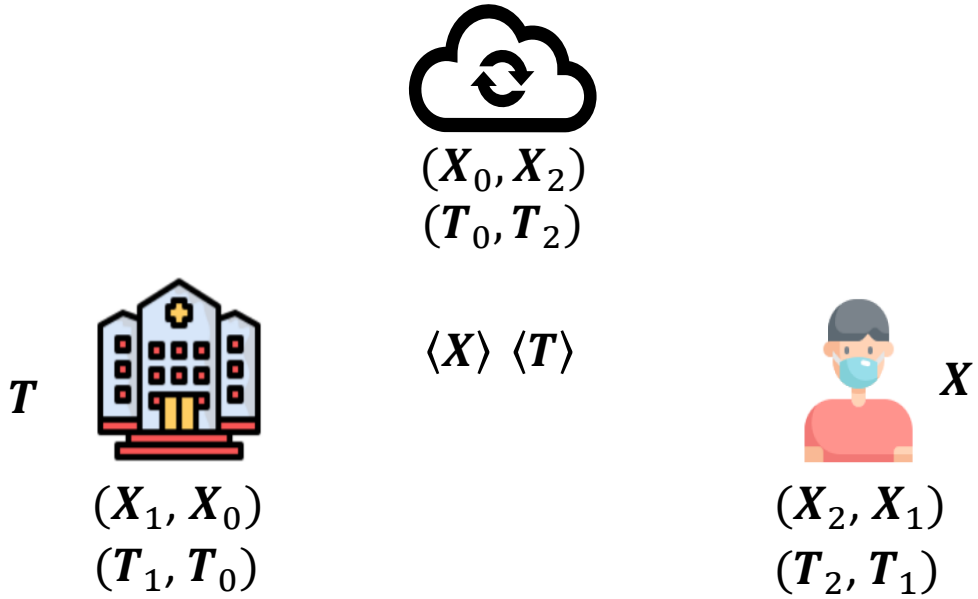
Mostree: High-level Description



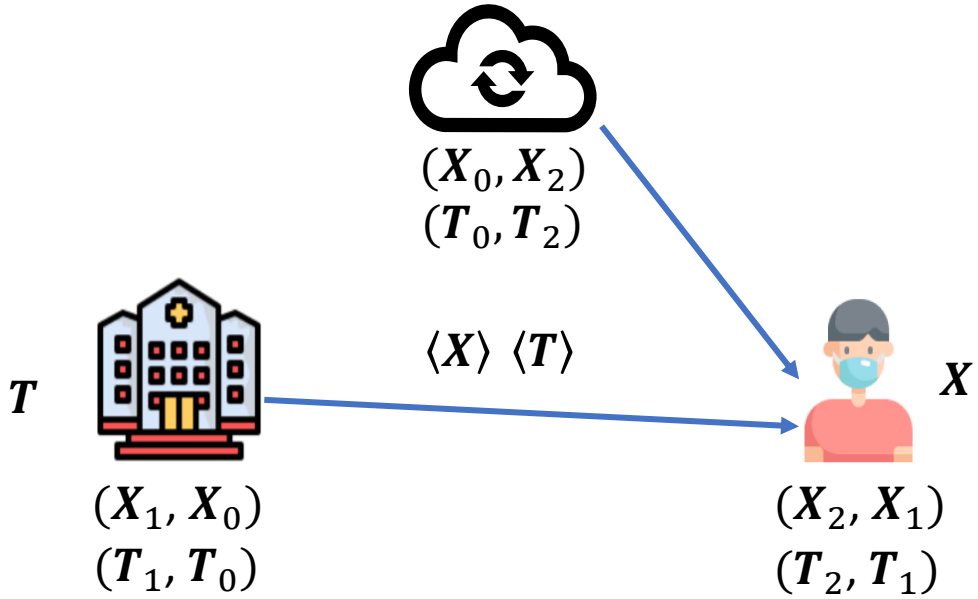
Mostree: High-level Description



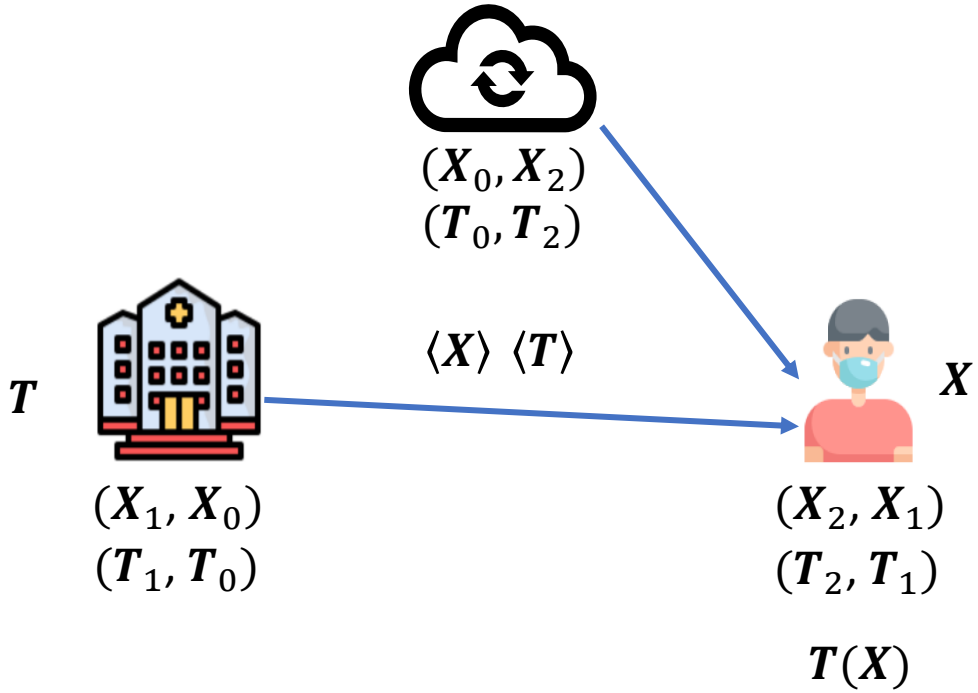
Mostree: High-level Description



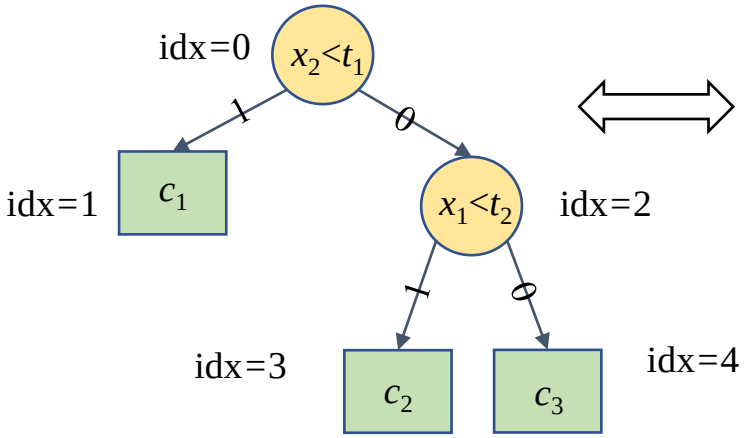
Mostree: High-level Description



Mostree: High-level Description

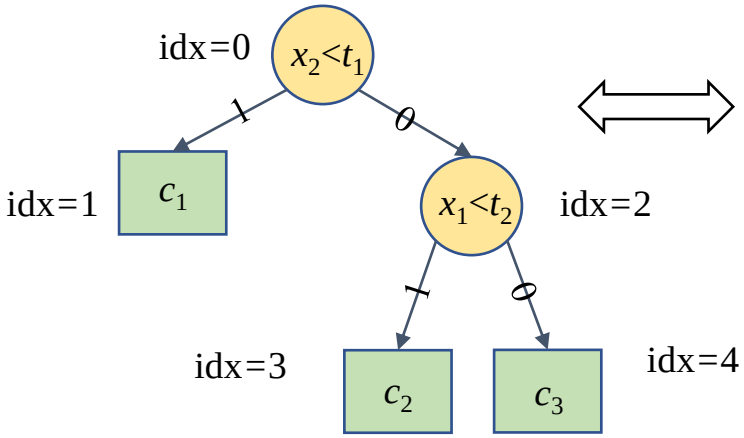


Decision Tree Evaluation



idx	*t*	*l*	*r*	*v*	*c*
0	t_1	1	2	2	⊥
1	⊥	⊥	⊥	⊥	c_1
2	t_2	3	4	1	⊥
3	⊥	⊥	⊥	⊥	c_2
4	⊥	⊥	⊥	⊥	c_3

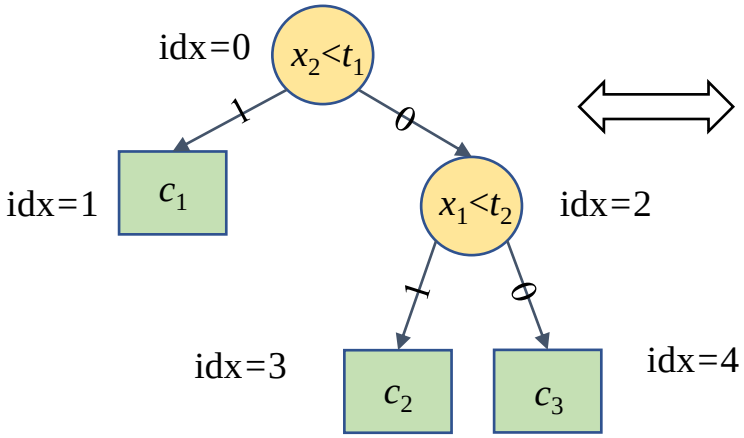
Decision Tree Evaluation



idx	*t*	*l*	*r*	*v*	*c*
0	t_1	1	2	2	⊥
1	⊥	⊥	⊥	⊥	c_1
2	t_2	3	4	1	⊥
3	⊥	⊥	⊥	⊥	c_2
4	⊥	⊥	⊥	⊥	c_3

- Decision tree encoding
  - t: threshold value. $t = \perp$ for decisive nodes.
  - l: index of left child. $l = \perp$ for decisive nodes.
  - r: index of right child. $r = \perp$ for decisive nodes.
  - v: index of attribute to be compared with t.
  - c: classification label. $c = \perp$ for non-leaf nodes.

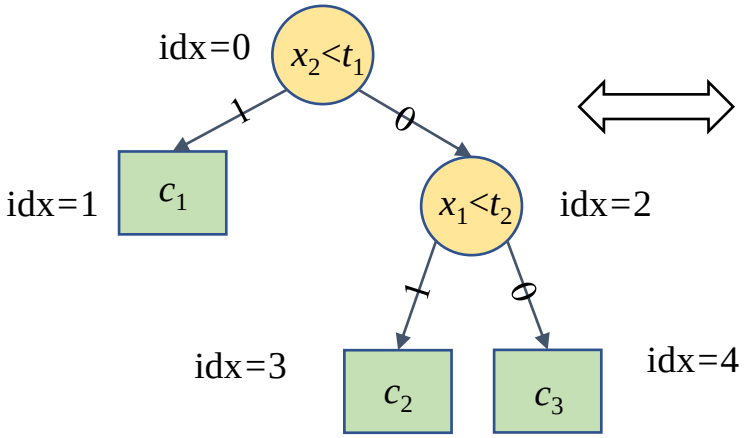
Decision Tree Evaluation



idx	*t*	*l*	*r*	*v*	*c*
0	t_1	1	2	2	$\perp$
1	$\perp$	$\perp$	$\perp$	$\perp$	c_1
2	t_2	3	4	1	$\perp$
3	$\perp$	$\perp$	$\perp$	$\perp$	c_2
4	$\perp$	$\perp$	$\perp$	$\perp$	c_3

- $\mathrm{DTE}(T, X)$:
  - $idx \leftarrow 0$
  - For $i \in [1, d]$:
    - $(t, l, r, v, c) \leftarrow T[idx]$
    - If $l = \perp$, return c
    - $x \leftarrow X[v]$
    - $b \leftarrow x < t$
    - $idx \leftarrow r \oplus b \cdot (r \oplus l)$

Decision Tree Evaluation

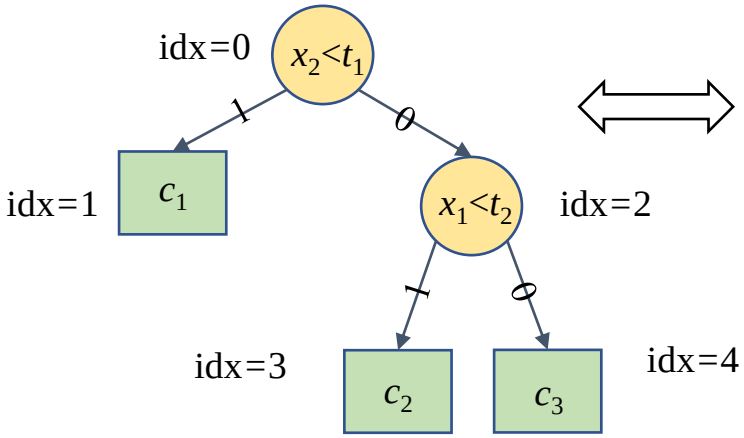


idx	*t*	*l*	*r*	*v*	*c*
0	t_1	1	2	2	$\perp$
1	$\perp$	$\perp$	$\perp$	$\perp$	c_1
2	t_2	3	4	1	$\perp$
3	$\perp$	$\perp$	$\perp$	$\perp$	c_2
4	$\perp$	$\perp$	$\perp$	$\perp$	c_3

- $\mathrm{DTE}(T, X)$:
  - $idx \leftarrow 0$
  - For $i \in [1, d]$:
    - $(t, l, r, v, c) \leftarrow T[idx]$
    - If $l = \perp$, return c
    - $x \leftarrow X[v]$
    - $b \leftarrow x < t$
    - $idx \leftarrow r \oplus b \cdot (r \oplus l)$

- Hiding intermediate values
  - Secured by MPC (RSS)

Decision Tree Evaluation

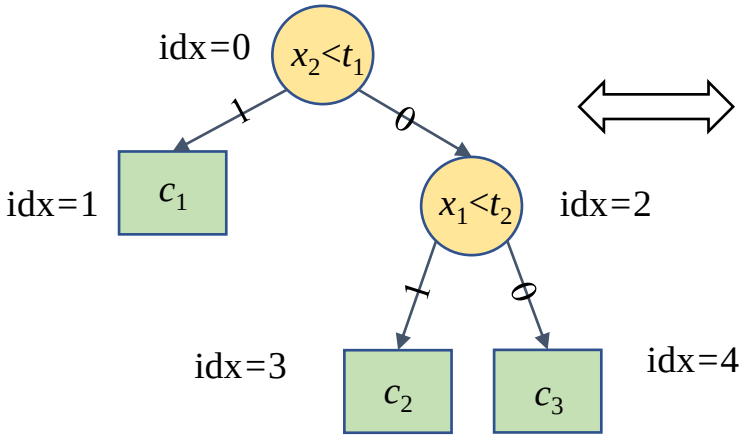


idx	*t*	*l*	*r*	*v*	*c*
0	t_1	1	2	2	$\perp$
1	$\perp$	$\perp$	$\perp$	$\perp$	c_1
2	t_2	3	4	1	$\perp$
3	$\perp$	$\perp$	$\perp$	$\perp$	c_2
4	$\perp$	$\perp$	$\perp$	$\perp$	c_3

- DTE(T, X):
  - $idx \leftarrow 0$
  - For $i \in [1, d]$:
    - $(t, l, r, v, c) \leftarrow T[idx]$
    - If $l = \perp$, return c
    - $x \leftarrow X[v]$
    - $b \leftarrow x < t$
    - $idx \leftarrow r \oplus b \cdot (r \oplus l)$

- Hiding intermediate values
  - Secured by MPC (RSS)
  - d secure comparison/MUX operations

Decision Tree Evaluation

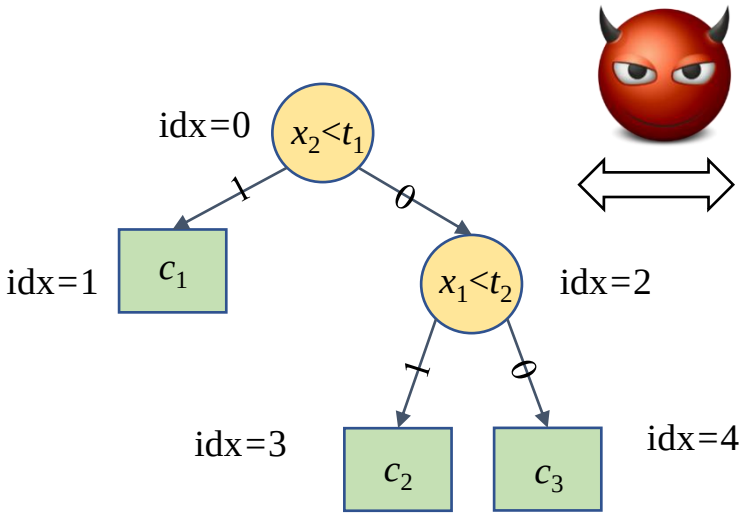


idx	*t*	*l*	*r*	*v*	*c*
0	t_1	1	2	2	$\perp$
1	$\perp$	$\perp$	$\perp$	$\perp$	c_1
2	t_2	3	4	1	$\perp$
3	$\perp$	$\perp$	$\perp$	$\perp$	c_2
4	$\perp$	$\perp$	$\perp$	$\perp$	c_3

- $\mathrm{DTE}(T, X)$:
  - $idx \leftarrow 0$
  - For $i \in [1, d]$:
    - $(t, l, r, v, c) \leftarrow T[idx]$
    - If $l = \perp$, return c
    - $x \leftarrow X[v]$
    - $b \leftarrow x < t$
    - $idx \leftarrow r \oplus b \cdot (r \oplus l)$

- Hiding access pattern

Decision Tree Evaluation

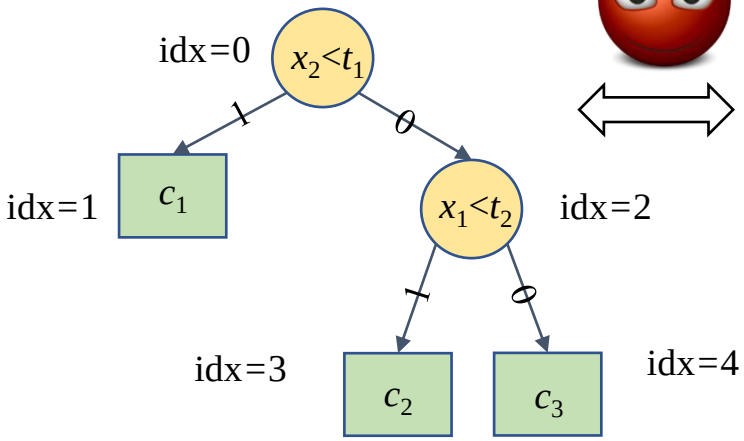


idx	*t*	*l*	*r*	*v*	*c*
0	t_1	1	2	2	$\perp$
1	$\perp$	$\perp$	$\perp$	$\perp$	c_1
2	t_2	3	4	1	$\perp$
3	$\perp$	$\perp$	$\perp$	$\perp$	c_2
4	$\perp$	$\perp$	$\perp$	$\perp$	c_3

- DTE(T, X):
  - $idx \leftarrow 0$
  - For $i \in [1, d]$:
    - $(t, l, r, v, c) \leftarrow T[idx]$
    - If $l = \perp$, return c
    - $x \leftarrow X[v]$
    - $b \leftarrow x < t$
    - $idx \leftarrow r \oplus b \cdot (r \oplus l)$

- Hiding access pattern

Decision Tree Evaluation

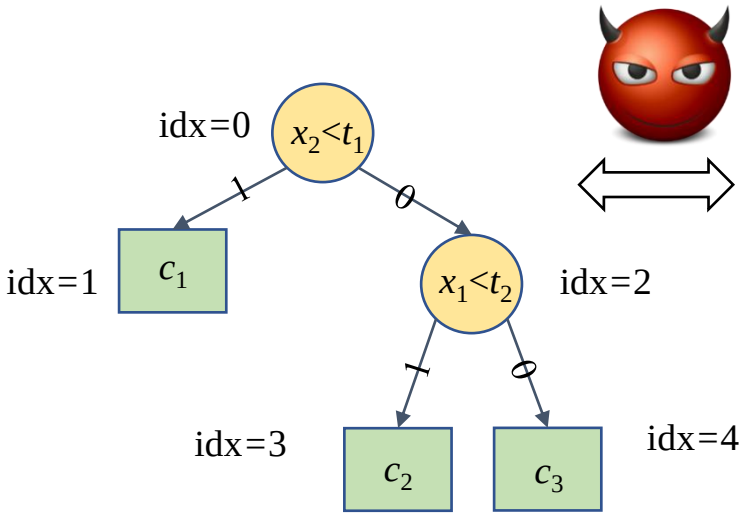


idx	*t*	*l*	*r*	*v*	*c*
0	t_1	1	2	2	$\perp$
1	$\perp$	$\perp$	$\perp$	$\perp$	c_1
2	t_2	3	4	1	$\perp$
3	$\perp$	$\perp$	$\perp$	$\perp$	c_2
4	$\perp$	$\perp$	$\perp$	$\perp$	c_3

- DTE(T, X):
  - $idx \leftarrow 0$
  - For $i \in [1, d]$:
    - $(t, l, r, v, c) \leftarrow T[idx]$
    - If $l = \perp$, return c
    - $x \leftarrow X[v]$
    - $b \leftarrow x < t$
    - $idx \leftarrow r \oplus b \cdot (r \oplus l)$

- Hiding access pattern

Decision Tree Evaluation

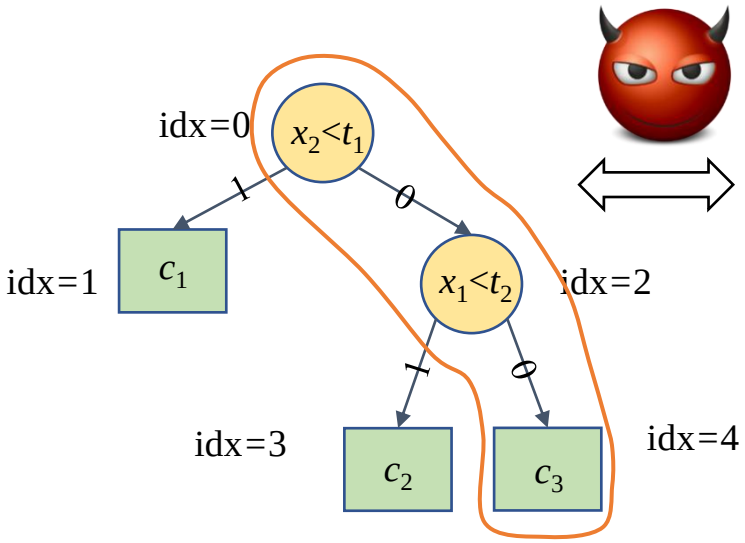


idx	*t*	*l*	*r*	*v*	*c*
0	t_1	1	2	2	$\perp$
1	$\perp$	$\perp$	$\perp$	$\perp$	c_1
2	t_2	3	4	1	$\perp$
3	$\perp$	$\perp$	$\perp$	$\perp$	c_2
4	$\perp$	$\perp$	$\perp$	$\perp$	c_3

- DTE(T, X):
  - $idx \leftarrow 0$
  - For $i \in [1, d]$:
    - $(t, l, r, v, c) \leftarrow T[idx]$
    - If $l = \perp$, return c
    - $x \leftarrow X[v]$
    - $b \leftarrow x < t$
    - $idx \leftarrow r \oplus b \cdot (r \oplus l)$

- Hiding access pattern

Decision Tree Evaluation

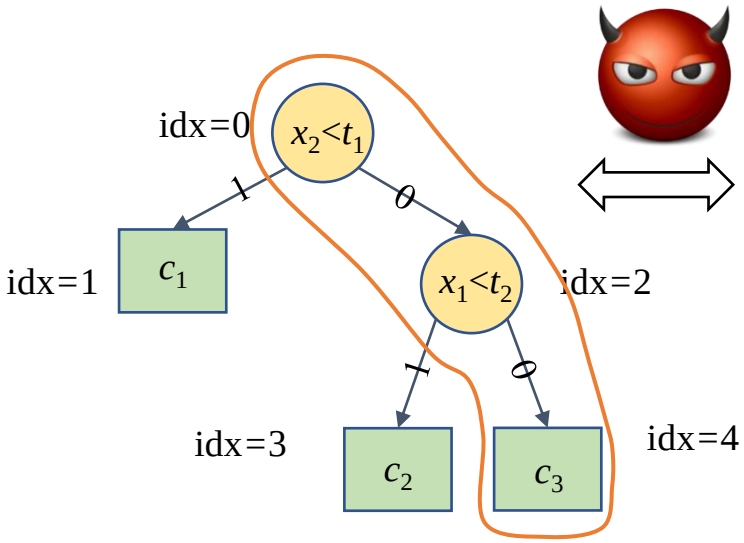


idx	*t*	*l*	*r*	*v*	*c*
0	t_1	1	2	2	$\perp$
1	$\perp$	$\perp$	$\perp$	$\perp$	c_1
2	t_2	3	4	1	$\perp$
3	$\perp$	$\perp$	$\perp$	$\perp$	c_2
4	$\perp$	$\perp$	$\perp$	$\perp$	c_3

- DTE(T, X):
  - $idx \leftarrow 0$
  - For $i \in [1, d]$:
    - $(t, l, r, v, c) \leftarrow T[idx]$
    - If $l = \perp$, return c
    - $x \leftarrow X[v]$
    - $b \leftarrow x < t$
    - $idx \leftarrow r \oplus b \cdot (r \oplus l)$

- Hiding access pattern

Decision Tree Evaluation

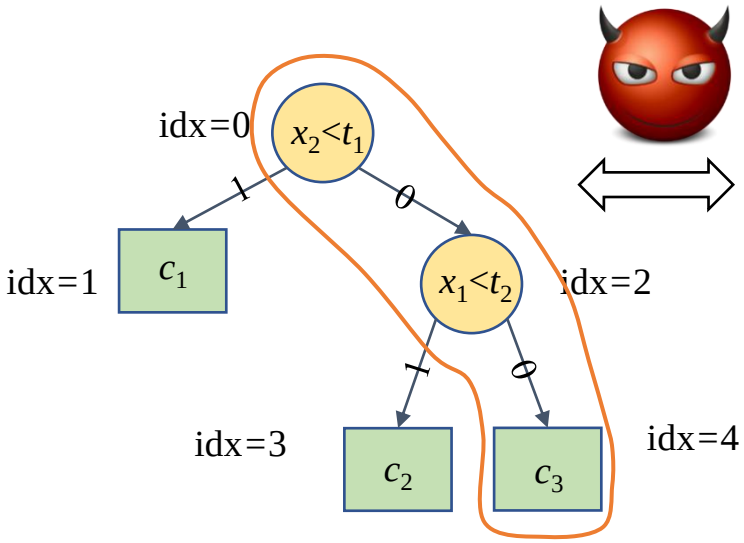


idx	*t*	*l*	*r*	*v*	*c*
0	t_1	1	2	2	$\perp$
1	$\perp$	$\perp$	$\perp$	$\perp$	c_1
2	t_2	3	4	1	$\perp$
3	$\perp$	$\perp$	$\perp$	$\perp$	c_2
4	$\perp$	$\perp$	$\perp$	$\perp$	c_3

- DTE(T, X):
  - $idx \leftarrow 0$
  - For $i \in [1, d]$:
    - $(t, l, r, v, c) \leftarrow T[idx]$
    - If $l = \perp$, return c
    - $x \leftarrow X[v]$
    - $b \leftarrow x < t$
    - $idx \leftarrow r \oplus b \cdot (r \oplus l)$

- Hiding access pattern
  - Linear scan: linear communication [DEFK19]
  - Oblivious RAM (ORAM):
    - Sublinear communication
    - Heavy overhead: evaluating ORAM circuit within MPC

Decision Tree Evaluation



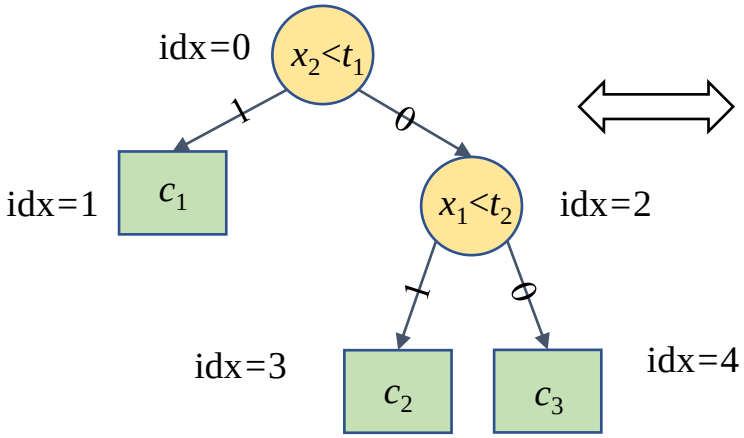
idx=4

idx	*t*	*l*	*r*	*v*	*c*
0	t_1	1	2	2	$\perp$
1	$\perp$	$\perp$	$\perp$	$\perp$	c_1
2	t_2	3	4	1	$\perp$
3	$\perp$	$\perp$	$\perp$	$\perp$	c_2
4	$\perp$	$\perp$	$\perp$	$\perp$	c_3

- DTE(T, X):
  - $idx \leftarrow 0$
  - For $i \in [1, d]$:
    - $(t, l, r, v, c) \leftarrow T[idx]$
    - If $l = \perp$, return c
    - $x \leftarrow X[v]$
    - $b \leftarrow x < t$
    - $idx \leftarrow r \oplus b \cdot (r \oplus l)$

- Hiding access pattern
  - Linear scan: linear communication [DEFK19]
  - Oblivious RAM (ORAM):
    - Sublinear communication
    - Heavy overhead: evaluating ORAM circuit within MPC
- Observation: DTE only involves read operations, i.e., selection
- Solution: more efficient oblivious selection protocol

Decision Tree Evaluation

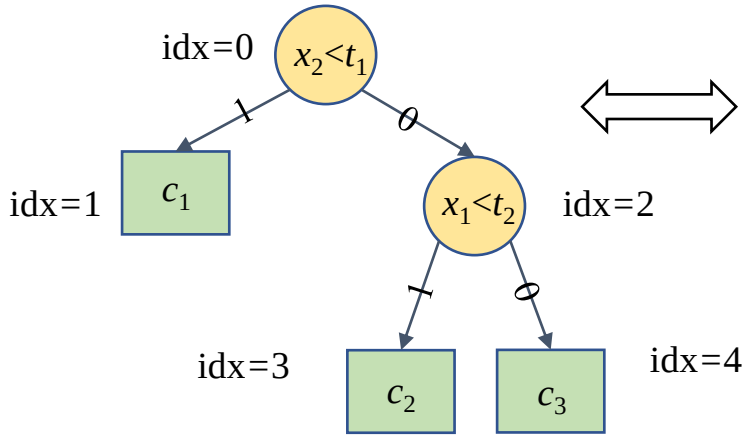


idx	*t*	*l*	*r*	*v*	*c*
0	t_1	1	2	2	$\perp$
1	$\perp$	$\perp$	$\perp$	$\perp$	c_1
2	t_2	3	4	1	$\perp$
3	$\perp$	$\perp$	$\perp$	$\perp$	c_2
4	$\perp$	$\perp$	$\perp$	$\perp$	c_3

- $\mathrm{DTE}(T, X)$:
  - $idx \leftarrow 0$
  - For $i \in [1, d]$:
    - $(t, l, r, v, c) \leftarrow T[idx]$
    - If $l = \perp$, return c
    - $x \leftarrow X[v]$
    - $b \leftarrow x < t$
    - $idx \leftarrow r \oplus b \cdot (r \oplus l)$

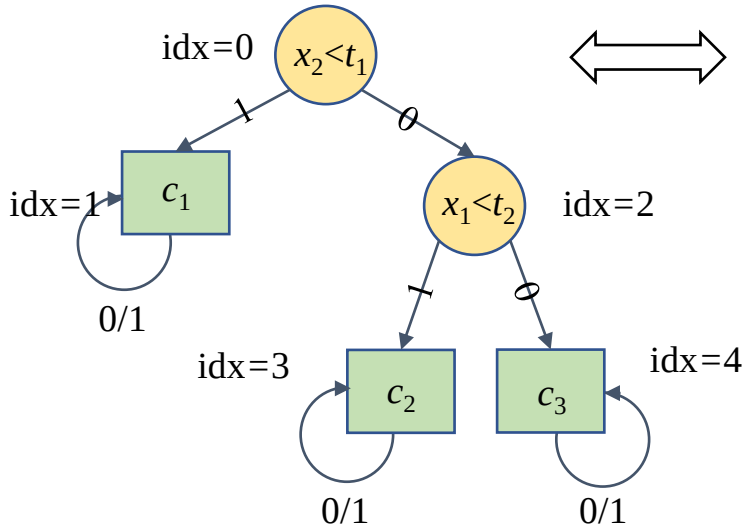
- Hiding path length
  - e.g., what if the evaluation return at $i = 2$?
- Solution: dummy evaluation
  - For a decision path with length d, do additional $d' - d$ iterations of dummy evaluation

Decision Tree Evaluation



idx	*t*	*l*	*r*	*v*	*c*
0	t_1	1	2	2	$\perp$
1	$\perp$	$\perp$	$\perp$	$\perp$	c_1
2	t_2	3	4	1	$\perp$
3	$\perp$	$\perp$	$\perp$	$\perp$	c_2
4	$\perp$	$\perp$	$\perp$	$\perp$	c_3

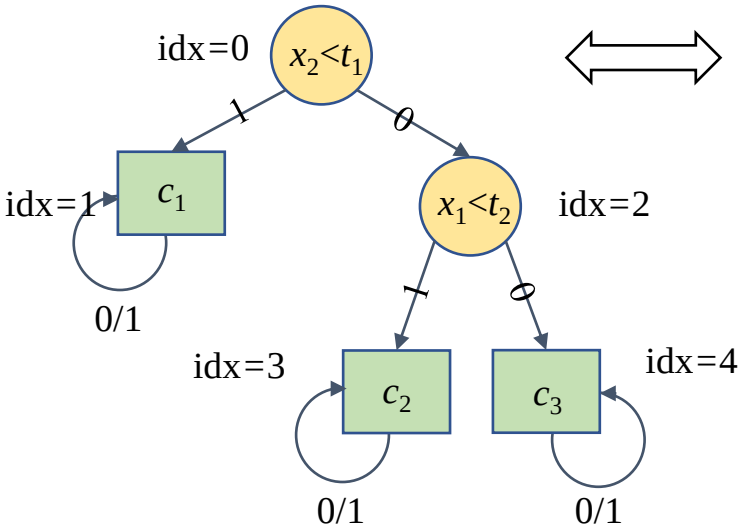
- DTE(T, X):
  - $idx \leftarrow 0$
  - For $i \in [1, d]$:
    - $(t, l, r, v, c) \leftarrow T[idx]$
    - If $l = \perp$, return c
    - $x \leftarrow X[v]$
    - $b \leftarrow x < t$
    - $idx \leftarrow r \oplus b \cdot (r \oplus l)$



idx	*t*	*l*	*r*	*v*	*c*
0	t_1	1	2	2	$\perp$
1	$\perp$	1	1	$\perp$	c_1
2	t_2	3	4	1	$\perp$
3	$\perp$	3	3	$\perp$	c_2
4	$\perp$	4	4	$\perp$	c_3

- DTE(T, X):
  - $idx \leftarrow 0, result \leftarrow \perp$
  - For $i \in [1, d']$: $//d' \geq d$
    - $(t, l, r, v, c) \leftarrow T[idx]$
    - $x \leftarrow X[v]$
    - $b \leftarrow x < t$
    - $idx \leftarrow r \oplus b \cdot (r \oplus l)$
    - $result \leftarrow c$
  - Return $result$

Private Decision Tree Evaluation

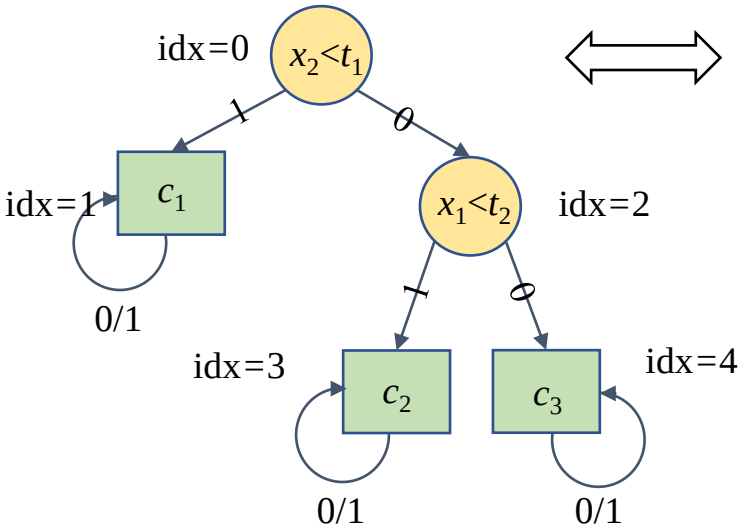


idx	*t*	*l*	*r*	*v*	*c*
0	t_1	1	2	2	$\perp$
1	$\perp$	1	1	$\perp$	c_1
2	t_2	3	4	1	$\perp$
3	$\perp$	3	3	$\perp$	c_2
4	$\perp$	4	4	$\perp$	c_3

- $\mathrm{DTE}(T, X)$:
  - $idx \leftarrow 0, result \leftarrow \perp$
  - For $i \in [1, d']$: $//d' \geq d$
    - $(t, l, r, v, c) \leftarrow T[idx]$
    - $x \leftarrow X[v]$
    - $b \leftarrow x < t$
    - $idx \leftarrow r \oplus b \cdot (r \oplus l)$
    - $result \leftarrow c$
  - Return $result$

- Hiding sensitive values of tree/feature
  - via MPC
- Hiding access pattern
  - via oblivious selection protocols
- Hiding length information
  - via dummy evaluation

Private Decision Tree Evaluation

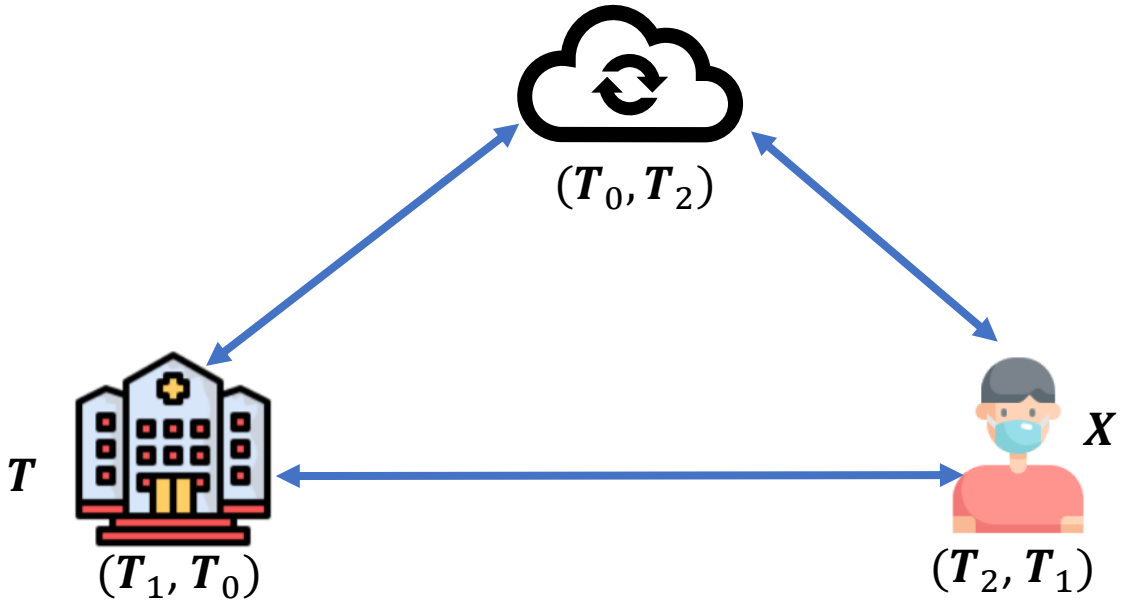


idx	*t*	*l*	*r*	*v*	*c*
0	t_1	1	2	2	$\perp$
1	$\perp$	1	1	$\perp$	c_1
2	t_2	3	4	1	$\perp$
3	$\perp$	3	3	$\perp$	c_2
4	$\perp$	4	4	$\perp$	c_3

- $\mathrm{DTE}(T, X)$:
  - $idx \leftarrow 0, result \leftarrow \perp$
  - For $i \in [1, d']$: $//d' \geq d$
    - $(t, l, r, v, c) \leftarrow T[idx]$
    - $x \leftarrow X[v]$
    - $b \leftarrow x < t$
    - $idx \leftarrow r \oplus b \cdot (r \oplus l)$
    - $result \leftarrow c$
  - Return $result$

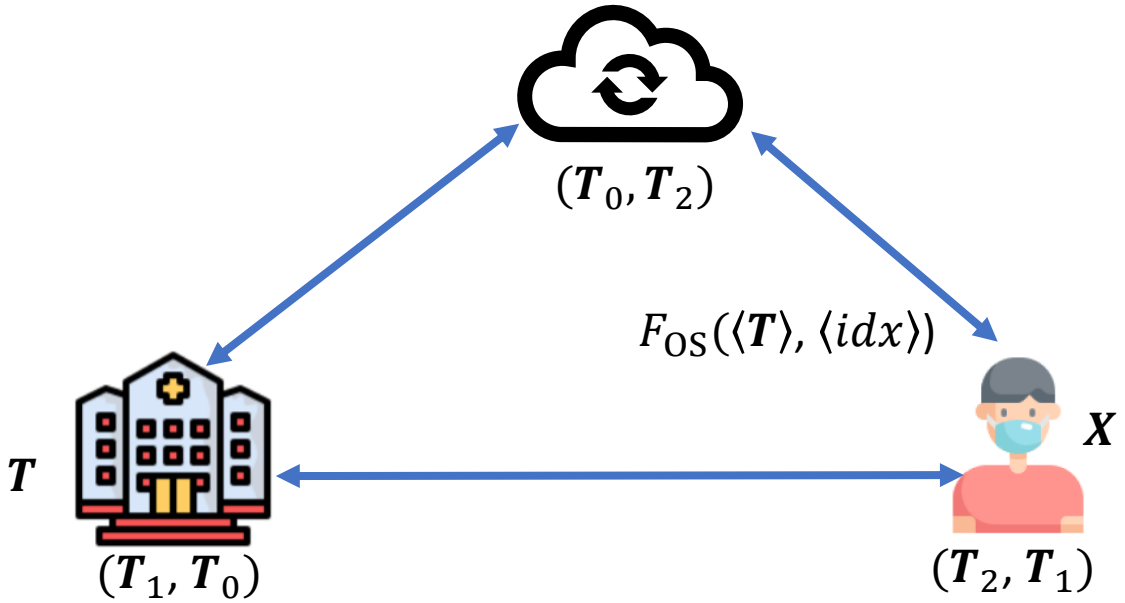
- Hiding sensitive values of tree/feature
  - via MPC
- Hiding access pattern
  - via oblivious selection protocols
- Hiding length information
  - via dummy evaluation

Three-party Oblivious Selection (OS)



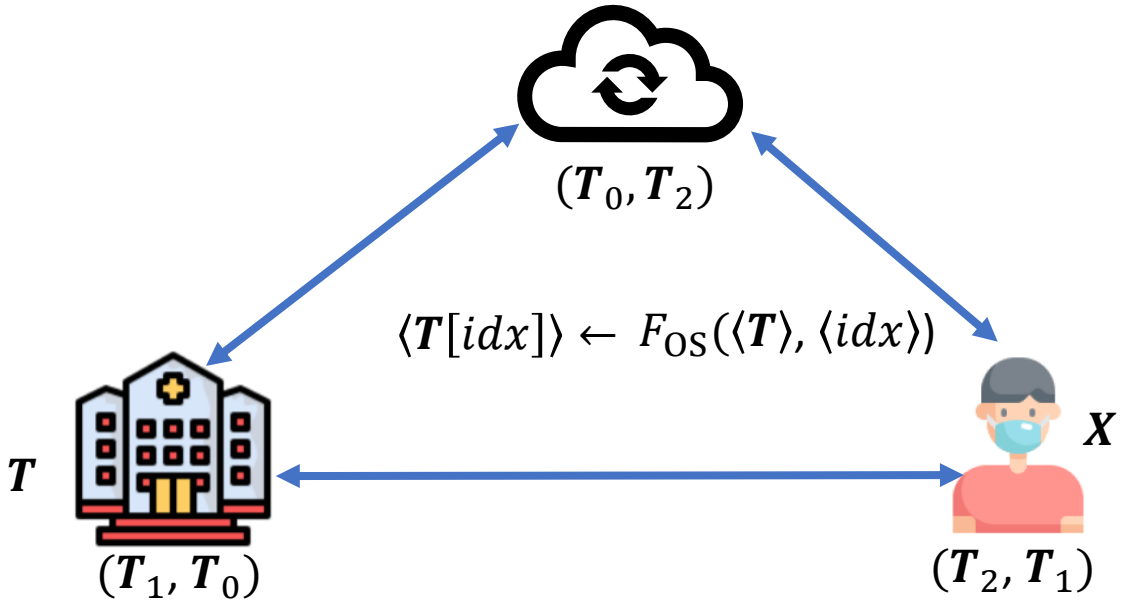
- Security
  - Privacy: No party knows idx nor $\boldsymbol{T}[idx]$
  - Correctness: The parties share the correct selection result $\boldsymbol{T}[idx]$
- Efficiency
  - Sublinear communication (in $|\boldsymbol{T}|$)

Three-party Oblivious Selection (OS)



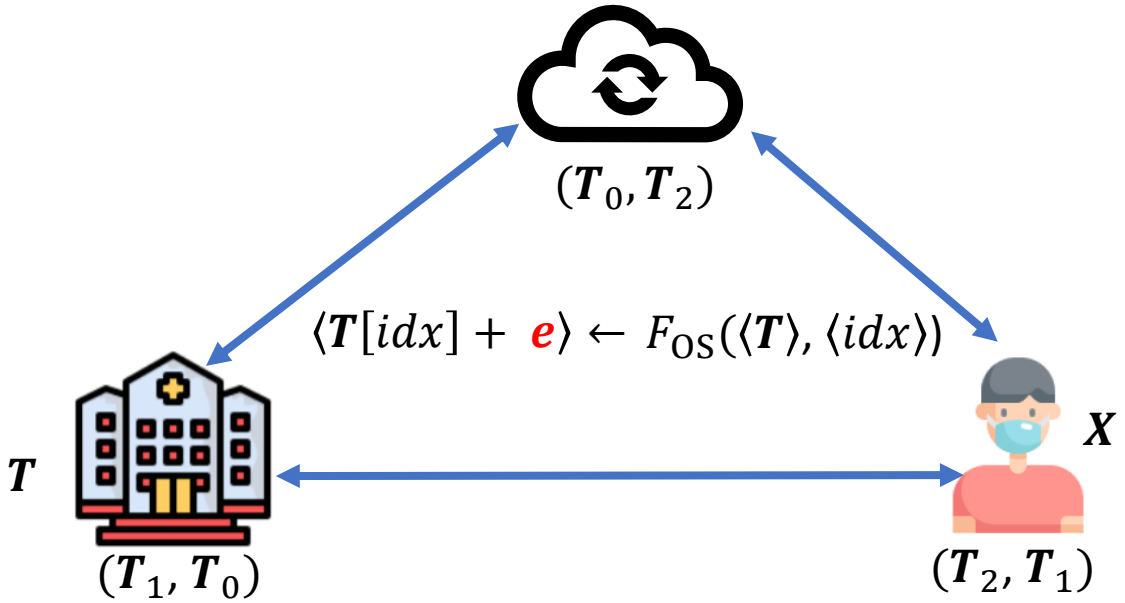
- Security
  - Privacy: No party knows idx nor $T[idx]$
  - Correctness: The parties share the correct selection result $T[idx]$
- Efficiency
  - Sublinear communication (in $|T|$)

Three-party Oblivious Selection (OS)



- Security
  - Privacy: No party knows idx nor $T[idx]$
  - Correctness: The parties share the correct selection result $T[idx]$
- Efficiency
  - Sublinear communication (in $|T|$)

Three-party Oblivious Selection (OS)



- Security
  - Privacy: No party knows idx nor $T[idx]$ ☺
  - Correctness: The adversary can add an error e to the result ☹
- Efficiency
  - Sublinear communication (in $|T|$)

3PC Oblivious Selection using Pure RSS

Oblivious Selection from Pure RSS

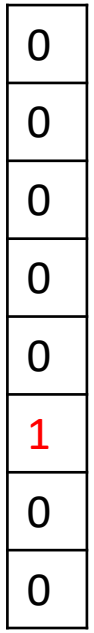
$$\boldsymbol{v}[i] = \begin{cases} 0, i \neq idx \\ 1, i = idx \end{cases}$$

- OS from inner-product computation.
  - Input: Vector sharing $\langle \boldsymbol{T} \rangle$; one-hot RSS-shared vector $(\langle \boldsymbol{v} \rangle, \langle idx \rangle)$.
  - Output: $\langle \boldsymbol{T}[idx] \rangle$.

- Security/Efficiency issues:
  1. $O(n)$ communication for generating $\langle \boldsymbol{v} \rangle$.
  2. $O(n)$ communication from n secret-shared multiplications.
  3. Additive attacks from a malicious party.

Derandomization for One-hot Vector

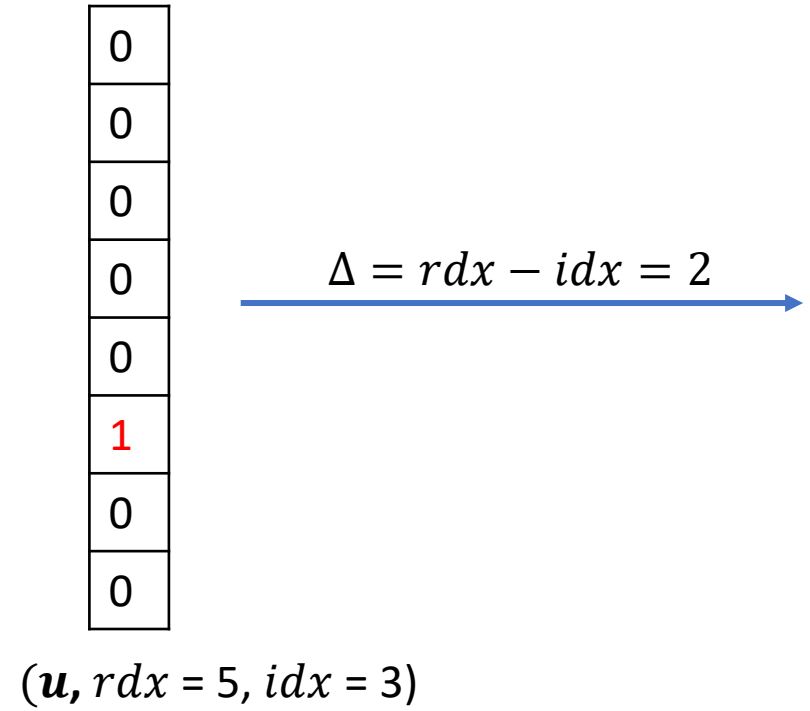
- Suppose the parties already share a one-hot vector $(\langle \boldsymbol{u} \rangle, \langle rdx \rangle)$



$(\boldsymbol{u}, rdx = 5, idx = 3)$

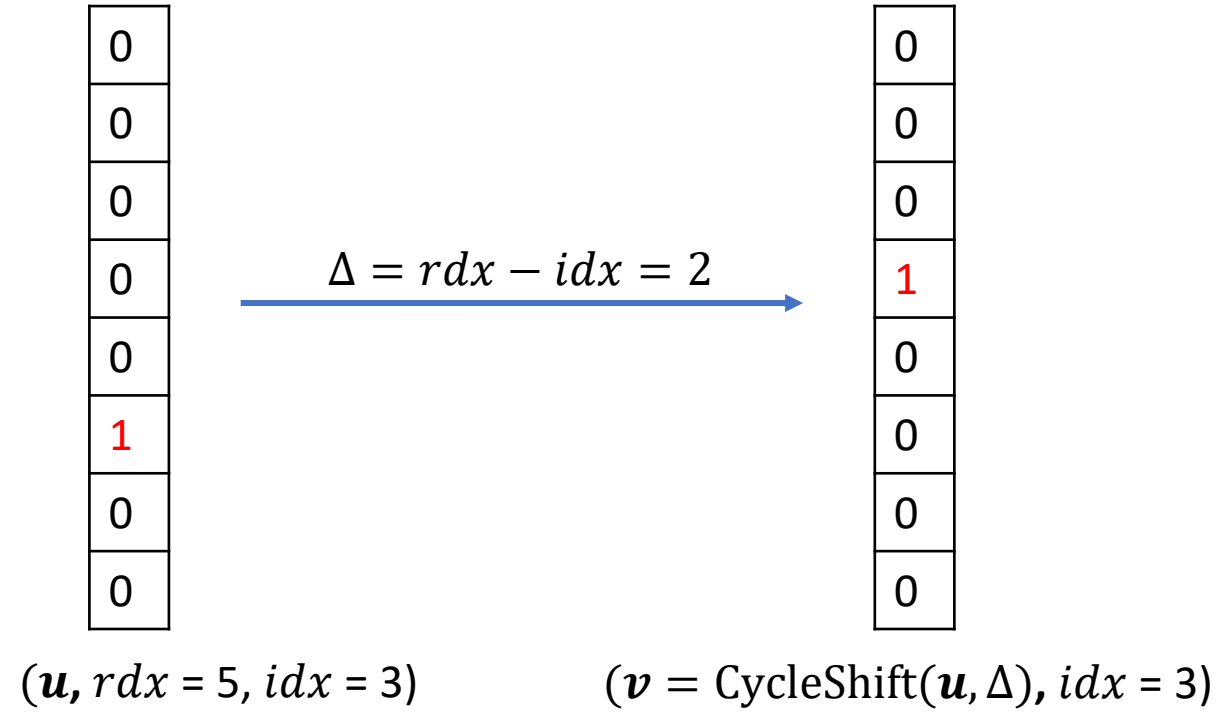
Derandomization for One-hot Vector

- Suppose the parties already share a one-hot vector $(\langle \boldsymbol{u} \rangle, \langle rdx \rangle)$



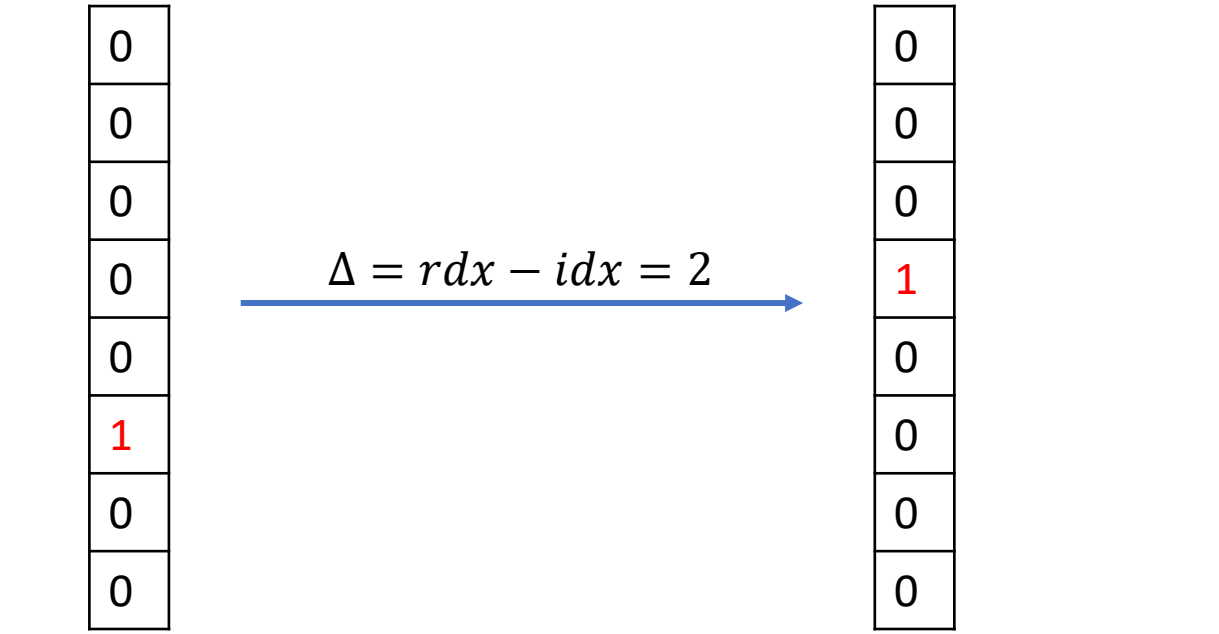
Derandomization for One-hot Vector

- Suppose the parties already share a one-hot vector $(\langle \boldsymbol{u} \rangle, \langle rdx \rangle)$



Derandomization for One-hot Vector

- Suppose the parties already share a one-hot vector $(\langle \boldsymbol{u} \rangle, \langle rdx \rangle)$



$(\boldsymbol{u}, rdx = 5, idx = 3)$ $(\boldsymbol{v} = \text{CycleShift}(\boldsymbol{u}, \Delta), idx = 3)$

$$\boldsymbol{v}[i] = \boldsymbol{u}[i + \Delta] = \begin{cases} 1, i = idx \\ 0, i \neq idx \end{cases}$$

Oblivious Selection with up to Additive Attacks

- Check correctness of multiplication:
  - Previous approach: Using an RSS triple $(\langle a\rangle, \langle b\rangle, \langle ab\rangle)$ to check correctness
  - $O(n)$ communication in total

Oblivious Selection with up to Additive Attacks

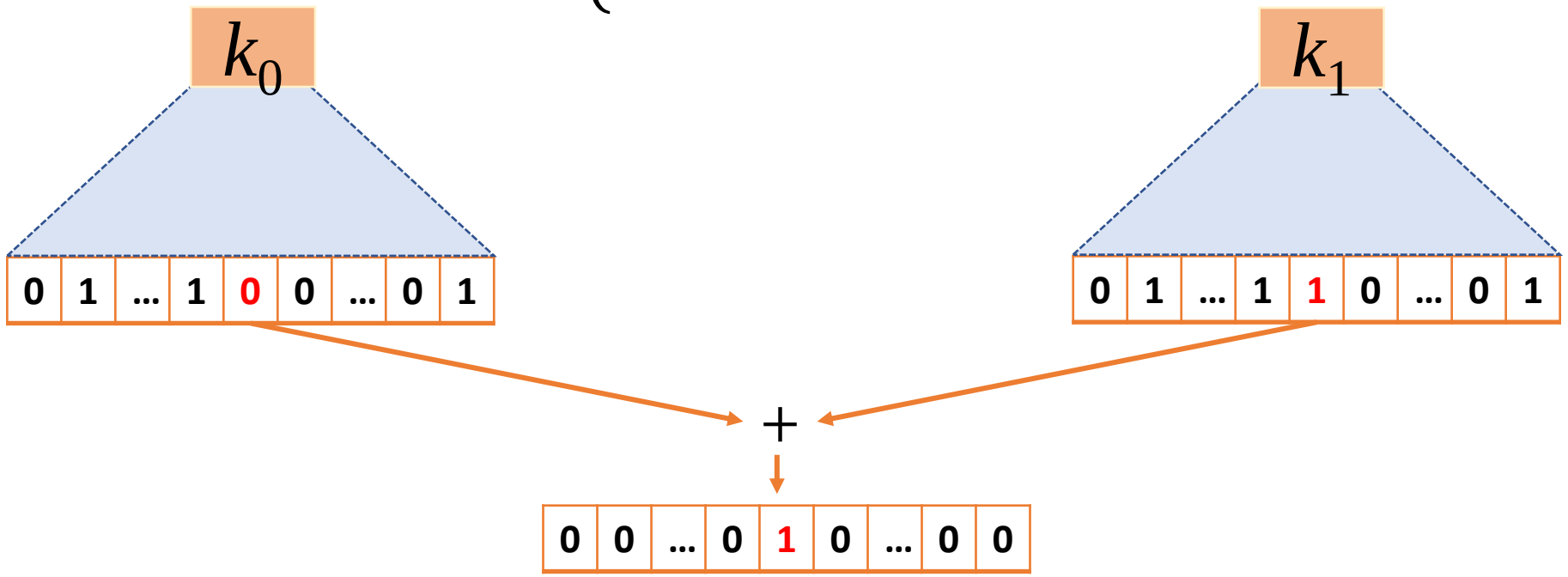
- Check correctness of multiplication:
  - Previous approach: Using an RSS triple $(\langle a\rangle, \langle b\rangle, \langle ab\rangle)$ to check correctness
  - $O(n)$ communication in total

- Our method
  - No correctness check for multiplication
  - Up to additive attacks
  - $O(1)$ communication for resharing

3PC Oblivious Selection using Distributed Point Functions

Tool: Distributed Point Function (DPF)

- A point function $f_{\alpha,\beta}(x) = \begin{cases} \beta, & \text{if } x = \alpha, \\ 0, & \text{otherwise.} \end{cases}$



Oblivious Selection from RSS+DPF

$(\boldsymbol{T}_0, \boldsymbol{T}_2)$

$(\boldsymbol{T}_1, \boldsymbol{T}_0)$

$(\boldsymbol{T}_2, \boldsymbol{T}_1)$

Oblivious Selection from RSS+DPF

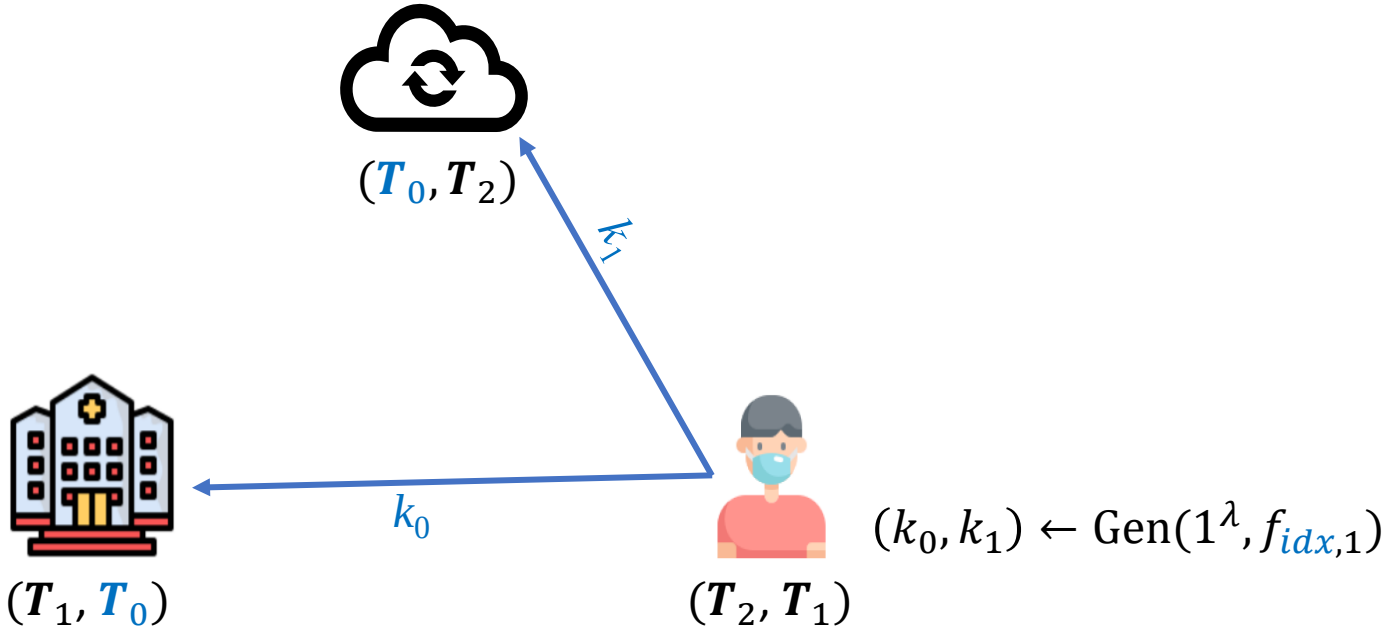
$(\boldsymbol{T}_0, \boldsymbol{T}_2)$

$(\boldsymbol{T}_1, \boldsymbol{T}_0)$

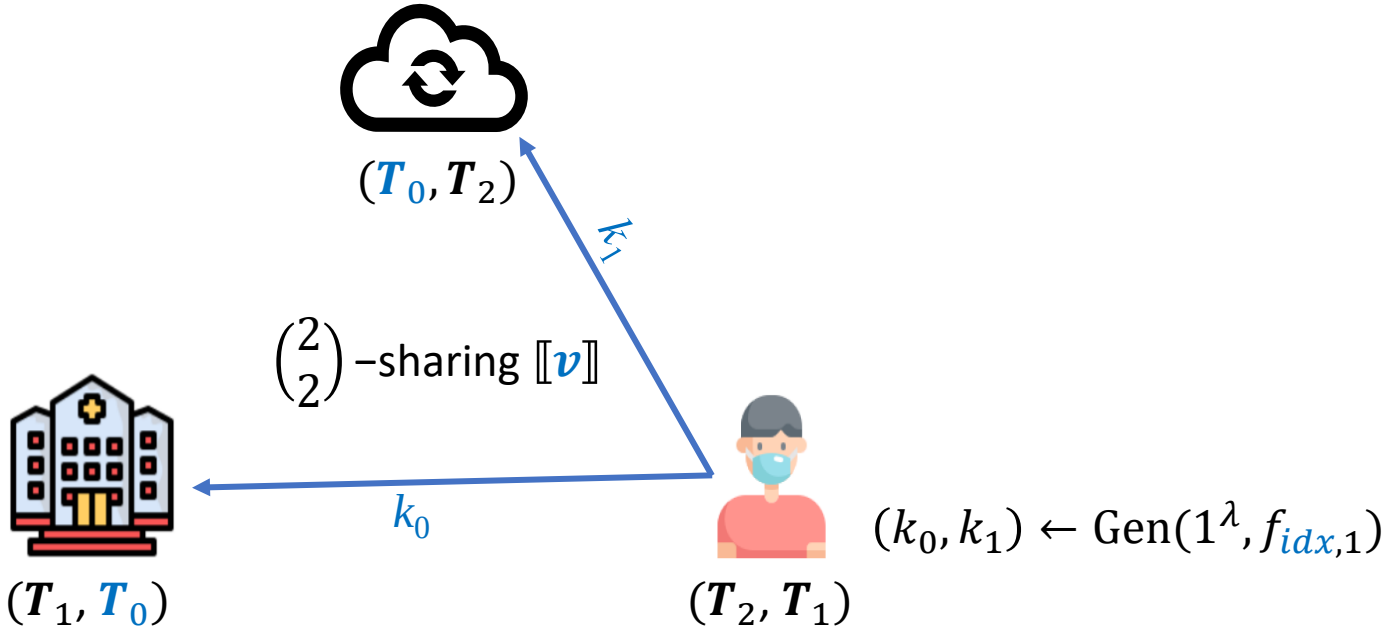
$(\boldsymbol{T}_2, \boldsymbol{T}_1)$

$(k_0, k_1) \leftarrow \mathrm{Gen}(1^\lambda, f_{idx,1})$

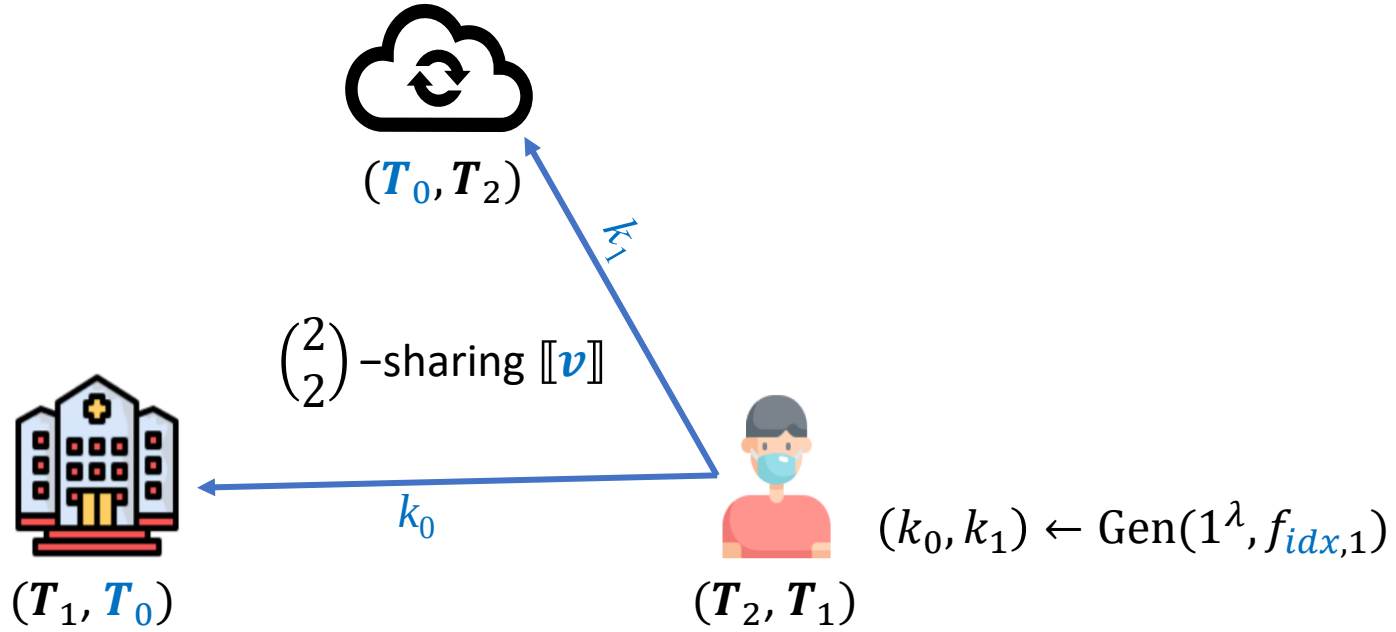
Oblivious Selection from RSS+DPF



Oblivious Selection from RSS+DPF

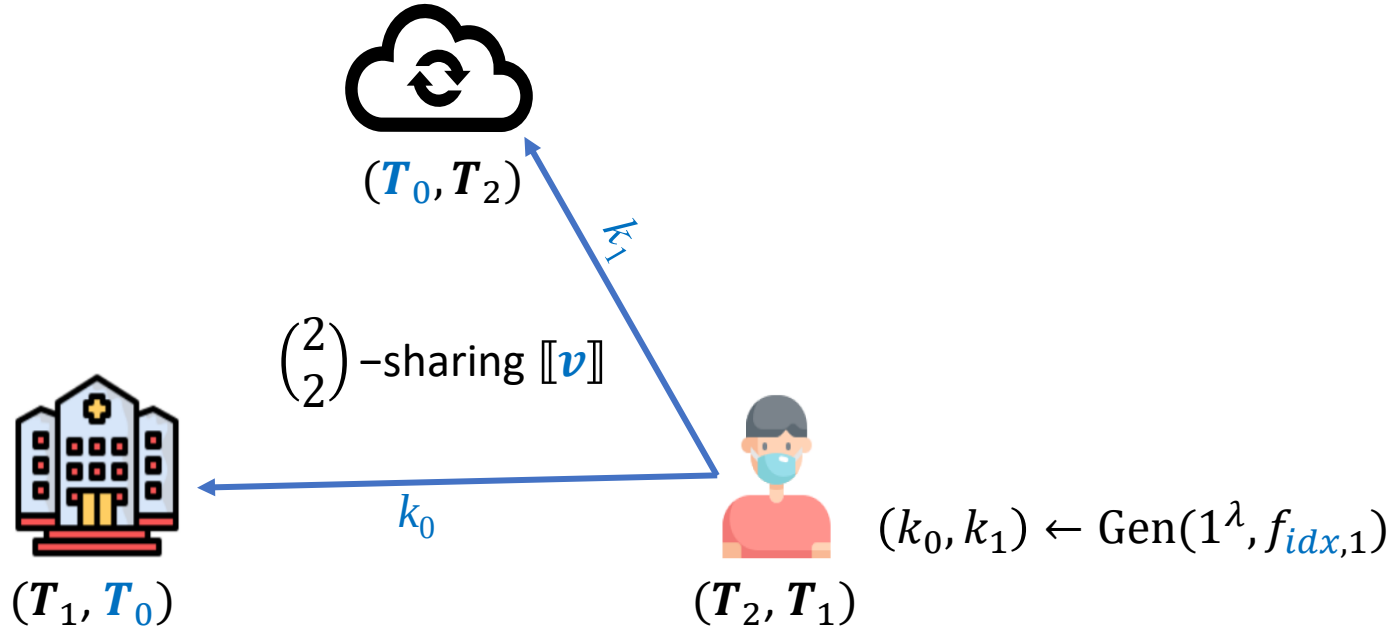


Oblivious Selection from RSS+DPF



- Two-party inner-product computation
  - Input: $[\![\boldsymbol{v}]\!]$ and $\boldsymbol{T}_0$
  - Output: $\binom{2}{2}$–sharing $[\![\boldsymbol{T}_0[idx]]\!]$

Oblivious Selection from RSS+DPF



- Two-party inner-product computation
  - Input: $[\![\boldsymbol{v}]\!]$ and $\boldsymbol{T}_0$
  - Output: $\binom{2}{2}$–sharing $[\![\boldsymbol{T}_0[idx]]\!]$
- Caveat: The dealer learns idx

Oblivious Selection from RSS+DPF

$(\boldsymbol{T}_0, \boldsymbol{T}_2)$

$(\boldsymbol{T}_1, \boldsymbol{T}_0)$

$(\boldsymbol{T}_2, \boldsymbol{T}_1)$

- Two-party inner-product computation
  - Input: $[\![\boldsymbol{v}]\!]$ and $\boldsymbol{T}_0$
  - Output: $\binom{2}{2}$−sharing $[\![\boldsymbol{T}_0[idx]]\!]$
- Caveat: The dealer learns idx
  - Solution: Derandomization for one-hot vector

Oblivious Selection from RSS+DPF

$(\boldsymbol{T}_0, \boldsymbol{T}_2)$

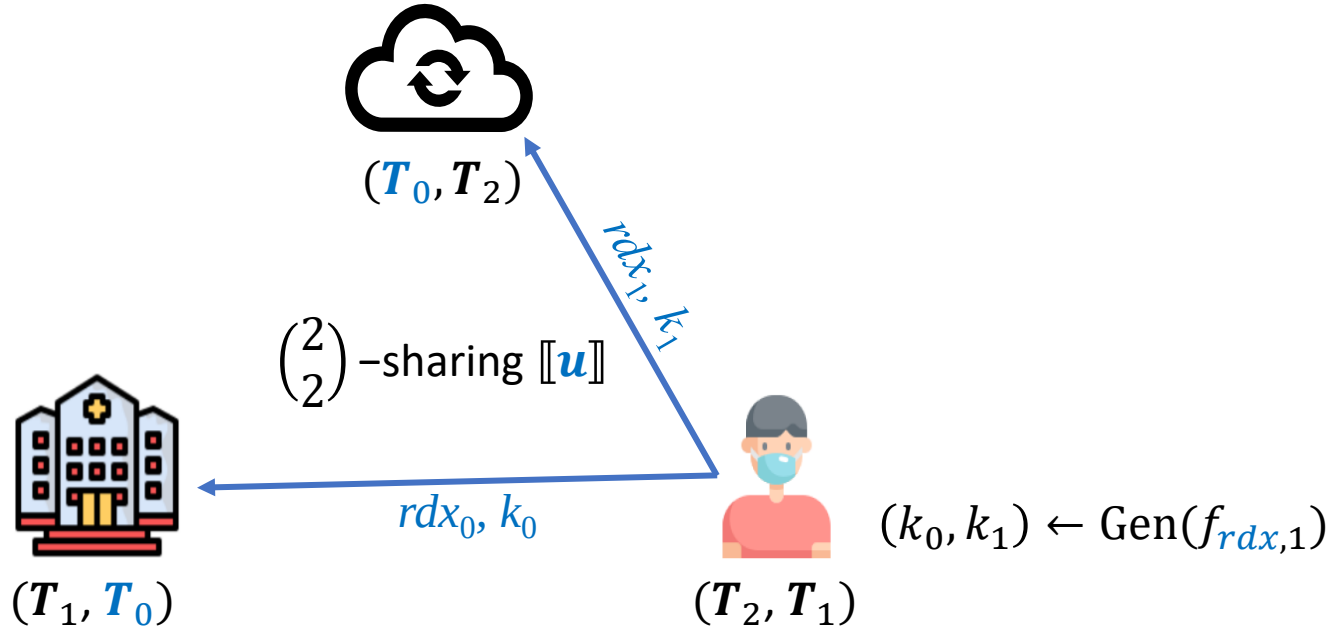
$(\boldsymbol{T}_1, \boldsymbol{T}_0)$

$(\boldsymbol{T}_2, \boldsymbol{T}_1)$

$(k_0, k_1) \leftarrow \mathrm{Gen}(f_{rdx,1})$

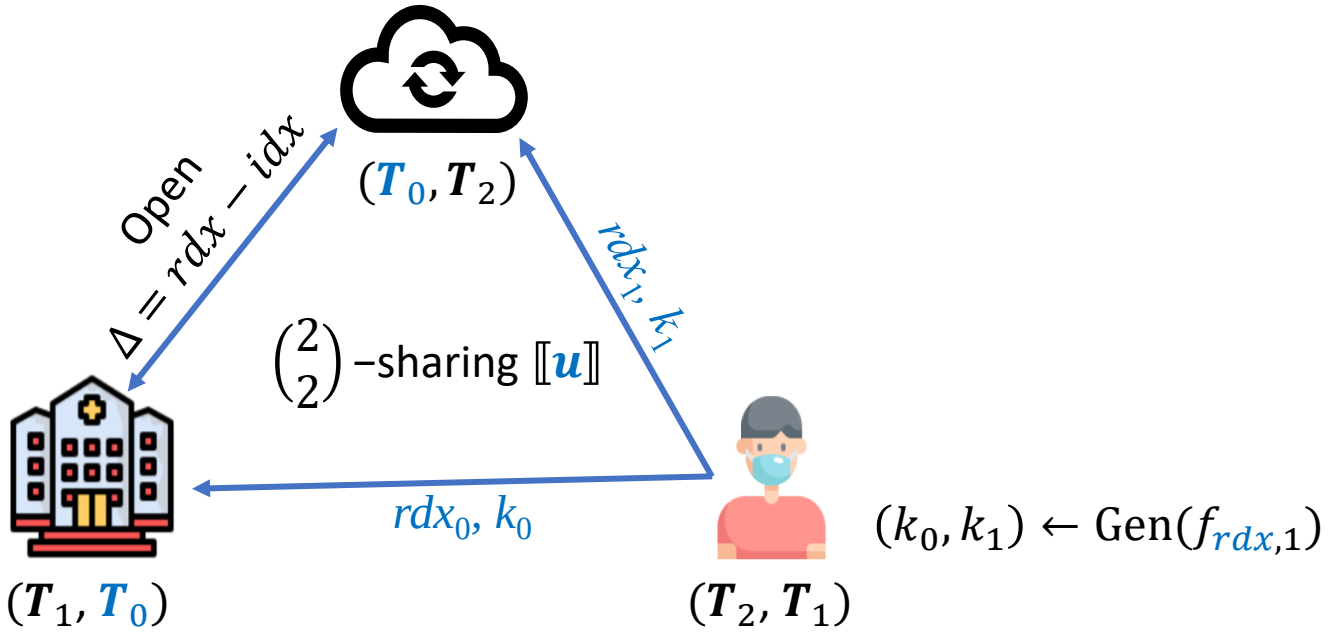
- Two-party inner-product computation
  - Input: $[\![\boldsymbol{v}]\!]$ and $\boldsymbol{T}_0$
  - Output: $\binom{2}{2}$−sharing $[\![\boldsymbol{T}_0[idx]]\!]$
- Caveat: The dealer learns idx
  - Solution: Derandomization for one-hot vector

Oblivious Selection from RSS+DPF



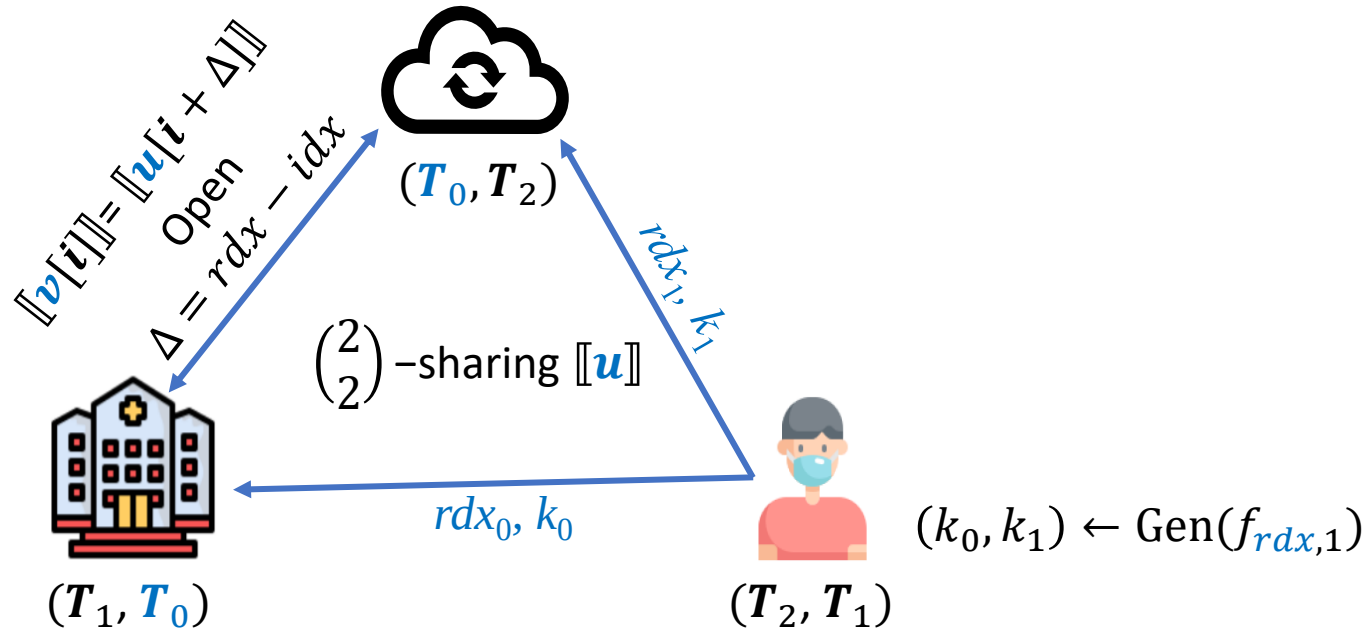
- Two-party inner-product computation
  - Input: $[\![\boldsymbol{v}]\!]$ and $\boldsymbol{T}_0$
  - Output: $\binom{2}{2}$–sharing $[\![\boldsymbol{T}_0[idx]]\!]$
- Caveat: The dealer learns idx
  - Solution: Derandomization for one-hot vector

Oblivious Selection from RSS+DPF



- Two-party inner-product computation
  - Input: $[\![\boldsymbol{v}]\!]$ and $\boldsymbol{T}_0$
  - Output: $\binom{2}{2}$–sharing $[\![\boldsymbol{T}_0[idx]]\!]$
- Caveat: The dealer learns idx
  - Solution: Derandomization for one-hot vector

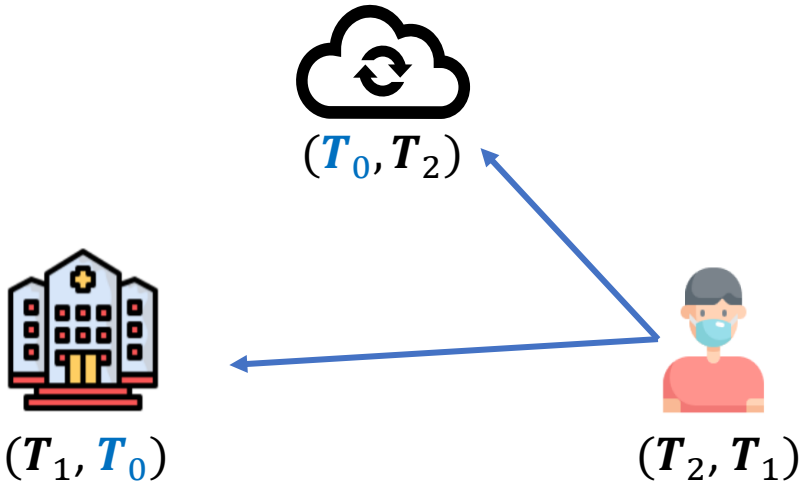
Oblivious Selection from RSS+DPF



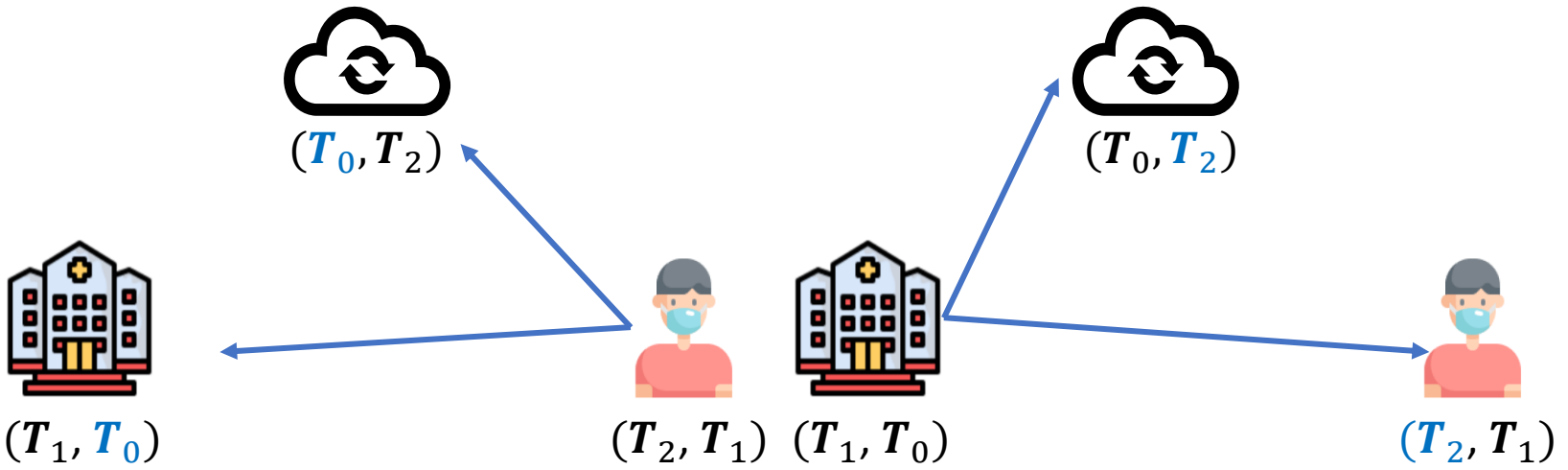
- Two-party inner-product computation
  - Input: $[\![v]\!]$ and T_0
  - Output: $\binom{2}{2}$−sharing $[\![T_0[idx]]\!]$
- Caveat: The dealer learns idx
  - Solution: Derandomization for one-hot vector

Oblivious Selection from RSS+DPF

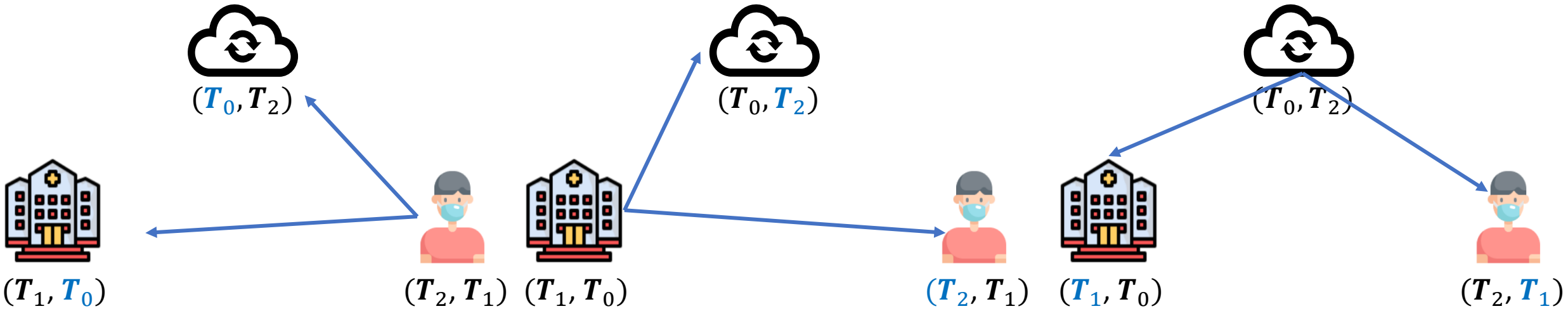
Oblivious Selection from RSS+DPF



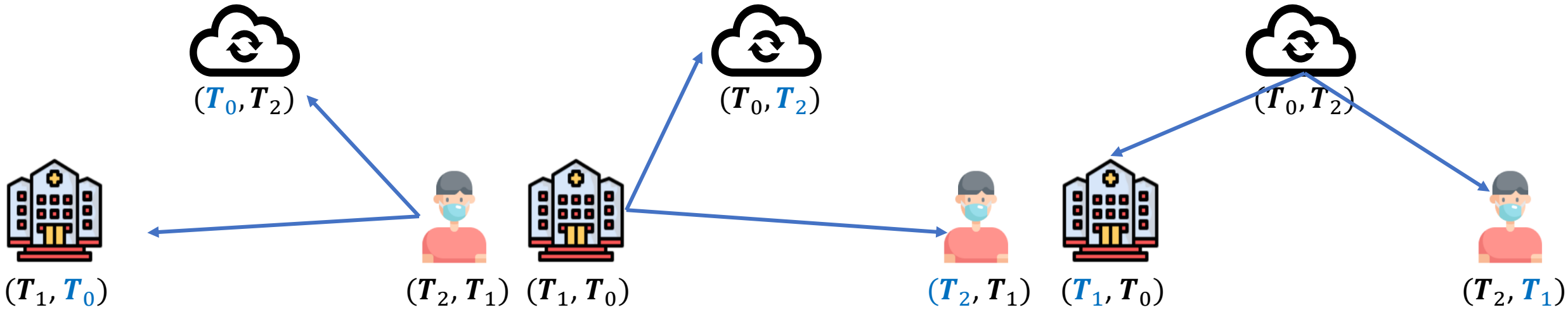
Oblivious Selection from RSS+DPF



Oblivious Selection from RSS+DPF

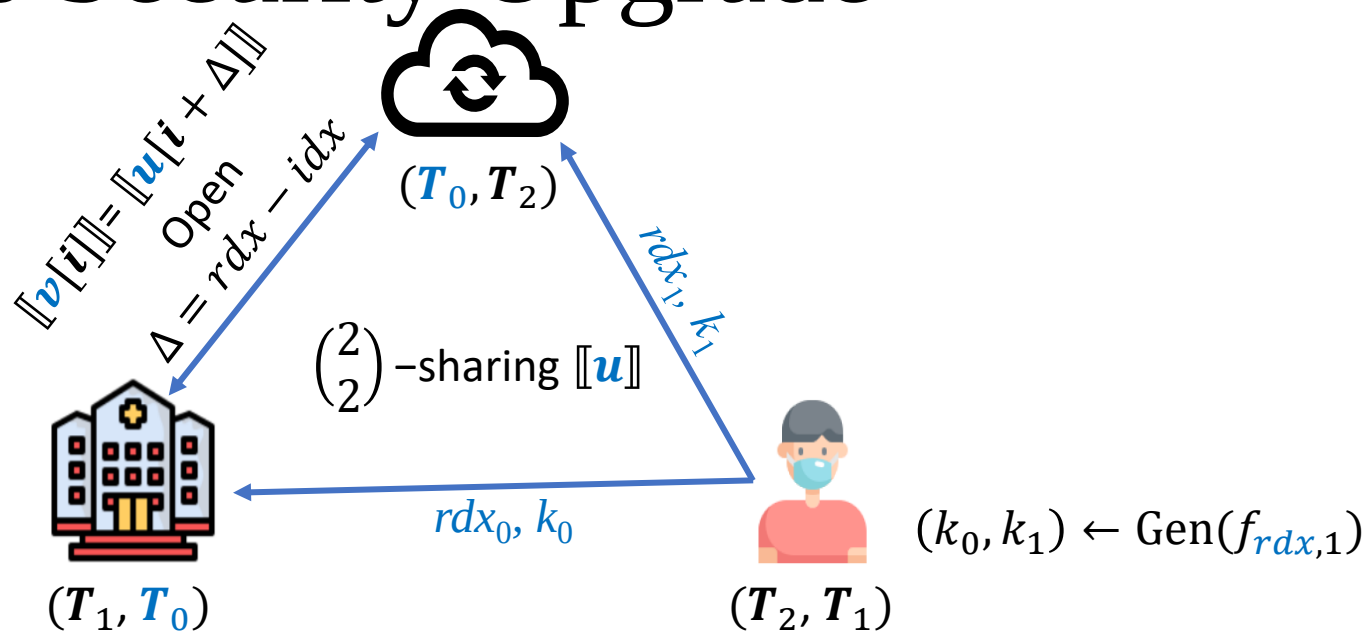


Oblivious Selection from RSS+DPF



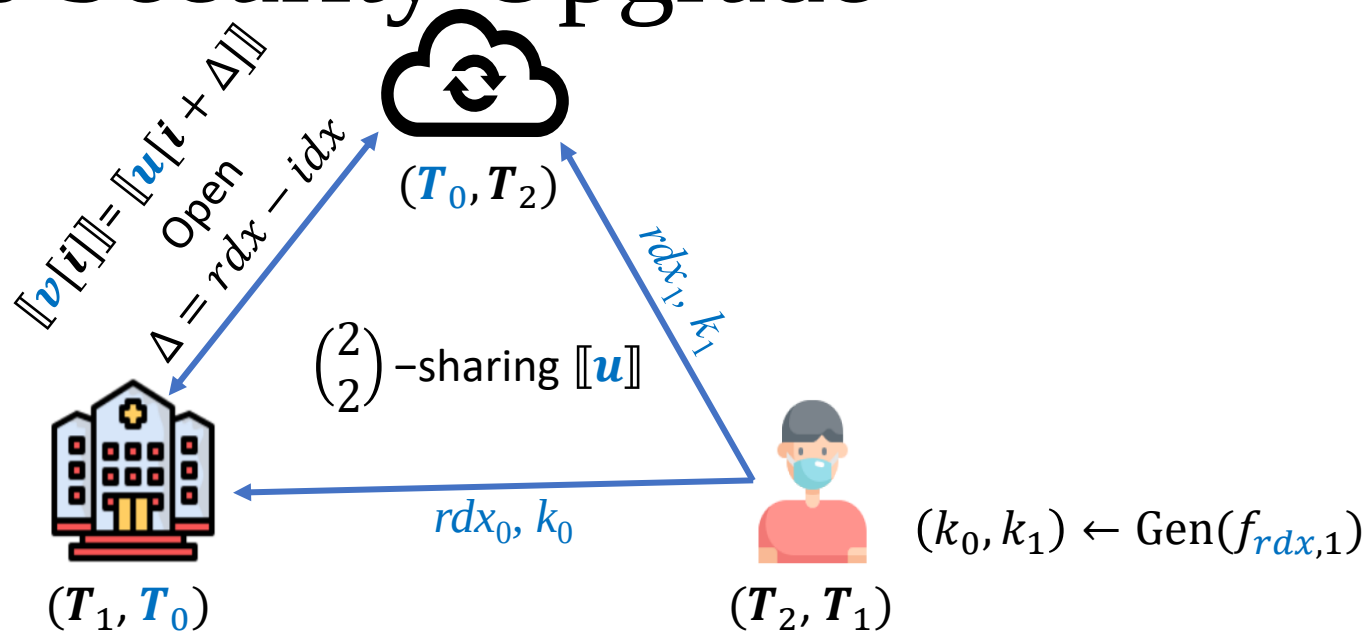
- Pair-wise two-party inner-product computation
  - Input: $\langle \boldsymbol{T} \rangle$ and $\langle idx \rangle$
  - Output: $\binom{3}{3}$–sharing $[\![\boldsymbol{T}[idx]]\!]$

Malicious Security Upgrade



- How to check DPF keys are correctly correlated?
  - Solution: Use Verifiable DPF (VDPF)
- How to check $[\![rdx]\!]$ is consistent with DPF keys?
  - Two evaluating parties check the following equations:
    - $rdx - \sum_i i \cdot \boldsymbol{u}[i] = 0$
- How to open Δ correctly?
  - Non-interactive conversion from $[\![rdx]\!]$ to $\langle rdx \rangle$
  - $\langle \Delta \rangle \leftarrow \langle rdx \rangle - \langle idx \rangle$

Malicious Security Upgrade



- How to check DPF keys are correctly correlated?
  - Solution: Use Verifiable DPF (VDPF)
- How to check $[\![rdx]\!]$ is consistent with DPF keys?
  - Two evaluating parties check the following equations:
    - $rdx - \sum_i i \cdot \boldsymbol{u}[i] = 0$
- How to open Δ correctly?
  - Non-interactive conversion from $[\![rdx]\!]$ to $\langle rdx \rangle$
  - $\langle \Delta \rangle \leftarrow \langle rdx \rangle - \langle idx \rangle$

$P_0: (0, rdx_1)$
$P_1: (rdx_0, 0)$
$P_2: (rdx_1, rdx_0)$

Detect Additive Attacks

- Oblivious selection up to additive errors

$$\langle \boldsymbol{T}[idx] + \boldsymbol{e} \rangle \leftarrow F_{\mathrm{OS}}(\langle \boldsymbol{T} \rangle, \langle idx \rangle)$$

- $\mathrm{DTE}(T, X)$:
  - $idx \leftarrow 0, result \leftarrow \perp$
  - For $i \in [1, d']$: $//d' \geq d$
    - $(t, l, r, v, c) \leftarrow T[idx]$
    - $x \leftarrow X[v]$
    - $b \leftarrow x < t$
    - $idx \leftarrow r \oplus b \cdot (r \oplus l)$
    - $result \leftarrow c$
  - Return $result$

Detect Additive Attacks

- Oblivious selection up to additive errors

$$\langle \boldsymbol{T}[idx] + \boldsymbol{e} \rangle \leftarrow F_{\mathrm{OS}}(\langle \boldsymbol{T} \rangle, \langle idx \rangle)$$

- Solution: SPDZ-like MAC
  - Given a shared value x, a SPDZ-like MAC is defined as $m(x) = \alpha \cdot x$ over a finite filed **F**
  - $m(x)$ is shared along with x

- DTE(T, X):
  - $idx \leftarrow 0, result \leftarrow \perp$
  - For $i \in [1, d']$: $//d' \geq d$
    - $(t, l, r, v, c) \leftarrow T[idx]$
    - $x \leftarrow X[v]$
    - $b \leftarrow x < t$
    - $idx \leftarrow r \oplus b \cdot (r \oplus l)$
    - $result \leftarrow c$
  - Return $result$

Detect Additive Attacks

- Oblivious selection up to additive errors

$$\langle \boldsymbol{T}[idx] + \boldsymbol{e} \rangle \leftarrow F_{\mathrm{OS}}(\langle \boldsymbol{T} \rangle, \langle idx \rangle)$$

- Solution: SPDZ-like MAC
  - Given a shared value x, a SPDZ-like MAC is defined as $m(x) = \alpha \cdot x$ over a finite filed **F**
  - $m(x)$ is shared along with x
- Our MAC scheme:
  - $m(x) = \alpha \cdot x$ over $\mathbf{F}_{2^l}$; l is the bit-length of elements in $\boldsymbol{T}$.

$$\langle \boldsymbol{T}[idx] + \boldsymbol{e} \rangle \leftarrow F_{\mathrm{OS}}(\langle \boldsymbol{T} \rangle, \langle idx \rangle)$$
$$\langle m(\boldsymbol{T}[idx]) + \boldsymbol{e'} \rangle \leftarrow F_{\mathrm{OS}}(\langle m(\boldsymbol{T}) \rangle, \langle idx \rangle)$$

- DTE(T, X):
  - $idx \leftarrow 0, result \leftarrow \perp$
  - For $i \in [1, M]$: $//M \geq d$
    - $(t, l, r, v, c) \leftarrow T[idx]$
    - $x \leftarrow X[v]$
    - $b \leftarrow x < t$
    - $idx \leftarrow r \oplus b \cdot (r \oplus l)$
    - $result \leftarrow c$
  - Return $result$

Malicious PDTE with Sublinear Communication

- Sublinear (online) communication
  - Tree encoding with optimizations.
  - Two OS protocols with constant online communication.

- Malicious security
  - Three-party honest-majority setting.
  - Lightweight consistency checks.
  - *Privacy*: Only the FO learns the classification result.
  - *Correctness*: Mostree always outputs the correct classification.

Experiment Setting

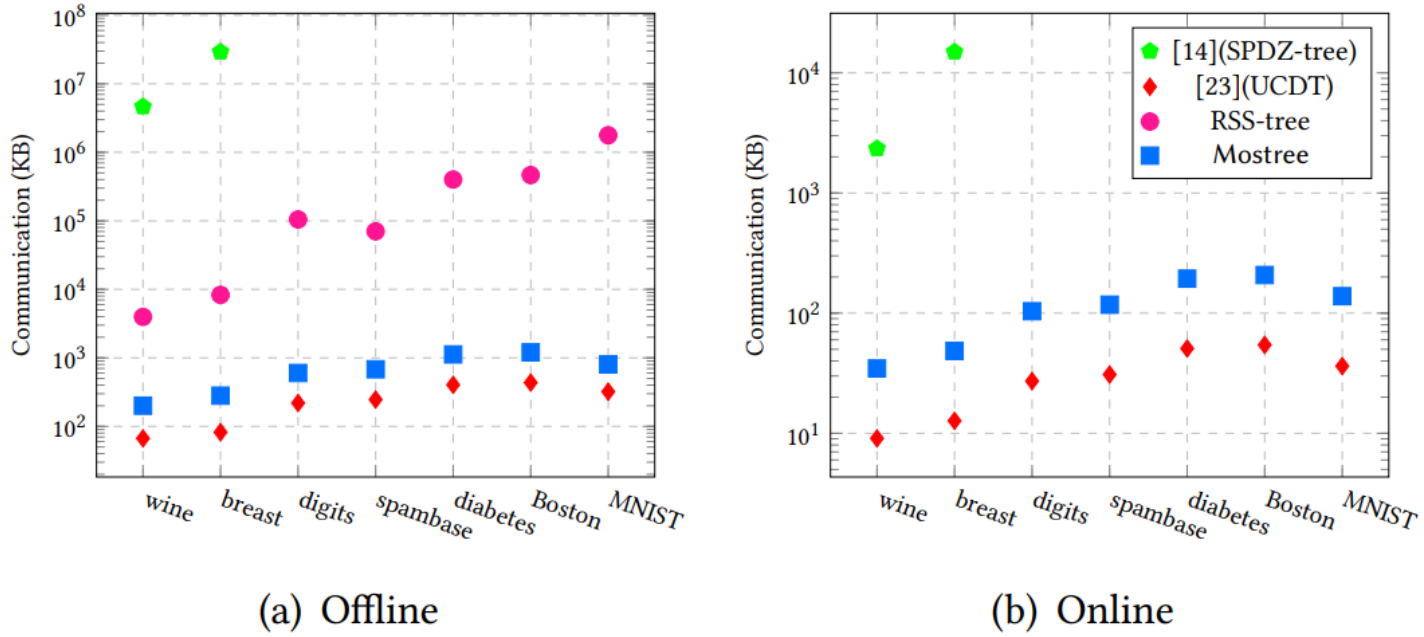
- Machine: Intel(R) Core i9-9900 CPU, 3.10 GHz×16, Ubuntu 20.04 LTS with 32 GB of memory
- Network setting:
  - LAN (RTT: 0.1ms, bandwidth: 1 Gbps)
  - MAN (RTT: 6ms, bandwidth: 100 Mbps)
  - WAN (RTT: 80ms, bandwidth: 40 Mbps)
- Dataset: UCI repository

Tree	Feature Dimension	Depth	# of Nodes
Wine	7	5	23
Linnerud	3	6	39
Breast	12	7	43
Digits	47	15	337
Spambase	57	17	171
Diabetes	10	28	787
Boston	13	30	851
MNIST	784	20	4179

https://github.com/Jbai795/Mostree-pub.git

Performance

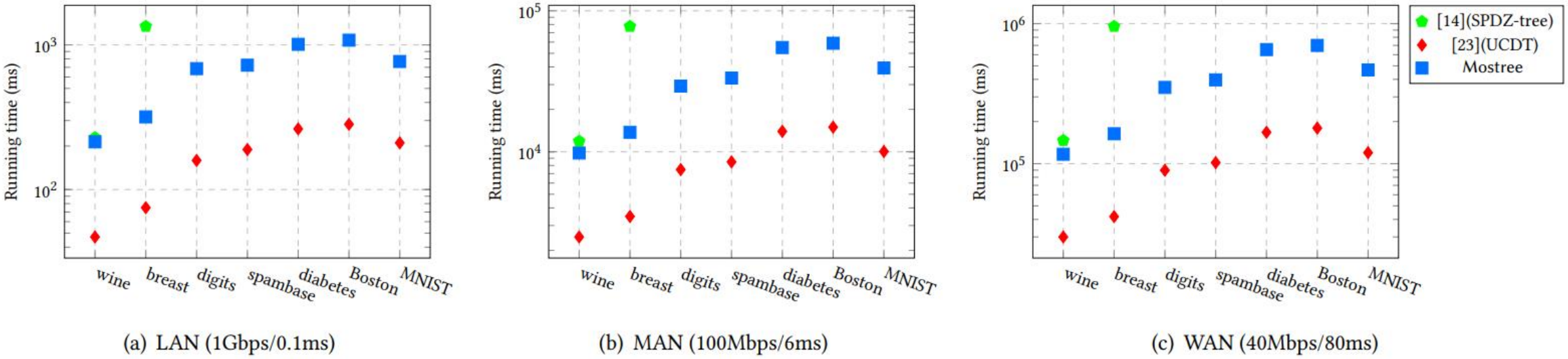
- Online communication



- Mostree requires ~4x online communication than UCDT over MNIST.
- RSS-Tree requires much more offline communication than Mostree.

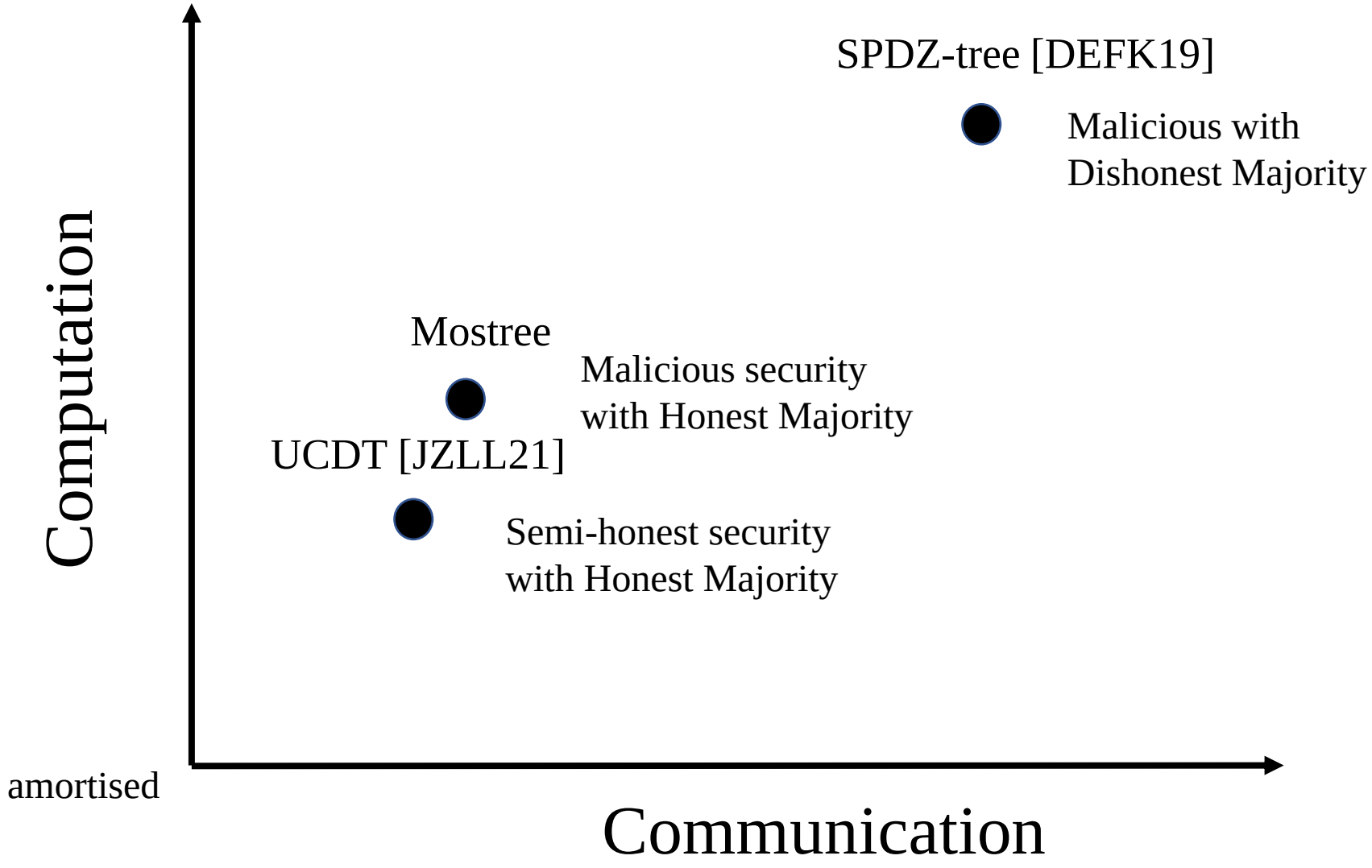
Performance

- Online running time



- ~4x overhead than the semi-honest version.
- In the LAN setting, Mostree requires ~0.7s running time over MNIST. ☺
- In the WAN setting, Mostree requires ~467s over MNIST. ☹

Comparison of Approaches



Conclusion

- We present a cryptographic solution to private decision tree evaluation problem.
- At the core of our scheme are oblivious selection protocols with sublinear (online) communication.
- Lightweight consistency check to ensure malicious security.

- Future work
  - Improving efficiency in the WAN setting → reducing round complexity.
  - Sublinear computation?
  - Applications to other areas, e.g., private database applications.

Thanks for your attention!