Differentially Private Online Task Assignment in Spatial Crowdsourcing: A Tree-based Approach

Qian Tao [1], Yongxin Tong [1], Zimu Zhou[2],
Yexuan Shi [1], Lei Chen[3], Ke Xu[1]

Outline

- Background and Motivation
- Problem Definition
- A Tree-based Framework
- Random Walk Acceleration
- Experimental Evaluation
- Conclusions

Background

- Spatial Crowdsourcing has penetrated in our life

Taxi Calling

Food Delivery

Logistics

Background

- Spatial Crowdsourcing has penetrated in our life
- Privacy leakage draws attraction in recent years

Taxi Calling	Food Delivery	Logistics
Uber	DOORDASH	UPS

Security Center > Emerging Threats > Uber announces new data breach affecting 57 million riders and drivers

Uber announces new data breach affecting 57 million riders and drivers

DoorDash confirms data breach affected 4.9 million customers, workers and merchants

Zack Whittaker @zackwhittaker / 4:21 am C

Image Credits: DoorDash / file photo

UPS Reveals Data Breach

POS Malware Compromises 105,000 Transactions at 51 Stores

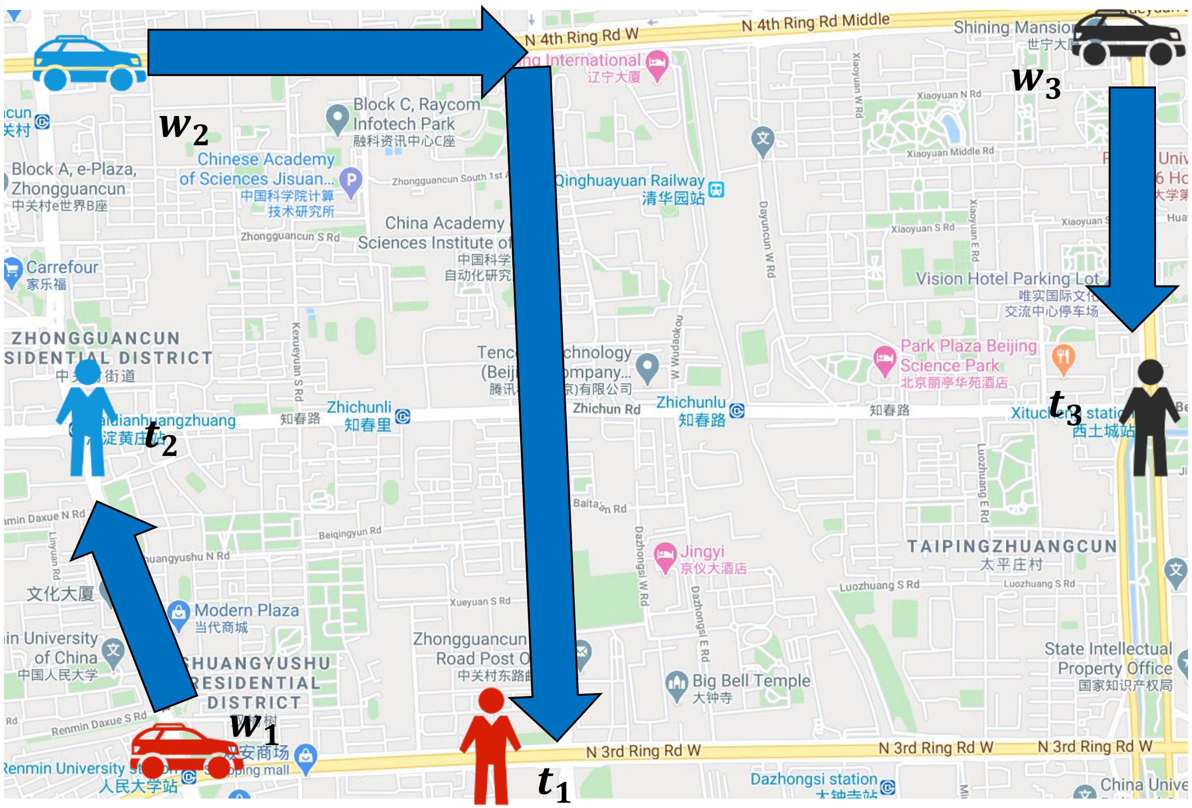
Mathew J. Schwartz (euroinfosec) • August 21, 2014

Background

- A core operation of spatial crowdsourcing is task assignment.

Type	Applications	Issue
Taxi Calling	Uber	Assign taxi-calling orders to drivers
Food Delivery	GRUBHUB	Assign food orders to proper deliverers
Logistics	ups	Assign delivery tasks to proper workers

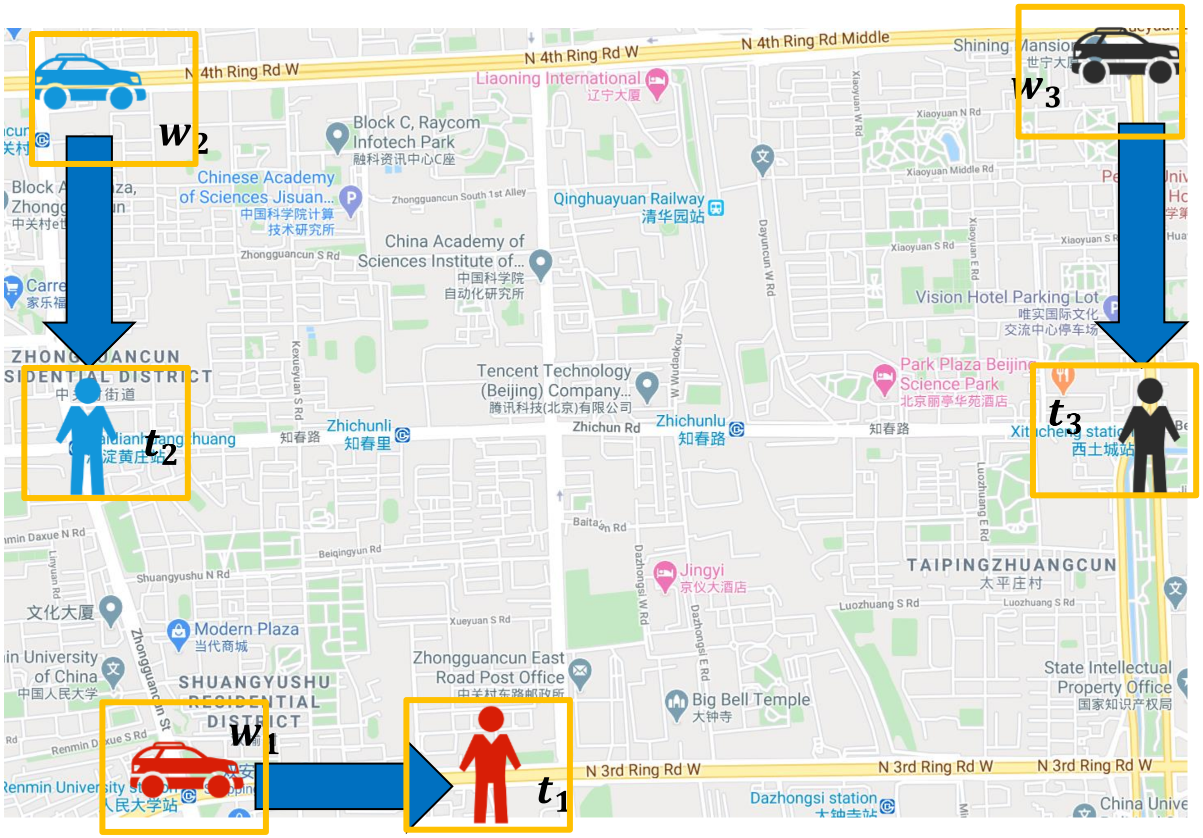
Background

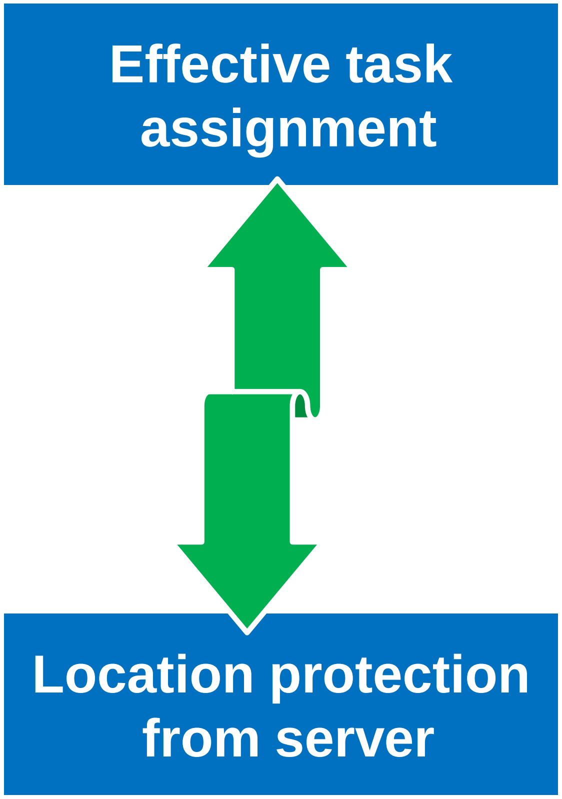


w_2
w_3
w_1
t_1
t_2
t_3
N 4th Ring Rd W
N 4th Ring Rd Middle
Shining Mansion
Block C, Raycom
Infotech Park
融科资讯中心C座
Block A, e-Plaza,
Zhongguancun
中关村e世界B座
Chinese Academy
of Sciences Jisuan...
中国科学院计算
技术研究所
Qinghuayuan Railway
清华园站
China Academy
Sciences Institute of
中国科学
自动化研究
Carrefour
家乐福
ZHONGGUANCUN
Vision Hotel Parking Lot
Park Plaza Beijing
Science Park
北京丽亭华苑酒店
Zhichunli
知春里
Zhichun Rd
Zhichunlu
知春路
Xitucheng
西土城站
TAIPINGZHUANGCUN
太平庄村
Jingyi
京仪大酒店
文化大厦
Modern Plaza
当代商城
of China
中国人民大学
Zhongguancun
Big Bell Temple
大钟寺
State Intellectual
Property Office
国家知识产权局
N 3rd Ring Rd W
Dazhongsi station
Renmin University
人民大学站

Background

- How to make effective task assignment while protecting the location privacy of the tasks and workers?





Limitations of Existing Works

- Ignore a widely-researched and practical objective: minimizing total distance
- Lack of theoretical analysis of the effectiveness of the task assignment

H. To et al, Privacy-preserving online task assignment in spatial crowdsourcing with untrusted server. In ICDE 2018.

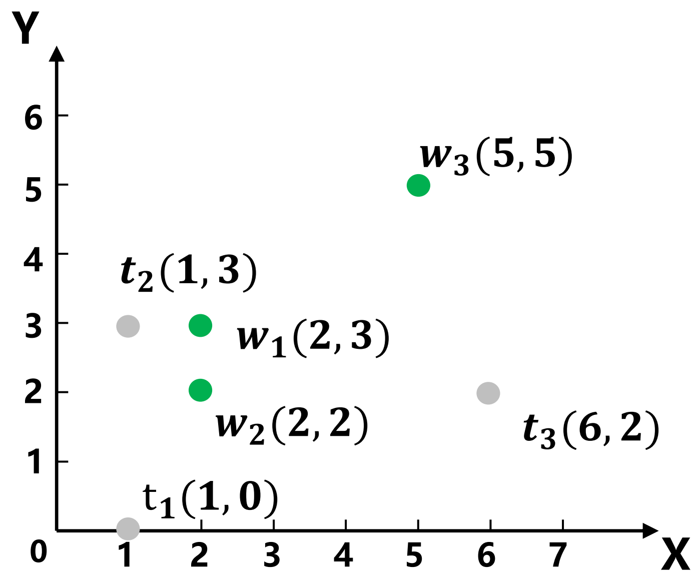
L. Wang et al, Location privacy-preserving task allocation for mobile crowdsensing with differential geo-obfuscation. In WWW 2017.

Outline

- Background and Motivation
- Problem Definition
- A Tree-based Framework
- Random Walk Acceleration
- Experimental Evaluation
- Conclusions

Problem Definition

- Crowd worker w
  - (x_w, y_w): location of the worker w
- Spatial task (Dynamically appears)
  - (x_t, y_t) : location of the task t



Problem Definition

The Privacy-preserving Online Minimum Bipartite Matching Problem is as follows.

POMBM Problem

Given a set of workers W and a set of dynamically appearing tasks T, we aims to design a privacy mechanism $\mathcal{M}$ such that

- **The mechanism guarantees the Indistinguishability of the locations**
- **The mechanism enables matching algorithms with minimum total distance**

Make effective task assignment

Location Protection from server

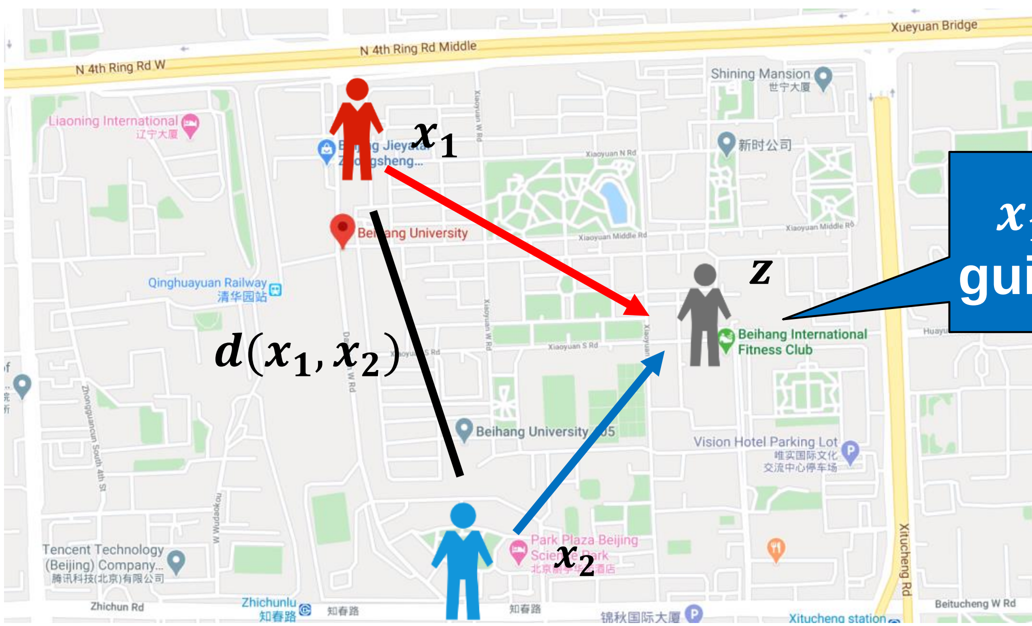
Problem Definition: Indistinguishability

We require a mechanism that satisfies Geo-Indistinguishability.

Geo-Indistinguishability

A mechanism is Geo-Indistinguishable on metric space $\mathcal{X}$ if for any $x_1, x_2 \in \mathcal{X}$ and $z \in \mathcal{Z}$, where $\mathcal{Z}$ is the projection space,

$$\mathcal{M}(x_1)(z) \leq e^{\epsilon d_{\mathcal{X}}(x_1,x_2)} \mathcal{M}(x_2)(z).$$



x_1 and x_2 cannot be distin-guished with high probability

$\mathcal{M}(x)(z)$: Probability of x projected to point z

Problem Definition

The Privacy-preserving Online Minimum Bipartite Matching Problem is as follows.

POMBM Problem

Given a set of workers W and a set of dynamically appearing tasks T, we aims to design a privacy mechanism $\mathcal{M}$ such that

- **The mechanism guarantees the Indistinguishability of the locations**
- **The mechanism enables matching algorithms with minimum total distance**

$$\min \sum_{(w,t)\in M} dis(w,t)$$

Make effective task assignment ⟷ **Location protection from server**

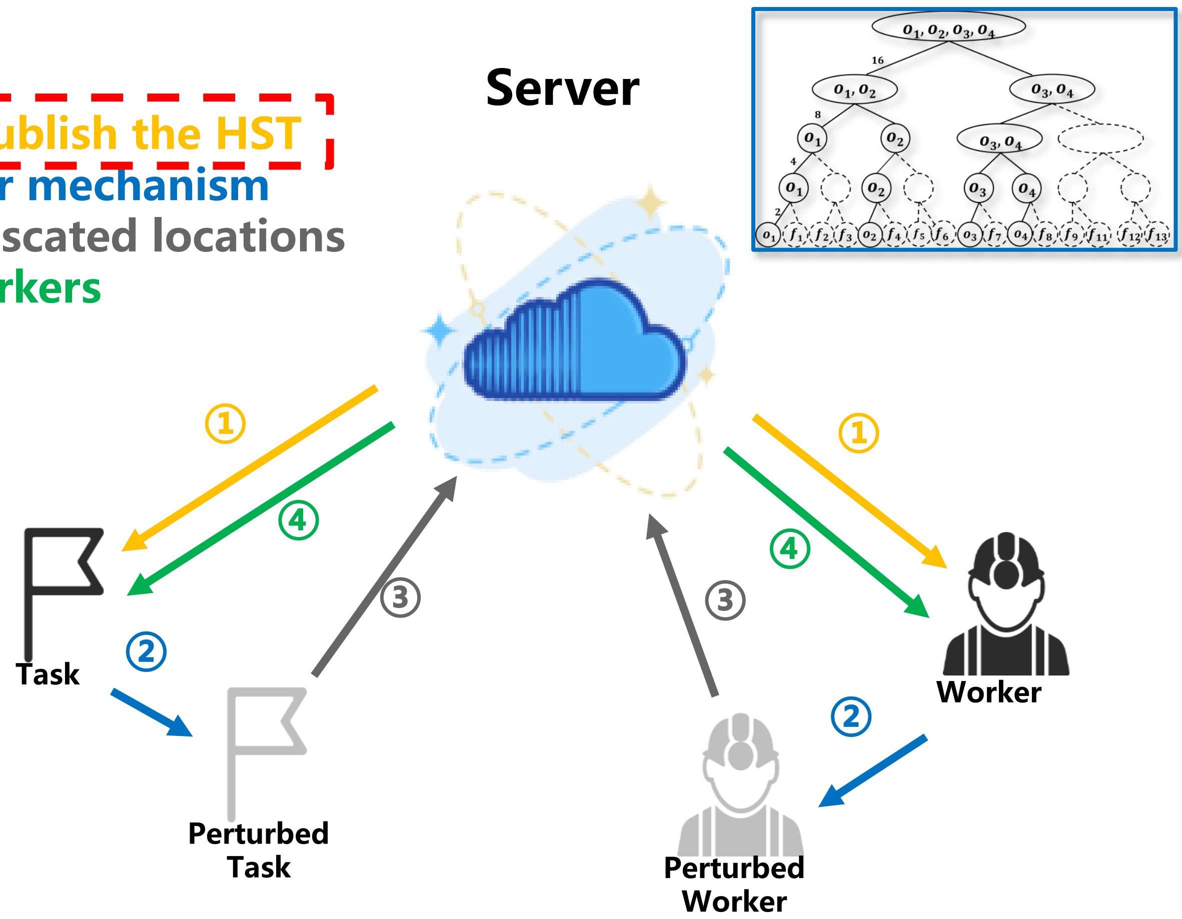
Outline

- Background and Motivation
- Problem Definition
- A Tree-based Framework
- Random Walk Acceleration
- Experimental Evaluation
- Conclusions

Tree-based Framework

- Our solution is devised based on a tree-based framework.

①Construct and publish the HST
②Add noise by our mechanism
③Publish the obfuscated locations
④Assign tasks/workers

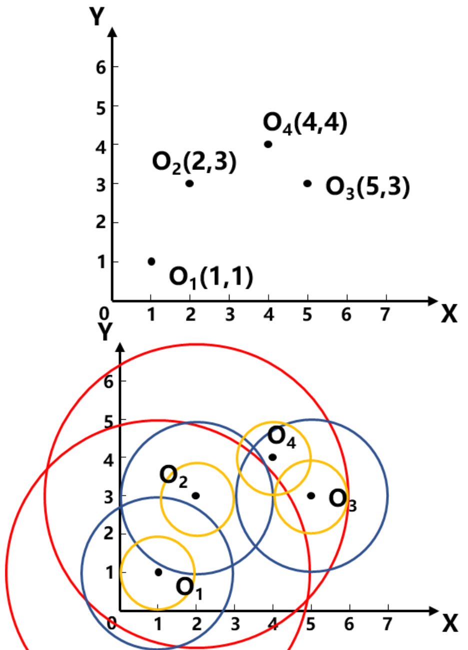


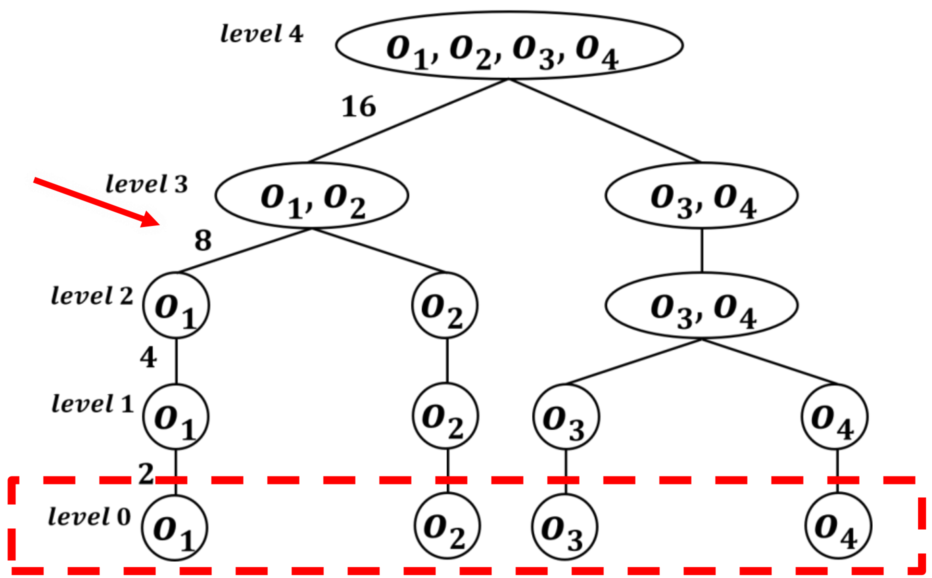
HST Construction

Hierarchical Well-Separated Tree (HST)

The Well-Separated Tree is a tree space $\mathcal{T} = (V_T, d_T)$ embedded from an arbitrary space (V, d) such that

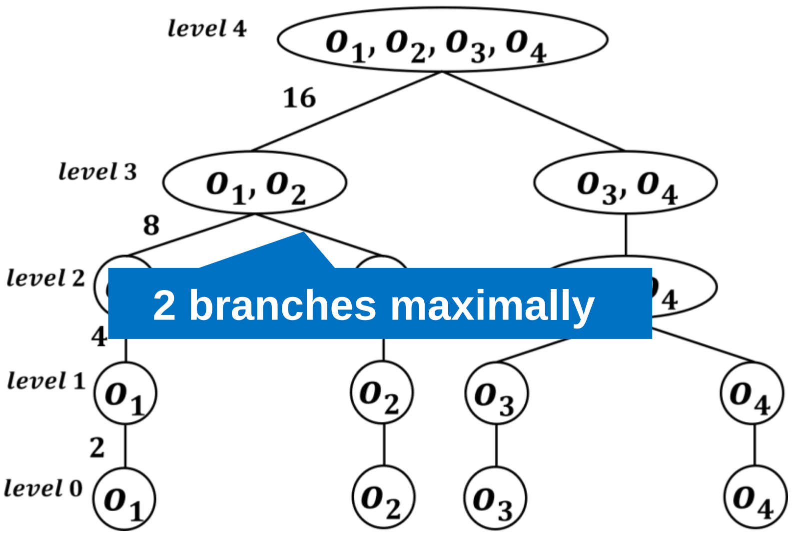
- **Each leaf node corresponds to a point in V**
- **The distance on the tree from a node at level i to its parent is 2^{i+1}**



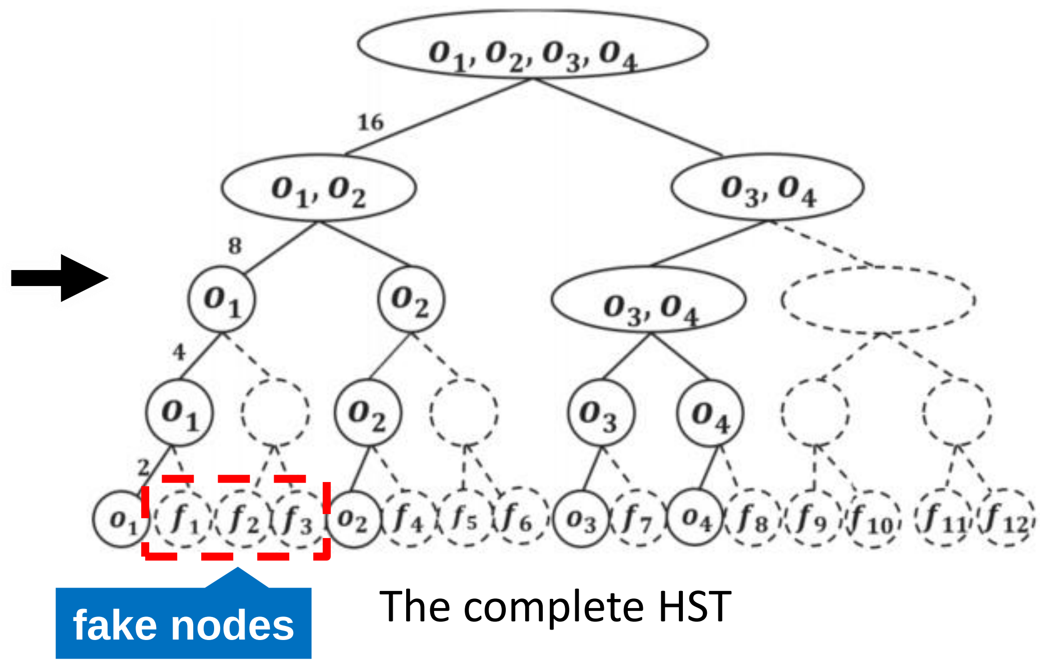


HST Construction

- Augment the HST to a complete one by adding fake nodes.



The original HST

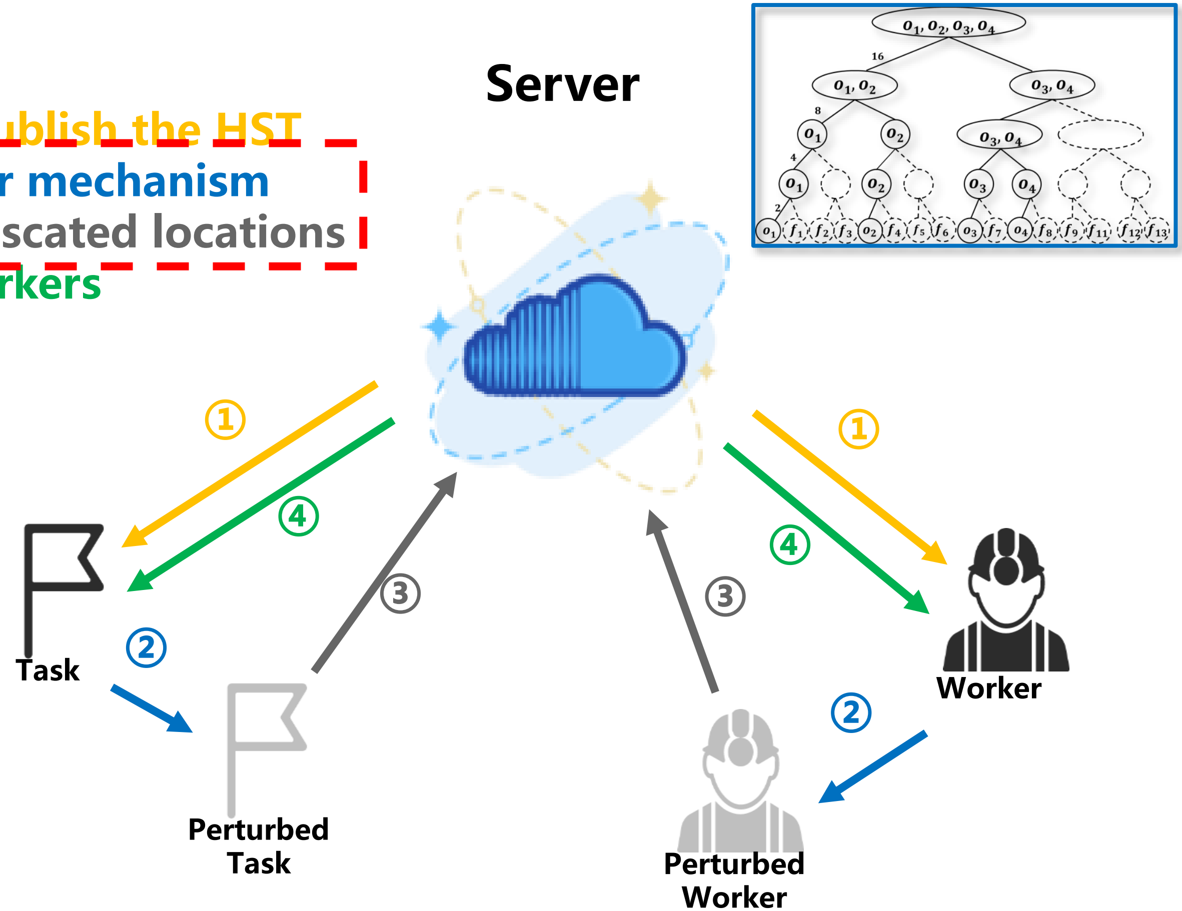


The complete HST

Tree-based Framework

- Our solution is devised based on a tree-based framework.

①Construct and publish the HST
②Add noise by our mechanism
③Publish the obfuscated locations
④Assign tasks/workers



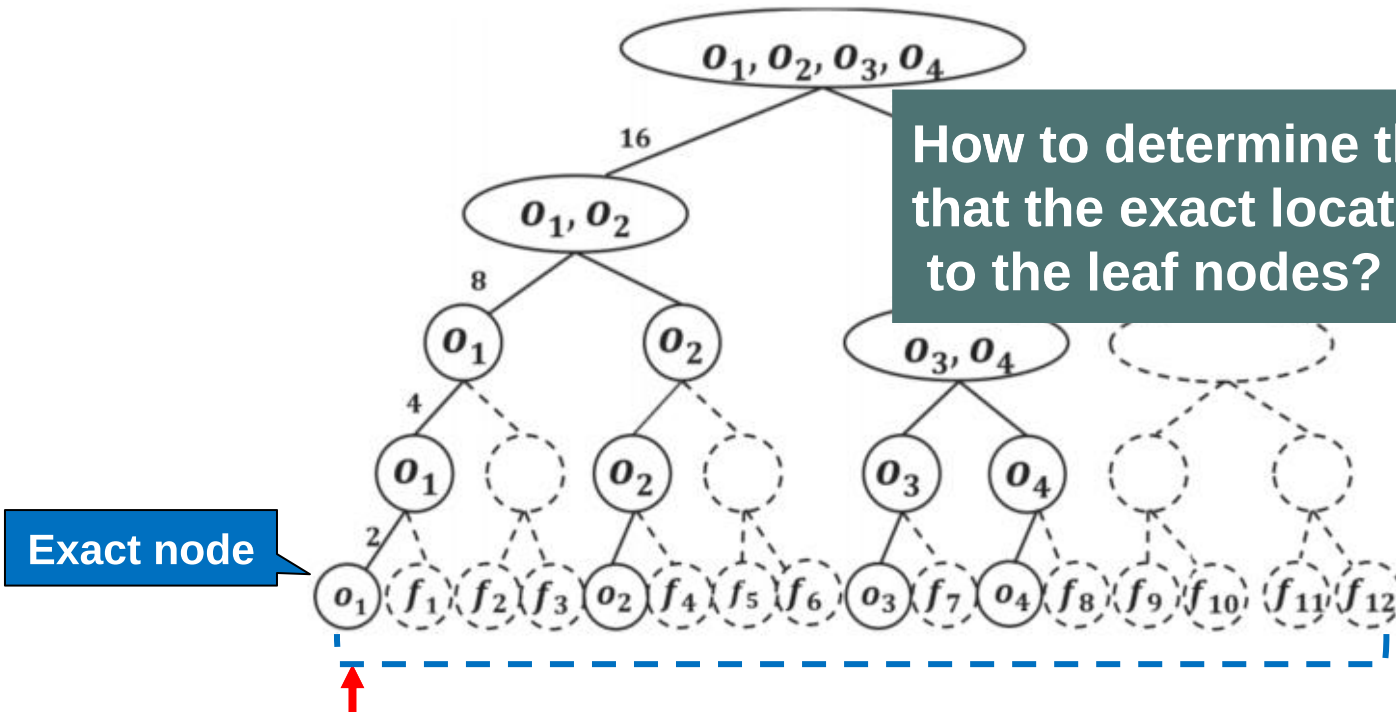
Privacy Mechanism Design

- Main Idea: Project the exact location to one of the **leaf nodes** such that Geo-I is satisfied.

Geo-Indistinguishability

A mechanism is Geo-Indistinguishability on metric space $\mathcal{X}$ if for any $x_1, x_2 \in \mathcal{X}$ and $z \in \mathcal{Z}$,

$$\mathcal{M}(x_1)(z) \leq e^{\epsilon d_{\mathcal{X}}(x_1,x_2)} \mathcal{M}(x_2)(z).$$

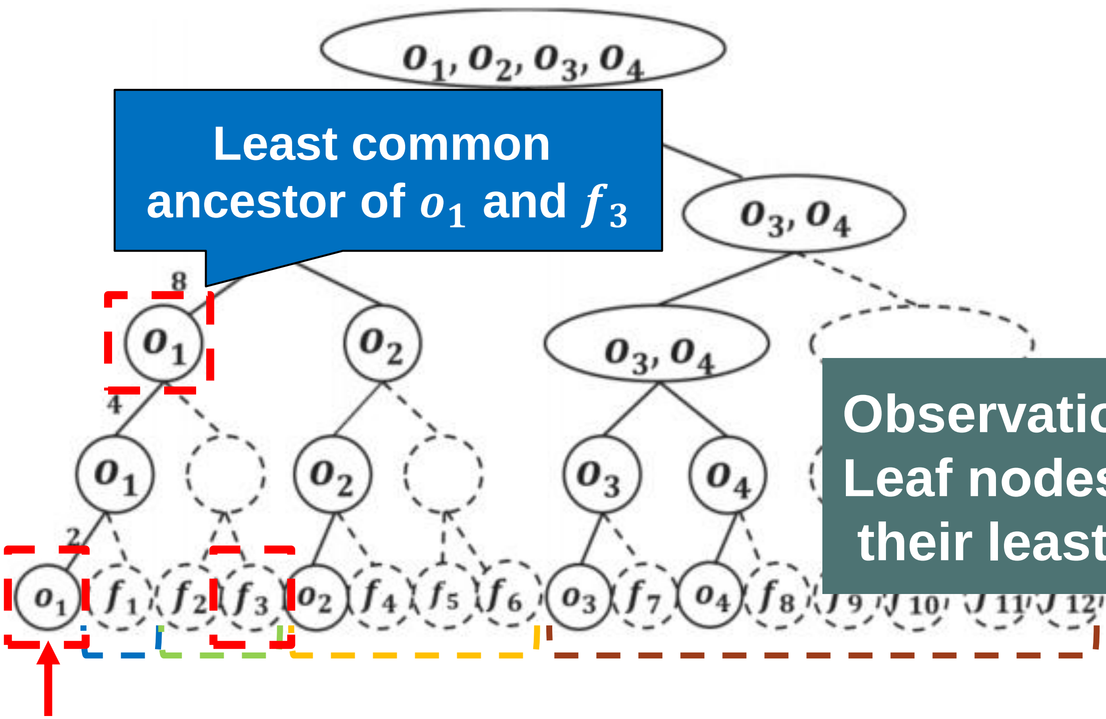


How to determine the probability that the exact location is projected to the leaf nodes?

Privacy Mechanism Design

- Assign different projection weights to leaf nodes based on the distance to the exact node.

nodes	distance
o_1	0
f_1	4
$f_2 - f_3$	12
$o_2, f_4 - f_6$	28
o_3-o_4, f_7-f_{12}	60



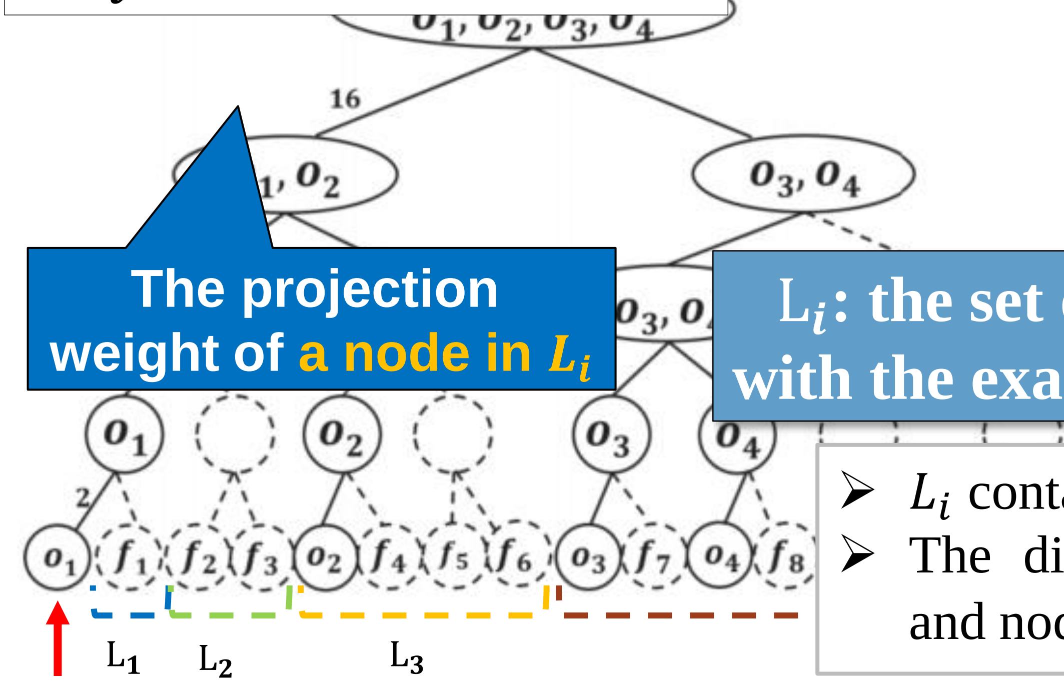
Observation:
Leaf nodes' distance to o_1 depends on their least common ancestor with o_1

Privacy Mechanism Design

- Key Point: Assign different projection weights to leaf nodes based on the distance to the exact node.

nodes	distance
$L_1: o_1$	0
$L_2: f_1$	4
$L_3: f_2 - f_3$	12
$L_4: o_2, f_4 - f_6$	28
$L_5: o_3\text{-}o_4, f_7\text{-}f_{12}$	60

$$wt_i = e^{-d_i\epsilon} = e^{(4-2^{i+2})\epsilon}$$



The projection weight of a node in L_i

L_i: the set of leaf nodes whose LCA with the exact node is located at level i

- L_i contains $c^{i-1}(c-1)$ nodes exactly
- The distance between the exact node and nodes in L_i is $d_i = 2^{i+2} - 4$

Privacy Mechanism Design

- Key Point: Assign different projection weights to leaf nodes based on the distance to the exact node.

$$Pr_i = \frac{wt_i}{WT}$$

The probability of a node in L_i being projected to

$$wt_i = e^{-d_i\epsilon} = e^{(4-2^{i+2})\epsilon}$$

$$WT = 1 + \sum_{i=1}^{D} c^{i-1}(c-1)wt_i$$

Total weight of all leaf nodes

The projection weight of a node in L_i

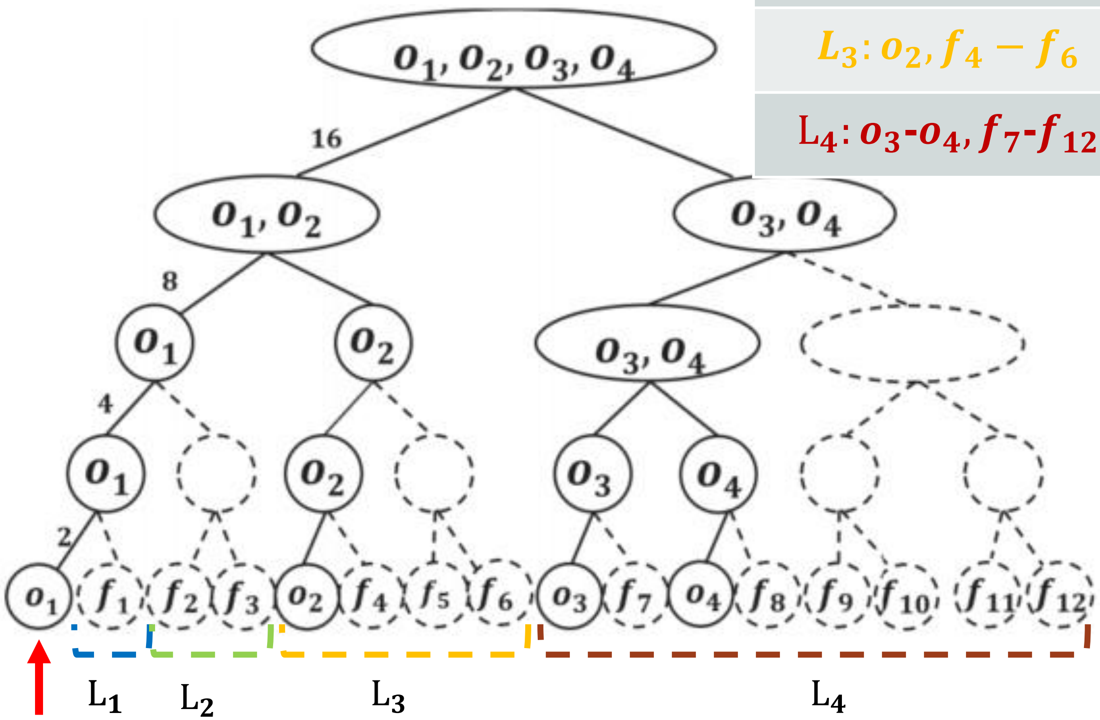
L_i: the set of leaf nodes whose LCA with the exact node is located at level i

Privacy Mechanism Design

- An Example

$\epsilon = 1$

nodes	distance	weights	Prob
$L_0: o_1$	0		
$L_1: f_1$	4		
$L_2: f_2 - f_3$	12		
$L_3: o_2, f_4 - f_6$	28		
$L_4: o_3\text{-}o_4, f_7\text{-}f_{12}$	60		



$$wt_i = e^{(4-2^{i+2})\epsilon}$$

$$Pr_i = \frac{wt_i}{WT}$$

Privacy Mechanism Design

- An Example

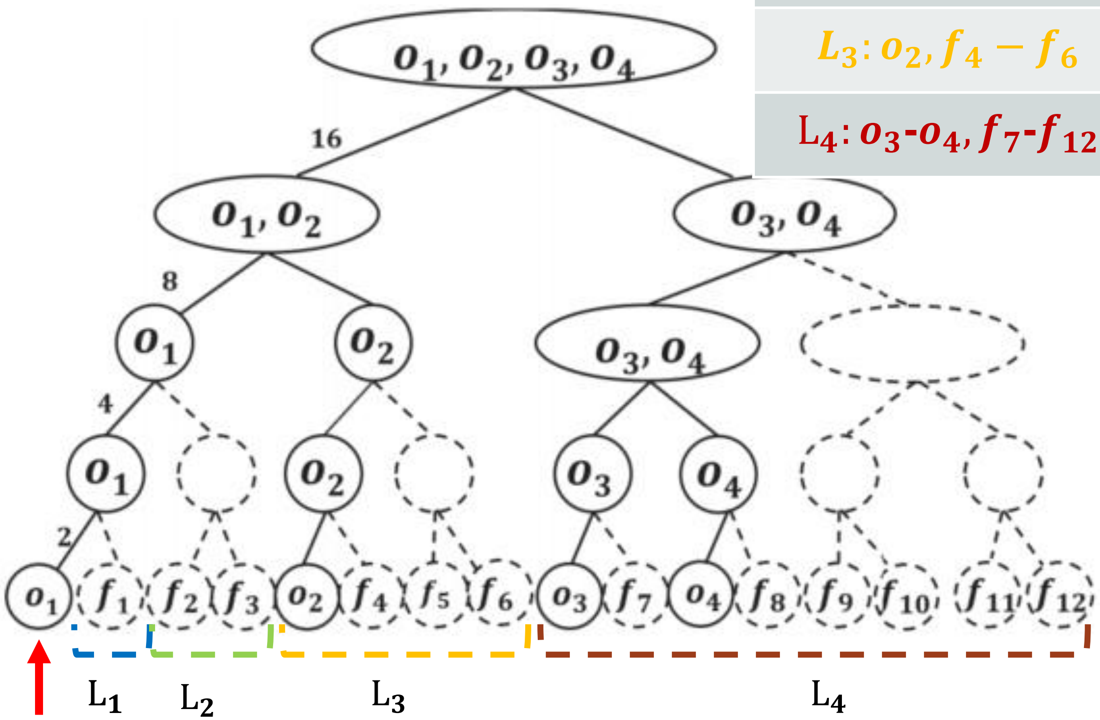
$\epsilon = 1$

e^0

nodes	distance	weights	Prob
$L_0: o_1$	0	1	
$L_1: f_1$	4		
$L_2: f_2 - f_3$	12		
$L_3: o_2, f_4 - f_6$	28		
$L_4: o_3\text{-}o_4, f_7\text{-}f_{12}$	60		

$$wt_i = e^{(4-2^{i+2})\epsilon}$$

$$Pr_i = \frac{wt_i}{WT}$$



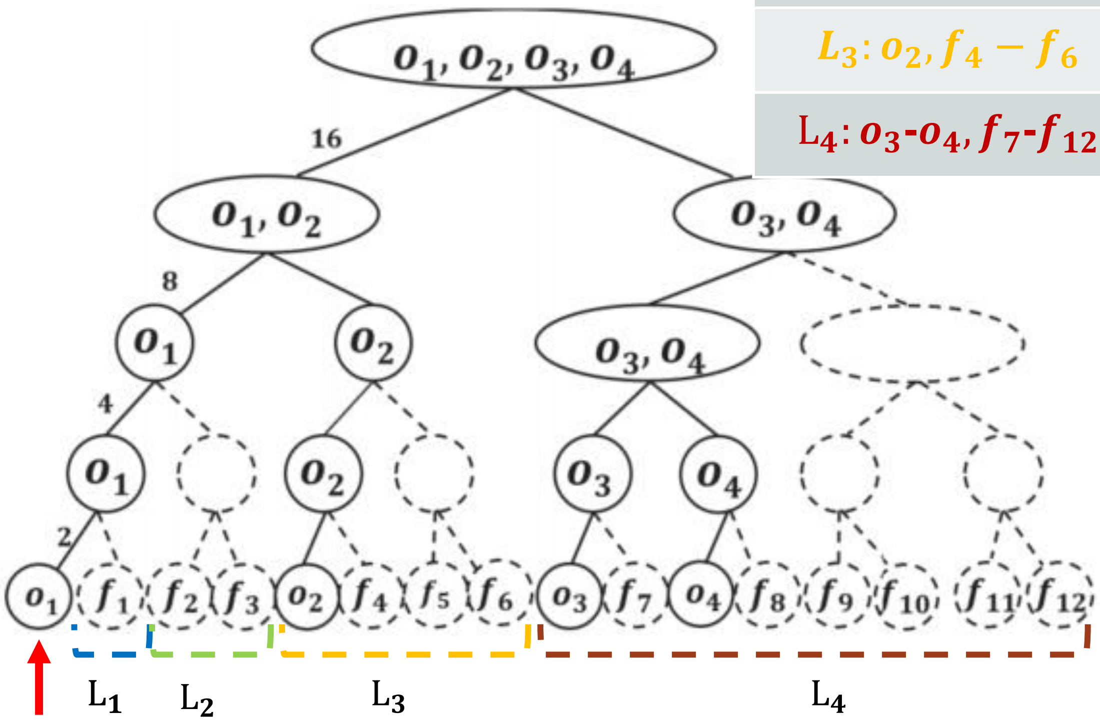
Privacy Mechanism Design

- An Example

$\epsilon = 1$

nodes	distance	weights	Prob
$L_0: o_1$		1	
$L_1: f_1$	4	0.670	
$L_2: f_2 - f_3$	12		
$L_3: o_2, f_4 - f_6$	28		
$L_4: o_3\text{-}o_4, f_7\text{-}f_{12}$	60		

e^{-4}



$$wt_i = e^{(4-2^{i+2})\epsilon}$$

$$Pr_i = \frac{wt_i}{WT}$$

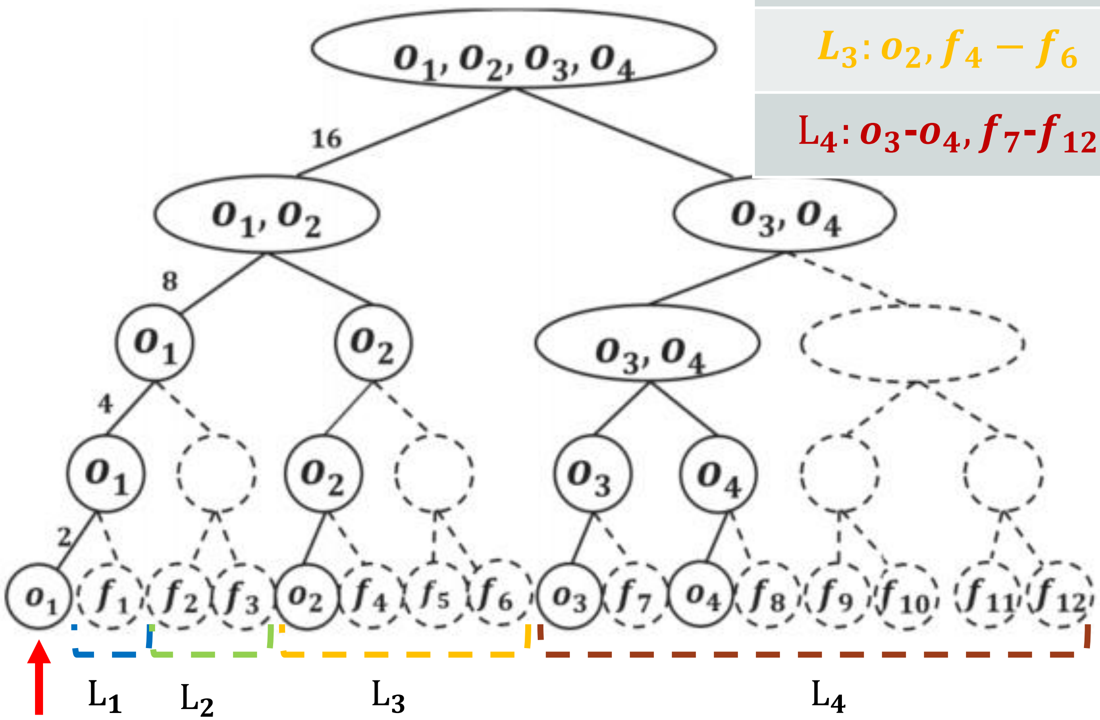
Privacy Mechanism Design

- An Example

$\epsilon = 1$

nodes	distance	weights	Prob
$L_0: o_1$	0	1	
$L_1: f_1$		0.670	
$L_2: f_2 - f_3$	12	0.301	
$L_3: o_2, f_4 - f_6$	28		
$L_4: o_3\text{-}o_4, f_7\text{-}f_{12}$	60		

e^{-12}



$$wt_i = e^{(4-2^{i+2})\epsilon}$$

$$Pr_i = \frac{wt_i}{WT}$$

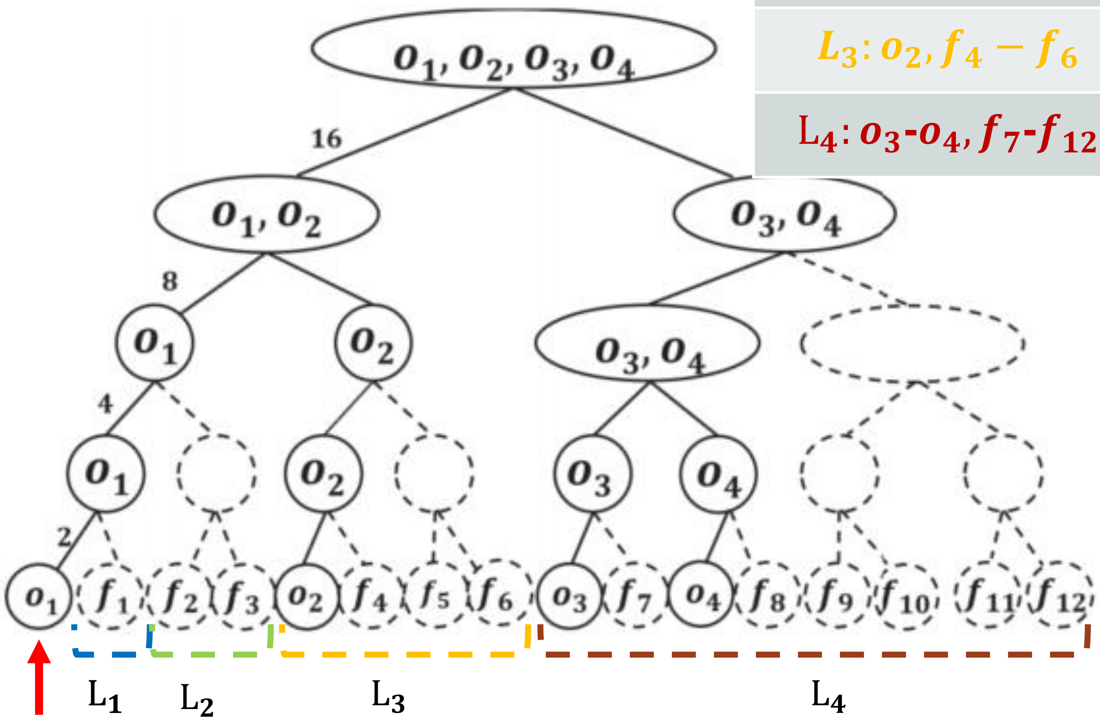
Privacy Mechanism Design

- An Example

$\epsilon = 1$

nodes	distance	weights	Prob
$L_0: o_1$	0	1	
$L_1: f_1$	4	0.670	
$L_2: f_2 - f_3$		0.301	
$L_3: o_2, f_4 - f_6$	28	0.061	
$L_4: o_3\text{-}o_4, f_7\text{-}f_{12}$	60		

e^{-28}



$$wt_i = e^{(4-2^{i+2})\epsilon}$$

$$Pr_i = \frac{wt_i}{WT}$$

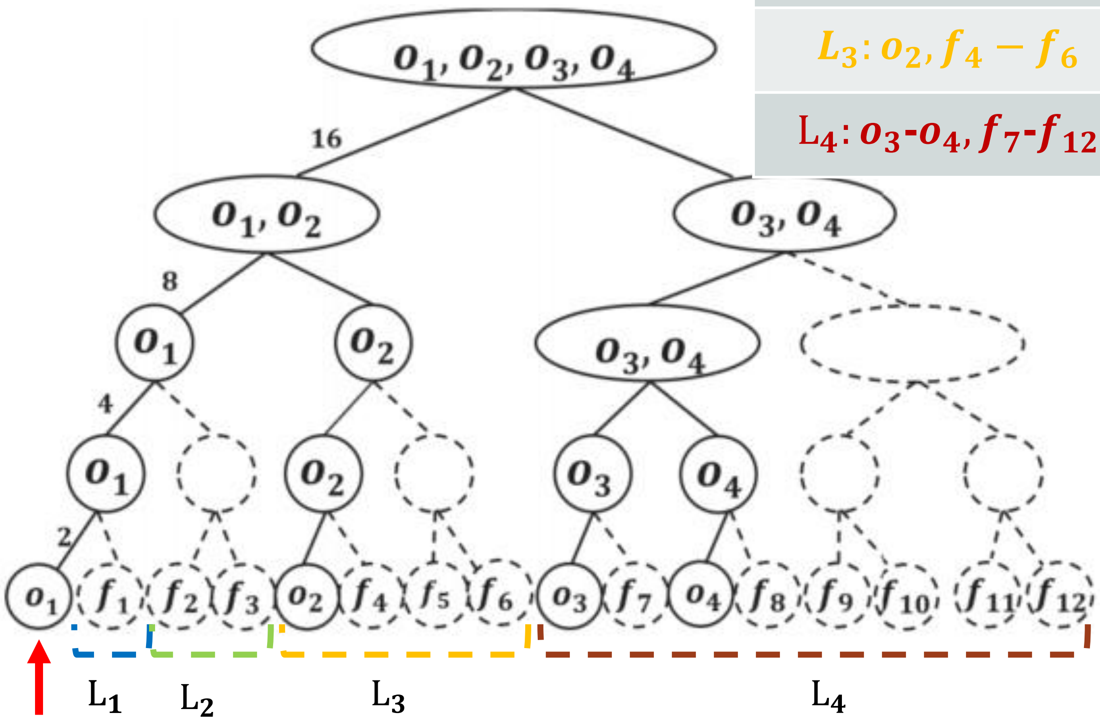
Privacy Mechanism Design

- An Example

$\epsilon = 1$

nodes	distance	weights	Prob
$L_0: o_1$	0	1	
$L_1: f_1$	4	0.670	
$L_2: f_2 - f_3$	12	0.301	
$L_3: o_2, f_4 - f_6$		0.061	
$L_4: o_3\text{-}o_4, f_7\text{-}f_{12}$	60	0.002	

e^{-60}



$$wt_i = e^{(4-2^{i+2})\epsilon}$$

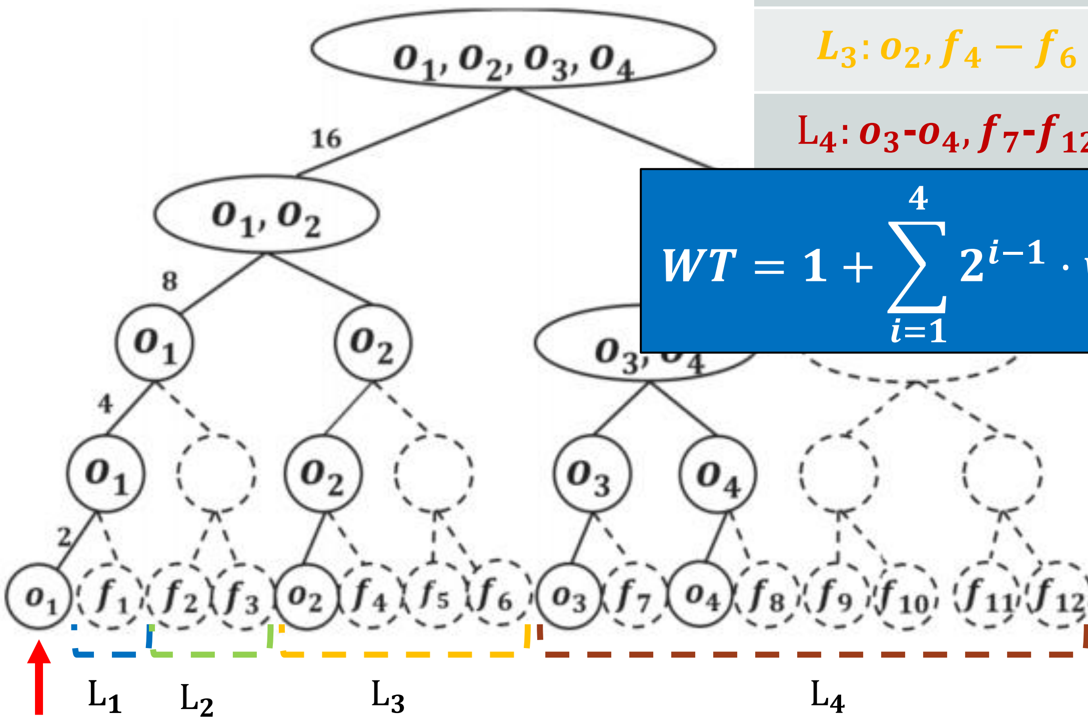
$$Pr_i = \frac{wt_i}{WT}$$

Privacy Mechanism Design

- An Example

$\epsilon = 1$

nodes	distance	weights	Prob
$L_0: o_1$	0	1	0.394
$L_1: f_1$	4	0.670	0.264
$L_2: f_2 - f_3$	12	0.301	0.119
$L_3: o_2, f_4 - f_6$	28	0.061	0.024
$L_4: o_3\text{-}o_4, f_7\text{-}f_{12}$	60	0.002	0.001



$$WT = 1 + \sum_{i=1}^{4} 2^{i-1} \cdot wt_i = 2.532$$

$$wt_i = e^{(4-2^{i+2})\epsilon}$$

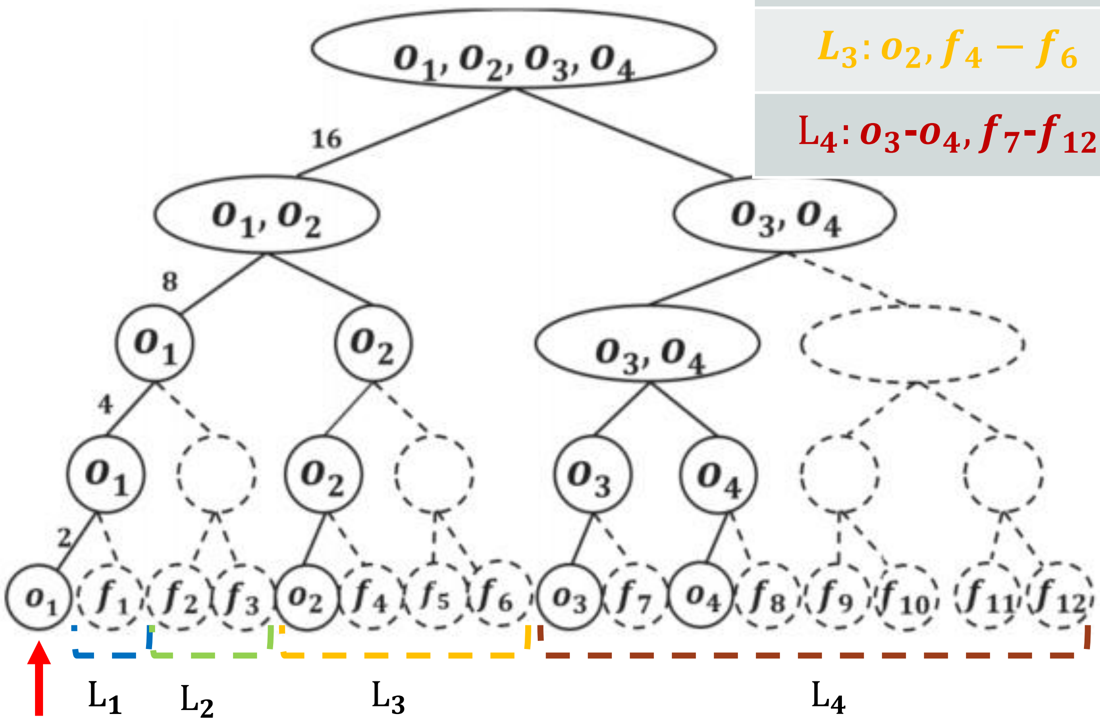
$$Pr_i = \frac{wt_i}{WT}$$

Privacy Mechanism Design

- An Example

$\epsilon = 1$

nodes	distance	weights	Prob
$L_0: o_1$	0	1	0.394
$L_1: f_1$	4	0.670	0.264
$L_2: f_2 - f_3$	12	0.301	0.119
$L_3: o_2, f_4 - f_6$	28	0.061	0.024
$L_4: o_3\text{-}o_4, f_7\text{-}f_{12}$	60	0.002	0.001



$$wt_i = e^{(4-2^{i+2})\epsilon}$$

$$Pr_i = \frac{wt_i}{WT}$$

Privacy Mechanism Design

- Proof of Geo-Indistinguishability

Geo-Indistinguishability

A mechanism is Geo-Indistinguishability on metric space $\mathcal{X}$ if for any $x_1, x_2 \in \mathcal{X}$ and $z \in \mathcal{Z}$,
$$\mathcal{M}(x_1)(z) \leq e^{\epsilon d_{\mathcal{X}}(x_1,x_2)} \mathcal{M}(x_2)(z).$$

$$wt_i = e^{(4-2^{i+2})\epsilon}$$

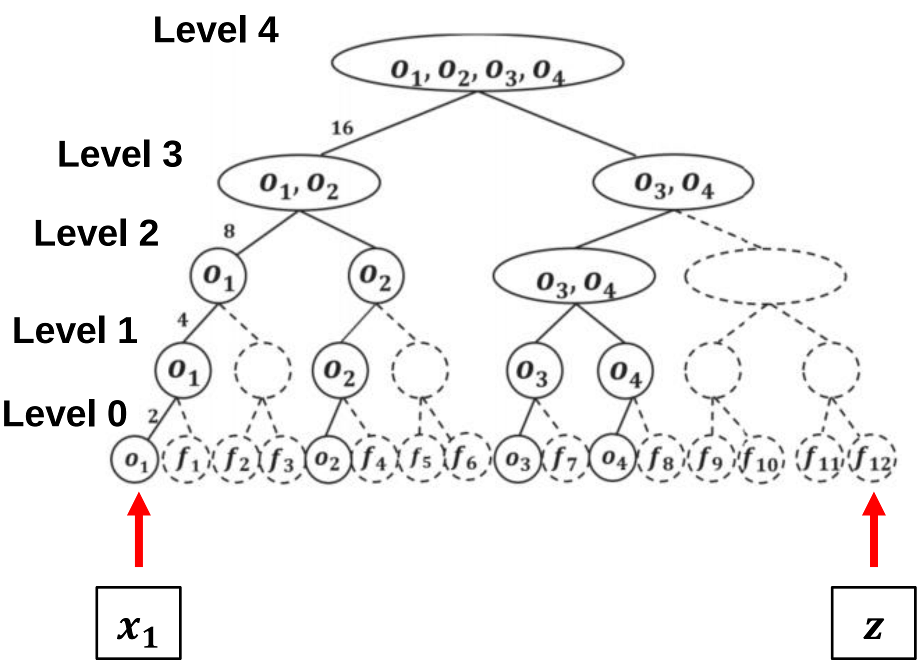
$$Pr_i = \frac{wt_i}{WT}$$

$lvl(a, b)$: level of the least common ancestor of a and b

$$\mathcal{M}(x_1)(z) = Pr_{lvl(x_1,z)}$$

$$\mathcal{M}(x_2)(z) = Pr_{lvl(x_2,z)}$$

$lvl(x_1, z) = 4$



Privacy Mechanism Design

- Proof of Geo-Indistinguishability

Geo-Indistinguishability

A mechanism is Geo-Indistinguishability on metric space $\mathcal{X}$ if for any $x_1, x_2 \in \mathcal{X}$ and $z \in \mathcal{Z}$,

$$\mathcal{M}(x_1)(z) \leq e^{\epsilon d_{\mathcal{X}}(x_1,x_2)} \mathcal{M}(x_2)(z).$$

$$wt_i = e^{(4-2^{i+2})\epsilon}$$

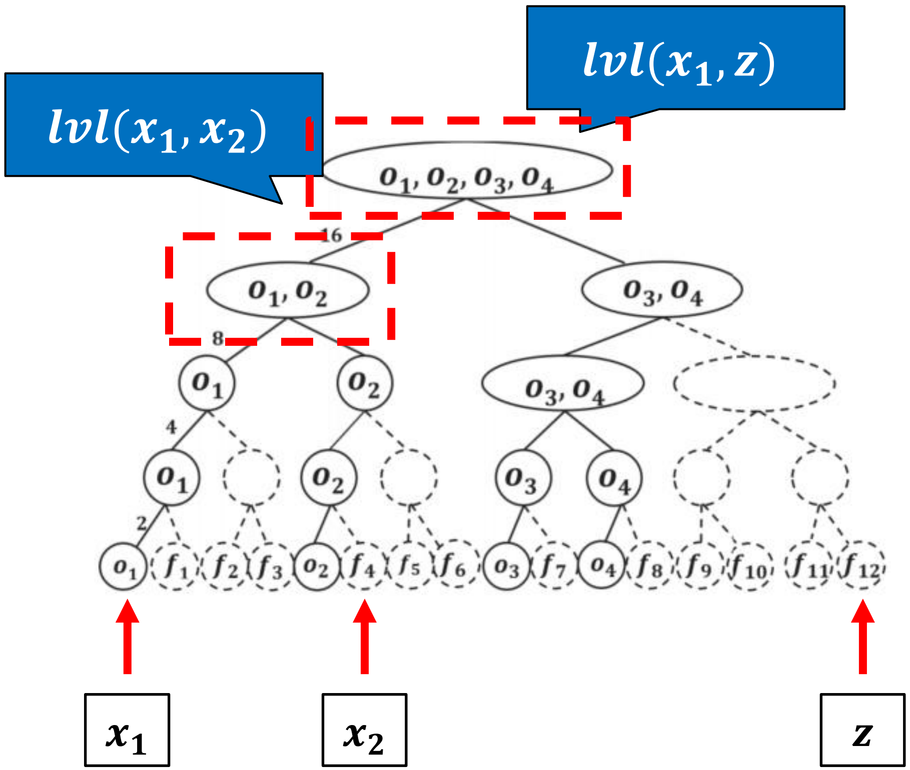
$$Pr_i = \frac{wt_i}{WT}$$

Case 1: $lvl(x_1, z) > lvl(x_1, x_2)$

$\Longrightarrow lvl(x_2, z) = lvl(x_1, z)$

$\Longrightarrow \mathcal{M}(x_1)(z) = \mathcal{M}(x_2)(z)$

$\Longrightarrow$ Case 1 proved



Privacy Mechanism Design

- Proof of Geo-Indistinguishability

Geo-Indistinguishability

A mechanism is Geo-Indistinguishability on metric space $\mathcal{X}$ if for any $x_1, x_2 \in \mathcal{X}$ and $z \in \mathcal{Z}$,

$$\mathcal{M}(x_1)(z) \le e^{\epsilon d_{\mathcal{X}}(x_1,x_2)} \mathcal{M}(x_2)(z).$$

$$wt_i = e^{(4-2^{i+2})\epsilon}$$

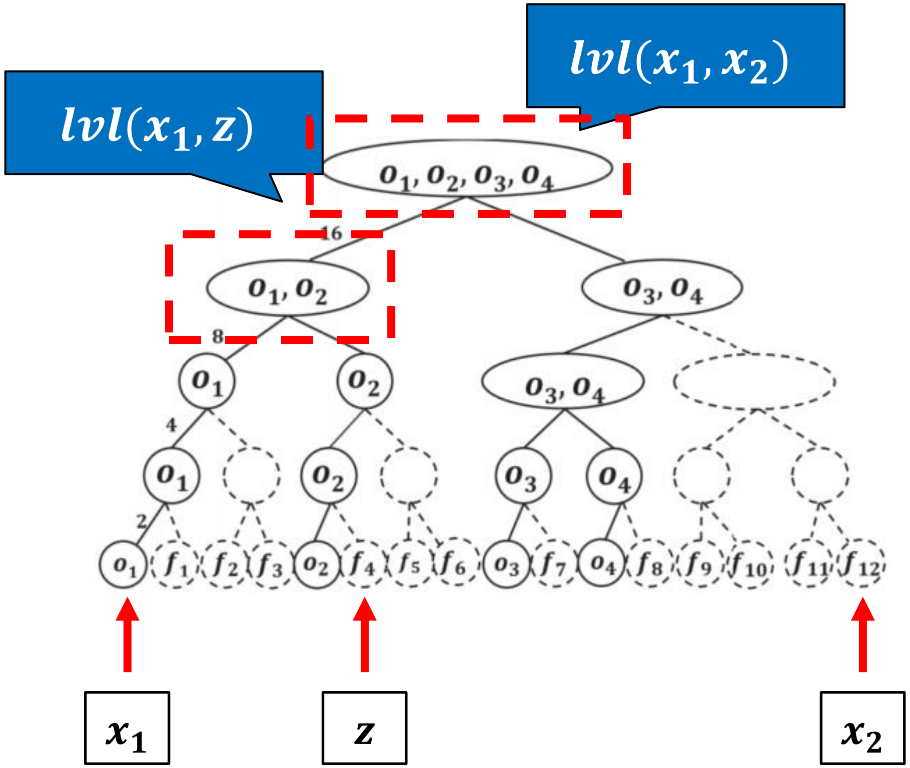
$$Pr_i = \frac{wt_i}{WT}$$

Case 2: $lvl(x_1, z) \le lvl(x_1, x_2)$

$\Longrightarrow lvl(x_2, z) \le lvl(x_1, x_2)$

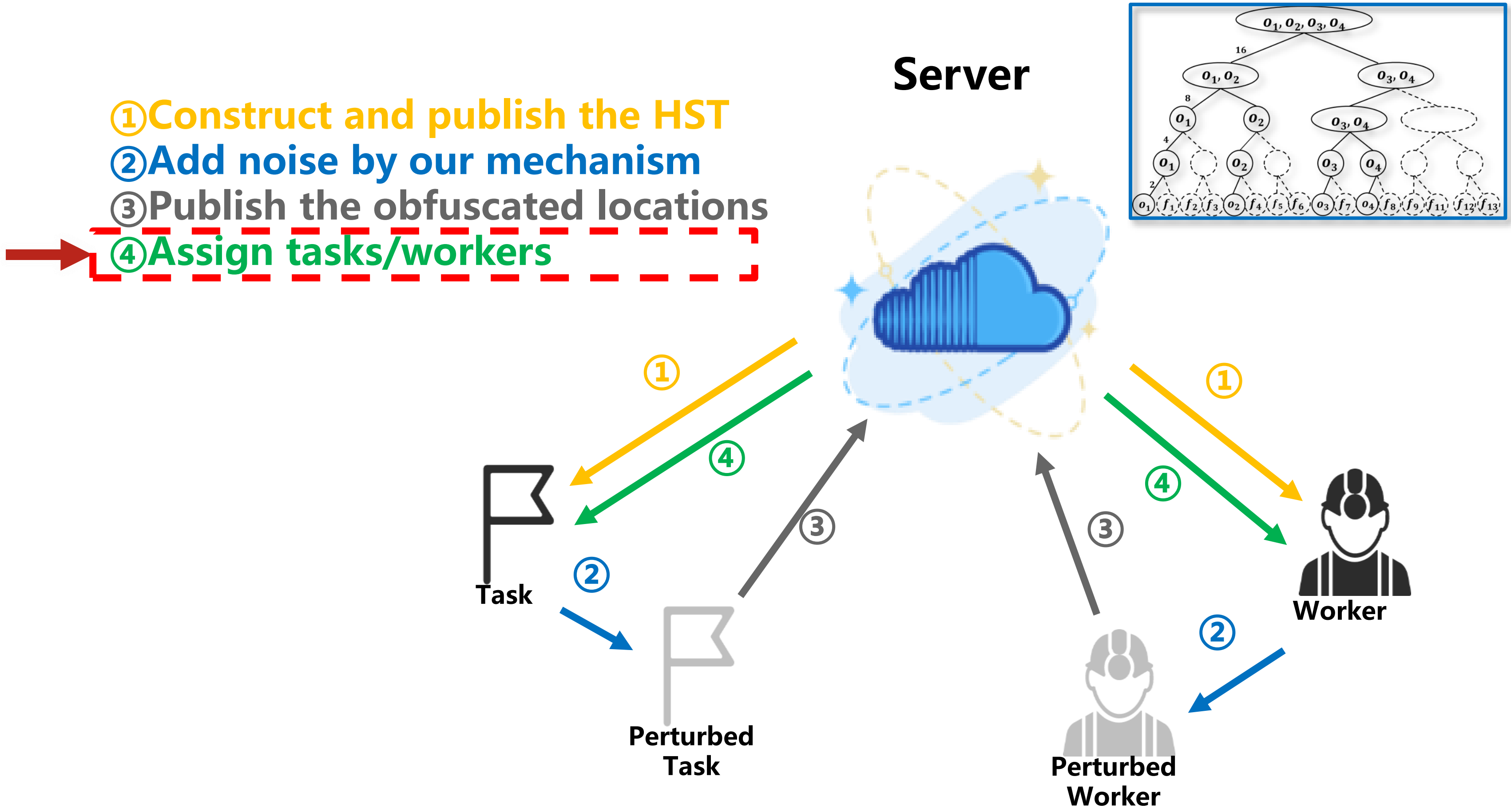
$$\mathcal{M}(x_1)(z)/\mathcal{M}(x_2)(z)$$
$$= \Pr_{lvl(x_1,z)}/\Pr_{lvl(x_2,z)}$$
$$= e^{(2^{lvl(x_2,z)+2} - 2^{lvl(x_1,z)+2})\epsilon}$$
$$\le e^{(2^{lvl(x_1,x_2)+2} - 2^2)\epsilon}$$
$$= e^{d_T(x_1,x_2)\epsilon}$$

$\Longrightarrow$ Case 2 proved



Tree-based Framework

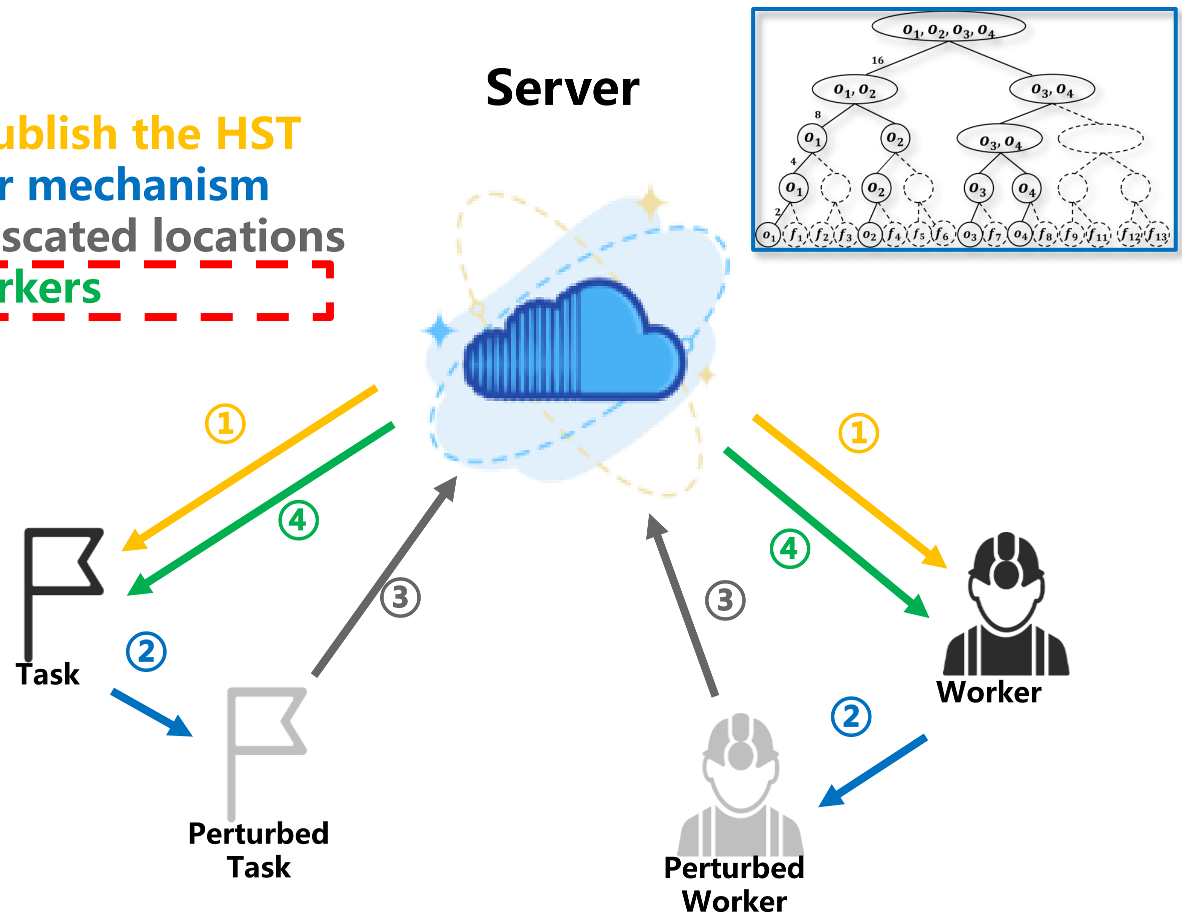
- Our solution is devised based on a tree-based framework.



Task Assignment

- Main Idea: Devise a greedy algorithm on the HST.

①Construct and publish the HST
②Add noise by our mechanism
③Publish the obfuscated locations
④Assign tasks/workers



Task Assignment

- Analysis of HST-based Greedy:
  - The competitive ratio of the Tree-based Framework can be bounded by

The Matching with server unknowing truth locations

$$\frac{M_{TBF}}{M_{OPT}} = O\left(\frac{1}{\epsilon^4} \cdot \log N \log^2 k\right)$$

The Optimal Matching even knowing all truth locations

An extra product related to privacy budget ϵ

The competitive ratio of HST-Greedy without privacy

N: Number of truth nodes in HST	
k: Number of tasks /workers	
ϵ: Privacy budget	

Outline

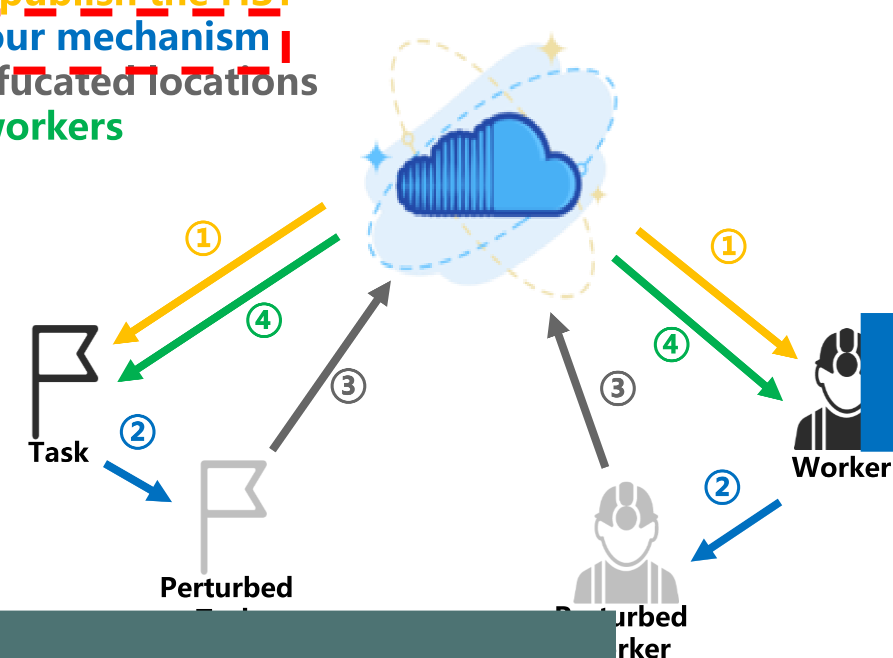
- Background and Motivation
- Problem Definition
- A Tree-based Framework
- Random Walk Acceleration
- Experimental Evaluation
- Conclusions

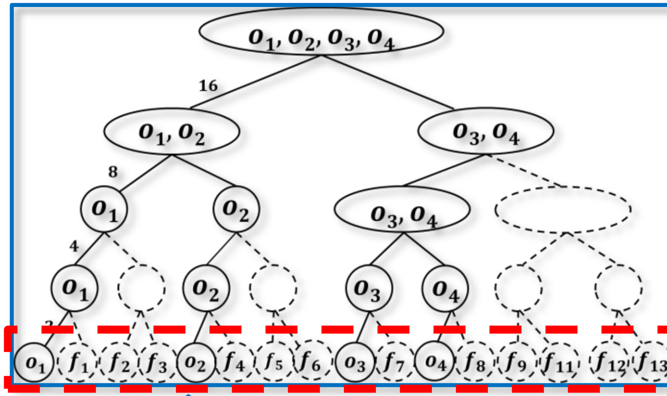
Privacy Mechanism Revisited

- It takes $O(c^D)$ to enumerate the probability of all leaf nodes.

c: **Number of branches of the HST**	
D: **Levels of the HST**	

①Construct and publish the HST
②Add noise by our mechanism
③Publish the obfucated locations
④Assign tasks/workers





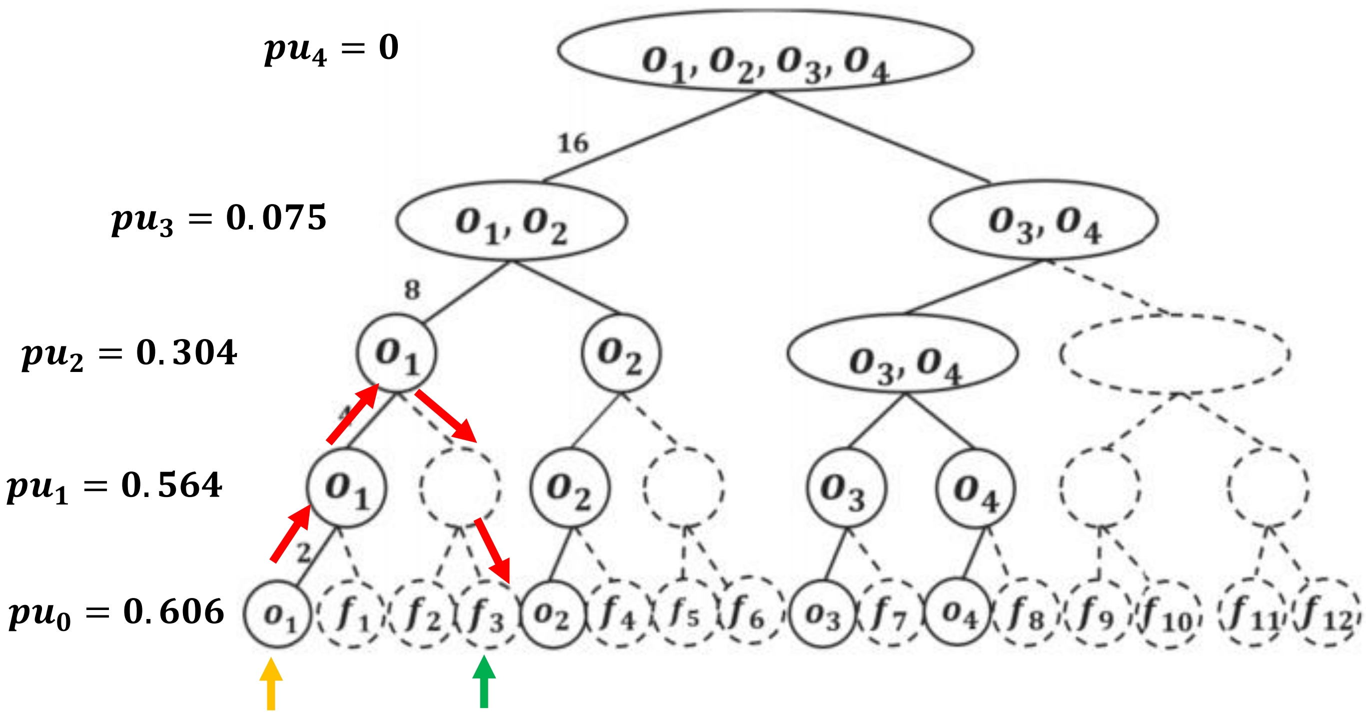
node	istance	weights	Prob
$f_2 - f_3$	12	0.301	0.119
$o_2, f_4 - f_6$	28	0.061	0.024
o_3-o_4, f_7-f_{12}	60	0.002	0.001

c^{D-1} leaf nodes in total

Question:
How to accelerate the mechanism?

A Random Walk Acceleration

- Main Idea:
  - Start from the exact node and randomly walk up or down with some probability at each node
  - Repeat until another leaf node is reached

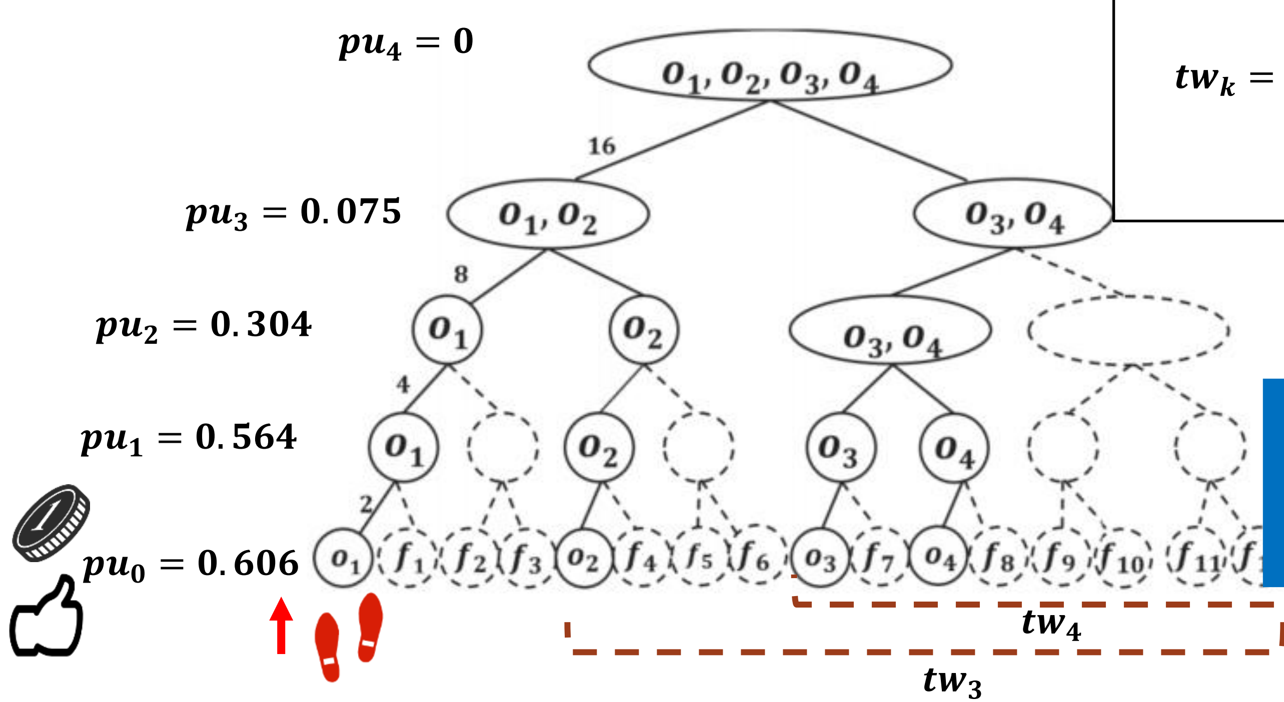


A Random Walk Acceleration

- Algorithm Details:
  - Phase I: Walk up until obtain a tail from the coin (at level k) with its head probability

$$pu_k = \frac{tw_{k+1}}{tw_k}$$

$$tw_k = \begin{cases} \sum_{i \geq k}^{D} c^{i-1}(c-1)wt_i, \textit{if } k > 0 \\ w_0 + \sum_{i=1}^{D} c^{i-1}(c-1)wt_i, \textit{if } k = 0 \end{cases}$$



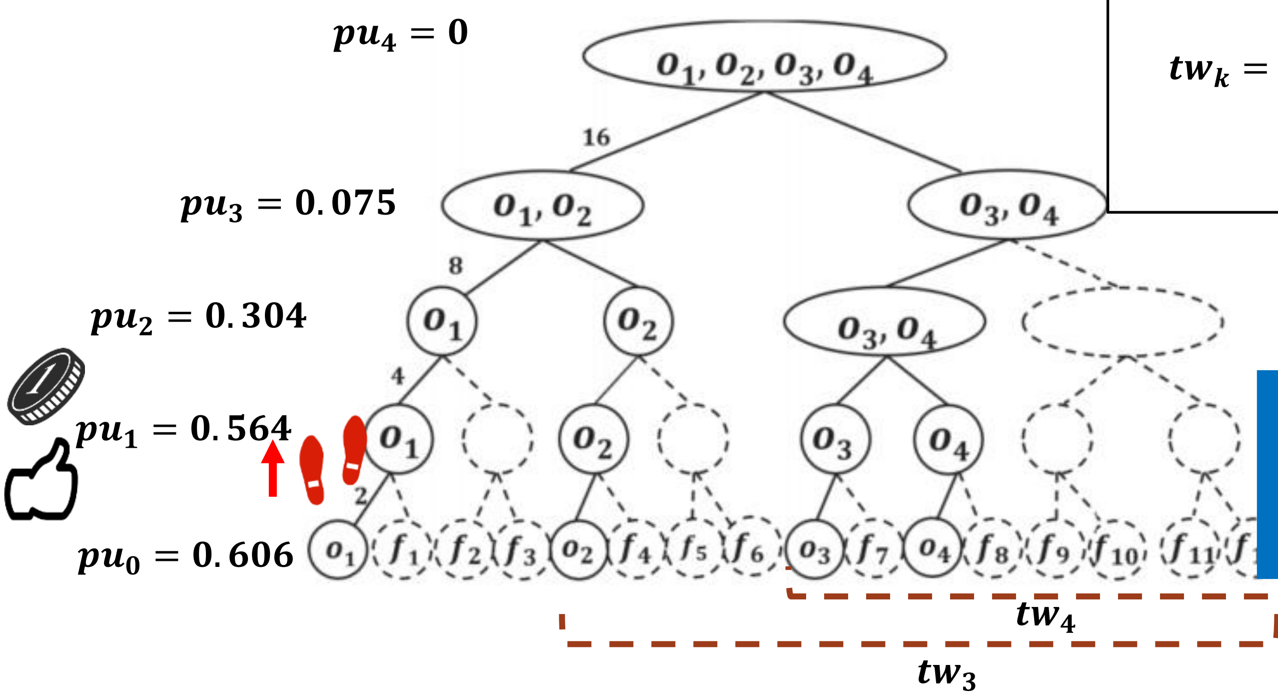
The total weights of leaf nodes outside level k (including level k)

A Random Walk Acceleration

- Algorithm Details:
  - Phase I: Walk up until obtain a tail from the coin (at level k) with its head probability

$$pu_k = \frac{tw_{k+1}}{tw_k}$$

$$tw_k = \begin{cases} \sum_{i \geq k}^{D} c^{i-1}(c-1)wt_i\,, if\ k > 0 \\ w_0 + \sum_{i=1}^{D} c^{i-1}(c-1)wt_i\,, if\ k = 0 \end{cases}$$



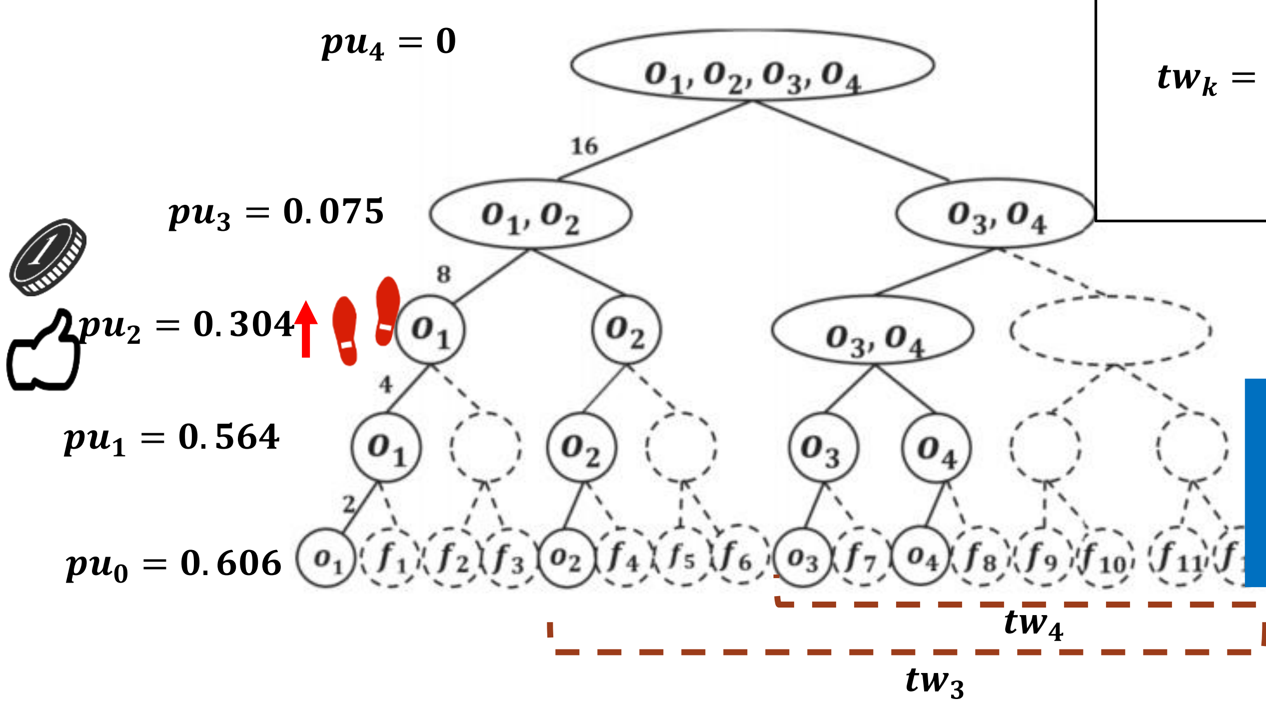
The total weights of leaf nodes outside level k (including level k)

A Random Walk Acceleration

- Algorithm Details:
  - Phase I: Walk up until obtain a tail from the coin (at level k) with its head probability

$$pu_k = \frac{tw_{k+1}}{tw_k}$$

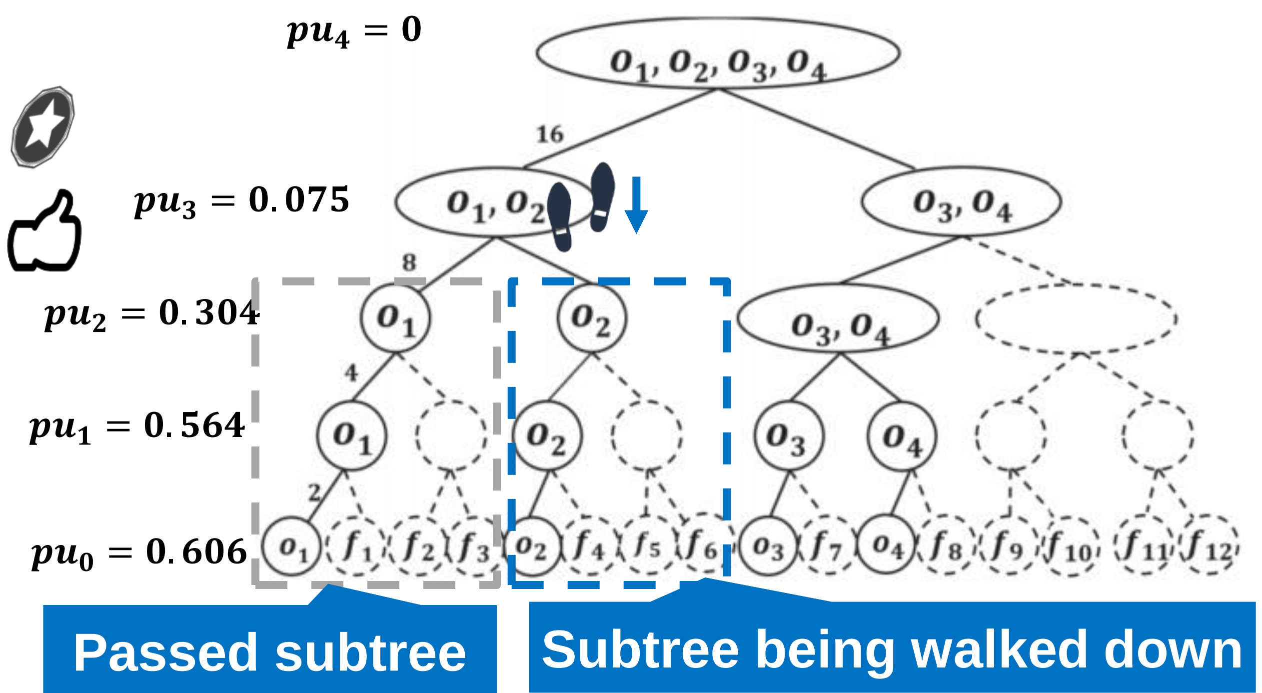
$$tw_k = \begin{cases} \sum_{i \geq k}^{D} c^{i-1}(c-1)wt_i, \text{if } k > 0 \\ w_0 + \sum_{i=1}^{D} c^{i-1}(c-1)wt_i, \text{if } k = 0 \end{cases}$$



The total weights of leaf nodes outside level k (including level k)

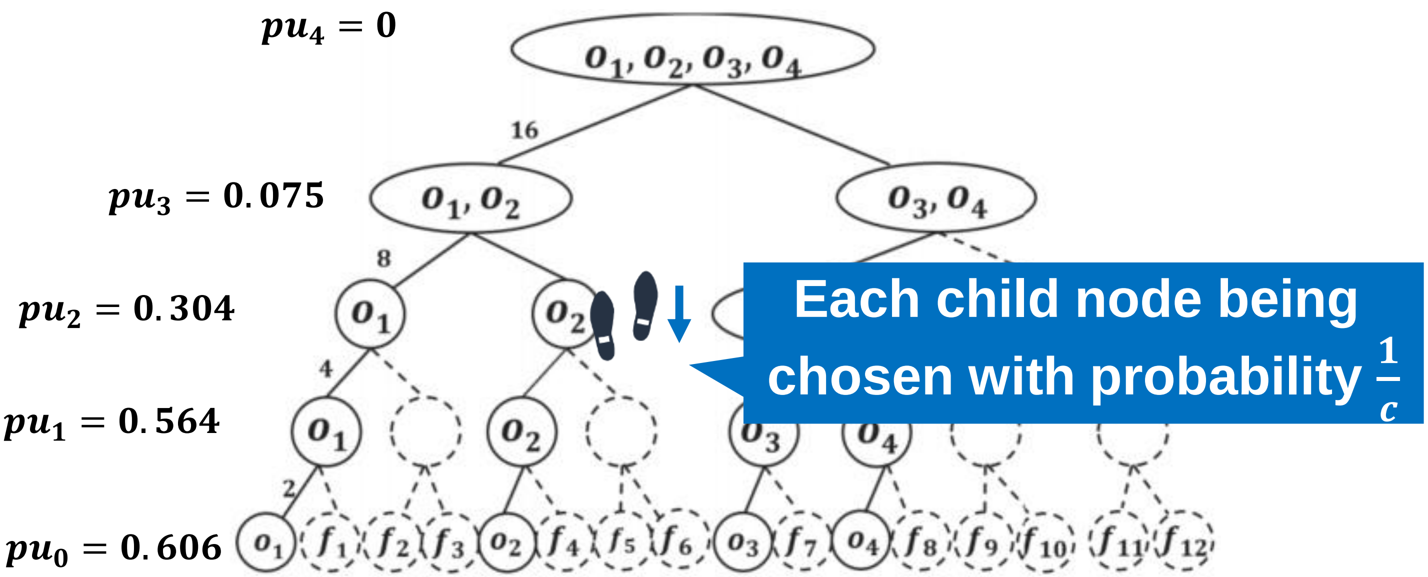
A Random Walk Acceleration

- Algorithm Details:
  - Phase II: Walk down uniformly (except the subtree that has been passed) until reaching a leaf node



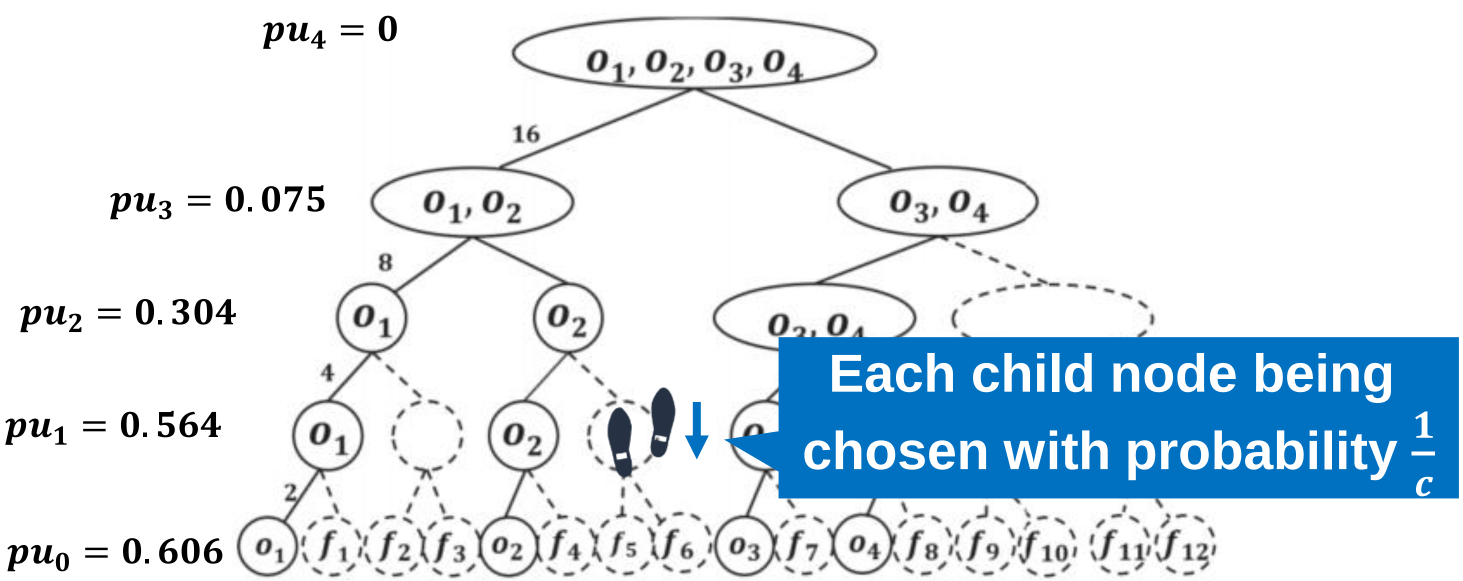
A Random Walk Acceleration

- Algorithm Details:
  - Phase II: Walk down uniformly (except the subtree that has been passed) until reaching a leaf node



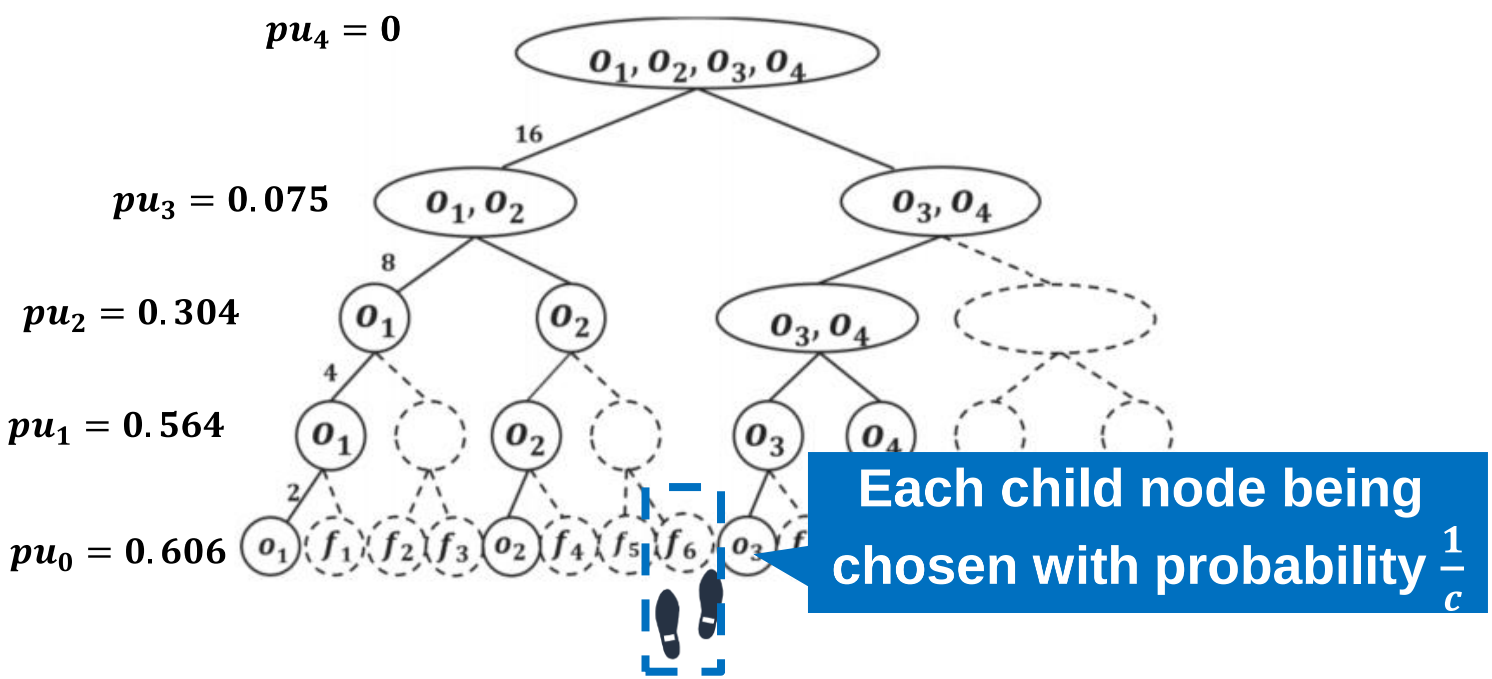
A Random Walk Acceleration

- Algorithm Details:
  - Phase II: Walk down uniformly (except the subtree that has been passed) until reaching a leaf node



A Random Walk Acceleration

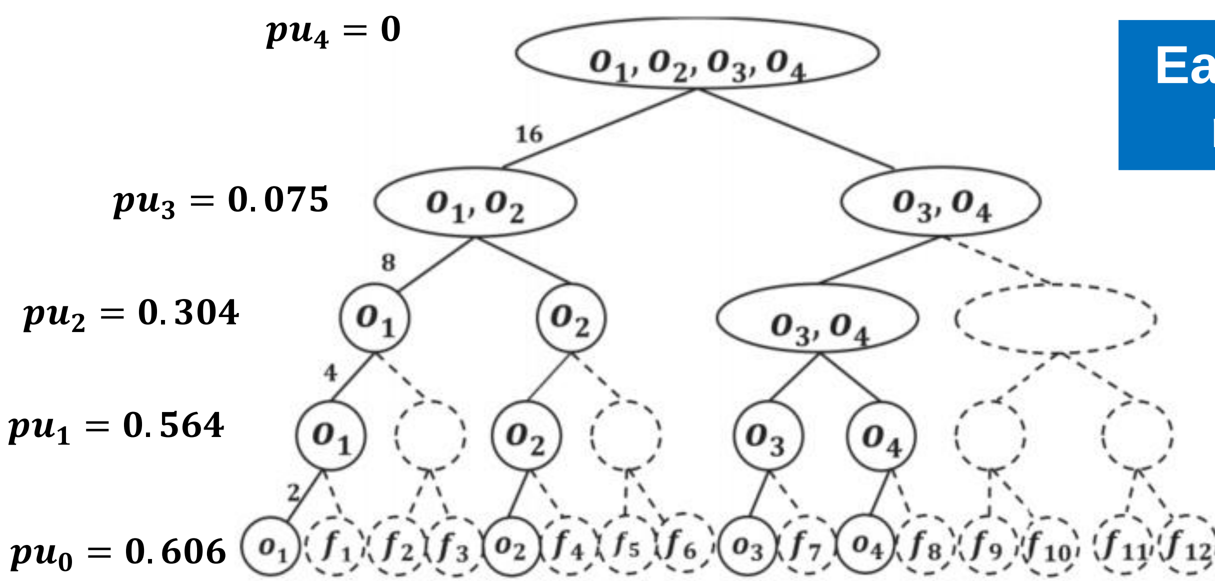
- Algorithm Details:
  - Phase II: Walk down uniformly (except the subtree that has been passed) until reaching a leaf node



A Random Walk Acceleration

- Time Complexity:
  - Phase I: Walk up until obtain a tail from the coin (at level k) with its head probability
  - Phase II: Walk down uniformly (except the subtree that has been passed) until reaching a leaf node

Each level is passed at most 2 times: $\boldsymbol{O(D)}$



Outline

- Background and Motivation
- Problem Definition
- A Tree-based Framework
- Random Walk Acceleration
- Experimental Evaluation
- Conclusions

Experimental Settings

- Compared Algorithms:
  - TBF:

    Our tree-based framework + the random walk acceleration

  - Lap-GR:

    State-of-the-art mechanism for location privacy

    Laplacian Mechanism + The Greedy Algorithm

    Representative task assignment algorithms with minimum total distance

  - LAP-HG:

    Laplacian Mechanism + The HST-Greedy Algorithm

Experimental Settings

- Datasets:
  - Synthetic datasets: 200x200 Euclidean space

Parameters	Settings
$\lvert T \rvert$	1000, 2000, **3000**, 4000, 5000
$\lvert W \rvert$	3000, 4000, **5000**, 6000, 7000
mean μ	50, 75, **100**, 125, 150
standard deviation σ	10, 15, **20**, 25, 30
privacy budget ϵ	0.2, 0.4, **0.6**, 0.8, 1
scalability ($\lvert T \rvert$	2×10^4, 4×10^4, 6×10^4, 8×10^4, 10×10^4

Parameters for Normal distribution

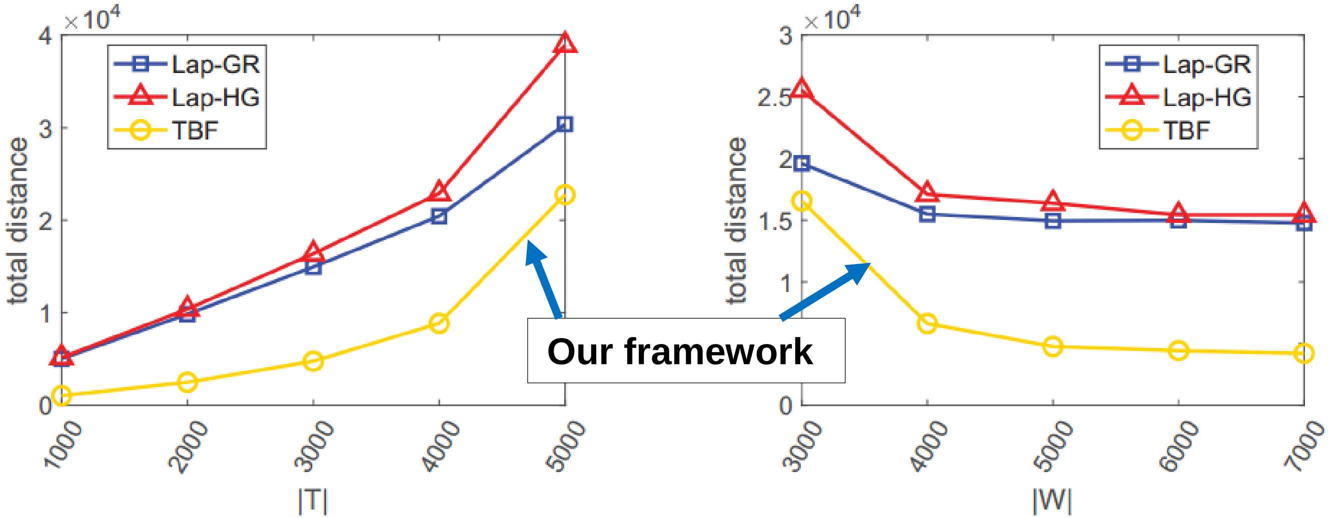
  - Real datasets:

Trip records of passengers from Didi Chuxing

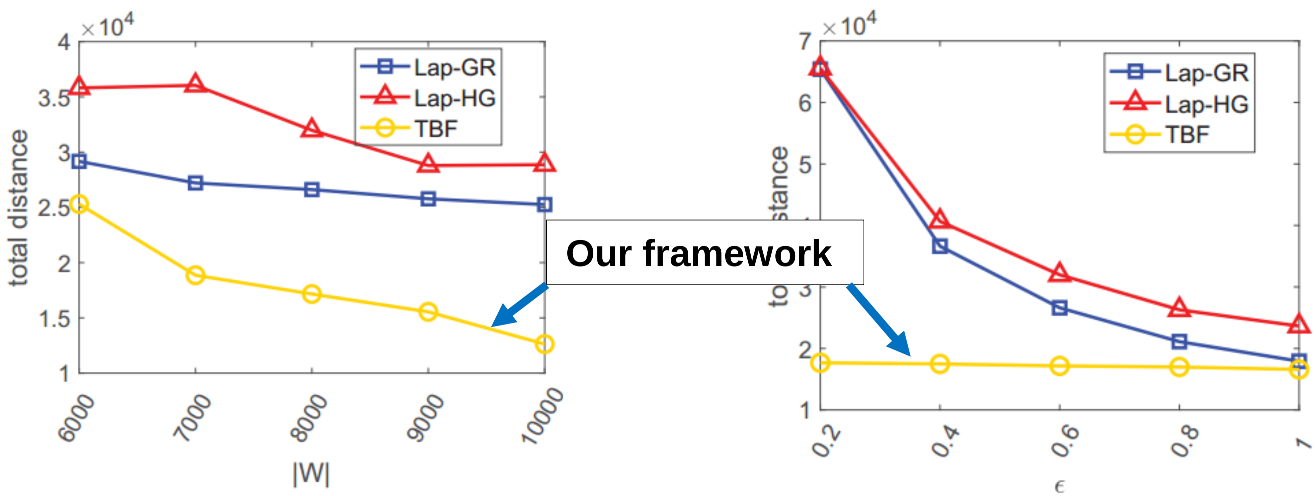
Parameters	Settings
collected date	2016/11/01, $\cdots$, 2016/11/30
$\lvert T \rvert$	range from 4245 to 5034
$\lvert W \rvert$	6000, 7000, **8000**, 9000, 10000
ϵ	0.2, 0.4, **0.6**, 0.8, 1

Experimental Results

- Results on synthetic datasets



- Results on seal datasets



Outline

- Background
- Problem Definition
- A Tree-based Framework
- Random Walk Acceleration
- Experimental Evaluation
- Conclusions

Contributions

- Devise a novel tree-based framework for private online task assignment
- Design a privacy mechanism to protect location privacy
- analyze the effectiveness of the framework
- Propose a random walk method for acceleration


dank u
ju faleminderit
Tack
Asante
谢谢
Tak
mulţumesc
kiitos
Gracias
Salamat!
Terima kasih
Aliquam
Merci
Dankie
Obrigado
ありがとう
köszönöm
grazie
Aliquam
Go raibh maith agat
děkuji
Thank you
gam