User Customizable and Robust Geo-Indistinguishability for Location Privacy

Primal Pappachan, Chenxi Qiu, Anna Squicciarini, Vishnu Sharma Hunsur Manjunath

Location: To share or not share?

Uber

Ride Hailing Apps

iNaturalist

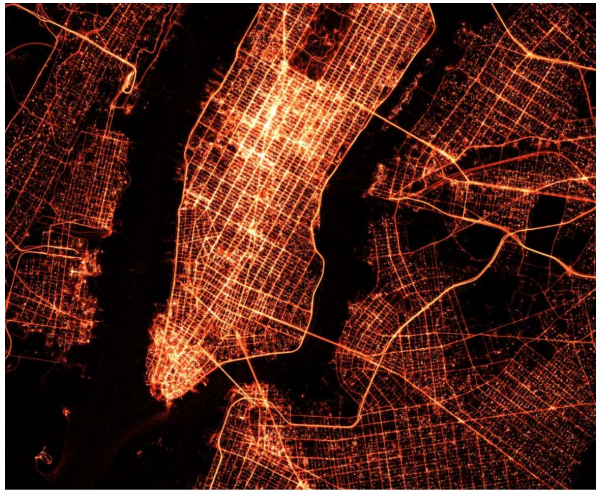
Citizen Science

yelp

Locality-based Search Engine

Pokémon GO

Games



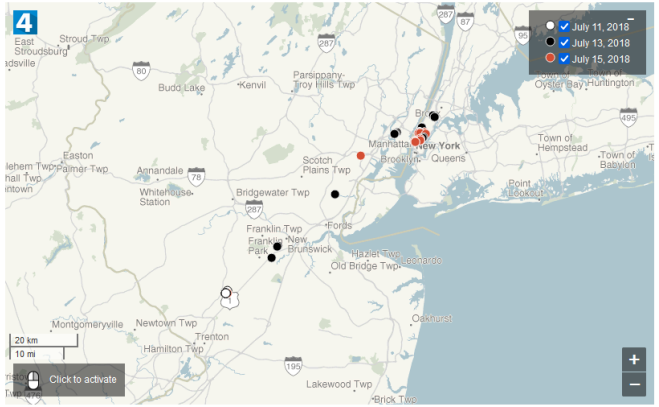
NYTimes

Your Apps Know Where You Were Last Night, and They're Not Keeping It Secret

AP

'Location history' off? Google's still tracking you

An AP investigation found that Google saves your location history even if you've paused "Location History" on mobile devices. This map shows where Princeton privacy researcher Gunes Acar travelled over several days, from data saved to his Google account despite "Location History" being off.



Google reaches record $392M privacy settlement over location data

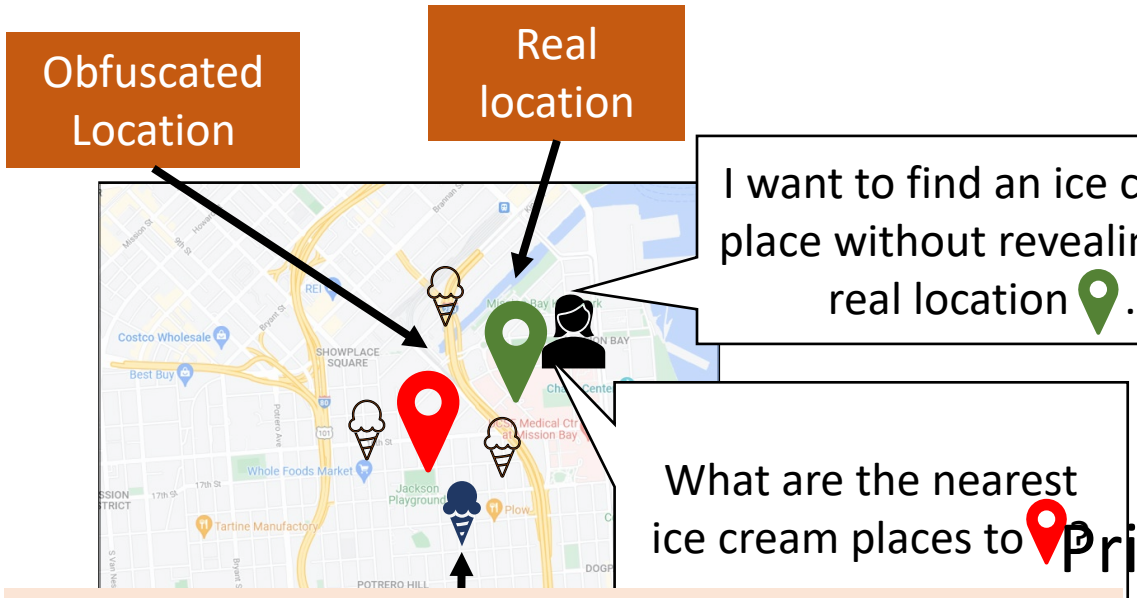
By Bryan Pietsch

November 15, 2022 at 2:45 a.m. EST

(Charles Platiau/Reuters)

Location obfuscation

Users get privacy while sacrificing some utility



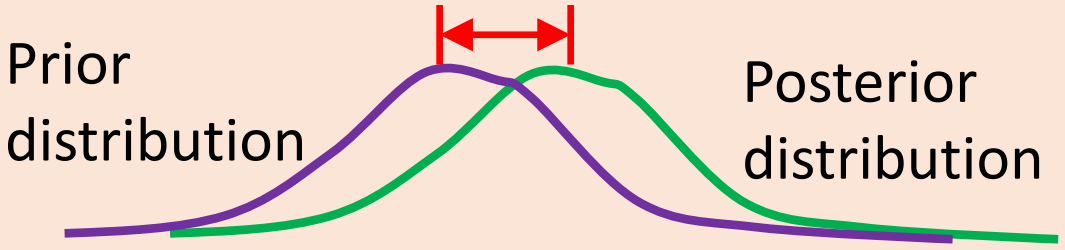
Utility Metric: **Quality Loss**

$$|distance(\,,\,) - distance(\,,\,)|$$

Privacy Criterion: **Geo-Indistinguishability [1]**

$$\frac{\Pr(X = \,|\, Y = \,)}{\Pr(X = \,|\, Y = \,)} \leq e^{\epsilon . distance(\,,\,)} \frac{\Pi(\,)}{\Pi(\,)}$$

The difference is quantified by ϵ



[1] Andrés, Miguel E., et al. "Geo-indistinguishability: Differential privacy for location-based systems (2013)

Problem Formulation

$$\boldsymbol{Z} = \begin{bmatrix} z_{1,1} & \cdots & z_{1,K} \\ \vdots & \ddots & \vdots \\ z_{K,1} & \cdots & z_{K,K} \end{bmatrix}$$

Area of Interest

Obfuscation matrix generation

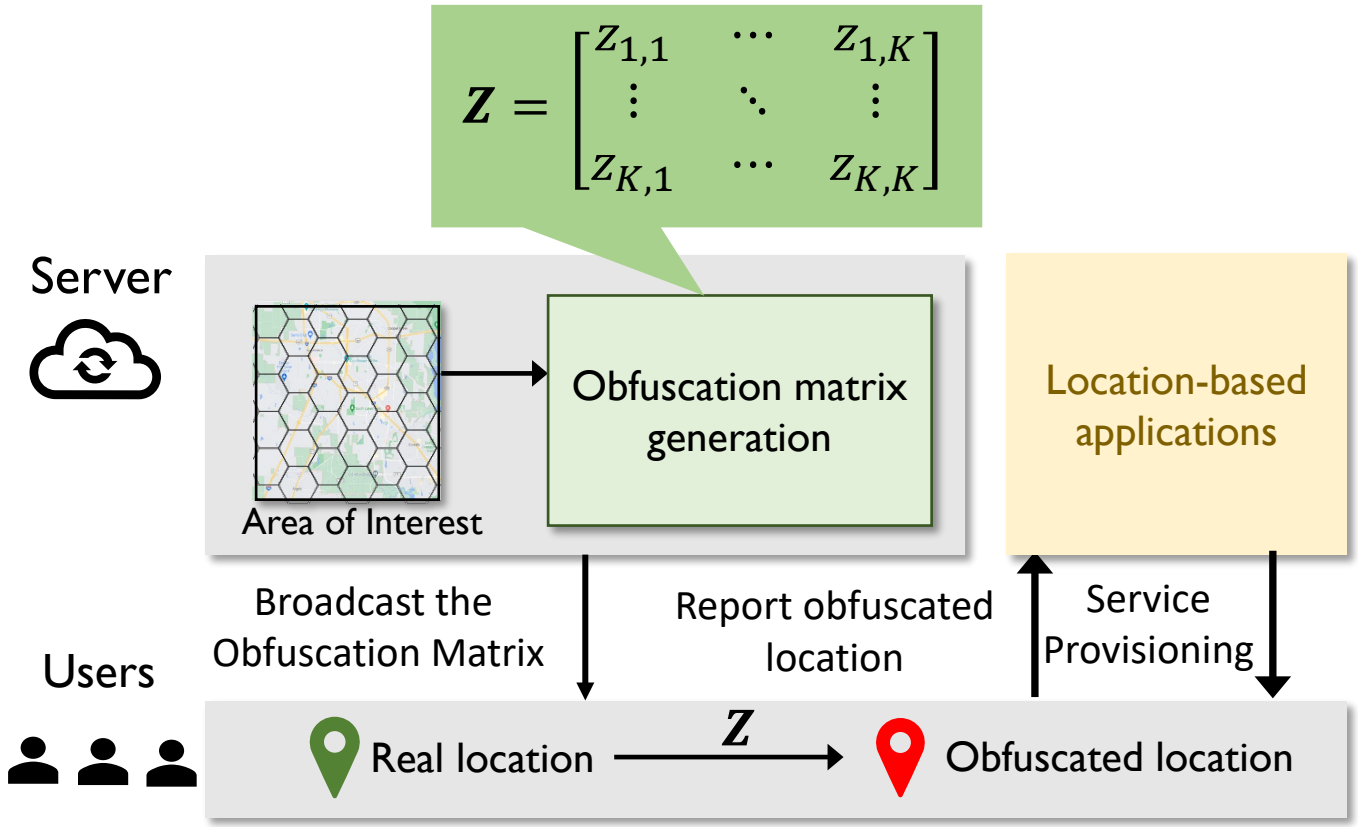
$z_{k,l}$: The probability of reporting **location l** as the obfuscated location given the real **location k**.

of target locations while satisfying the **Privacy Criterion** for each pair of locations.

Complexity: The number of constraints = $O(K^3)$

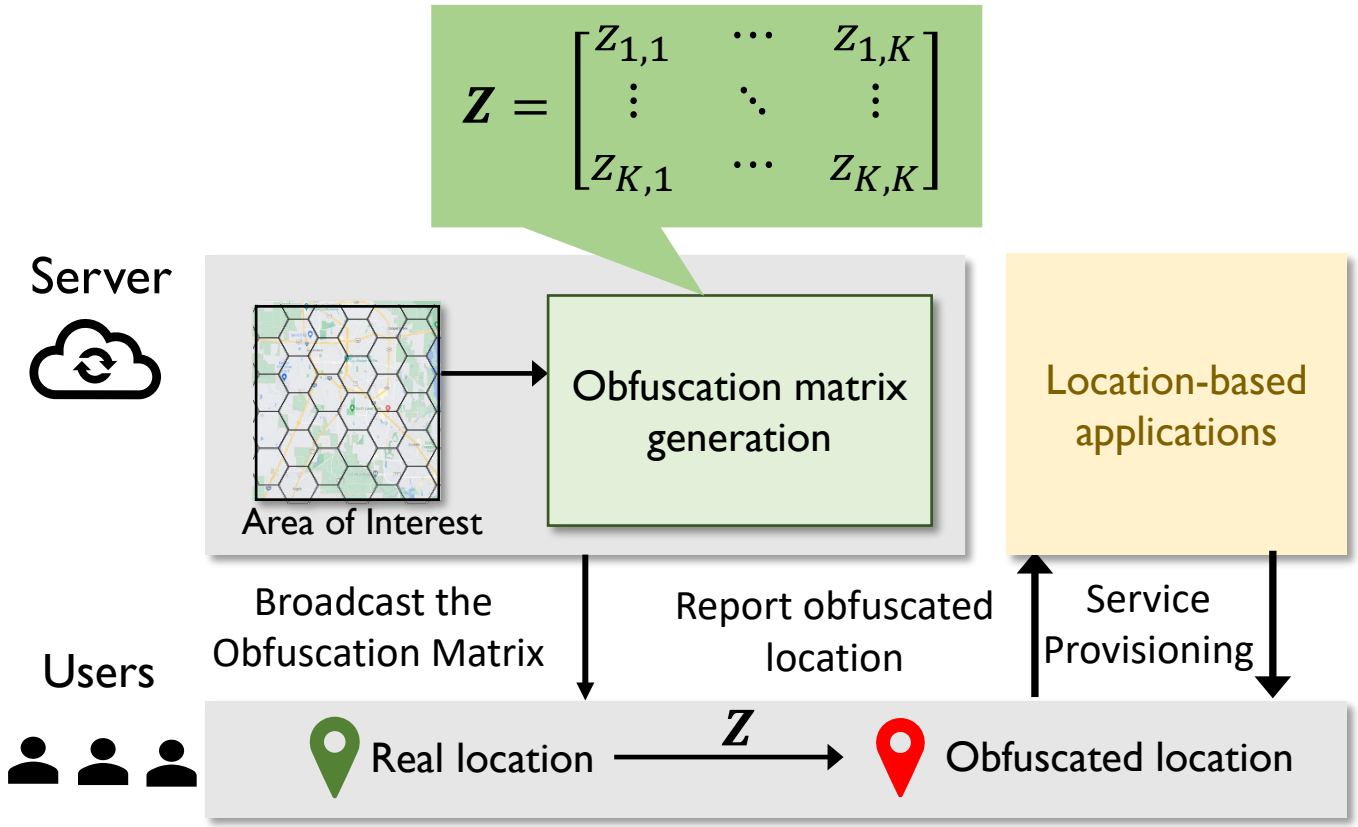
Sample a set of discrete locations over the area of interest: 1, 2, …, *K*.

Location Obfuscation Workflow



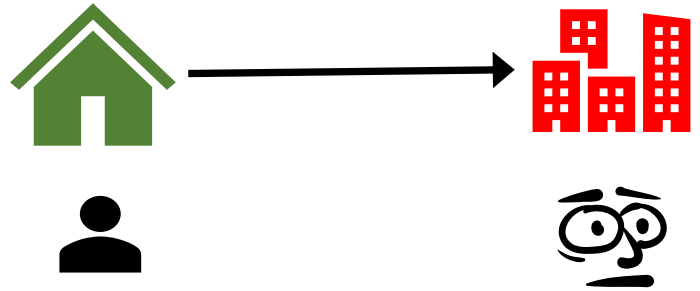
Qiu et. al (2020), Shokri et.al. (2012), Wang et.al. (2017)

Location Obfuscation Workflow



Problem #1: **Customization** not supported

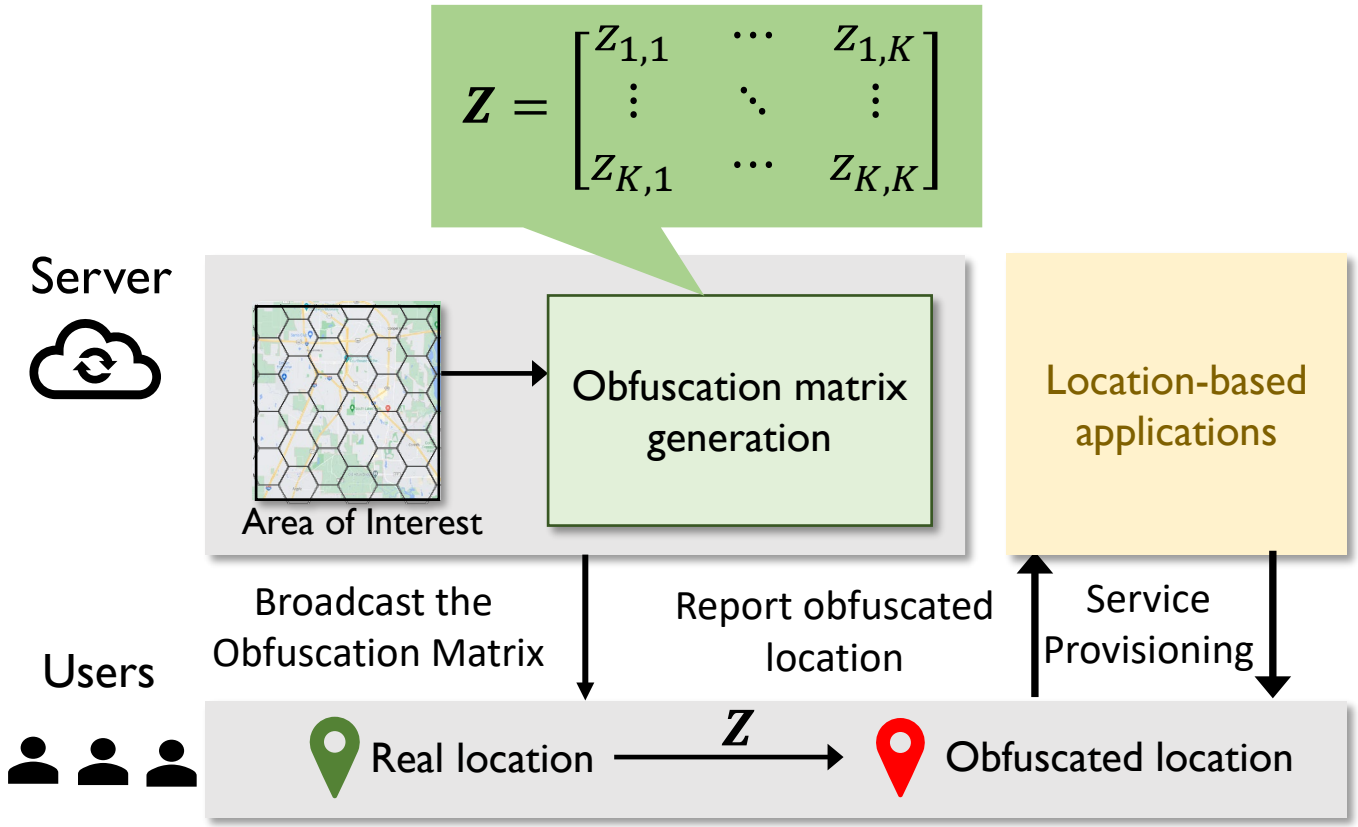
Users cannot state their preferences for range of locations used for reporting.



Users could be mapped to undesirable locations leading to poor quality of service

Qiu et. al (2020), Shokri et.al. (2012), Wang et.al. (2017)

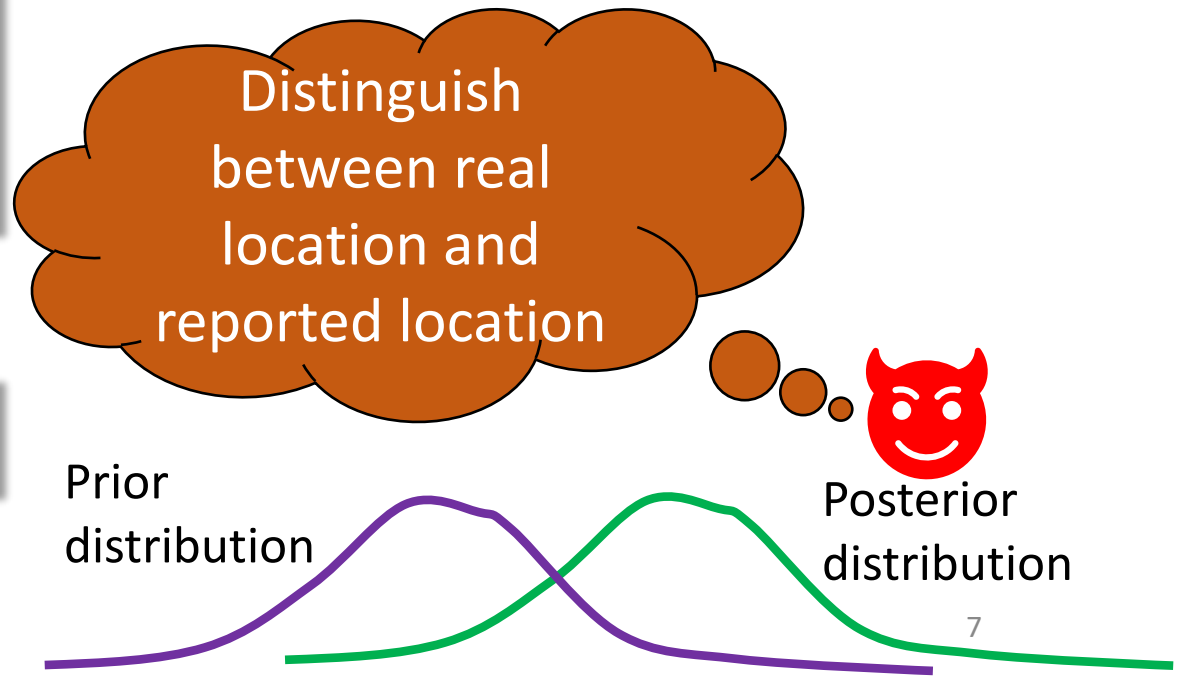
Location Obfuscation Workflow



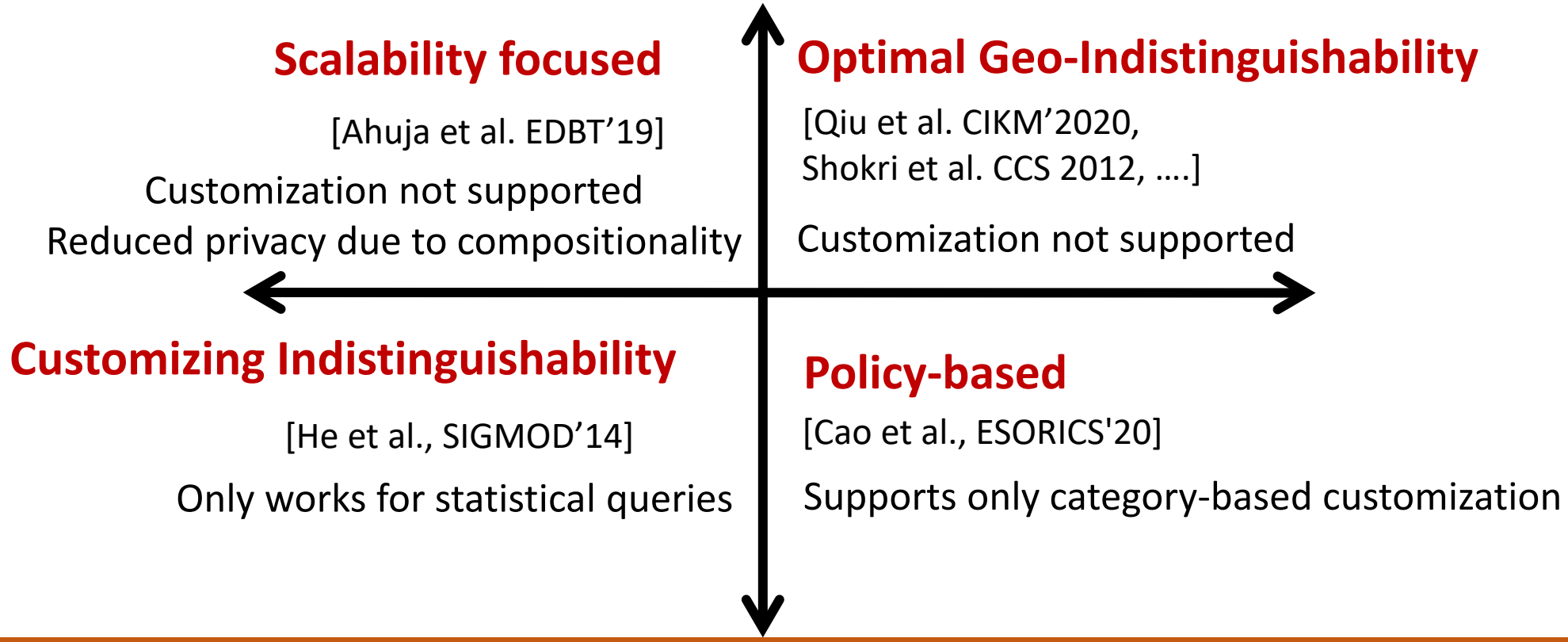
Qiu et. al (2020), Shokri et.al. (2012), Wang et.al. (2017)

Problem #2: Lack of **Robustness**

If a user modifies the range of locations used for reporting, it leads to violating the privacy guarantees of Geo-Indistinguishability.



Prior works



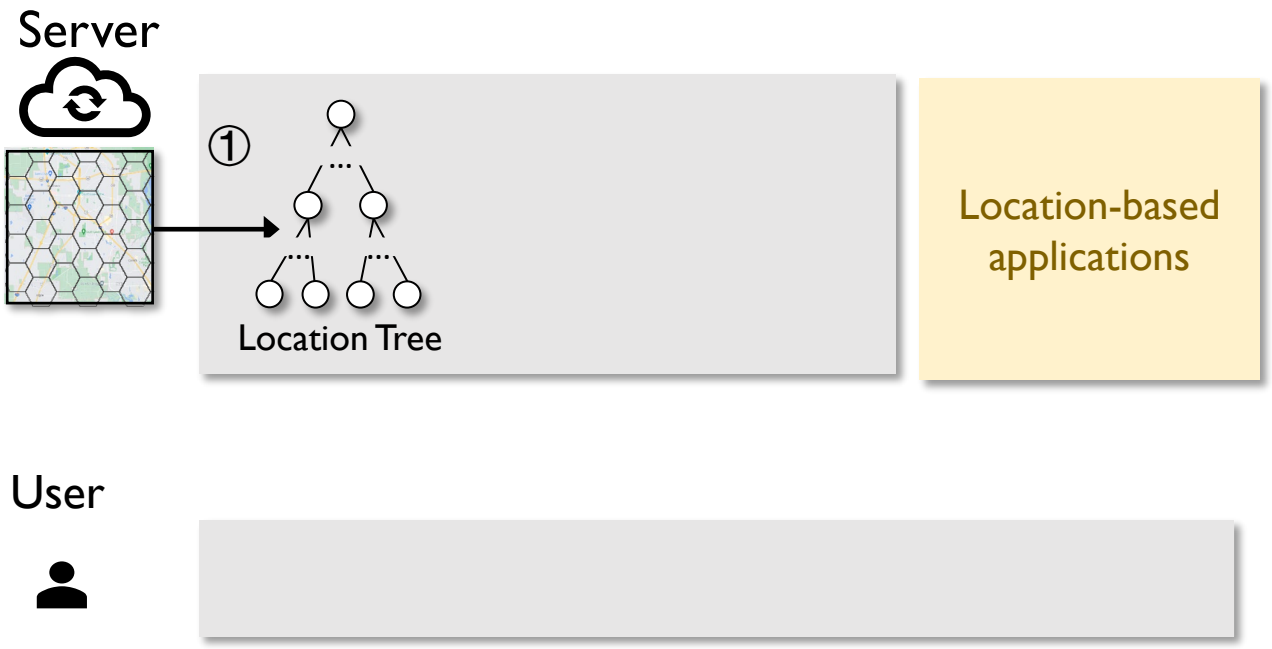
None of the prior works have proposed a general model for user customizable and robust location obfuscation

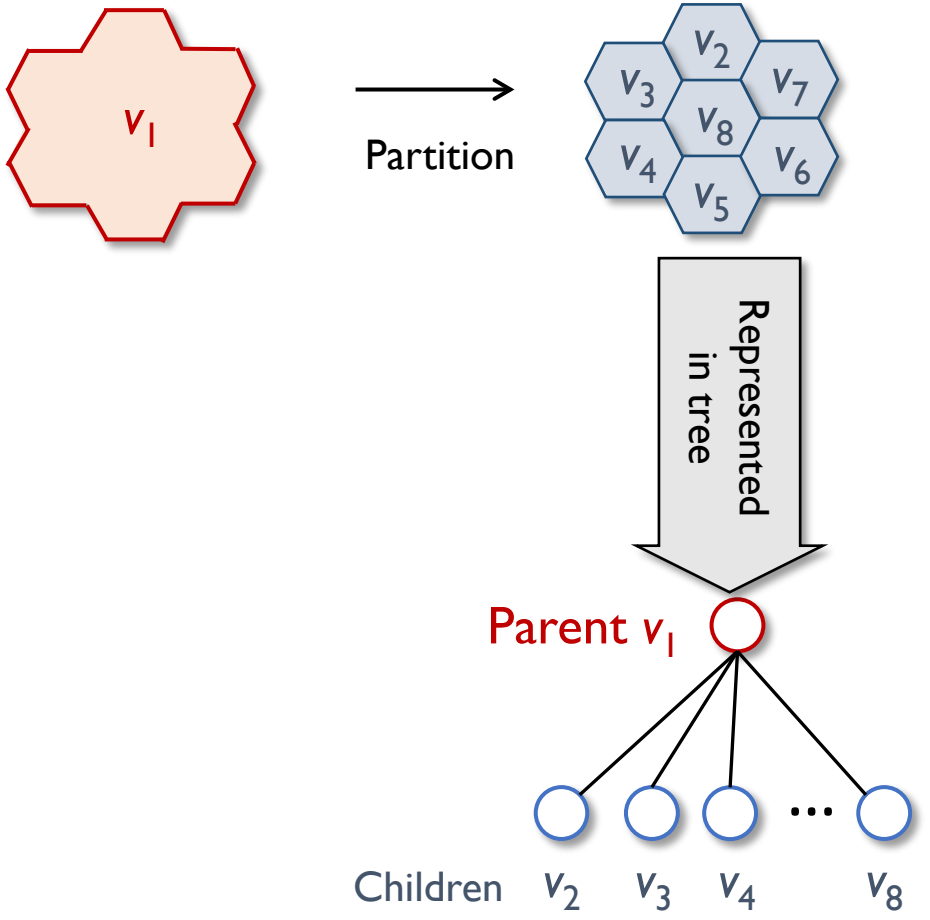
Our framework

CORGI: **C**ust**o**mizable **R**obust **G**eo-**I**ndistinguishability

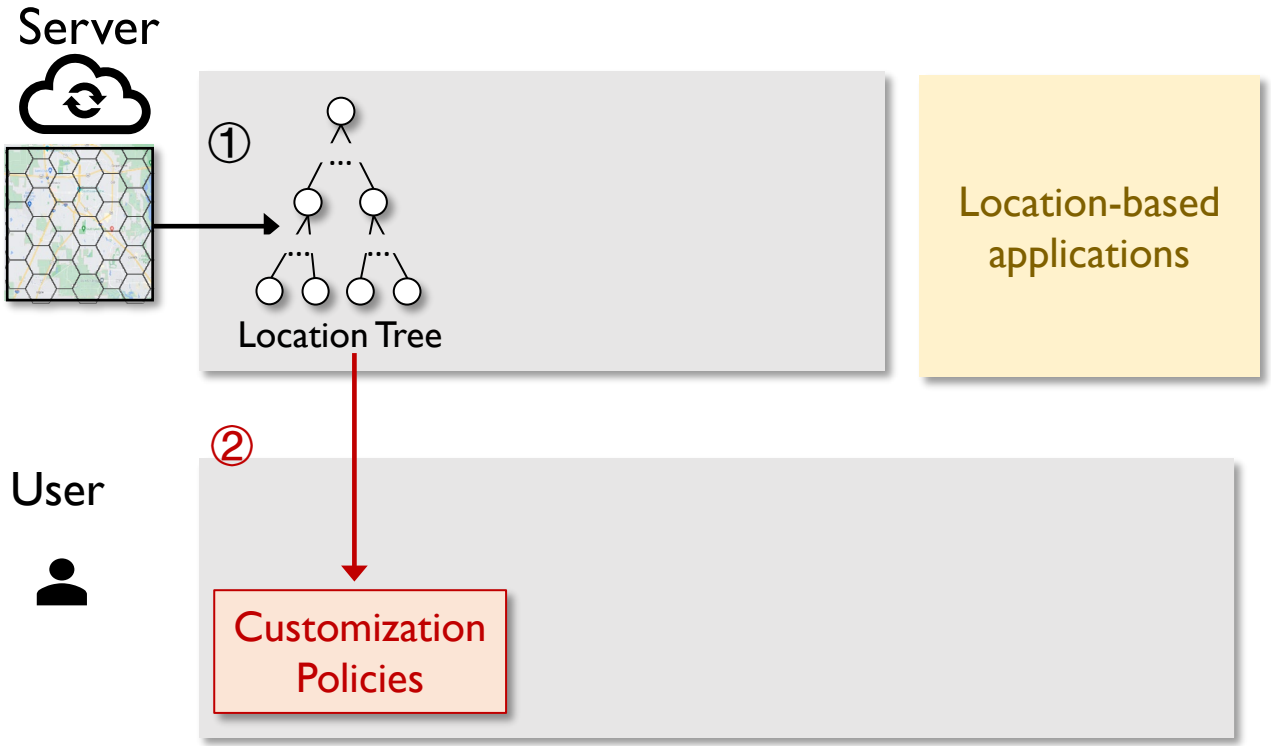
- Allow users to **customize** obfuscation matrices
- Ensure **robustness** of matrix after customization
- Improve efficiency of the workflow using optimization techniques

CORGI framework





CORGI framework



User Customization Policy

- ***Privacy_level:*** the maximum range of locations used for reporting (subtree)

e.g., Privacy_level = 3

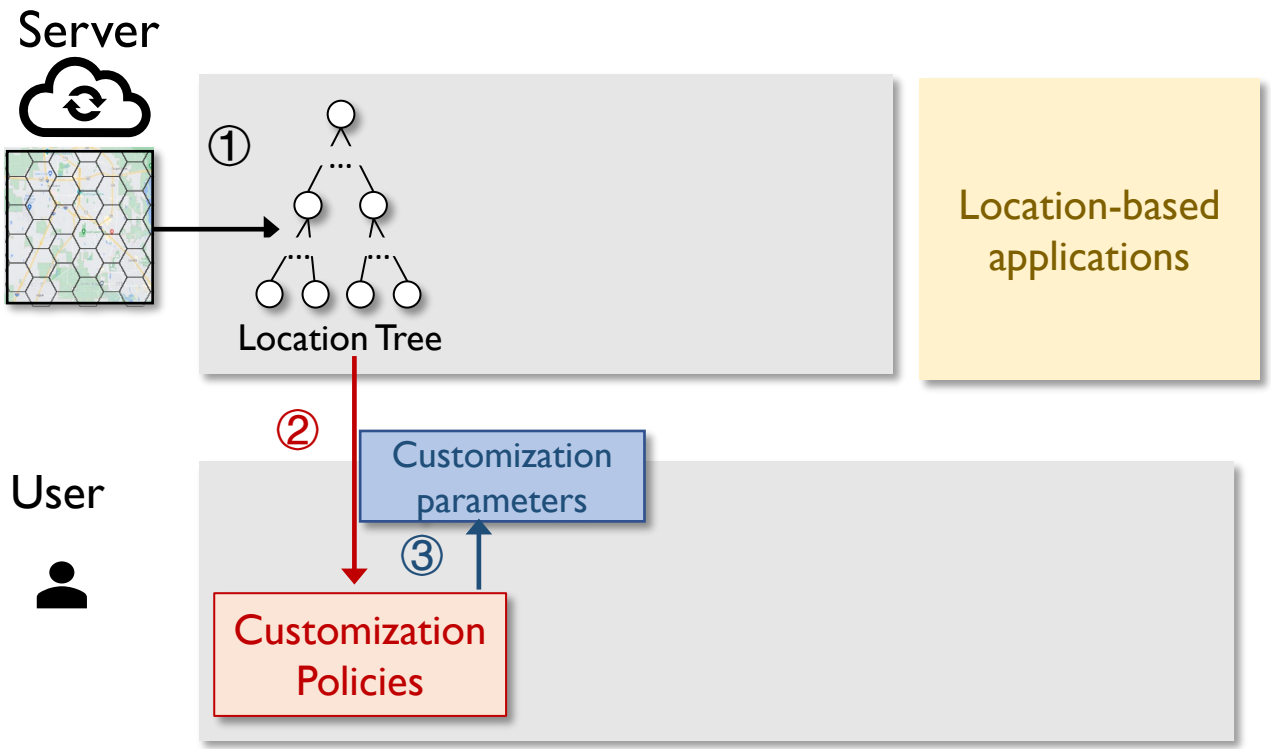
- ***Precision_level:*** granularity of location sharing (height of the tree)

e.g., Precision_level = 0

- ***User_Preferences:*** Specific locations to be removed from being reported (nodes to be pruned from the tree)

e.g., location = home, driving_distance $>$ 5 miles

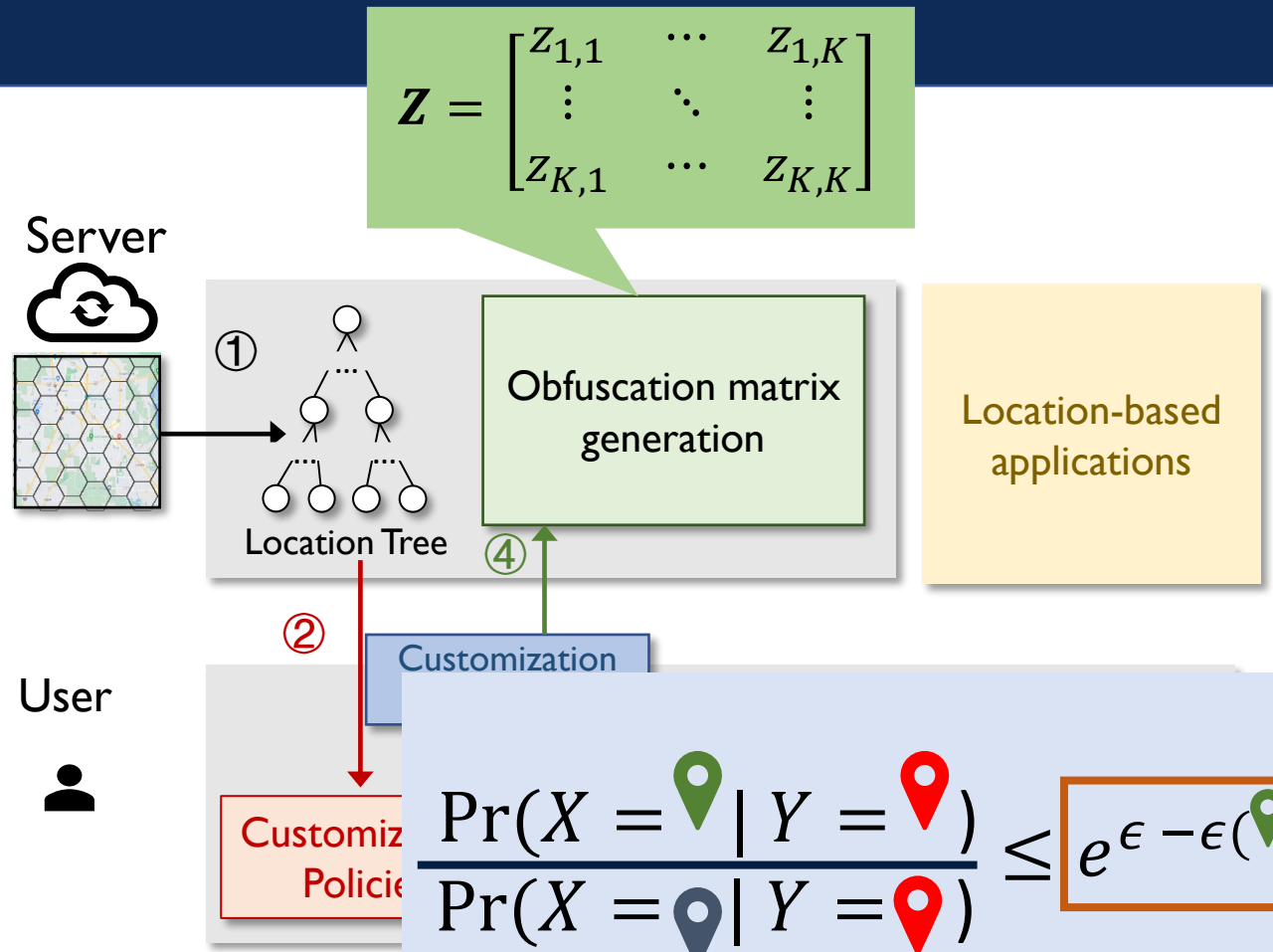
CORGI framework



Customization Parameters

- Server and the communication channel are **untrusted**
- Customization Policies are **sensitive** and cannot be shared directly with the server
- On the user side, **CORGI** evaluates the *User_Preferences* and only shares the **maximum number of locations (δ)** that could be removed

CORGI framework

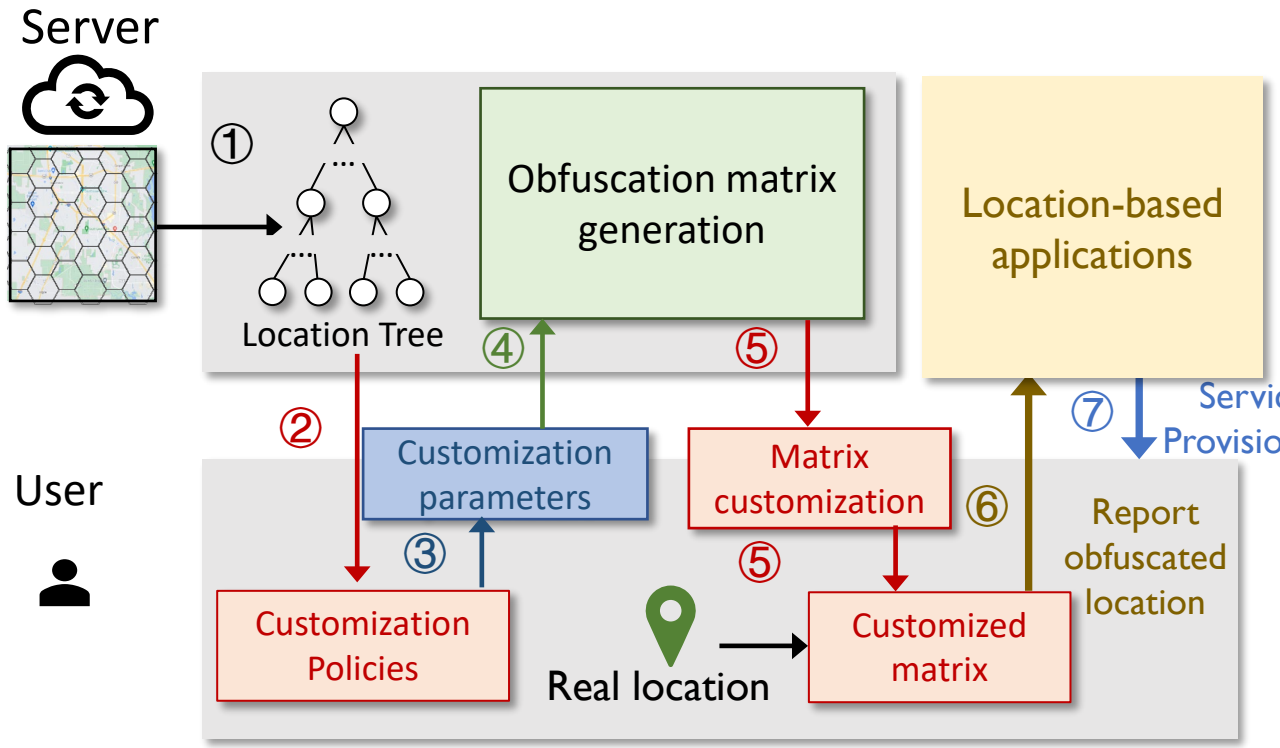


Generating the obfuscation matrix

- ***δ-prunable*** obfuscation matrix → if after removal of δ locations, the reported location still satisfies the Geo-Ind condition
- On the server side, **CORGI** adjusts the privacy metric by determining **reserved**

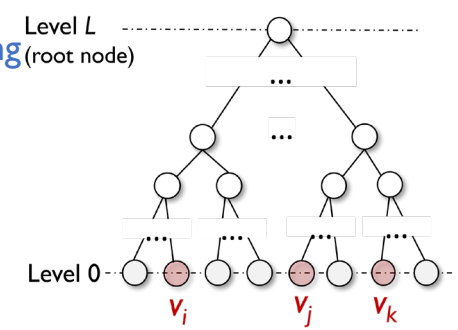
problem to
able obfuscation

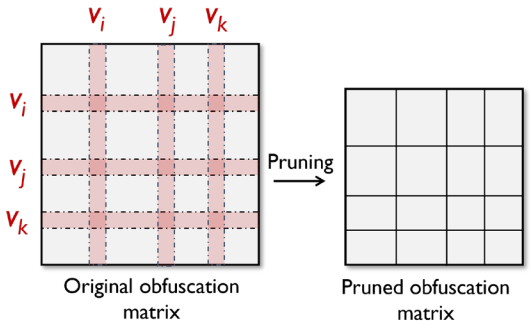
CORGI framework



Customizing the matrix

- Removing rows and columns corresponding to locations that do not satisfy *User_ Preferences*





Experimental Setup

- **Dataset**: Gowalla dataset [1] containing user check-ins at location
- **Location Tree** generated using H3 library*
- Prior Probability computed by counting # check-ins per location
- Baseline: Non-robust approach for matrix generation [2, 3, 4]

Source code available on GitHub 

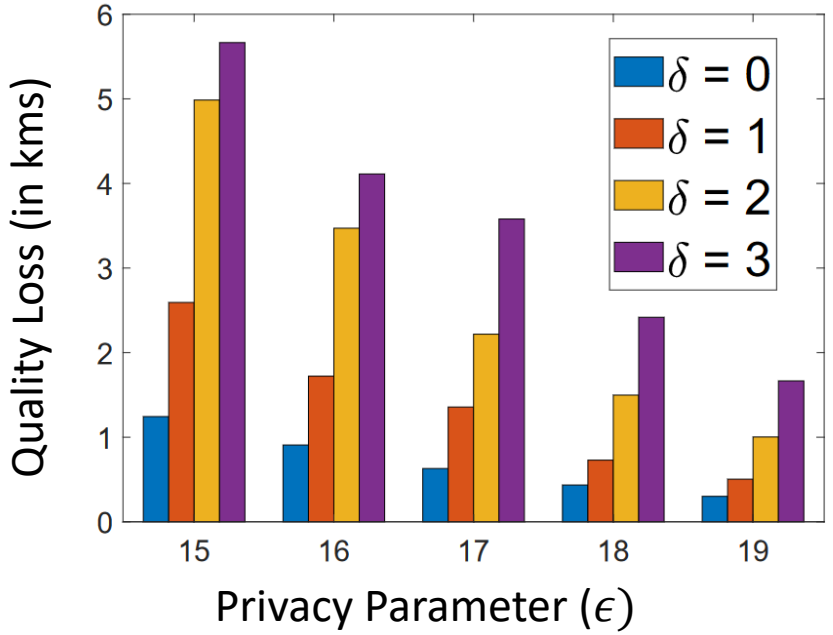
[1] Cho et. al (2011)

[2] Qiu et. al (2020), [3] Shokri et.al. (2012), [4] Wang et.al. (2017)

*https://h3geo.org/

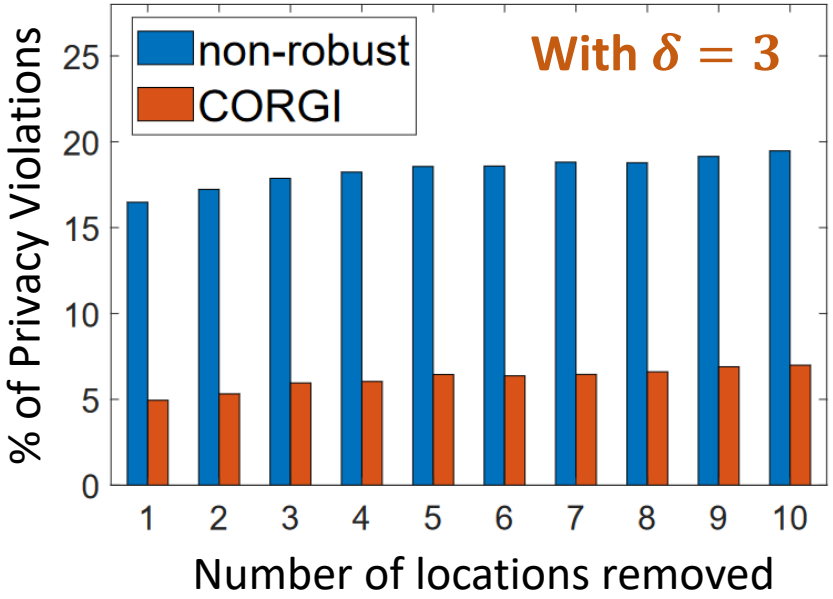
Experimental Results

Utility impact



- Quality loss **decreases as higher ϵ** implies weaker constraint/privacy
- Quality loss **increases as higher δ** introduces higher privacy budget

Privacy impact



Number of violations in **baseline is ~6X compared to CORGI**.

Takeaways

User Customization leads to weakening privacy guarantees of obfuscation functions

CORGI supports customization of location obfuscation functions while preserving their Geo-Ind guarantees

Future work: 1) enable user customization with improved utility; 2) support sharing of trajectories