IMMoA 2012 Workshop

Privacy Preservation for Location-Based Services Based on Attribute Visibility

Masanori Mano, Xi Guo, Tingting Dong, Yoshiharu Ishikawa

Nagoya University

Outline

- Background
- Motivation
- Related work
- Overview of the approach
- Anonymization algorithm
- Experimental evaluation
- Conclusions and future work

Background

Location-Based Services (LBSs)

- LBSs are useful and popular

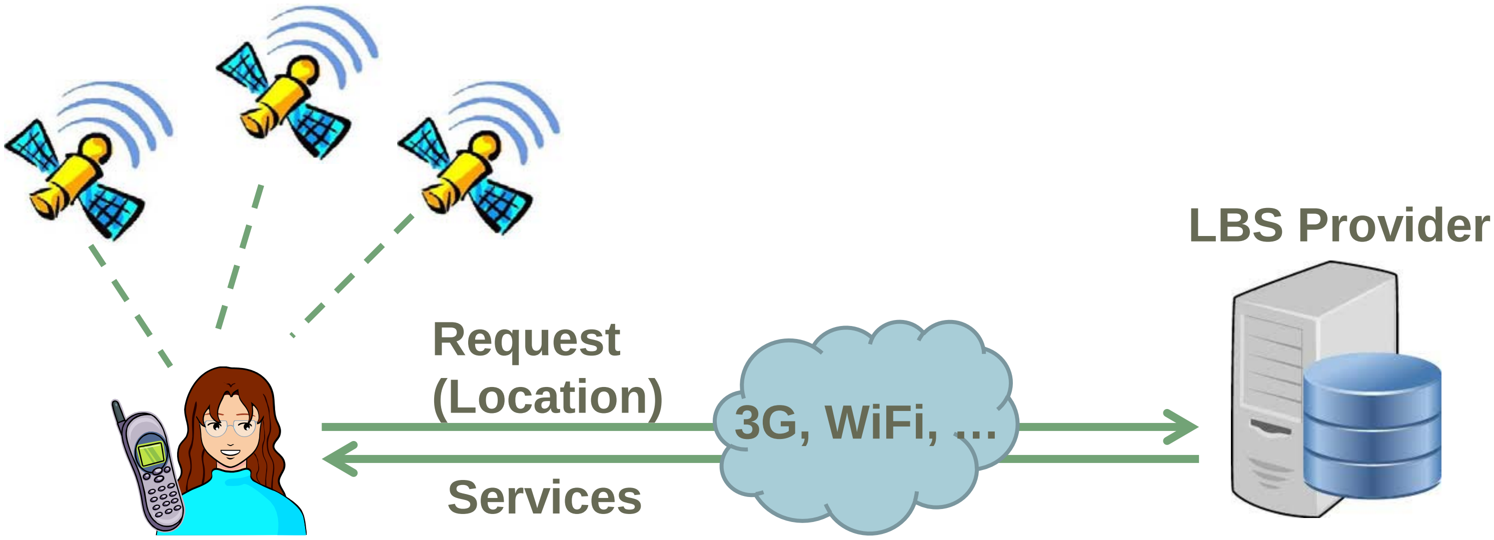
  Provide services to mobile users according to their geographical locations

  - Show nearby cafés, gas-stations, restaurants. … foursquare
  - Compute the best route to the destination Google Maps
  - Send coupons provided by nearby restaurants O_2

Japan
Yamagata
Sendai
Niigata
Fukushima
Nagano
Maebashi
Tokyo
Nagoya
Mito
B
Kofu
A
Chiba
Totton
Kyoto
Shizuoka
Yokohama
Tsu
Osaka
Wakayama

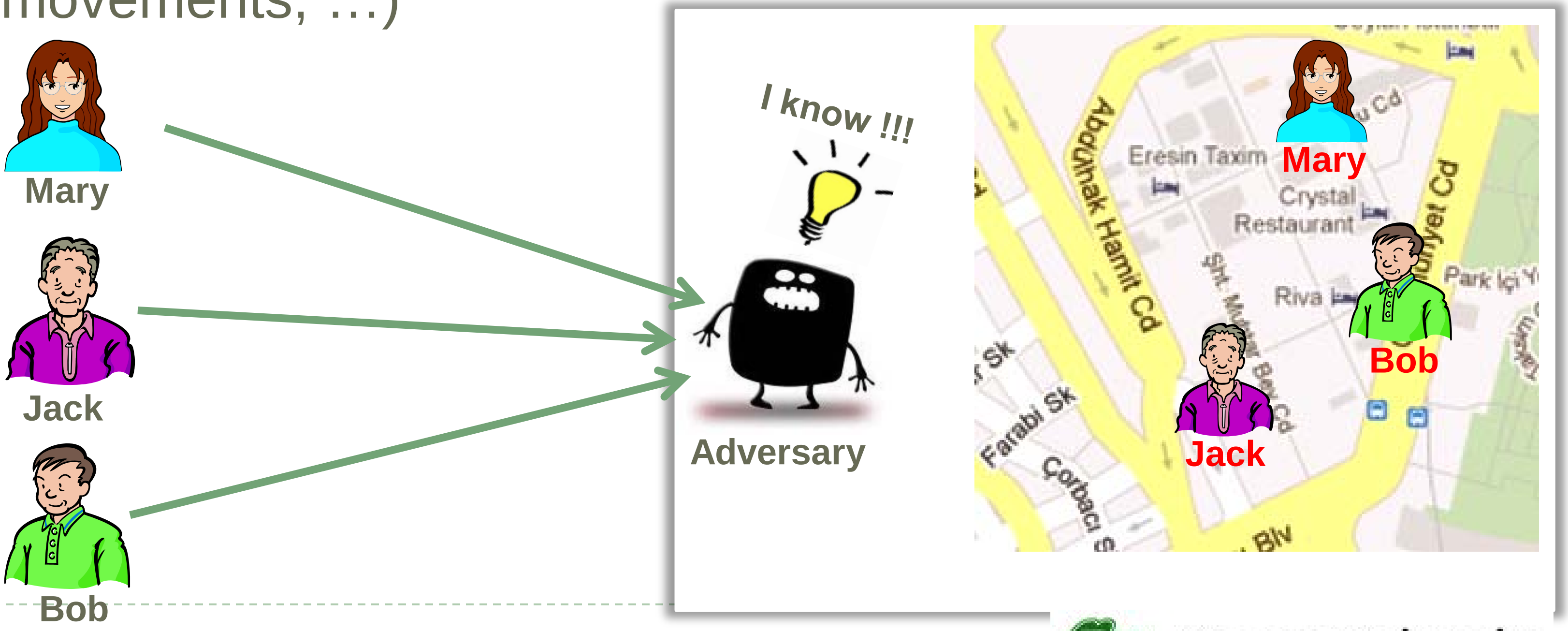
Technologies Supporting LBSs

- Positioning technology: obtain users' locations
  - Example: GPS chips/satellites, cellphone triangulation, …
- Networking technology: access to Internet everywhere
  - Example: 3G, WiFi, …
- Database technology: develop colorful applications



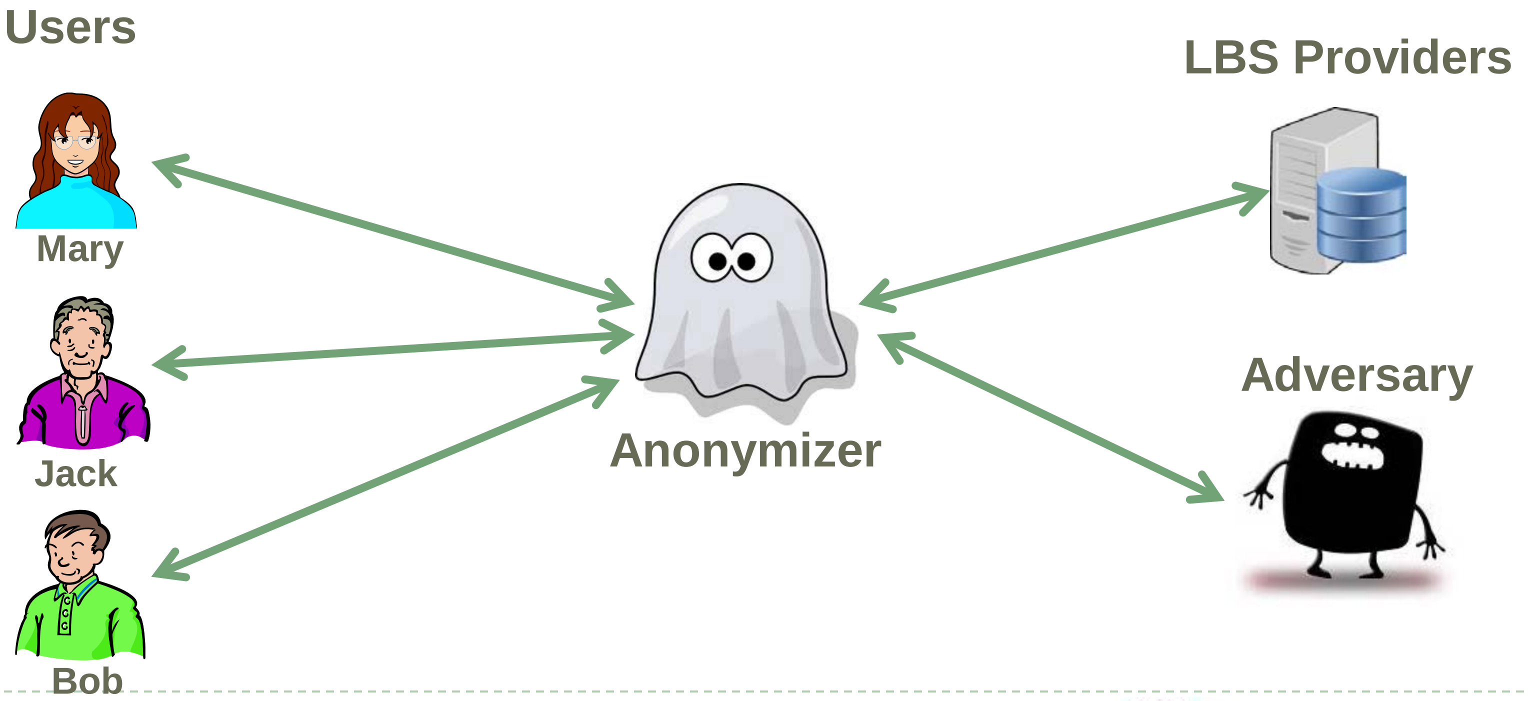
Privacy Issue

- However, the LBS providers might be un-trusted or even adversaries
  - Identity (E.g., name, phone number, IP address, …)
  - Sensitive location (E.g., home, night club, clinic, …)
  - Malicious usage (E.g., keep and sell users’ logs, track users’ movements, …)



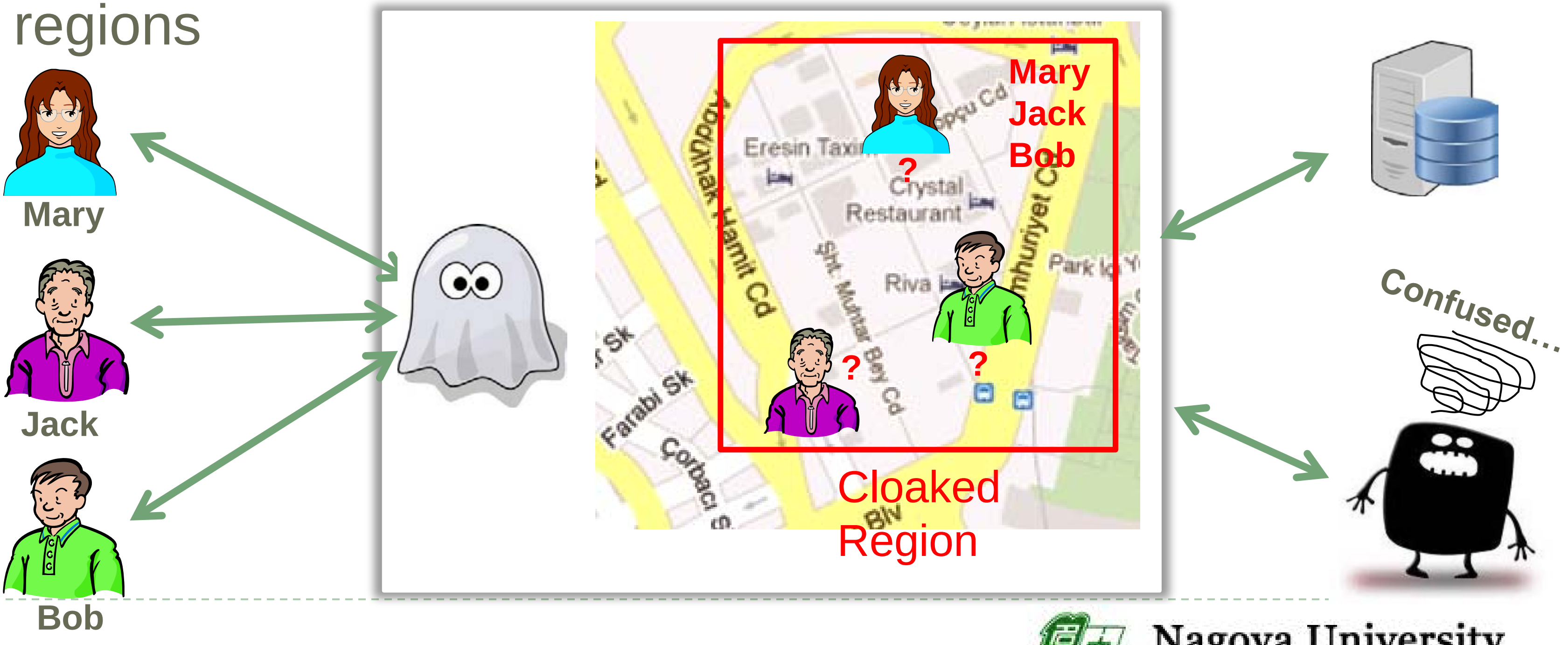
Protect Privacy

- Anonymizer, a trusted third party server
  - Place in-between users and LBS providers
  - Protect privacy by anonymizing users
  - Spatial cloaking [MobiSys03, VLDB06, WWW08]



Spatial Cloaking

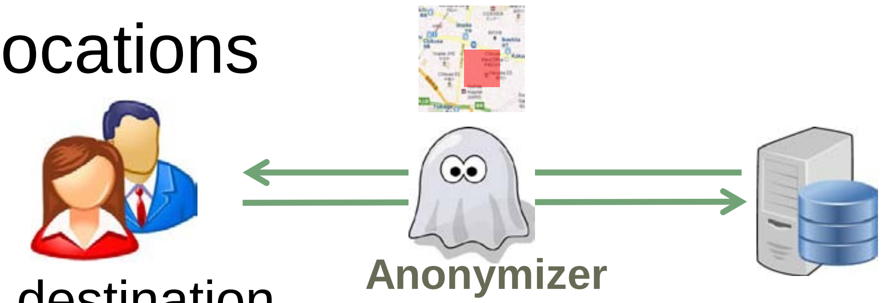
- Anonymizer groups *k* near users and send the group information to LBS providers
  - Prevent the adversary from identifying an individual with probability above 1/*k*
  - Guarantee service quality by limiting the size of cloaked regions



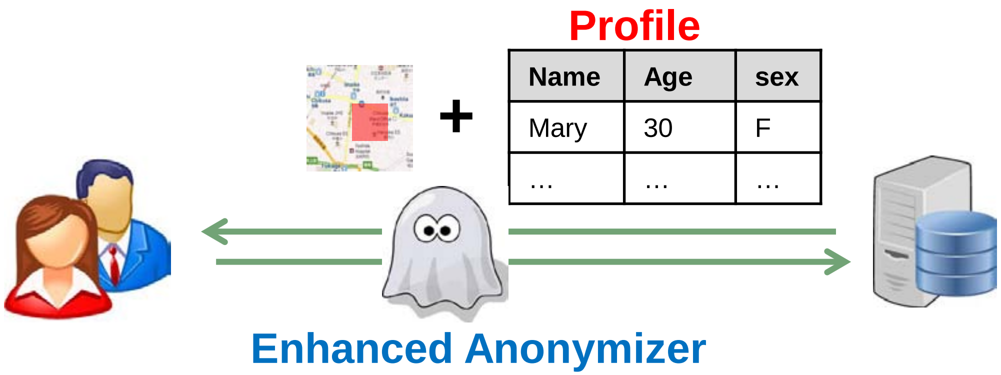
Motivation

Personalized LBSs

- LBSs typically utilize user locations
  - Applications
    - Show restaurants nearby
    - Compute the best route to the destination
  - Protect privacy
    - Spatial cloaking



- Personalized LBSs utilize both locations and profiles
  - Profile: age, sex, occupation, … .
  - Applications
    - Mobile shopping
    - Mobile advertising
  - Protect privacy ?



Profile

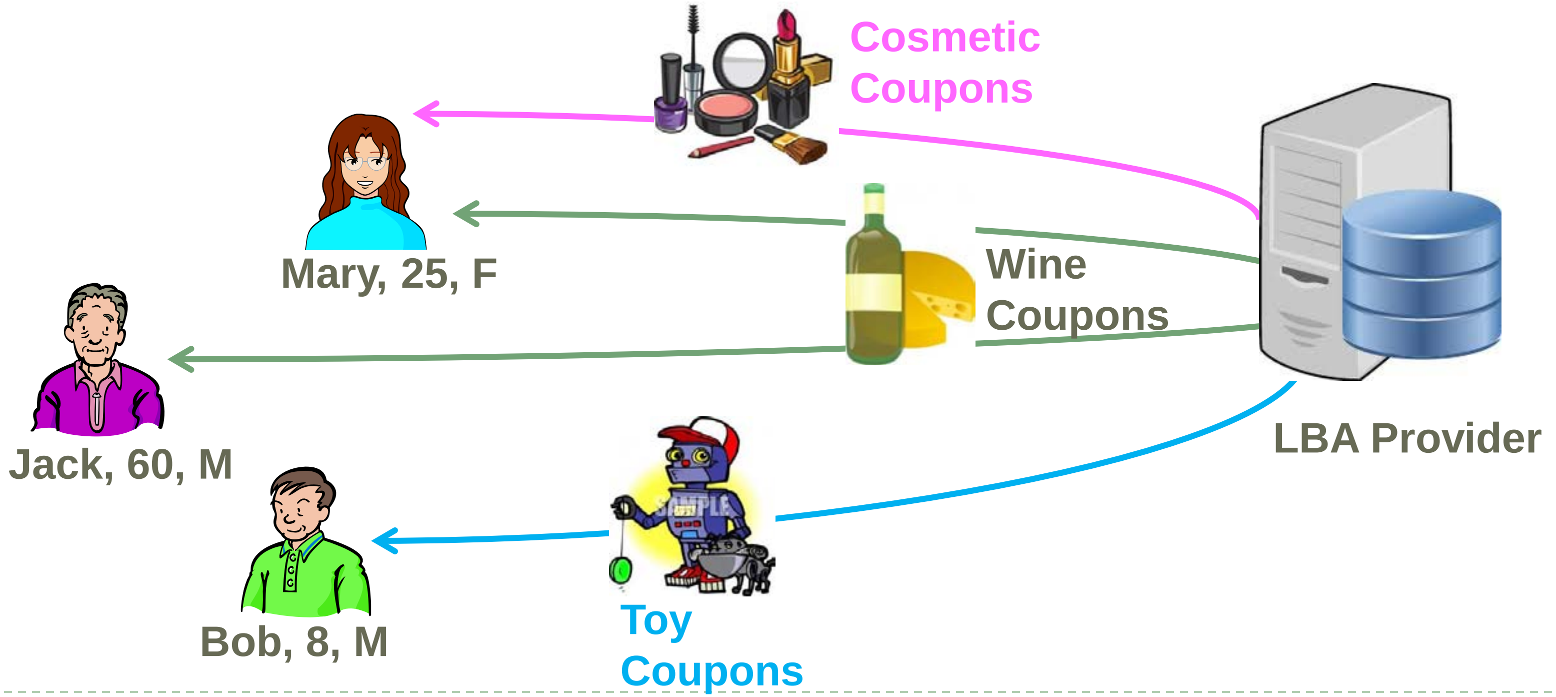
Name	Age	sex
Mary	30	F
…	…	…

Personalized LBS Example

- Location-based advertising (LBA)

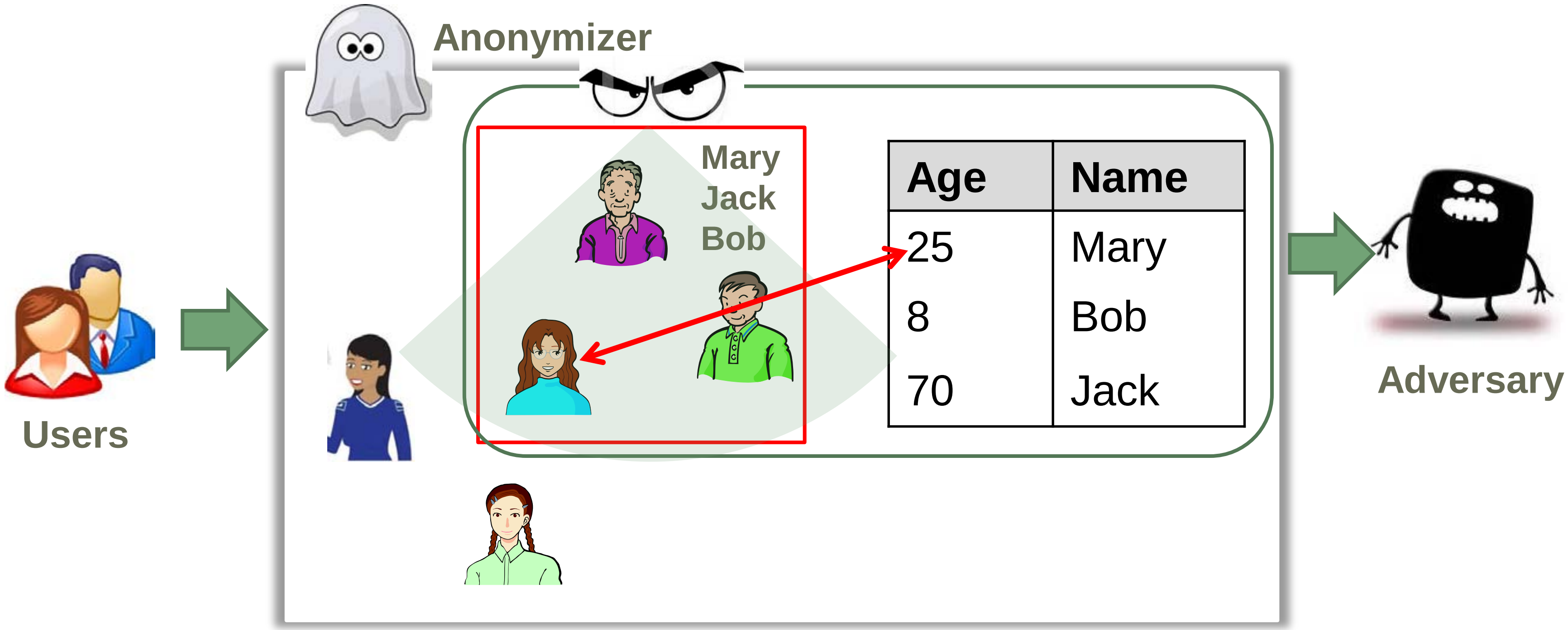
  Provide local advertisements to appropriate persons

  - Use location information to attract nearby users
  - Use profiles to avoid spam that make users unhappy



Privacy Issue in Personalized LBSs (cont.)

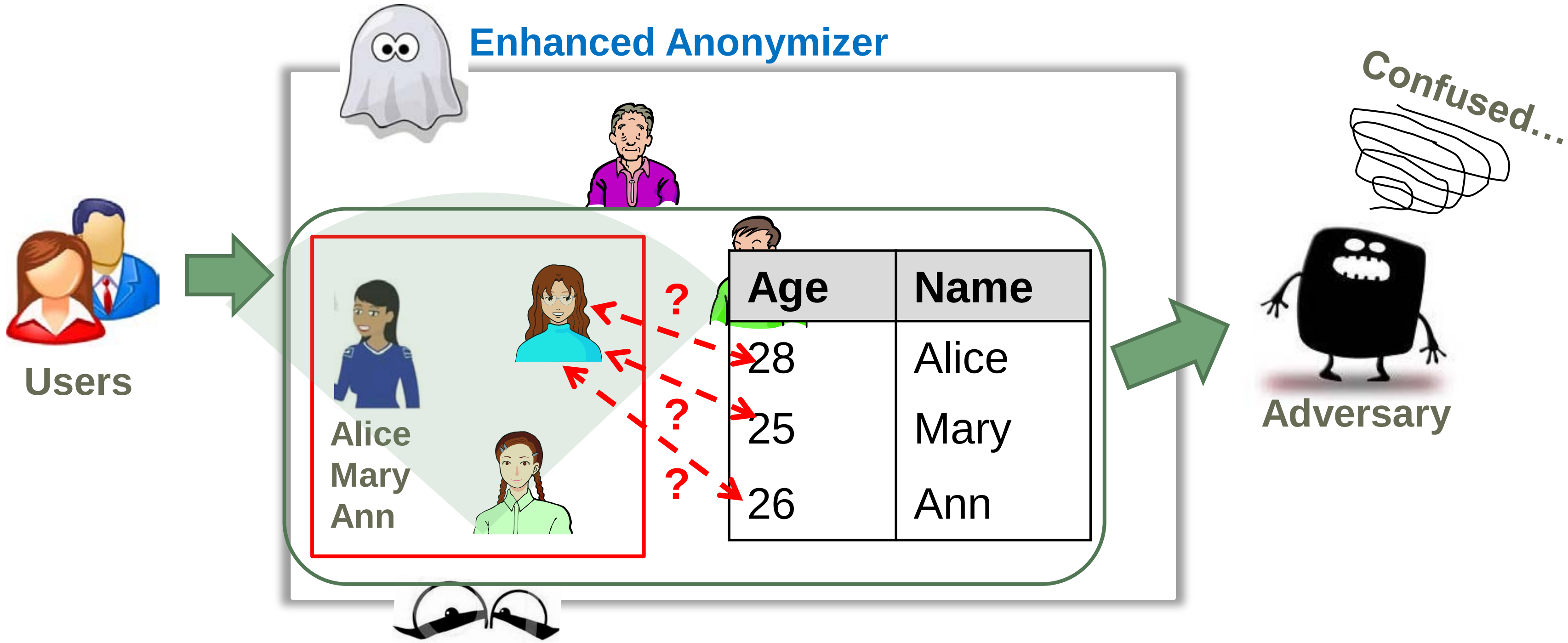
- However, the adversary can distinguish users
  - Associate users with profiles by watching the target area



Age	Name
25	Mary
8	Bob
70	Jack

Our Idea to Protect Privacy

- Group the near users with similar profiles
  - Reduce the identification probability
  - Guarantee the quality of service (unchanged size of the cloaked region)



Age	Name
28	Alice
25	Mary
26	Ann

Related Work

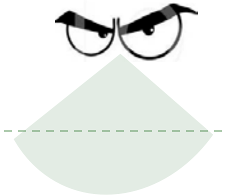
Protect Privacy in LBSs

- In traditional LBSs

  [MobiSys03], [VLDB06], [WWW08], [TMC08]

  - Spatial cloaking
  - Construct cloaked regions that contain near users

- In personalized LBSs [MDM08]
  - Most anonymization methods do not consider users' profiles
  - One exception is [MDM08], but it does not consider the attribute observability
    - Adversaries can associate profiles with users by watching



Personalized Anonymization

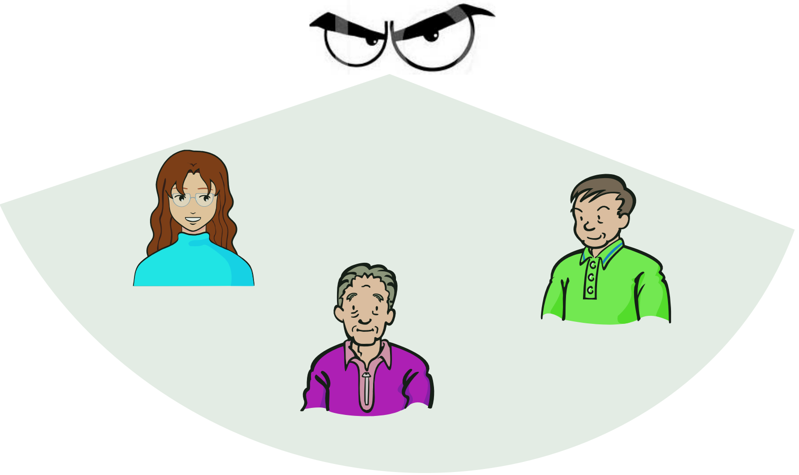
- Users specify their preferences of the attribute disclosure levels [SIGMOD06]
  - Static databases
  - Construct a hierarchical taxonomy for each attribute

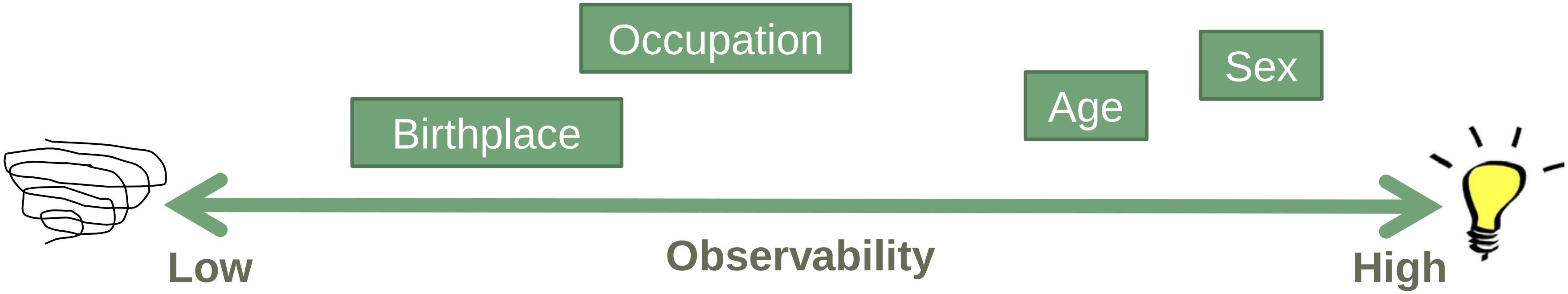
- Our work
  - Spatial databases
    - Service request stream
    - Moving users
  - Hierarchical taxonomy

Details of the Approach

Attribute Observability

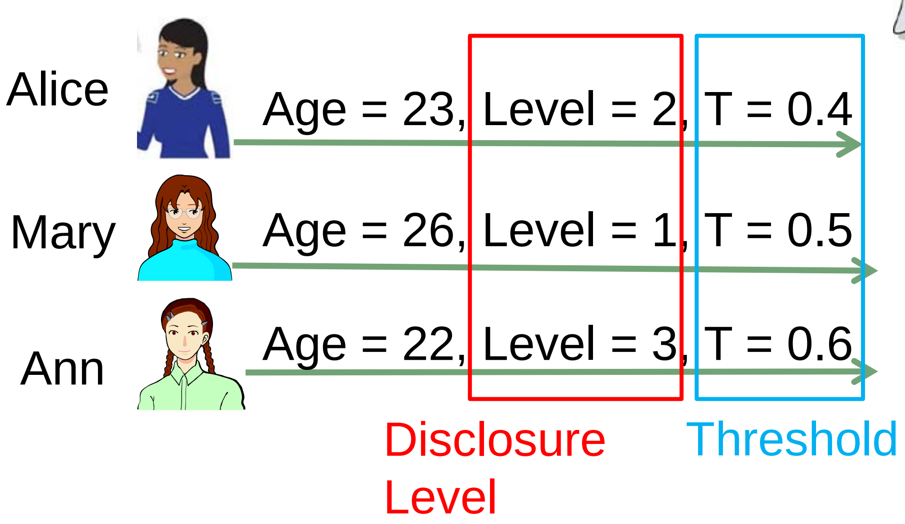
- Observability measures the easiness that adversaries can guess attribute values by observing
  - High observability
    - "Age", "Sex", …
  - Low observability
    - "Birthplace", "Occupation" …





Personalized Anonymization

- Users specify their anonymization preferences
  - Attribute disclosure level (Lower level, disclose less)
  - Identification probability threshold
- According to the preferences, anonymizer construct cloaked regions and the anonymized profiles



Alice
Age = 23, Level = 2, T = 0.4
Mary
Age = 26, Level = 1, T = 0.5
Ann
Age = 22, Level = 3, T = 0.6
Disclosure Level
Threshold

Cloaked Regions

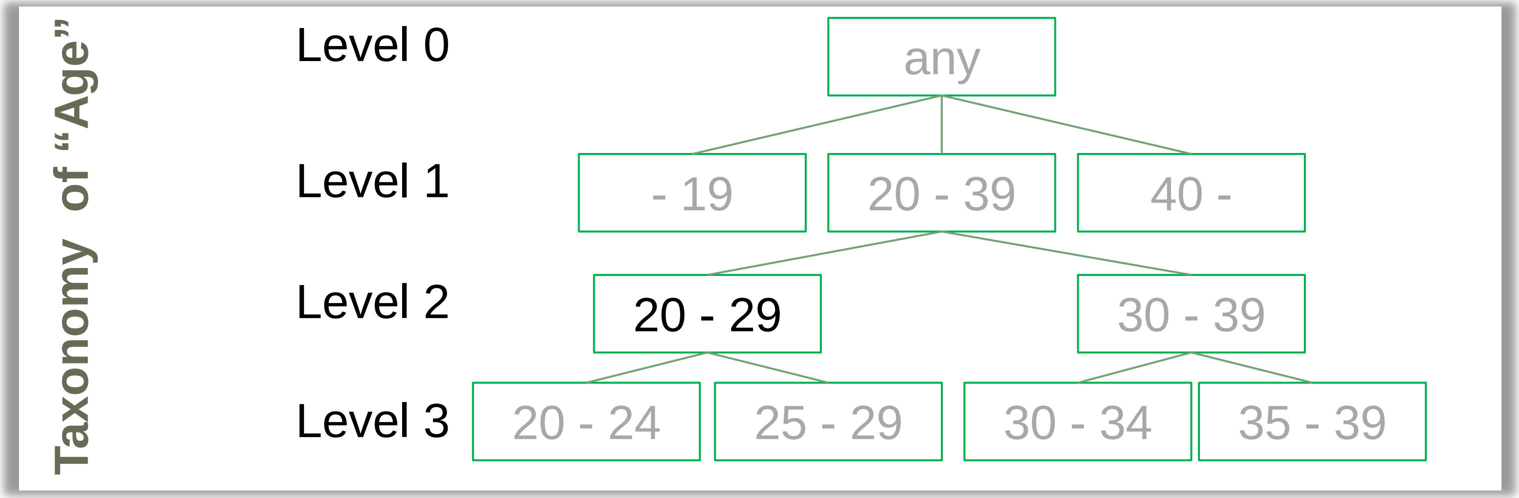
+

Anonymized Profiles

Name	Age	T
Alice	[20-29]	0.4
Mary	[20-39]	0.5
Ann	[20-24]	0.6

Attribute Disclosure Level

- Generalize attribute values by hierarchical taxonomy



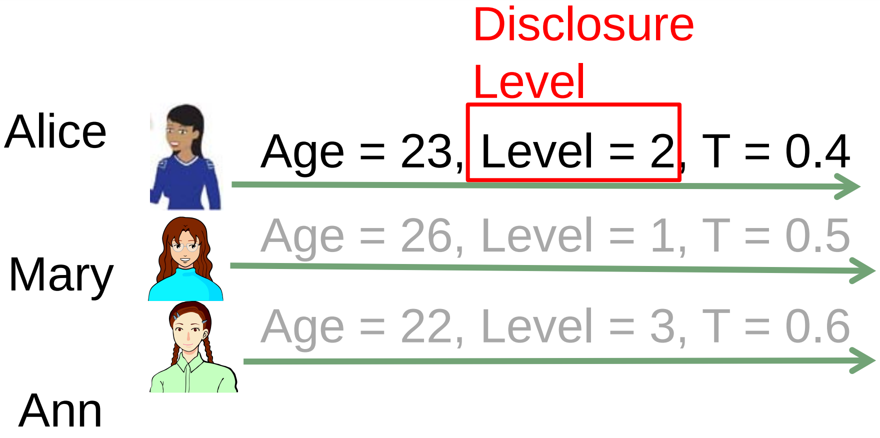
Taxonomy of “Age”

Level 0: any

Level 1: - 19 | 20 - 39 | 40 -

Level 2: 20 - 29 | 30 - 39

Level 3: 20 - 24 | 25 - 29 | 30 - 34 | 35 - 39



Disclosure Level

Alice: Age = 23, Level = 2, T = 0.4

Mary: Age = 26, Level = 1, T = 0.5

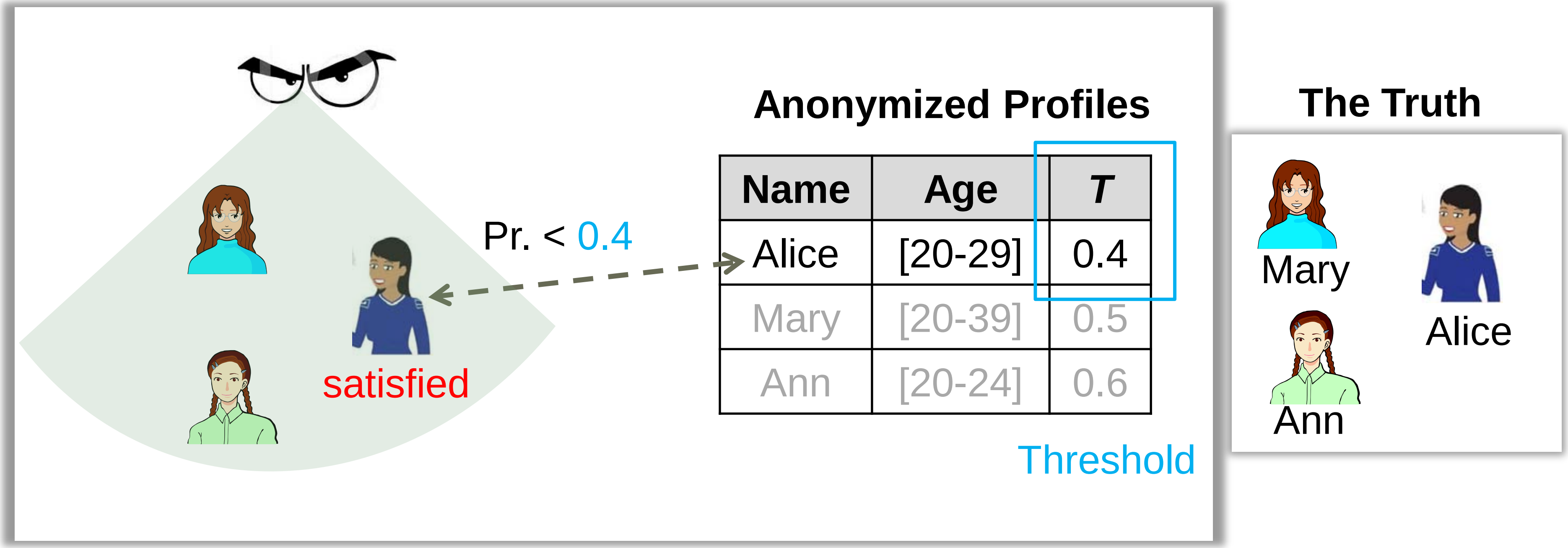
Ann: Age = 22, Level = 3, T = 0.6

Anonymized Profiles

Name	Age	T
Alice	[20-29]	0.4
Mary	[20-39]	0.5
Ann	[20-24]	0.6

Identification Probability Threshold

- Identification probability (*Pr.*)
  - The probability that the individual is identified
- Threshold (*T*)
  - The highest probability permitted by the user



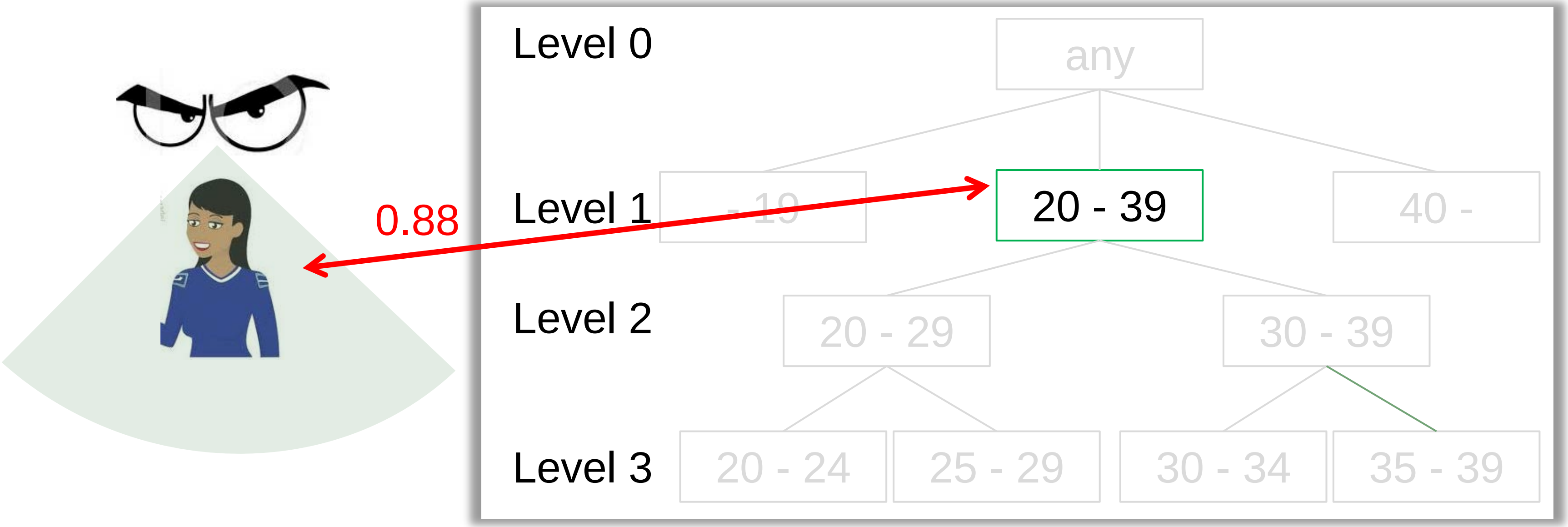
Anonymized Profiles

Name	Age	*T*
Alice	[20-29]	0.4
Mary	[20-39]	0.5
Ann	[20-24]	0.6

The Truth

Matching Degree

- The probability that a user can be related to an attribute value by watching
  - The probability is an empirical value
  - Describe the observability of an attribute value



Matching Degree Table

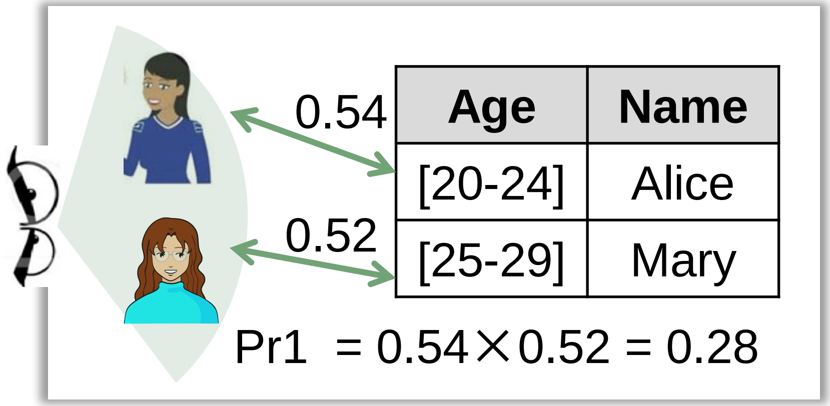
- Record all the matching degrees between users and nodes in the taxonomy tree
  - Anonymizer owns the matching degree table

Matching Degree Table

ID	Level 1	Level 2		Level 3			
	[20-39]	[20-29]	[30-39]	[20-24]	[25-29]	[30-34]	[35-39]
	0.88	0.88	0.00	0.54	0.34	0.00	0.00
	1.00	0.90	0.10	0.38	0.52	0.10	0.00
	0.79	0.79	0.00	0.56	0.23	0.00	0.00
…	…	…	…	…	…	…	…

Calculate Identification Probability (cont.)

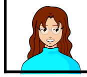
- Calculate the identification probabilities by looking up the matching degree table

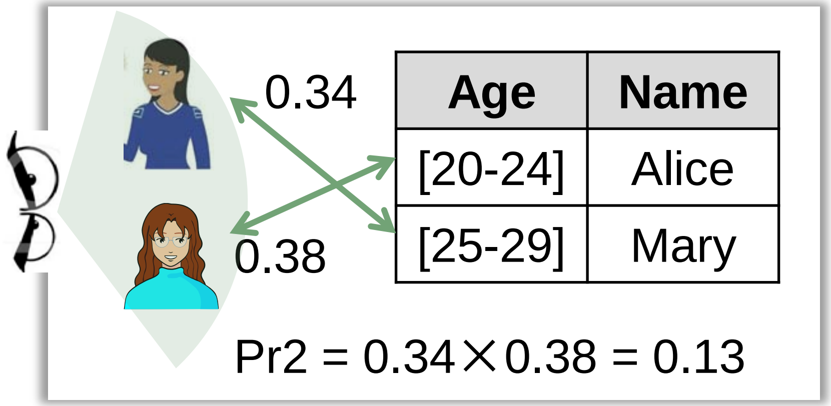


Age	Name
[20-24]	Alice
[25-29]	Mary

Pr1 = 0.54×0.52 = 0.28

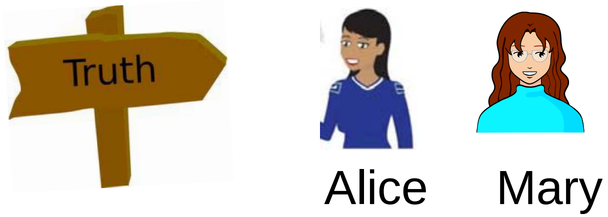
Matching Degree Table

u_i	…	Level 3		
	…	[20-24]	[25-29]	…
	…	0.54	0.34	…
	…	0.38	0.52	…



Age	Name
[20-24]	Alice
[25-29]	Mary

Pr2 = 0.34×0.38 = 0.13

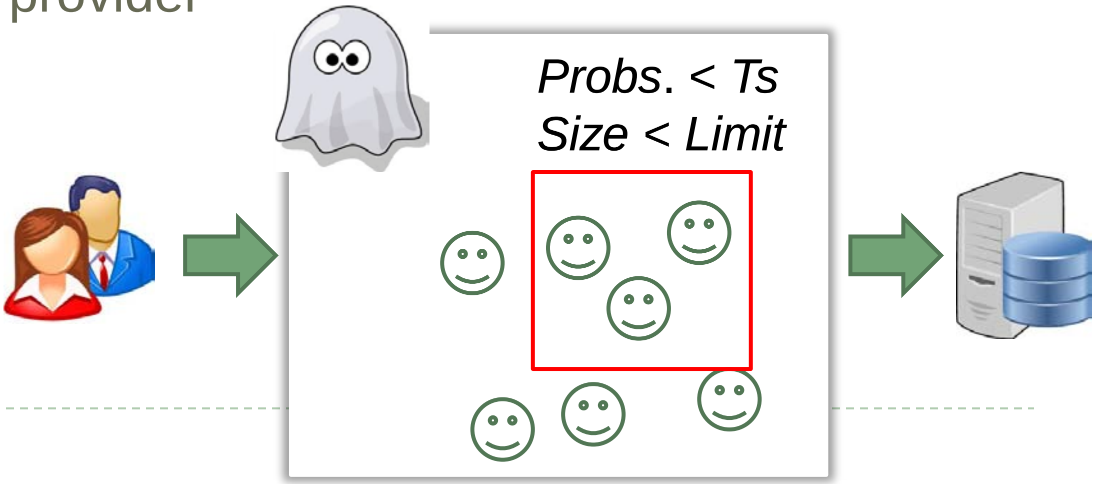


Identification Probability
= Pr1 / (Pr1 + Pr2)
= 0.69

Anonymization Algorithm

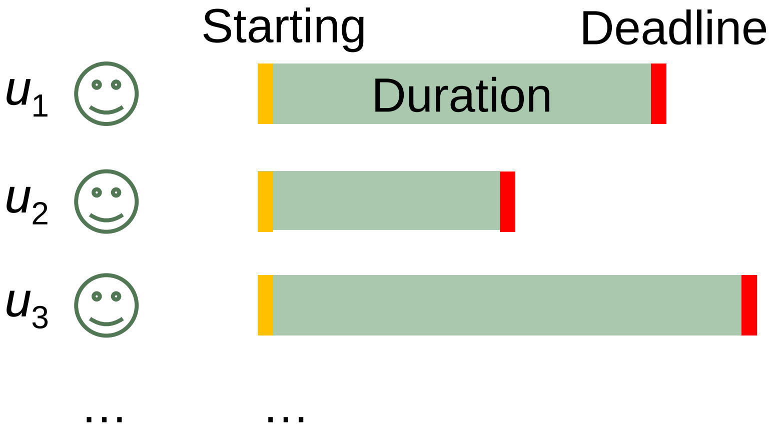
Anonymization Process

- Input (sporadic user requests)
  - Profile (name, age, …)
  - Location (geographical coordinate)
  - Anonymization preference (disclosure level, threshold)
- Construct candidate group
  - The identification probability (*Pr*.) of each user should be lower than the threshold (*T*) permitted by her
  - The cloaked region should be smaller than the maximum size specified by the service provider
- Output
  - Candidate group



Temporal Information of User Requests

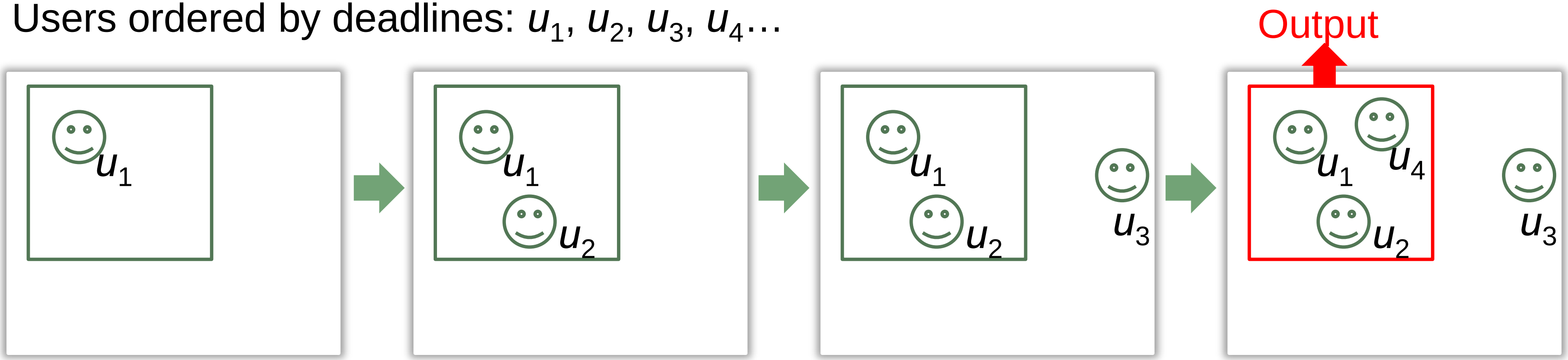
- Starting time
  - When the user requests the service
- Duration
  - How long the user is willing to wait
- Deadline
  - Starting time + Duration



Naïve Approach

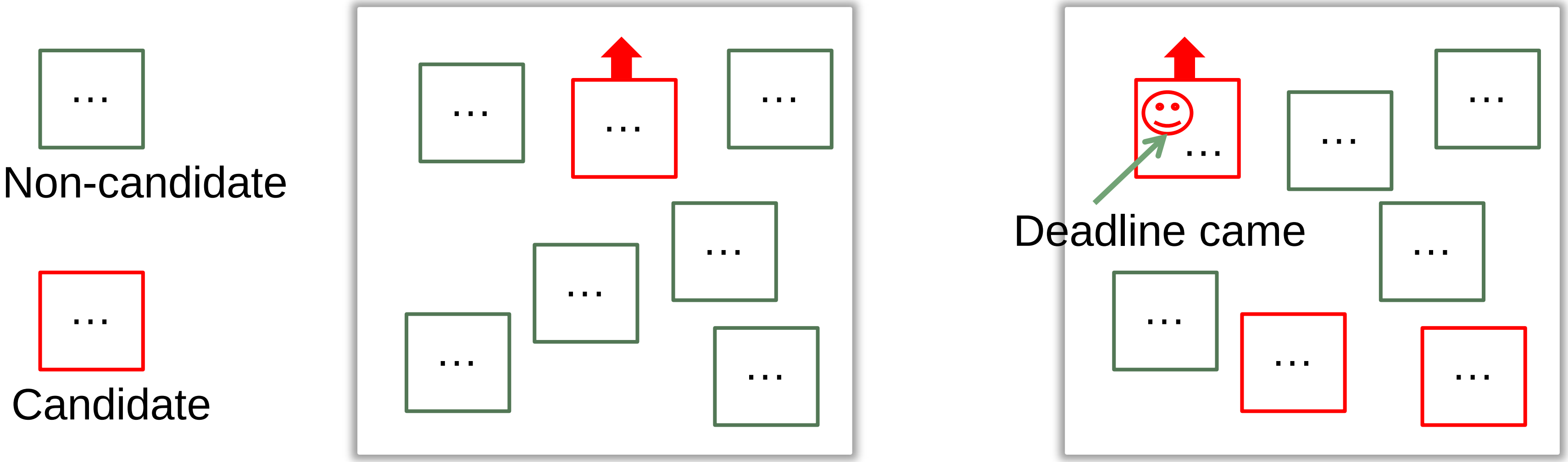
- Process requests in the order of their deadlines
- When a candidate group is constructed successfully, output it immediately

Users ordered by deadlines: u_1, u_2, u_3, u_4…



Optimization Idea

- Wait for the appearance of a better candidate group until the earliest deadline came
  - Six different approaches

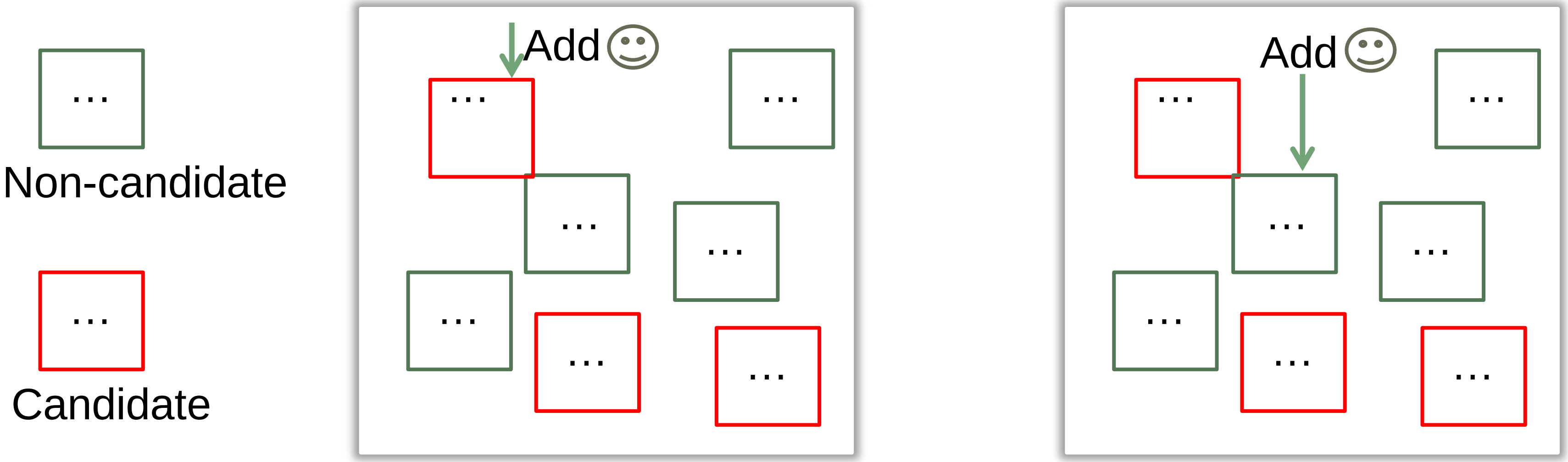


Naïve

Optimization

Optimization Approaches (2/6)

- Deadline-based (candidate first)
  - Add the new user into the existing candidate groups
  - If no candidate group can merge it, construct new groups
- Lazy (non-candidate first)
  - Add the new user into the existing non-candidate groups to make the groups satisfying the thresholds

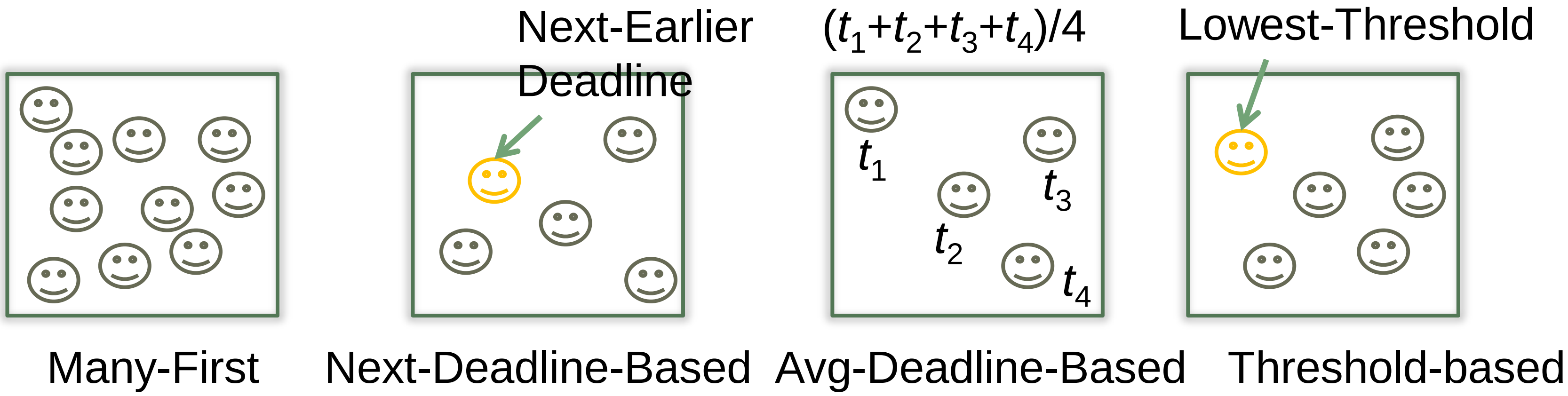


Deadline-Based

Lazy

Optimization Approaches (4/6)

- Many-first: Output the candidate group containing the largest number of users
- Next-deadline-based: Output the candidate group containing the next-earliest deadline user
- Avg-deadline-based: Output the candidate group with the earliest average deadline
- Threshold-based: Output the candidate group containing the lowest-threshold user



Experiments

Settings

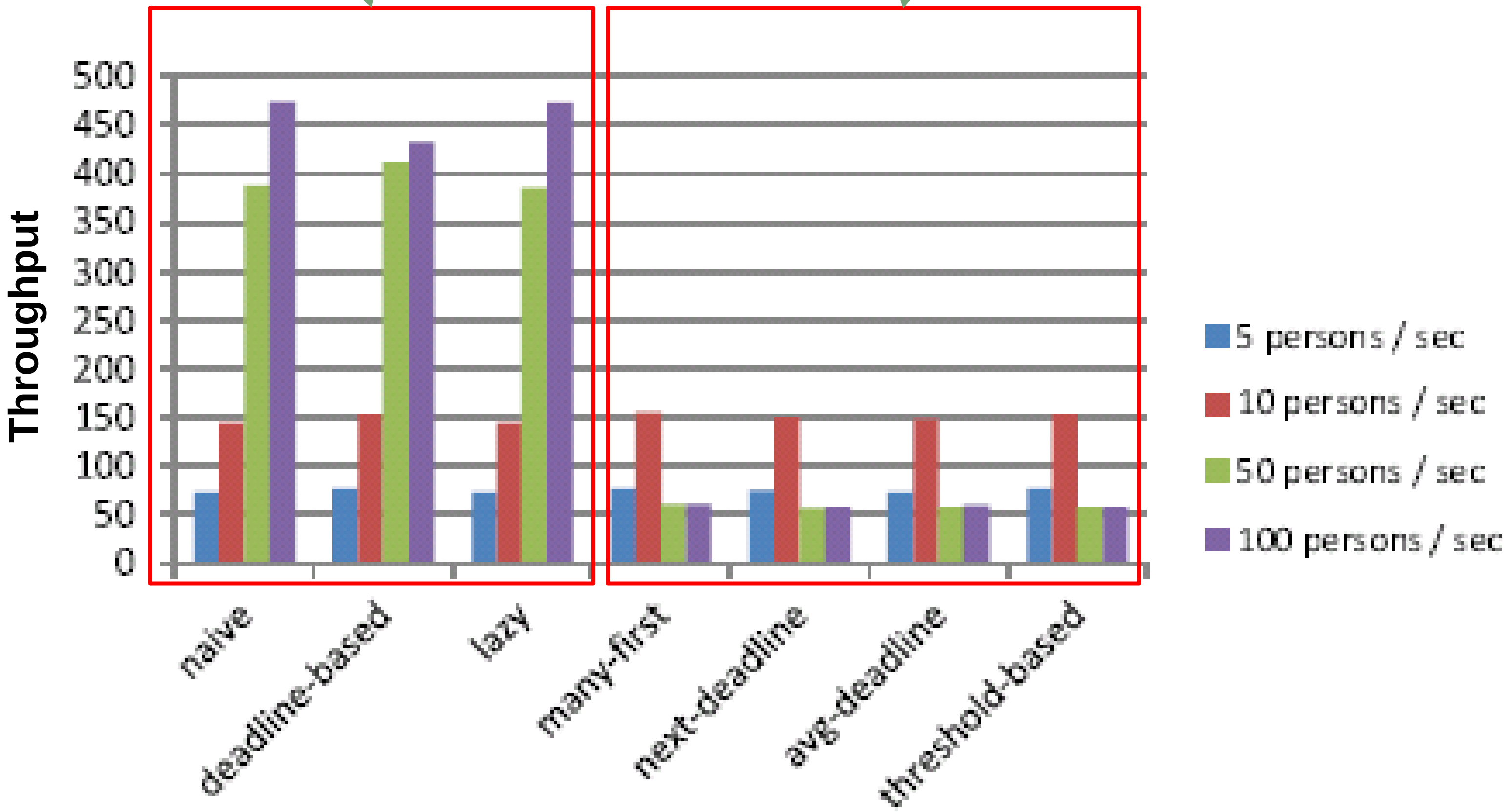
Experimental parameters	Value
Number of users	1000
Request frequencies	10 times/s (default)
Expiration duration (deadline)	10s ∓10% (default)
Used attribute	Age
Age range	[20, 39]
Disclosure level	1, 2, 3
Threshold probability	0.3, 0.4, 0.5 (default)
Cloaked area size limit	1000 × 1000 (default)

Evaluation criteria	Meaning
Throughput	The number of users successfully anonymized
Quality	The average disclosure level

Varying Request Frequencies

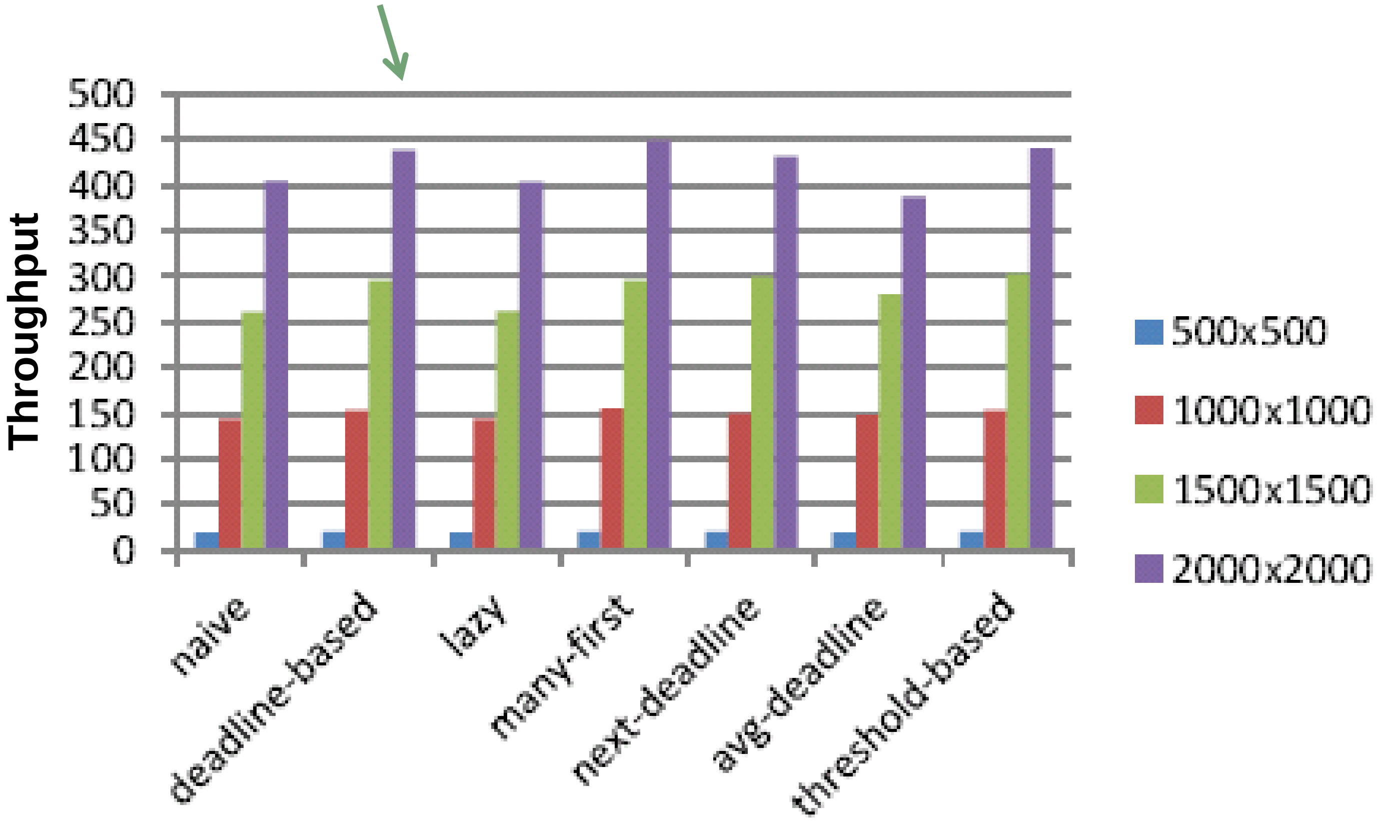
Good throughput with the increase of frequencies

Bad throughput with the increase of frequencies

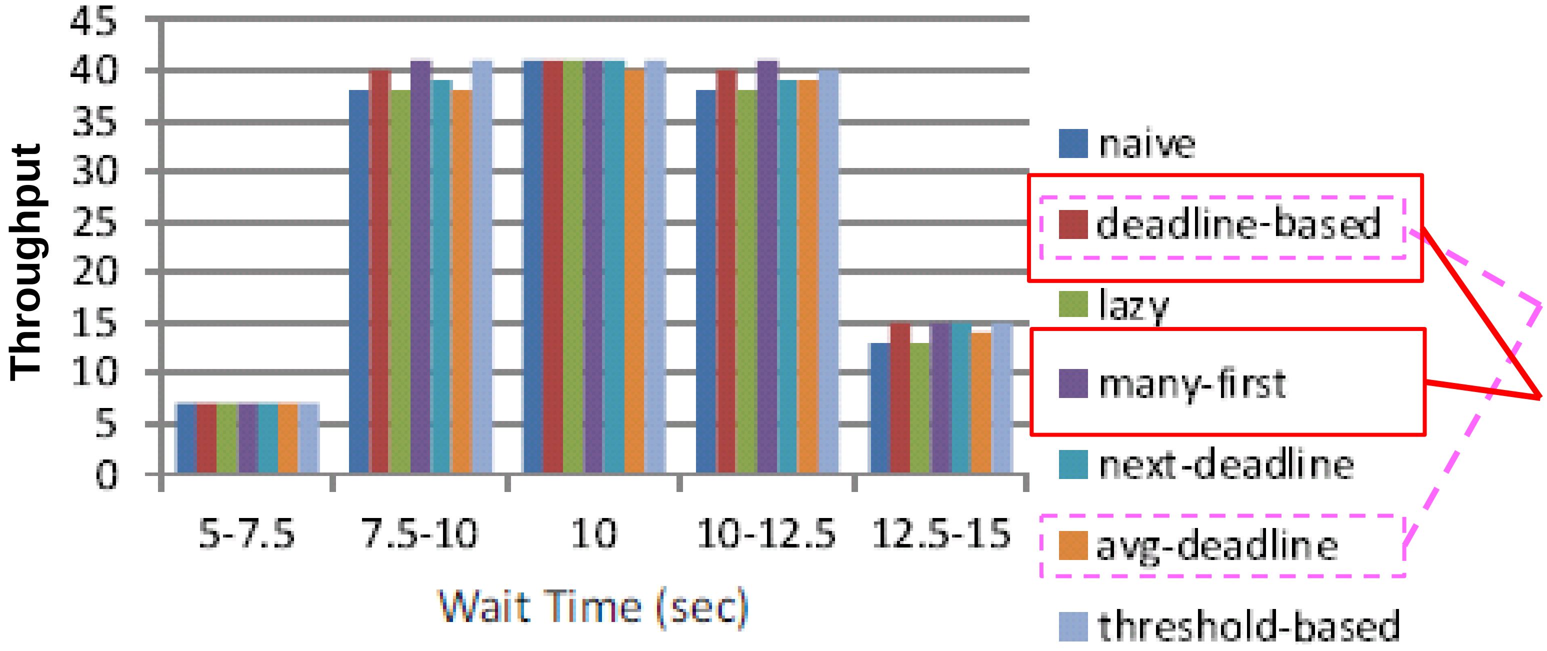


Varying Maximum Size of Cloaked Region

Good throughput with the increase of the size



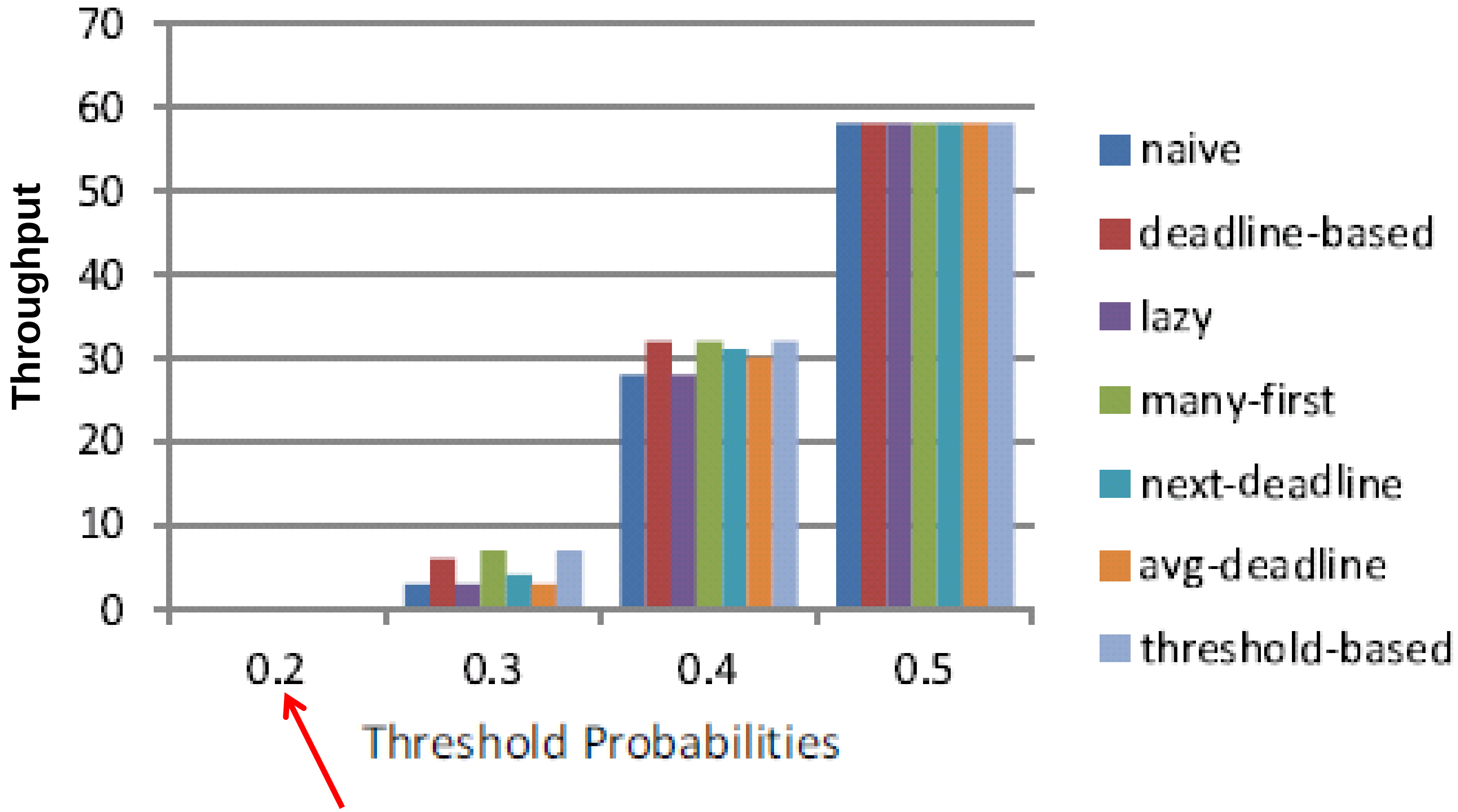
Varying Durations



Anticipate

But, in fact

Varying Probability Thresholds



Failed

Conclusions and Future Work

- Conclusions
  - Propose a new personalized anonymization method for LBSs considering not only locations but also the attribute observability
  - Propose several variations of strategies to implement the new anonymization method
  - Conduct experiments to evaluate the strategies
- Future work
  - Develop high-throughput strategies that can anonymize users with low thresholds

Thank you!