Privacy Preservation for Location-Based Services Based on Attribute Visibility

Masanori Mano*
Graduate School of Information Science
Nagoya University
mano@db.itc.nagoya-u.ac.jp

Xi Guo
Graduate School of Information Science
Nagoya University
guoxi@db.itc.nagoya-u.ac.jp

Tingting Dong
Graduate School of Information Science
Nagoya University
dongtt@db.itc.nagoya-u.ac.jp

Yoshiharu Ishikawa
Information Technology Center / Graduate School of Information Science
Nagoya University
y-ishikawa@nagoya-u.jp

ABSTRACT

To provide a high-quality mobile service in a safe way, many techniques for *location anonymity* have been proposed in recent years. Advanced location-based services such as mobile advertisement services may use not only users' locations but also users' attributes. However, the existing location anonymization methods do not consider attribute information and may result in low-quality privacy protection. In this paper, we propose the notion of *visibility*, which describes the degree that an adversary can infer the identity of the user by an observation. Then we present an anonymization method which considers not only location information but also users' attributes. We show several strategies for the anonymization process and evaluate them based on the experiments.

1. INTRODUCTION

1.1 Background

In recent years, *location anonymization* has become one of the important topics in location-based services and mobile computing [6]. The issue concerned is that a user should send her location information to receive a high-quality service in general. However, if the service provider is an adversary, the detailed location information may be used for non-intended purposes. In an extreme case, the user's identity may be estimated by combining the location information with additional information sources. The use of location anonymization would solve the problem in some sense, but it may result in the degradation of service quality; an appropriate anonymization method is required.

1.2 Location-based services that use attribute information

For a typical location-based service which only utilizes location information, the conventional notion of location anonymity is effective for privacy protection. However, advanced location-based services may use additional *attribute information* such as user's age, sex, and occupation. For illustrating our motivation, let us consider an example of a *mobile advertisement service*.

In this service, we assume that a mobile user issues a request for an advertisement and it is delivered to an appropriate advertiser. Then the advertiser sends corresponding advertisements to the user. In this sense, the advertisement service is a pull-based service. The matching service (called the *matchmaker*) plays the role of a mediator between users and advertisers, and uses users' attribute information for selecting appropriate advertisers. Since the success of an advertisement is charged by the matchmaker, advertisers would like to perform effective advertisements with low investments. If an advertiser can specify the type of the target users (e.g., young women), then the effectiveness of the advisement would increase.

Figure 1 illustrates the overview of a mobile advertisement service assumed in this paper. The *matchmaker* between mobile users and advertisers is a trusted third party and manages each user's information as her *profile*. As described later, the matchmaker is responsible for anonymization. When a mobile user issues a request for a service (i.e., an advertisement), the matchmaker anonymizes the location and profile of the user and sends them to the advertisers. Then appropriate advertisers send corresponding advertisements to the user via the matchmaker. By the obtained

*Current Affiliation: NTT DOCOMO Inc.

advertisement, the user can receive benefits such as coupons and discounts. In this paper, we focus on the anonymization part in this scenario.

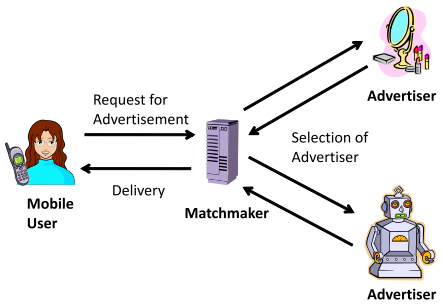

Figure 1: Location-based anonymization framework

1.3 Privacy issues

In the system architecture, we should note that an advertiser is not necessarily reliable and it may be an *adversary*. If the exact location is notified to an adversary, there is a risk that the advertiser identifies the user by watching the location. For this problem, we may be able to apply a conventional location-based anonymization method, but the following problem happens if we consider users' attributes.

Assume that users in Fig. 2 issue requests of advertisements with the order $u_1, u_2, \ldots, u_5$. Their profile information is also shown in the figure. The matchmaker needs to consider tradeoffs between requirements of users, who want to preserve privacy, and advertisers, who want to know the details of user information to improve the service quality. One idea is to apply the *k-anonymization* technique; it groups k users based on proximity. For example, given $k = 3$, we can perform anonymization as $\{u_1, u_2, u_4\}$ as an example. If the matchmaker provides the users' profiles, the received advertiser would know three persons with ages 23, 26, and 38 are requesting advertisements. The problem is that the advertiser easily identifies user with age 38 by watching the target area.

ID	Age
u_1	23
u_2	26
u_3	22
u_4	38
u_5	24

Figure 2: Users and their profiles

If the matchmaker considers not only user proximity but also user profiles, we have another option. Consider the grouping $\{u_1, u_2, u_5\}$. In this case, it is not easy to determine who corresponds to each profile entry. Therefore, this anonymization is better than the former one.

1.4 Research objectives

In this paper, we propose a location-based anonymization method that also considers users' attributes. For this purpose, an important point is whether we can guess each user attribute with an observation. To represent this idea, we incorporate a new criterion called *observability*. In addition, since different users may have different privacy policies, we provide an anonymization method which considers users' preferences.

The preliminary version of the paper was appeared in [7]. In this paper, we revised the problem setting and the method proposed is a totally novel one.

2. RELATED WORK

2.1 Anonymization for location-based services

There have been many proposals on privacy preservation in location-based services. A popular approach in this field is *spatial cloaking*, in which an anonymizer constructs a *cloaked region* which contains target users. For example, [4] uses the notion of *k-anonimity* [9], which is often used in database publishing. The notion of k-anonymity is used in many proposals and there are variations such as the use of graph structure [3] and cell decompositions [1, 8]. In this paper, we extend the idea for our context.

Most of the anonymization methods for location-based services do not consider users' properties. One exception is [10], in which an attribute vector is constructed for each user based on her attribute values. In the anonymization process, a user group is constructed based on the proximity of vectors. The problem of the approach is that it does not consider difference of attributes in terms of observability so that attribute values tend to be over-generalized and results in low-quality services.

2.2 Classification of attributes

In traditional privacy-preservation methods based on k-anonymity [9], user attributes are classified into the following three categories:

- *Sensitive attribute*: It represents privacy information such as disease names.
- *Identifier*: It is used for uniquely identifying individuals such as names and addresses.
- *Quasi-identifier*: Like age and sex attributes, it does not identify individuals directly, but their combinations with other attributes may reveal the identity.

In contrast to the assumption of traditional data publishing, an adversary in our context is not easy to identify individuals using quasi-identifiers and external information (e.g., telephone directory) because it is difficult to determine the candidate users who appear in the target location for the given time. In contrast, visual observation is more problematic in our context. If an adversary watches the target area, he may be able to identify the person who requested the service.

For this problem, we need to enhance the traditional treatment of attributes. In the context of privacy protection in social networks, [5] considered two properties of attributes:

- *Sensitivity*: It describes how the attribute is related to privacy violation. For example, "address" is more sensitive than "birthplace" because the latter is not so useful for identifying people. [5] assumes that sensitivity of each attribute does not depend on a specific user and takes a constant value in the system.

- *Visibility*: It is used as a criterion of how much a user can disclose a detailed value for the attribute. Visibility preference depends on each user and each attribute. For example, different users may have different disclosure policies for "Birthdate".

The notion of visibility cannot be applied to our context. In a location-based service, an adversary can observe some of the user properties even if the user does not want that—it means that visibility is not controllable. In contrast, *observability* of an attribute, which means how much we can estimate the actual value of the attribute from the observation, is more important. We describe the notion in detail later.

2.3 Personalized anonymization

For our context, a personalized privacy-protection mechanism is required because the exposure of user profiles depends on each user's preference. However, most of the existing data anonymization techniques do not consider personalization. [11] proposed a personalized privacy preservation method for a static database. In this method, a hierarchical *taxonomy* is constructed for each attribute. Every user can specify the level of detail in the hierarchy for each attribute and then she can represent her preference. In this paper, we extend the idea considering our context.

3. OVERVIEW OF THE APPROACH

3.1 Objectives of anonymization

We employ the following policies to take trade-off between privacy preservation and service quality.

- *Identification probability*: The probability represents how a user is related with a profile. A user prefers a low identification probability, but an advertiser would expect to high identification probability for the good service. Thus, we assume that each user can specify the *threshold* of the identification probability in her profile. In our approach, the identification probability of an anonymization result should be as large as possible with the constraint that the probability should be smaller than the threshold.

- *Attribute generalization*: Attribute generalization is a fundamental method for protecting privacy. However, excessive generalization results in low service quality, and preference on attribute generalization depends on each user. Therefore, we consider that each user can specify a preferred *disclosure level* for each attribute; the anonymization algorithm should not violate this restriction and tries to group users with similar attribute values.

- *Area size*: A cloaked region with a large size results in a poor service quality. We assume that the system sets the maximum area size for a cloaked region.

3.2 Taxonomy for attribute domain

The taxonomy for an attribute domain is used in the process of generalization. We assume that there exists a hierarchical taxonomy for each attribute domain. Figure 3 shows an example for "age" domain. The root node *any* at level 0 represents all the domain values and the leaf nodes correspond to the most detailed information. Note that Fig. 3 only shows only the descendants of node [20-39] for simplicity. We assume that taxonomies are available for other domains (e.g., ZIP code).

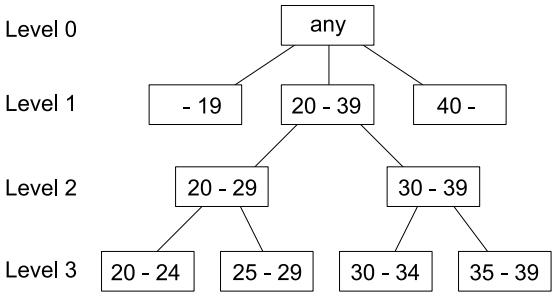

Figure 3: Taxonomy for "Age" domain

We also assume that each user can specify a *disclosure level* for each attribute. For example, consider a user with age 23. The user can specify node [20-29] as her disclosure level for the age domain. If the selected node is near the leaf level, the user can receive more personalized advertisements, but the privacy may not be well protected.

3.3 Profile

Each mobile user constructs a *profile* to represent her preferences on service quality and privacy levels. The trusted matchmaker maintains profiles. An example of user profiles is shown in Fig. 4.

ID	Age		Sex		Threshold Prob.
u_1	23	[20-29]	M	[Any]	0.4
u_2	26	[20-39]	M	[M]	0.5
u_3	22	[20-24]	F	[F]	0.6
u_4	38	[30-39]	F	[Any]	0.5
u_5	24	[20-24]	M	[M]	0.5

Figure 4: Example of profiles

The contents of profiles are as follows:

- *Attribute value*: It represents the attribute value of the user (e.g., Age = 23 for user u_1)

- *Attribute disclosure level*: The level is given by specifying a taxonomy node (e.g., [20-29] for user u_1's Age attribute)

- *Threshold for identification probability*: The user requests that her identification probability should be smaller than this value.

3.4 Attribute observability

Now we introduce a new criterion called *observability*.

Definition 1 (Observability). *Attribute* observability *is a measure of how we can guess its actual value by visually observing the user.* ■

For example, "Sex" is easy to guess, but "Birthplace" is difficult to estimate by an observation. In this case, the observability of "Sex" is higher than "Birthplace". In this

paper, we assume that the observability of an attribute domain (e.g., age) is represented by a probability and takes a system-wide constant value.

We take the following approach for other two properties on attribute privacy.

- A user can specify the disclosure level of each attribute to reflect her preference on *sensitivity*. For example, if a user considers that her age is very sensitive, she can specify "any" node in Fig. 3. Note that a user cannot fully control her sensitivity because an adversary may watch the user directly.
- A user can control *visibility* by specifying the disclosure level of each attribute. If we select the leaf-level node, the visibility is the highest, but it depends on the attribute domain whether the attribute is actually observable.

3.5 Matching degree

To use the notion of observability in an anonymization algorithm, we need to introduce a method to measure the observability of an attribute. We take the following approach: we measure the degree considering taxonomy nodes. For example, consider attribute "Age". The attribute value $age = 21$ is highly related with node [20-24], but has little relationship with node [30-34]. We call the degree that user u_i and taxonomy node n_k match their *matching degree* and define it as follows:

$$match(u_i \rightarrow n_k) = \Pr(n_k \mid u_i). \quad (1)$$

When there are K nodes in a level of the taxonomy, the aggregated matching degree is defined as follows:

$$\sum_{k=1}^{K} match(u_i \rightarrow n_k) = \sum_{k=1}^{K} \Pr(n_k \mid u_i). \quad (2)$$

In this paper, we assume that the matchmaker holds the predefined matching degrees between all the combination of attribute values and taxonomy nodes. Figure 5 shows an example. Due to the limited space, we omit the level 0 node [any] and only show some representative nodes.

	$l=1$	$l=2$		$l=3$			
ID	[20-39]	[20-29]	[30-39]	[20-24]	[25-29]	[30-34]	[35-39]
u_1	0.88	0.88	0.00	0.54	0.34	0.00	0.00
u_2	1.00	0.90	0.10	0.38	0.52	0.10	0.00
u_3	0.79	0.79	0.00	0.56	0.23	0.00	0.00
u_4	0.64	0.00	0.64	0.00	0.00	0.11	0.53
u_5	0.97	0.95	0.02	0.51	0.44	0.02	0.00

Figure 5: Matching degrees

In this paper, we assume that each attribute in a profile is independent. Therefore, the total matching degree can be calculated by multiplying attribute-wise matching degrees.

3.6 Identification probability

An *identification probability* is a probability that a user is identified by watching the users in the target area with the anonymized profiles. If the identification probability is lower than the threshold probability specified by the user, we can say that the requirement of the user is satisfied. As described below, an identification probability is calculated using matching degrees.

3.6.1 Computing identification probability for two users

We first consider a simpler case when there are two users (u_1, u_2) and their anonymized profiles are given as Fig. 6. Note that an adversary does not know which user corresponds to which of the profile entries. Therefore, the adversary should consider two cases $(u_1 : p_1, u_2 : p_2)$ and $(u_1 : p_2, u_2 : p_1)$. Clearly, the following equation holds:

$$\Pr(u_1 : p_1, u_2 : p_2) + \Pr(u_1 : p_2, u_2 : p_1) = 1. \quad (3)$$

pid	Taxonomy Node
p_1	[20-24]
p_2	[25-29]

Figure 6: Anonymized profiles

For computing the probability, we consider the following idea. We play a dice for each user u_i. A dice has a face corresponding to each taxonomy node and its occurrence probability obeys the matching degree. In this example, we play two dices for u_1, u_2 at the same time and there are four patterns of the results: $(u_1 : p_1, u_2 : p_1)$, $(u_1 : p_1, u_2 : p_2)$, $(u_1 : p_2, u_2 : p_1)$, and $(u_1 : p_2, u_2 : p_2)$. The occurrence probability of $(u_1 : p_1, u_2 : p_2)$ is calculated as

$$\Pr(p_1|u_1) \times \Pr(p_2|u_2) = 0.54 \times 0.52 = 0.281, \quad (4)$$

and the probability of $(u_1 : p_2, u_2 : p_1)$ is given as

$$\Pr(p_2|u_1) \times \Pr(p_1|u_2) = 0.34 \times 0.38 = 0.129. \quad (5)$$

Since $(u_1 : p_1, u_2 : p_1)$ and $(u_1 : p_2, u_2 : p_2)$ are prohibited patterns (one profile entry does not correspond to multiple users), we omit when these patterns occur. Thus, the identification probabilities are given as

$$\Pr(u_1 : p_1, u_2 : p_2) = \frac{0.281}{0.281 + 0.129} = 0.69 \quad (6)$$

$$\Pr(u_1 : p_2, u_2 : p_1) = \frac{0.129}{0.281 + 0.129} = 0.31. \quad (7)$$

3.6.2 Computing identification probability for general case

The basic idea is similar to the former case. For example, if the number of users is three, we should consider six combination patterns.

For the anonymization, we need to consider an identification probability of each user. Consider users u_1, u_2, u_3 and profiles p_1, p_2, p_3 are given. User u_1 is only interested in her identification probability is lower than the specified threshold and does not care the identification probabilities of u_2 and u_3. As an example, the probability that user u_1 and profile p_1 is related with is calculated as

$$\begin{aligned}\Pr(u_1 : p_1) = {} & \Pr(u_1 : p_1, u_2 : p_2, u_3 : p_3) \\ & + \Pr(u_1 : p_1, u_2 : p_3, u_3 : p_2). \end{aligned} \quad (8)$$

In the following, we use the term *identification probability* in this sense.

4. ANONYMIZATION ALGORITHM

Table 1 shows the symbols used for describing the algorithm. The algorithm consists of two components: profile generalization and user group construction.

Table 1: Symbols and their definitions

Symbol	Definition
u_i	Mobile user
p_j	Profile
n_k	Taxonomy node
u_q	User who requested an advertisement
$u_q.t$	The time when u_q issued a request
$u_q.e_t$	Request duration time for u_q
$u_q.th$	Threshold probability of u_q
U_R	Set of users in a cloaked region
$\mathcal{U}_C$	Candidate set for U_R
H_U	Priority heap of users who requested advertisements
P_R	Profiles for users in U_R

4.1 Generalization of profiles

For lowering the identification probability for each user, we perform *generalization* of user profiles in a target cloaked region. A profile is, as described above, a set of taxonomy nodes. Since we assume that attributes are independent, the process results in generalization of each attribute in the corresponding taxonomy. Note that the minimum identification probability obtained by generalization is $1/N$ when N users are in the candidate cloaked region.

Algorithm 1 shows the generalization algorithm when N users exist in the cloaked region. $\mathrm{LUB}(n_1, n_2, ..., n_N)$ returns the least upper bound of taxonomy nodes $n_1, \ldots, n_N$ for the target attribute. In Fig. 3 for example, we get

$$\begin{aligned} \mathrm{LUB}([20\text{-}25], [25\text{-}29]) &= [20\text{-}29] \\ \mathrm{LUB}([20\text{-}25], [30\text{-}39], [40\text{-}]) &= [any] \\ \mathrm{LUB}([20\text{-}29], [20\text{-}25]) &= [20\text{-}29]. \end{aligned}$$

GENERALIZE is a function which generalizes n_i to the specified level. Given the least upper bound node and the disclosure level specified by the user, it employs the highest one for the generalization.

Algorithm 1 Taxonomy Node Generalization

```
1: procedure GENERALIZENODE
2:     ñ ← LUB(n_1, n_2, ..., n_N)
3:     for all i such that 1 ≤ i ≤ N do
4:         n'_i ← GENERALIZE(n_i, max(u_i.discl_level, ñ.level))
5:     end for
6:     return {n'_1, n'_2, ..., n'_N}
7: end procedure
```

4.2 User group construction

Algorithm 2 shows the outline of the anonymization process when a user requests a service. At line 2, we insert the user id into priority heap H_U. H_U is ordered by the expiration time, which is the sum of the service request time and the duration time. At line 5, we check whether the bounding box for the grouped users is larger than the maximum limit size. GENERALIZEPROFILE at line 6 performs generalization of profiles. It uses the aforementioned GENERALIZENODE function for node generalization. From line 7 to 12, we check whether the identification probability is lower than the threshold. If it is successful, we remove all S's (the sets that contain the finished users) from the candidate set $\mathcal{U}_C$. Function CHECKEXPIRATION from line 17 is for checking and managing the expireation of user requests.

Algorithm 2 Anonymization

```
1: procedure ANONYMIZE(u_q)
2:     Add user id into H_U
3:         ▷ heap entries are ordered by {u_q, u_q.t + u_q.e_t}
4:     for all U_R such that U_R ∈ U_C do
5:         U_R ← U_R ∪ u_q
6:         if GETMBRSIZE(U_R) ≤ MAX_RECT_SIZE then
7:             P_R ← GENERALIZEPROFILE(U_R)
8:             if ∀u_i ∈ U_R, ∀p_j ∈ P_R, Pr(u_i : p_j) ≤ u_i.th then
9:                 ∀S ∈ U_R, remove S from U_C
10:                return {U_R, P_R}
11:            else
12:                U_C ← U_C ∪ U_R
13:            end if
14:        end if
15:    end for
16: end procedure

17: procedure CHECKEXPIRATION
18:    while true do
19:        {u, deadline} ← POP(H_U)
20:        if deadline > now then
21:            Remove all the sets that contain u from U_C
22:        else
23:            break
24:        end if
25:    end while
26: end procedure
```

We illustrate how the algorithm works using Fig. 2. Assume that the requests are issued with the order u_1, u_2, u_3, u_4, u_5. The process of candidate maintenance in the matchmaker is shown in Fig. 7, where "Ev" represents "Event". We can see that the candidates of cloaked regions increase during the process until the output of the user group $\{u_1, u_2, u_5\}$, which corresponds to a cloaked region. Note that each candidate of cloaked region consists of users, their profiles, and their identification probabilities.

Ev	Candidate Groups
init	$g_0 = \emptyset$
u_1	$g_1 = g_0 \cup \{\{u_1[20\text{-}24] : 1.0\}\}$
u_2	$g_2 = g_1 \cup \{\{u_2[25\text{-}29] : 1.0\},$ $\{u_1[20\text{-}29] : 0.5, u_2[20\text{-}29] : 0.5\}\}$
u_3	$g_3 = g_2 \cup \{\{u_3[20\text{-}24] : 1.0\}\}$
u_4	$g_4 = g_3 \cup \{\{u_4[30\text{-}34] : 1.0\},$ $\{u_1[20\text{-}29] : 1.0, u_4[30\text{-}39] : 1.0\},$ $\{u_2[20\text{-}39] : 0.91, u_4[30\text{-}39] : 0.91\},$ $\{u_1[20\text{-}29] : 0.55, u_2[20\text{-}39] : 0.5, u_4[30\text{-}39] : 0.95\}\}$
u_5	$g_5 = g_4 \cup \{\{u_5[20\text{-}24] : 1.0\},$ $\{u_1[20\text{-}24] : 0.5, u_5[20\text{-}24] : 0.5\},$ $\{u_2[20\text{-}29] : 0.56, u_5[20\text{-}24] : 0.56\},$ $\{u_1[20\text{-}29] : 0.4, u_2[20\text{-}29] : 0.37, u_5[20\text{-}24] : 0.34\}\}$
out	$\{u_1, u_2, u_5\}$ is output. After the output, candidate groups are $g_6 = \{\emptyset, \{u_3[20\text{-}24] : 1.0\}, \{u_4[30\text{-}34] : 1.0\}\}$.

Figure 7: Management of candidates

At the initial state, the candidate set is empty: $\mathcal{U}_C = \emptyset$. As requests arrive, the number of candidates increases, and the algorithm performs profile generalization and identification probability calculation. For example, since the threshold probability of u_1 is 0.4 in Fig. 4, if the calculated identification probability for u_1 is less than 0.4, the anonymization is considered successful for u_1. Note that the maximum size of MBR is defined by the system parameter. Therefore, user u_3, which is far away from u_1 and u_2, is not grouped with them.

In the example of Fig. 2, we cannot get a satisfactory grouping until u_4 arrives. When u_5 requests a service, we can get an anonymization group $\{u_1, u_2, u_5\}$, which satisfies the constraints of identification probabilities. The matchmaker sends the constructed group to an appropriate advertiser and then removes the candidates which include u_1, u_2, and u_5 from $\mathcal{U}_C$. The remaining users u_3 and u_4 should wait the forthcoming user requests.

4.3 Processing strategies and evaluation criteria

The algorithm shown in Subsection 4.2 was the baseline (naive) algorithm. It outputs an anonymized group when a group of users that satisfies the constraints can be constructed. We can consider other option such that we wait the decision for a better grouping until the earliest deadline of users is reached. For selecting an appropriate strategy, it is important how to evaluate an anonymization result. We employ the following evaluation criteria:

- *Throughput*: It is the ratio how many users can be anonymized among all the requested users. A large throughput is preferrable.
- *Quality (Detailedness)*: From the perspective of an advertiser, detailed information is better. For evaluating the detailedness, we use the average level of taxonomy nodes after the anonymization process. For example, assume that we only have "Age" attribute and there are two generaliation results: $r_1 = \{[20\text{-}24], [20\text{-}24], [25\text{-}29]\}$ and $r_2 = \{[20\text{-}24], [20\text{-}29], [20\text{-}29]\}$. Since the levels of [20-24] and [25-29] are three and the level of [20-29] is two, the average levels of r_1 and r_2 are 3 and 2.33, respectively. We can deduce that r_1 is better than r_2 in quality.

5. EXPERIMENTAL EVALUATION

5.1 Setting of experiments

We evaluate the performance of different strategies using synthetic data and simulation-based data. The synthetic data is generated by multiple two-dimensional Gaussians with different centers and variances. The simulation-based data is obtained from the road network of Oldenburg city used in Brinkhoff's moving objects generator [2]. Although the generator generates moving histories of moving objects, we only use their first appearance places since we do not consider movement of users.

The basic settings of simulation parameters are shown in Table 2. In the default setting, we assume that requests are issued based on a Poisson arrival and a new user requests a service in every 1/100 second with the probability parameter $\lambda = 0.1$ (if two users issue requests at the same time, one of the users should wait other one's process). Once a user issues a request, she does not issue another request later. In the simulation, we assume that there is only "Age" attribute in the profiles. The range of age is from 20 to 39, and the matching degrees are set based on Fig. 5 (the lacked entries in the figure are filled). We extend the taxonomy shown in Fig. 3 and selects disclosure levels from 1 (node [20-39]) to 3 (leaf nodes).

Table 2: Basic parameters and their settings

Name	Value
Number of users	1000
Unit time of advertisement request	1/100 s
Advertisement request frequencies	10 times/ s
Used attribute	Age
Range of user age	$[20, 39]$
Disclosure level	$1, 2, 3$
Threshold probability	$0.3, 0.4, 0.5$
Expiration duration	$10\,\text{s} \pm 10\%$
Maximum area of a cloaked region	1000×1000

5.2 Strategies for anonymization

Based on the idea shown in Subsection 4.3, we consider the following seven strategies:

- *Naive*: This is the algorithm in Algorithm 2. We process each user based on the arrival order and then output a group immediately when we can construct it.
- The following two strategies share the same idea. We do not output a constructed group immediately and wait the appearance of a better group.
  - *Deadline-based*: This strategy maintains the candidate groups until the earliest deadline of the current users approaches. If a new user arrives, we try to add this user into the existing candidate groups. If the existing groups cannot merge the user, we try to construct new groups with the existing non-grouped users based on Algorithm 2.
  - *Lazy*: This is similar to *deadline-based*. When we add a new user, *deadline-based* checks the existing groups which satisfy the threshold probabilities first. In contrast, this strategy checks the groups which do not satisfy the threshold probabilities first. The lazy strategy can be said as a variation of *naive* which waits the deadline and cares users who are not in the current candidate groups.
- The following two strategies are also based on the same idea. They maintain all the candidate groups that satisfy threshold probabilities. When the earliest deadline of users approaches, they select one group from the existing candidates. The groups selected and output are different as follows:
  - *Many-first*: The group which has the largest number of users among the groups that contain the user.
  - *Next-deadline-based*: The group which contains the user with the next-earlier deadline. The intuition is that we care the user whose deadline approaches near future.

– *Avg-deadline-based*: The group with the earliest average deadline.

– *Threshold-based*: The group which contains the lowest threshold probability.

5.3 Experiment 1: Users' request frequencies

In this experiment, we change the frequencies of user requests and we check the number of users whose anonymization processes are successful. Increase of request frequency results in a large number of users in the target area, and we can estimate that many groups will be generated. We consider four cases of request frequencies: 5, 10, 50, and 100 times per second. This experiment is done using the synthetic data and we use the parameter settings shown in Table 2. The experimental result is shown in Fig. 8. Three methods *naive*, *deadline-based*, and *lazy* have good throughputs as the increase of request frequency. In contrast, *many-first*, *next-deadline-based*, *avg-deadline-based*, and *threshold-based* have bad performance especially for 50 / 100 times per second. The reason is that the four methods maintain all the candidate groups so that their number rapidly increases as the increase of users.

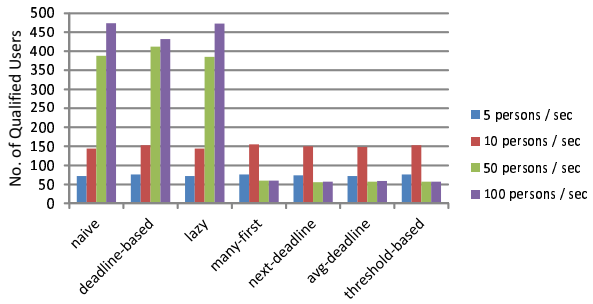

Figure 8: Request frequencies and number of qualified users

Figure 9 shows the number of users of two types: 1) whose process is delayed more than 0.1 seconds due to the foregoing users' processes do not finish, and 2) whose process is expired since the wait time reaches the deadline. We consider four strategies *naive*, *deadline-based*, *lazy*, and *many-first*. We can see that delays happen in *deadline-based* and especially in *many-first*. Note that *next-deadline-based*, *avg-deadline-based*, and *threshold-based* have almost the same result with *many-first*. Since *many-first*, *next-deadline-based*, *avg-deadline-based*, and *threshold-based* contain all the groups which satisfy the threshold probabilities, the increase of the number of candidates results in delays for the requests.

5.4 Experiment 2: Changing maximum area size

We perform experiments by changing the maximum area size of a cloaked region (MAX_RECT_SIZE in Algorithm 2) from 500 × 500 to 2000 × 2000.

Figure 10 shows the number of qualified users for the synthetic data and the uniform attribute distribution. When the maximum size is 2000 × 2000, delays happen only for *avg-deadline-based* and results in the low the number of qualified users. The number of qualified users are large for *many-first*, *deadline-based*, and *threshold-based*. Figure 11 shows how user attributes are generalized. In this figure,

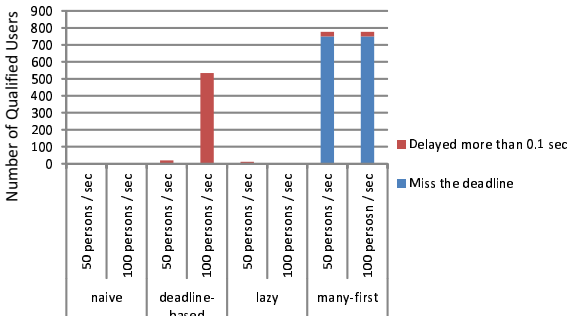

Figure 9: Request frequencies and delays

naive and *lazy* provide reults with good quality in which moderate generalization is performed.

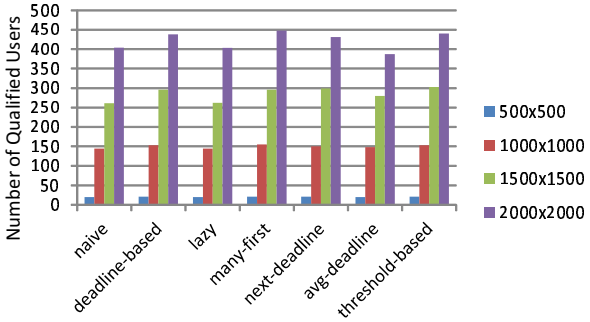

Figure 10: Maximum area sizes and number of qualified users

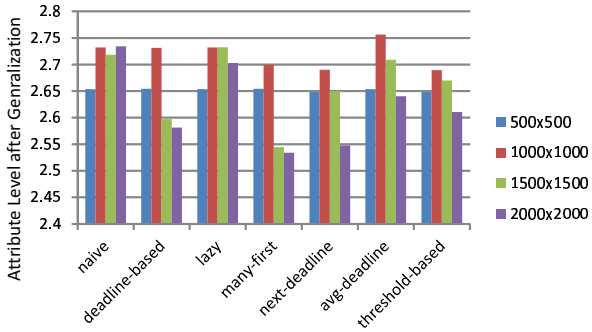

Figure 11: Averaged attribute generalization levels

Additionally, we performed similar experiments using the simulation-based dataset and the correlated distributions, but the trends were similar.

5.5 Experiment 3: Changing user conditions

In this experiment, we observe the behaviors when we change deadline and identification parameters in Table 2 using the synthetic data. First, we change the deadline to $10 \pm 50\%$. Figure 12 shows the qualified users for each deadline setting. We anticipate that *next-deadline-based* and *avg-deadline-based* have good results, but the results are different—*deadline-based* and *many-first*, which do not care deadlines, perform well. Detailed analysis reveals that deadline-based strategies could output users with nearly expiring, but failed to output groups which contain many users.

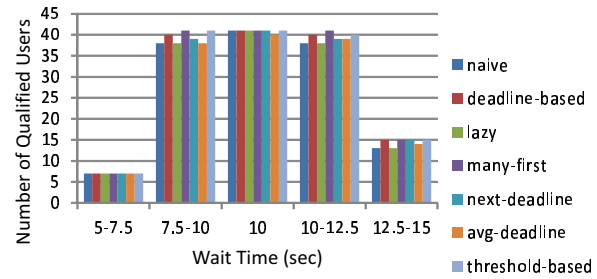

Figure 12: Number of qualified users for each deadline

Next, we change the deadline setting to the original one ($10 \pm 10\%$), but add 0.2 to threshold probabilities. Figure 13 shows the number of suceeded users for each threshold probability setting. In contrast to the case above, *threshold-based*, which tries to output low threshold ones, shows a good result for the threshold setting of 0.3. However, it is worse than *deadline-based* and *many-first*, which do not care thresholds and try to output groups with many users. All the strategies could not make a group for users with threshold settings lower than 0.2.

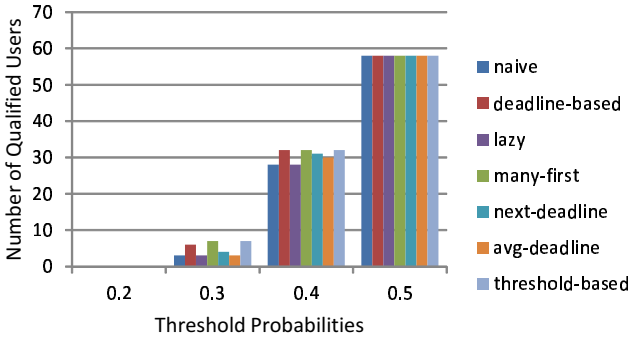

Figure 13: Number of qualified users for each threshold probability setting

5.6 Discussion

In terms of throughputs, *many-first* showed good performance. Compared to the strategies that considers deadline and threshold (*avg-deadline-based*, *next-deadline-based*, and *threshold-based*), the quality of the generated groups were better. However, these four strategies have a common problem when request frequency is high due to the increase of the number of candidate groups. For such a heavy-traffic case, the *naive* strategy might be a better choice since it can achieve high successful rate with low cost. It may be possible to change strategies considering the traffic.

In terms of the availability of cloaked regions, *lazy* was good. In this strategy, since generalization is not performed agressively, the quality of the results was generally good. This is a good property for advertisers. In addition, the strategy can support many users without serious delays.

6. CONCLUSIONS

In this paper, we have proposed a new anonymization method for location-based services. The feature is that we consider not only location information but also user attributes. For that purpose, we defined a new criteria called observability and introduced the notion of a matching degree. We proposed several variations of strategies and evaluated their performance based on the experiments.

Future work includes the development of robust and high-throuput method and a new algorithm which can anonymize users with low threshold settings.

7. ACKNOWLEDGMENTS

This research was partly supported by the Funding Program for World-Leading Innovative R&D on Science and Technology (First Program).

8. REFERENCES

[1] B. Bamba, L. Liu, P. Pesti, and T. Wang. Supporting anonymous location queries in mobile environments with PrivacyGrid. In *Proc. of WWW*, pages 237–246, 2008.

[2] T. Brinkhoff. A framework for generating network-based moving objects. *GeoInformatica*, 6:153–180, 2002.

[3] B. Gedik and L. Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Transactions on Mobile Computing*, 7(1):1–18, 2008.

[4] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proc. MobiSys*, pages 31–42, 2003.

[5] K. Liu and E. Terzi. A framework for computing the privacy scores of users in online social networks. In *Proc. ICDM*, pages 288–297, 2009.

[6] L. Liu. Privacy and location anonymization in location-based services. *SIGSPATIAL Special*, 1(2):15–22, 2009.

[7] M. Mano and Y. Ishikawa. Anonymizing user location and profile information for privacy-aware mobile services. In *Proc. the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks (LBSN '10)*. pages 68–75, 2010.

[8] M. F. Mokbel, C.-Y. Chow, and W. G. Aref. The New Casper: Query processing for location services without compromising privacy. In *Proc. VLDB*, pages 763–774, 2006.

[9] P. Samarati. Protecting respondents' identities in microdata release. *IEEE TKDE*, 13(6):1010–1027, 2001.

[10] H. Shin, V. Atluri, and J. Vaidya. A profile anonymization model for privacy in a personalized location based service environment. In *Proc. MDM*, pages 73–80, 2008.

[11] X. Xiao and Y. Tao. Personalized privacy preservation. In *Proc. ACM SIGMOD*, pages 229–240, 2006.