δ-Calculus: A New Approach to Quantifying Location Privacy[☆]

Lihua Yin[1], Ran Li[1, 2, *], Jingquan Ding[3, 4, 5, *], Xiao Li[3, 4, 5], Yunchuan Guo[2], Huibing Zhang[6] and Ang Li[7]

Abstract: With the rapid development of mobile wireless Internet and high-precision localization devices, location-based services (LBS) bring more convenience for people over recent years. In LBS, if the original location data are directly provided, serious privacy problems raise. As a response to these problems, a large number of location-privacy protection mechanisms (LPPMs) (including formal LPPMs, FLPPMs, etc.) and their evaluation metrics have been proposed to prevent personal location information from being leakage and quantify privacy leakage. However, existing schemes independently consider FLPPMs and evaluation metrics, without synergizing them into a unifying framework. In this paper, a unified model is proposed to synergize FLPPMs and evaluation metrics. In detail, the probabilistic process calculus (called δ-calculus) is proposed to characterize obfuscation schemes (which is a LPPM) and integrate α-entropy to δ-calculus to evaluate its privacy leakage. Further, we use two calculus moving and probabilistic choice to model nodes' mobility and compute its probability distribution of nodes' locations, and a renaming function to model privacy leakage. By formally defining the attacker's ability and extending relative entropy, an evaluation algorithm is proposed to quantify the leakage of location privacy. Finally, a series of examples are designed to demonstrate the efficiency of our proposed approach.

Keywords: Location privacy, δ-calculus, relative entropy.

1 Introduction

With the widespread usage of mobile devices equipped with high-precision localization

[☆]This is an extended version of our conference paper [Ding, Li, Guo et al. (2018)].

[1] Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, 510006, China.

[2] Institute of Information Engineering, CAS, Beijing, 100093, China.

[3] Xinjiang Technical Institute of Physics & Chemistry, Chinese Academy Science, Urumqi, 830011, China.

[4] Key laboratory of Speech Language Information Processing of Xinjiang, Urumqi, 830046, China.

[5] University of Chinese Academy of Sciences, Beijing, 100049, China.

[6] Guilin University of Electronic Technology, Guilin, 541004, China.

[7] Department of Electrical and Computer Engineering, Duke University, Durham, 27708, USA.

[*] Corresponding Authors: Ran Li. Email: liran@iie.ac.cn; Jingquan Ding. Email: dingjq@ms.xjb.ac.cn.

Received: 22 January 2020; Accepted: 28 February 2020.

CMC.doi:10.32604/cmc.2020.09667

capabilities, such as mobile phones [Yin, Guo, Zhang et al. (2019)], intelligent cars [Tian, Gao, Su et al. (2019)], and wearable devices [Tian, Luo, Qiu et al. (2020)], location-based services (LBS) have gained great success in the mobile wireless Internet. LBS (e.g., navigation, point of interest (POI), and motion data publishing) is changing our daily lives at an unprecedented speed. Specifically, users can guide themselves to places that they have never been. With the help of LBS, they also can query nearby POIs with location. Recently, the publishing of running or cycling trajectories has become a new fashion in the circle of friends. Furthermore, trajectories publishing can also help optimize the city resource and prevent traffic congestion. In addition, LBS can be easily integrated in many other fields, such as crowdsensing systems [Li, Sun, Lu et al. (2020)], edge computing systems [Tian, Shi, Wang et al. (2019)], and IoT-Based network [Yin, Luo, Zhu et al. (2020)].

However, as users enjoy the convenience of LBS, location privacy has become a major concern. Location-based Service providers can infer the users' preferences and behavior habits by counting uses' location information and their search history. What's more, adversaries can obtain users' trajectories by monitoring their communication, thus carry out trail, rob or theft of empty houses, which seriously threaten the safety of users' life and property [Yin and Liu (2019)].

Nowadays, a large number of efforts have been spent on preserving location privacy, which can be roughly be divided into two categories: (1) the threat analysis of location privacy, the formalization of related attacks, and the design of the appropriate LPPMs [Zheng, Cai, Li et al. (2017)], and (2) the evaluation and measurement of location privacy [Olteanu, Huguenin, Shokri et al. (2017)]. Beyond that, several efforts have been spent on acquiring the true location of users from the anonymized locations.

In the first category, three schemes, the elimination scheme (ES) [Abul, Bonchi, Nanni et al. (2014); Pandit, Polina, Kumar et al. (2014); Dong and Pi (2018)], the anonymity scheme (AS) [Sweeney (2002); Freudiger, Manshaei, Hubaux et al. (2013)], and the obfuscation scheme (OS) [Xiao and Xiong (2015); Zhang, Zhong, Han et al. (2016)] have been proposed. ES is to confuse the linkage relationship of locations at continuous time series through eliminating the trajectory of the real user, thus, preventing trajectory from being leakage. In the anonymity scheme, Sweeney [Sweeney (2002)] interrupted the connection between the true identity and privacy information. Through this approach, the true identity is protected. In OS, noise is added into the user information and prevents attackers from discovering the relevance between user's location and identities, thus degrading the possibility of location attack [Wang, Yang, Han et al. (2017)]. However, these schemes do not formally and theoretically verify their efficiencies.

To address the above problem, a large number of efforts are spent on adopting (or designing) formal methods to mine vulnerability and improve the protective efficacy of LPPMs [Arapinis, Chothia, Ritter et al. (2010); Guo, Zhang, Zhang et al. (2018)]. Additionally, many of the metrics Emara et al. [Emara, Woerndl, Schlichter et al. (2015); Niu, Li, Zhu et al. (2015)] are proposed to measure evaluate the effectiveness (including accuracy, correctness and certainty) of the LPPMs.

The requirements for adopting formal methods to design LPPMs and evaluate their degree of privacy have been widely recognized. However, in the existing approaches, the design scheme is often separated from evaluation metrics, and this separation might cause that the selected formal tool for evaluating LPPMs might not match the designed LPPMs. As a result, users' location privacy cannot be measured precisely. This makes it harder to guarantee location privacy. Therefore, we should closely combine assessments into the designed LPPMs to guarantee location privacy.

In this paper, we propose a probabilistic process calculus, called δ-calculus, to formalize the LPPMs and measure the privacy level of LPPMs by using the relative entropy. The main contributions of this paper are as follows: Through adding location calculus into π calculus, we propose a δ-calculus to formalize the obfuscation schemes. In detail, a suit of syntax and their semantics is designed to formally describe LPPM. Specially, we design the moving calculus to model nodes' mobility and probabilistic choice calculus to compute the probability distribution of nodes' locations. Two examples show that our proposed calculus can efficiently evaluate location traces. We propose the renaming function to model information leakage. Further, by formally defining the ability of an attacker and extending the relative entropy, we propose the evaluation algorithm to evaluate the degree of location privacy. We use the proposed scheme to evaluate the protection scheme (DUMMY-T) of trace privacy. The results demonstrate that our scheme can accurately quantify location privacy.

The rest of the paper is organized as follows: Related work is introduced in Section 2, and Section 3 presents the probabilistic automata for this paper. We propose the syntax and semantics of our δ-calculus in Section 4. Section 5 provides some evaluation results of our proposed location privacy measuring. Section 6 shows some experiment issues. Finally, we conclude our work in Section 7.

2 Related work

In this paper, we mainly focus on location-privacy protection mechanisms and their metrics methods, so we also discuss the related work in these two aspects.

2.1 Location-privacy protection mechanisms

In general, protection mechanisms of location-privacy can be divided into 3 categories: the elimination scheme (ES) [Abul, Bonchi, Nanni et al. (2014); Pandit, Polina, Kumar et al. (2014); Dong and Pi (2018)], the anonymity scheme (AS) [Sweeney (2002); Freudiger, Manshaei, Hubaux et al. (2013);], and the obfuscation scheme (OS) [Xiao and Xiong (2015); Zhang, Zhong, Han et al. (2016)].

In the aspect of ES, by exploiting the inherent uncertainty of the whereabouts of the moving object, Abul et al. [Abul, Bonchi, Nanni et al. (2014)] designed a co-localization-based LPPMs to eliminate the outlier trajectories of users', and cluster them, thus, enhancing users' privacy. Pandit et al. [Pandit, Polina, Kumar et al. (2014)] proposed a novel server-central framework (called CLOPRO) to generalize a new query content and protected location privacy in continuous LBS by eliminating some attributes from the original query and confusing the temporal link of locations. Dong et al. [Dong and Pi (2018)] presented a frequent-path-based approach (called TOPF) for preserving privacy in trajectory data publishing. In their work, information of infrequent road was removed and all trajectories were divided into candidate groups, and provided a balance between the data usability and data privacy. However, in ES, attackers can use the uneliminated content of the original trajectory to infer the real location of users.

In the anonymity scheme, location privacy is guaranteed by hiding the relationship between users' true identity and their sensitive location. Generally, the anonymity scheme can be divided into two categories: cloaking methods [Sweeney (2002)], and pseudonym change [Freudiger, Manshaei, Hubaux et al. (2013)], where *k*-anonymity is a common method in cloaking techniques where require at least *k* users in the anonymity set. Through *k*-anonymity protection, an attacker cannot distinguish the user from the other *k*-1 users [Roberto and Rakesh (2005); Niu, Li, Zhu et al. (2014)], thus providing anonymity. Except basic *k*-anonymity schemes, several variants are proposed to protect location privacy. Ye et al. [Ye, Li, Xu et al. (2014)] proposed an *l*-diversity-based LPPMs to maintain the heterogeneity of anonymity trajectories and depersonalized user's characteristic. In the aspect of pseudonym change, mix zone where users collectively changed their pseudonyms is one of the frequently used solutions to protect location privacy. By frequently changing pseudonyms, Beresford et al. [Beresford and Stajano (2003)] proposed the Mixzone to prevent the locations they visit from being identified. Besides, many efforts are spent on the combination of *k*-anonymity with pseudonyms. For example, Liao et al. [Liao, Sun, Zhang et al. (2017)] hid user's real trajectory by combining k-anonymity [Pramanik, Lau, Zhang et al. (2016)], Mix-zone [Liu, Zhao, Pan et al. (2012)], MixGroup [Yu, Kang, Huang et al. (2016)] together. Although this kind of work has disrupted the relationship between user ID and query, attackers with background knowledge can still guess the real location of the user.

OS reduced the location accuracy by adding noise into users' information. Using geo-obfuscation, Wang et al. [Wang, Yang, Han et al. (2017)] proposed a location privacy-preserving framework for assigning tasks to protect users' locations. By adding noise, it is difficult for attackers to guess the real location by analyzing the query results. Xiao et al. [Xiao and Xiong (2015)] proposed a systematic solution to preserve location privacy with rigorous privacy guarantee. Their work reduced the sensitivity of a single node transmission by rendering indistinguishability between the real events and the fake ones. To balance the quality of service and privacy protection, the noise should be accurately added, which required to quantify and evaluate the similarity between the obfuscated trajectory and the real trajectory. However, it is an important challenge to quantify them.

2.2 Formal analysis of location privacy and its metrics

Many efforts are spent on formally analyzing and discovering location privacy to decrease the risk brought by the vulnerability of LPPMs. Arapinis et al. [Arapinis, Chothia, Ritter et al. (2010)] used the applied π calculus to analyze the unlinkability and anonymity of identities and demonstrated that the RFID e-passport of French is linkable. In consequence, a person who uses this e-passport can be traced physically by a malicious attacker. Brusó et al. [Brusó, Chatzikokolakis, Hartog et al. (2010)] defined both untraceability and forward privacy, and they formally proved the privacy guarantees of the OSK protocol (an encryption method named after the authors: Miyako Ohkubo, Koutarou Suzuki and Shingo Kinoshita). Dahl et al. [Dahl, Delaune, Steel et al. (2010)] also used the applied π calculus to demonstrate that the cryptographic mix-zones (CMIX) protocol doesn't provide privacy guarantees in specific scenarios. Guo et al. [Guo, Zhang, Zhang et al. (2018)] measured the degree of privacy disclosure by adopting evaluating the privacy leakage level with uncertainty of the adversary's speculating the user's identity by formalizing the proportion of users using the pseudonym algorithm in the system. Liu et al. [Liu, Zhao, Pan et al. (2012)] proposed a metric method to quantify the system's resilience to the side information. An optimization formulation with cost and traffic constraints is presented to model the multiple mix zones placement problem. However, the formal privacy protection model LPPMs is not enough. How to measure the effect of different privacy protection algorithms is an important standard in the design of privacy protection algorithms.

To measure the LPPMs, a large amount of effort has been spent on studying metrics (accuracy, correctness and certainty) to measure location privacy for specific scenarios [Yin, Sun, Wang et al. (2018)]. For example, Emara et al. [Emara, Woerndl and Schlichter (2015)] used the uncertainty to describe the ambiguity of the actual location that can be disguised by posterior distributions. Through this approach, location privacy of a given user can be quantified [Niu, Li, Zhu et al. (2015)]. The validity of the uncertainty metric relies on the knowledge of probability mass believed by an adversary. However, the changes of the assigned probabilities will be influenced by the inaccuracy of context information. As a result, the choice of the attacker may be tainted with uncertainty [Fischer, Katzenbeisser, Eckert et al. (2008)].

Since uncertainty cannot be adopted to accurately evaluate location privacy, many new metric (e.g., inaccuracy) have been proposed, where for an observed location and its distributions estimated by an adversary, the inaccuracy is defined as the discrepancy between the actual posterior distributions of its possible location. Because, the tracking error is taken into account in the inaccuracy metric, the metric is an appropriate approach to evaluating the privacy. Unfortunately, there is still a gap between the current location privacy protection model and the privacy protection effect evaluation algorithm. When the model is not consistent with the rules of evaluation index, it will be difficult to ensure the effect of LPPM. There is an urgent need to design a way to integrate the two in a unified architecture.

3 Probabilistic automata

In our work, we use probabilistic automata to describe the formal semantics of the proposed δ-Calculus. To achieve this goal, we compactly retrospect the probabilistic automata [Herescu and Palamidessi (2000); Deng, Pang and Wu (2006)] in this section, as follows.

Let X be a set of discrete events and pb be a probability function over X, pair (X, pb) is a discrete probabilistic space, that is, $X \rightarrow (0,1]$ with constrain $\sum_{x \in X} pb(x) = 1$. Given a set Y of discrete events, its set of probabilistic spaces are defined on $Y: Prob(Y) = \{(X, pb) \mid X \subseteq Y\}$.

Definition 1. The tuple $M = \{S, s_0, A, \Delta\}$ is a probabilistic automata, where

(1) S is a set of the pre-defined states;

(2) $s_0 \in S$ is the initial state;

(3) A is a set of actions;

(4) $\boldsymbol{\Delta \subseteq S \times Prob(A \times S)}$ is sub-set of Cartesian product between S and $\boldsymbol{Prob(A \times S)}$, and the element of Δ is called transition group.

For a probabilistic automata $\boldsymbol{M = \{S, s_0, A, \Delta\}}$, a tree (denoted as $tree(M)$ is obtained by unfolding M, as follows. Informally, we first mark the root n_0 of $tree(M)$ as s_0; next, if node n of $tree(M)$ is marked as s, then for transition group $\boldsymbol{(s,(X, pb))} \in \Delta$ and any $(a,s) \in X$, then there exists a node n' in $tree(M)$ and an arc from n to n' such that this arc is marked as action a and probability p, where $p = pb(a,s)$. Given probabilistic automata M, the set of nodes of $\boldsymbol{tree(M)}$ is defined as $\boldsymbol{nodes(M)}$.

Generally, there may exist multiple groups for a given state of probabilistic automata. In this case, schedulers will select one group during an automata running. Formally, the scheduler for M can be represented by function ξ: $\boldsymbol{nodes(M) \to Prob(A \times S)}$, defined as follows: $\boldsymbol{\xi(n) = (X, pb)}$ if $(s,(X,pb)) \in \Delta$, where n is marked as s. Informally, each node of the tree of M is assigned to a transition group by the scheduler. Given a probabilistic automata $M = \{S, s_0, A, \Delta\}$ and a scheduler ξ, we can obtain the execution tree of M under the scheduler ξ (written as $etree(M;\xi)$) by removing all the relevant arcs for transitions groups which do not be chosen by ξ. Formally, given a $tree(M)$, its sub-tree (written as $etree(M;\xi)$) is defined as follows: (1) the root node of $etree(M;\xi)$ is the root of $tree(M)$ and the label of the root node of $etree(M;\xi)$ is the same with that of $tree(M)$. (2) Let n be a node of $etree(M;\xi)$, $\xi(n) = (X, pb)$ holds if and only if for any $(a,s) \in X$, there is an arc from n to n' (where n' is in $etree(M;\xi)$) such that this arc is tagged with action a and probability p (where $p = pb(a,s)$). For simplicity, we define a shorthand for the notation, as follows.

$$s\{\xrightarrow[p_i]{\alpha_i} s_i \mid i \in I\}. \tag{1}$$

If and only if $(s,(\{(a_i,s_i) \mid i \in I\}, pb)) \in \Delta$ and for each $i \in I$, $p_i = pb(a_i,s_i)$ holds, where I is an index set. If I is irrelevant, we will replace it with the notation $s\{\xrightarrow[p_i]{\alpha_i} s_i\}_i$.

4 δ-calculus

4.1 Syntax

Communication devices (also called *nodes*) which might be comprised in the mobile wireless Internet run at locations and they may move from locations the other locations. For simplicity, we use notations M or N to denote the set of devices and use notations P or Q to denote the set of processes. The syntax for nodes of δ-calculus is defined as follows.

$$N,M ::= z[|P|](\textstyle\sum_i p_i \cdot loc_i, rad) \,|\, M \,|\, N \,|\, (\nu \cdot loc)N \,|\, 0 \tag{2}$$

In term $z[|P|](\sum_i p_i \cdot loc_i, rad)$, z denotes the node name (e.g., node ID) and rad denotes the *communication radius*. P stands for a process and p_i is a positive probability, that is, $p_i \in (0,1]$ and $\sum_i p_i = 1$; Notation loc_i represents the possible location. $z[|P|](\sum_i p_i \cdot loc_i, rad)$ means that node z runs process P with probability p_i at location loc_i, and the maximum communication distance of node z is rad. M|N denotes that nodes M and N run in parallel. The *restriction* operator is denoted by symbol ν, and $(\nu \cdot loc)$ is used to constrain the range of locations. 0 denotes an inactive node. A process is defined as followed:

$$\begin{aligned} P,Q ::= & \,\overline{S}T.P \mid S(x).P \mid \textstyle\sum_i p_i P_i \mid\, !P \mid (\nu \cdot x)P \\ & \mid \mathrm{MV} f.P \mid nil \end{aligned} \tag{3}$$

where processes $\overline{S}T.P$ and $S(x).P$ denote "sending T over channel S, then running as P" and "receiving x over channel S, then running as P", respectively. *Probabilistic* process $\sum_i p_i P_i$ denotes that process P_i is selected to run with probability p_i, where $p_i \in (0,1]$ and $\sum_i p_i = 1$. $!P$ is used to replicate P and $(\nu \cdot x)$. P is used to restrict x in P. $\mathrm{MV}f.P$ is a move calculus, where function $f: Loc \to Prob(Loc)$ maps a location to probabilistic location spaces. Let $f(loc) = (Loc', pb)$ and $Loc' = \{loc'_0 \ldots loc'_n\}$, assuming that P are now at loc, $\mathrm{MV}f.P$ is used to denotes that process P will reach location loc'_i with probability $pb(loc'_i)$, $1 \le i \le n$. Notation nil is an empty process. In δ-calculus, we use both S and T to denote terms and their syntax are defined as followed.

$$S,T ::= x \mid a \tag{4}$$

Where x is an element of the countable set of variables and a is an element of the countable set of *channel names*.

4.2 Semantics

Combining normal π calculus with node names, locations and communication distances, we can get δ-calculus. Using a transition system tagged by actions α, β, we define its operational semantics, as follows:

$$\boldsymbol{\alpha,\beta := \tau \mid \overline{a}(T) \mid a(x)} \tag{5}$$

where τ is a silent action, $\overline{a}(M)$ and $a(x)$ denote the output of term M and the input of x on channel a, respectively. Generally, an attacker may deduce the true location of nodes by monitoring their communications. To simulate this behavior, $s\{\xrightarrow[p_i]{\alpha_i} s_i\}_i$ is extended to $s\{\xrightarrow[z_i,p_i,l_i;z_j]{\alpha_i} s_i\}_i$, indicating that receiver z_j knows that node z_j at loc_i has performed α_i. Note: node z_j can obtain this knowledge only after completing the communication with node z_j, thus this extension is used in the COM rule. In other rules, z_j, loc_i and z_j should be neglected, denoted by "-". We use the LOC-SUM rule to simulate the *location-choice* behavior. In the LOC-SUM, through executing τ, $z[\vert P \vert](\sum_i p_i \cdot loc_i, rad)$ becomes $z[\vert P \vert](loc_i, rad)$ with probability loc_i. Because the LOC-SUM does not require communication, no node can record the location of node z, denoted by "-" as shown in Tab. 1. In the PRO-SUM rule, processes are randomly selected, that is, $z[\vert \sum_i p_i P_i \vert](loc, rad)$ becomes $z[\vert P \vert](loc_i, rad)$ with probability loc_i by executing τ. From the OUT rule, we can see that: (1) $z[\vert \overline{a}T.P \vert](loc, rad)$ outputs T over channel a with probability loc and then it runs as $z[\vert P \vert](loc, rad)$, (2) an attacker cannot access z's names and locations, because nodes z only outputs terms and it does not communicate with other nodes. From the IN rule, we can see that, after $z[\vert a(x).P \vert](loc, rad)$ accepts x over channel a, it runs as $z[\vert P \vert](loc, rad)$. The COM rule simulates the interaction between two nodes, in detail, given two nodes z' and z'', their interaction completes if the following conditions are satisfied: (1) nodes z'' is in the communication ranges of z' (that is, $\Vert loc'' - loc' \Vert < rad'$, where $\vert loc'' - loc' \Vert$ denotes the physical distance between location loc' and location loc'') and (2) z' synchronizes with z''. If the above conditions are satisfied simultaneously, z'' will accept data sent by z' (that is, z'' will use x to substitute T). As shown in the COM rule, z'' will get the name and location of z' after interacting with z'. In the PAR rule, wildcard "*" is either "-", a location or a node name. Note: after PAR is used, new information about locations and names cannot be obtained by interactive nodes. The REP rule (the replication rule) denotes that a process repeatedly executes at the given location. We use RES1 to denote the constraint at locations, that is, actions at the restricted location should be allowed to be executed. RES2 means that actions on the channels different from y are allowed.

Table 1: Operational semantics of δ-Calculus

LOC-SUM	$z[\vert P \vert](\sum_i p_i \cdot loc_i, rad)\left\{\xrightarrow[-,p_i,-,-]{\tau} z[\vert P \vert](loc_i, rad)\right\}_i$
PRO-SUM	$z[\vert \sum_i p_i P_i \vert](loc, rad)\left\{\xrightarrow[-,p_i,-,-]{\tau} z[\vert P_i \vert](loc, rad)\right\}_i$
OUT	$z[\vert \overline{a}T.P \vert](loc, rad)\left\{\xrightarrow[-,1,-,-]{\overline{a}T} z[\vert P \vert](loc, rad)\right\}$
IN	$z[\vert a(x).P \vert](loc, rad)\left\{\xrightarrow[-,1,-,-]{a(x)} z[\vert P \vert](loc, rad)\right\}$
COM	$\dfrac{\begin{array}{c} z[\vert P \vert](loc', rad')\{\xrightarrow[-,1,-,-]{\overline{a}T} z[\vert P' \vert](loc', rad')\} \\ z''[\vert Q \vert](loc'', rad'')\{\xrightarrow[-,1,-,-]{\overline{a}(x)} z''[\vert Q''](loc'', rad'')\} \end{array}}{P \vert Q \xrightarrow[z,1,loc',z'']{\tau} z[\vert P' \vert](loc', rad') \vert z''[\vert Q''\{T/x\}](loc'', rad'')}$ if $\Vert loc'' - loc' \Vert < rad'$
PAR	$\dfrac{z[\vert P \vert](loc, rad)\{\xrightarrow[*,p_i,*,-]{a_i} z[\vert P_i \vert](loc', rad)\}_i}{z[\vert Q \vert](loc, rad) \vert z''[\vert P \vert](loc'', rad'')\{\xrightarrow[-,p_i,-,-]{a_i} z[\vert P_i \vert](loc', rad) \vert z'}$

REP	$\dfrac{z[\| P \|](loc,rad)\{\xrightarrow[*,p_i,*,-]{\alpha_i} z[\| P_i \|](loc',rad)\}_i}{z[\| !P \|](loc,rad)\{\xrightarrow[-,p_i,-,-]{\alpha_i} z[\| P_i \|](loc',rad) \mid z[\| !P \|](loc',rad)\}_i}$
RES1	$\dfrac{z[\| P \|](\sum_i p_i \cdot loc_i, rad)\{\xrightarrow[p_i,-,-,-]{\alpha_i} z[\| P \|](loc_i', rad)\}_{i:I}}{v \cdot loc \cdot z[\| P \|](\sum_i p_i \cdot loc_i, rad)\{\xrightarrow[p_i',-,-,-]{\alpha_i} v \cdot loc \cdot z[\| P \|](loc_i', rad')}$ *if* $I > 1$ *and* $\forall i.p_i' = p_i / \sum_{j:l_j \neq I} p_j$
RES2	$\dfrac{z'[\| P \|](loc',rad')\{\xrightarrow[p_i,-,-,-]{\alpha_i} z'[\| P_i \|](loc'',rad')\}_i}{z'[\| vxP \|](loc',rad')\{\xrightarrow[p_j,-,-,-]{\alpha_i} z'[\| vxP_i \|](loc'',rad')\}_{i:x \notin fn(\alpha_i)}}$ *if* $\exists i.x \notin fn(\alpha_i)$ *and* $\forall i.p_i' = p_i / \sum_{j:y \notin fn(\alpha_j)} p_j$

If we use δ-calculus to simulate a LPPM S, then S's behavior, denoted by $tds(S)$, can be considered as a set of trace distributions (denoted as $tds(S)$). We obtain $tds(S)$ via unfolding the δ-calculus. Next we give two simple examples to show the use of our δ-calculus.

Example 1. We assume two entities (i.e., a node and a sink, denoted by z_1 and z_a, respectively) exist in a wireless communication system. Node z_1 at location loc_1 delivers information $info$ to z_a with probability p_1, and at location loc_1' with probability $1-p_1$; Let z_a be always in the range of communication of z_1, (i.e., z_a can always receive $info$ from z_1). We can use δ-calculus to simulate this system, as follows:

$$SYS\_1 = Node \mid Sink$$

$$Node = z_1[\| \overline{send}(info) \|](p_1 \cdot loc_1 + (1-p_1)loc_1', r_1) \tag{6}$$

$$Sink = \{z\_a\}[\| send(x) \|](loc, rad)$$

where $\| loc_1 - loc \| < rad$ and $\| loc_1' - loc \| < rad$.

Fig. 1 shows the probabilistic execution of Example 1. In Fig. 1, there are 8 of execution sequences (i.e., $t_1 \ldots t_8$), where trace t_1 indicates that if the send action is asynchronous with the receive action (i.e., they don't shake hands), then node z_a does not get the information about z_1 from t_1. In traces t_3 and t_4, z_a records locations (loc_1 and loc_1', respectively) of z_1. In the two traces, z_a observes two traces: (p, z_1, loc) and $(1-p, z_1, loc')$, that is, z_a knows that z_1 stayed either at location loc_1 with probability p_1, or at location loc_1' with probability $1-p_1$.

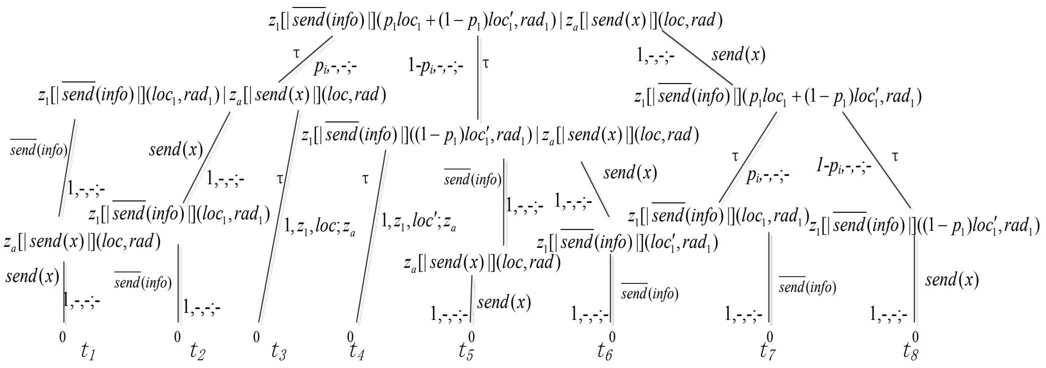
$z_1[|\overline{send}(info)|](p_1loc_1+(1-p_1)loc_1',rad_1)\,|\,z_a[|send(x)|](loc,rad)$
τ
p_i,-,-;-
1-p_i,-,-;-
τ
1,-,-;-
$send(x)$
$z_1[|\overline{send}(info)|](loc_1,rad_1)\,|\,z_a[|send(x)|](loc,rad)$
$z_1[|\overline{send}(info)|](p_1loc_1+(1-p_1)loc_1',rad_1)$
$z_1[|\overline{send}(info)|]((1-p_1)loc_1',rad_1)\,|\,z_a[|send(x)|](loc,rad)$
$\overline{send}(info)$
$send(x)$
1,-,-;-
1,-,-;-
τ
τ
$\overline{send}(info)$
$send(x)$
τ
p_i,-,-;-
1-p_i,-,-;-
τ
$z_1[|\overline{send}(info)|](loc_1,rad_1)$
$z_a[|send(x)|](loc,rad)$
$1,z_1,loc;z_a$
$1,z_1,loc';z_a$
1,-,-;-
1,-,-;-
$z_1[|\overline{send}(info)|](loc_1,rad_1)$
$z_1[|\overline{send}(info)|]((1-p_1)loc_1',rad_1)$
$z_1[|\overline{send}(info)|](loc_1',rad_1)$
$z_a[|send(x)|](loc,rad)$
$send(x)$
$\overline{send}(info)$
$\overline{send}(info)$
$send(x)$
$\overline{send}(info)$
$\overline{send}(info)$
$send(x)$
1,-,-;-
1,-,-;-
1,-,-;-
1,-,-;-
1,-,-;-
1,-,-;-
0
t_1
0
t_2
0
t_3
0
t_4
0
t_5
0
t_6
0
t_7
0
t_8

Figure 1: The probabilistic execution of Example 1

In Example 1, if the side constraint $\| loc_1 - loc \| < rad$ becomes $\| loc_1 - loc \| > rad$, and the remaining constraints keep unchanged, then, according to on the operational semantics, the only knowledge owned by z_a is that z_1 stays at loc_1'. In another word, z_1 doesn't realize that z_1 stayed at location loc_1.

Example 2. Consider the system:

$$SYS\_2 = Node \,|\, Sink$$

$$\begin{aligned} &Node = z_1[|\overline{send}(req_{auth}).receive(x).MVf. \\ &\overline{send}(info)|](p_0 \cdot loc_0 + (1-p_0)loc_1, rad_1) \end{aligned} \tag{7}$$

$$Sink = z_a[|send(x).\overline{receive}(ack_{auth}).send(x)|](loc,rad)$$

where (1) $f(loc_0) = (\{loc_2, loc_3, loc_4\}, pb_{l_0})$, $pb_{loc_0}(loc_2) = p_2$, $pb_{loc_0}(loc_3) = p_3$, $pb_{loc_0}(loc_4) = p_4$; (2) $f(loc_1) = (\{loc_5, loc_6, loc_7\}, pb_{loc_1})$, $pb_{loc_1}(loc_5) = p_5$, $pb_{loc_1}(l_6) = p_6$, and $pb_{loc_1}(loc_7) = p_7$. Assume that $\| loc_i - loc \| < rad, rad_1$ ($i = 0 \ldots 7$).

In this example, z_1 can infer that z_a stayed at loc with probability 1. Accordingly, z_a will infer that the traces of z_1 are $loc_0 \to loc_2$, $loc_0 \to loc_3$, $loc_0 \to loc_4$, $loc_1 \to loc_5$, $loc_1 \to loc_6$ and $loc_1 \to loc_7$, with probabilities p_0p_2, p_0p_3, p_0p_4, p_1p_5, p_1p_6, and p_1p_7, respectively.

5 Measuring location privacy

Surely, a LPPM always reveals location information more or less. In general, in an attack, if the amount of location leakage is less than a given threshold value, then this leakage can be accepted. This involves two issues (that is, the attacker model, and quantifying location leakage). In this section, we discuss them separately.

5.1 Modeling attackers

An adversary may infer location information by playing the role of normal users to interact with them, and monitoring their communication. Generally, adversaries are divided into two categories: strong attackers and weak attackers. If a LPPM under a strong attack is secure, then it is also secure under a weak attack. Thus, we should simulate the strong attacker. Informally, an attacker is to be strong, if it can gather all locations of the normal users at anywhere or anytime.

Definition 2. An attacker is strong, if an *output* action (i.e., $\overline{a}(T)$) is performed anywhere or anytime and the attacker can gather the location where action a happens.

In Examples 1 and 2, if an attacker can act as the sink, then it is strong (because it can obtain all locations once the *output* action happens). Next, we illustrate an example of a weak attacker.

Example 3. Consider the system:

$$SYS\_3 = Node \mid Attacker$$

$$Node = z_1[\mid \overline{send}(req_{auth}) \mid](p_0 \cdot loc_0 + (1-p_0)loc_1, rad_1) \qquad (8)$$

$$Attacker = z_a[\mid send(x) \mid](loc, rad)$$

where, $\| loc_0 - loc \| > rad_1$ or $\| loc_1 - loc \| > rad_1$.

In Examples 1 and 2, if an attacker can act as the sink, then it is strong (because it can obtain all locations once the *output* action happens). Next, we illustrate an example of a weak attacker.

Example, if z_1 is at loc_0, the attacker z_a cannot track z_1 's location loc_0 (because $\| loc_0 - loc \| > rad_1$, and z_a is not in the communication range of z_1). Similarly, when z_1 is at location loc_1, z_a cannot record z_1 's location. So, z_a is not strong.

5.2 Quantifying location privacy

Given LPPM M and an attacker ATT simulated by δ-calculus, we use a set of trace distributions (obtained by unfolding $M \mid ATT$, written as $tds(M \mid ATT)$) to describe ATT 's interactions with M. An attacker implicitly obtains nodes' locations of by recording these traces.

Given a set X of trace distributions, a metric D on a set X can be defined as function $D : X \times X \to R^+$. Generally, metric D is required to satisfy the following three axioms: non-negative (i.e., for all $x_1, x_2 \in X$, formula $D(x_1, x_2) \geq 0$ holds), coincidence (i.e., for all $x_1, x_2 \in X$, $D(x_1, x_2) = 0$ if and only if $x_1 = x_2$), symmetry (i.e., for all $x_1, x_2 \in X$, $D(x_1, x_2) = D(x_2, x_1)$) and subadditivity (i.e., for all $x_1, x_2, x_3 \in X$, $D(x_1, x_2) +$ $D(x_2, x_3) \geq D(x_1, x_3)$), where R^+ is the set of non-negative real numbers.

To protect location privacy, many LPPMs add false locations into true locations to prevent an attacker from inferring the true location. For simplicity, we use LOC to denote the set of locations. To measure location privacy, we define the re-naming function as $f_{LOC}: LOC \to LOC$, which permutes $loc \in LOC$ to $loc' \in LOC$ ($loc \neq loc'$). That is, for each location in LOC, the following conditions are satisfied: (1) $f(loc) \neq loc$ forever, and (2) $loc_1 \neq loc_2$ implies $f_{LOC}(loc_1) \neq f_{LOC}(loc_2)$. We use F_{LOC} to represent the set of all renaming functions f_{LOC} on LOC.

Definition 3. Given a LPPM M and a metric D, M is privacy-preserved under D on a set of locations LOC if formula $\forall f_{loc} \in F_{LOC}: D(M, f_{loc}(M)) = 0$ holds; M is called ς-privacy if $\forall f_{loc} \in F_{LOC}: D(M, f_{loc}(M)) \leq \varsigma$.

Theorem. Given two metrics D_1 and D_2, and a LPPM M, if M is privacy-preserved under D_1, M is strong privacy- preserved under D_2. ς-privacy preservation of M under D_1 doesn't imply ς-privacy preservation under D_2.

Proof. Because M is the privacy-preserved under D_1, $D_1(M, f_{loc}(M)) = 0$ holds. According to the coincidence axiom of metric spaces, we have $M = f_{loc}(M)$, thus, $D_2(M, f_{loc}(M)) = 0$. This means that M is the privacy- preserved under D_2. The second part is direct.

In our paper, we use α-entropy as a quasi-metric to evaluate location privacy, because relative entropy meets the axiom of nonnegative and coincidence.

Definition 4. For discrete probability distributions u and u', the relative entropy of u' from u is defined to be

$$D_{KL}(u \| u') = \sum_i \log(\frac{u(i)}{u'(i)})u(i) \tag{9}$$

where $0\log\frac{0}{0} = 0$, $0\log\frac{0}{q} = 0$, $0\log\frac{q}{0} = \infty$ and $i \in I$ is an index set. Because the behavior of a node is simulated as a set of trace distributions, D_{KL} is extended to D_{EKL} as follows (Similarly to Kapus [Kapus (2017)]).

Definition 5. Given two sets ($U = \{u_i\}_i$ and $U' = \{u'_j\}_j$) of probability distributions, the relative entropy of U' from U is defined as

$$D_{EKL}(\mathrm{U} \| \mathrm{U}') = \sup_i \inf_j D_{KL}(u_i \| u'_j) \tag{10}$$

where $\inf \phi = \infty$ and $\sup \phi = 0$.

Measuring location privacy: Assume that node M is protected under the LPPM and $tds(M)$ denotes M's trace distribution, then the amount of leakage of local privacy is

$$\sup_{f_{loc} \in F_{LOC}} D_{EKL}(tds(M) \| tds(f_{loc}(M))^{\cdot}$$

Example 4. Considering that a wireless communication system owns a base station z_b and nodes Z_1. Assuming that node z_1 at location loc_1 sends data to z_b, and the goal of attacker z_a is to get z_1's location by observing their communications. We also assume that in LPPM, false locations (e.g., false location loc'_1) is used to protect z_1. The system can be simulated as:

$$\begin{aligned} &M = Node \mid BaseStation \mid Attacker \\ &Node = z_1[| \overline{send}(info) |](p_1 \cdot loc_1 + (1-p_1)loc'_1, rad_1) \\ &BaseStation \ = \ \{z\_b\}[| send(x) |](loc, rad) \\ &Attacker \ = \ \{z\_a\}[| send(x) |](loc, rad) \end{aligned} \tag{11}$$

That is, node z_1 sends data to z_b at location loc_1 with probability p_1 or at location loc'_1 with probability 1-p_1. Base station z_b at location accepts the data loc. If z_a is in the range of communication radius of z_1 (that is, the distance between loc_1 and loc'_1, and loc is less than communication radius rad_1 of z_1), and then attacker z_a can accept the data sent by z_1 and conjecture z_1's location. Let the only permutation function f on LOC be $f(loc_1) = loc'_1$ and $f(loc'_1) = loc_1$. That is,

$$f_{loc}(Node) = z_1[| \overline{send} \ (\text{info})|](p_1 loc'_1 + (1-p_1) loc_1, rad_1), \tag{12}$$

The leakage amount of location privacy will be

$$D_{KL}(tds(M) \parallel tds(f_{loc}(M)) = p_1 \log \frac{p_1}{1-p_1} + (1-p_1) \log \frac{1-p_1}{p_1}. \tag{13}$$

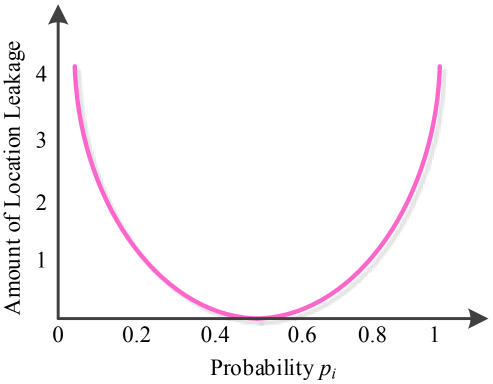

Figure 2: Amount of location leakage

Fig. 2 shows the amount of location leakage of M when p_i changes. From this figure, we can see that: the amount gained by z_a equals 0 if $p_i = 0.5$ (that is, z_a cannot deduce the location of node z_1) and the leakage amount becomes infinite if either $p_i \to 0$ or $p_i \to 1$ holds (This means that z_a can accurately deduce the true location of z_1). This is accordance with our intuition, (i.e., the capability of the LPPM is zero if p_i approaches to 0 or 1, and it reaches 1 if p_i approaches to 0.5). This shows that our measurement is accurate. According to the above analyses, we summarize our proposed approach to measuring location privacy, as shown in Tab. 2.

Table 2: Operational semantics of δ-Calculus

1	For the LPPM P to be measured, the LPPM P is formally defined using the δ-calculus syntax as shown in Section 4.1.
2	Computing all the trajectories of the LPPM P (including both the truth and the dummy) with a probability distribution, using the δ-calculus semantics as shown in Section 4.2.
3	Defining a set F_{LOC} of renaming functions, where the mapping function $f \in F_{LOC}$ on LOC satisfies two conditions: (1) $f(loc) \neq loc$ and (2) $loc_1 \neq loc_2$ implies $f(loc_1) \neq f(loc_2)$.
4	Computing the probability distribution $tds(P)$ and $tds(f(P))$ of traces of P and $f(P)$, respectively.
5	Quantifying location privacy by using $D_{RKL}\left(tds(P)\,\|\,\{tds(f_1(P)),\ \ldots,\ tds(fn(P))\}\right)$, where $f_i \in P$ and $1 \leq i \leq n$.

6 Experiments

In this section, we use the proposed δ-calculus to evaluate trace privacy protected by DUMMY-T [Niu, Gao, Li et al. (2016)]. In DUMMY-T, a set of realistic dummy locations for each snapshot is generated to guarantee the minimum cloaking region and resist from attacks performed by adversaries with background information. Then, DUMMY-T connects the dummy locations together into the dummy paths with considering the location reachability.

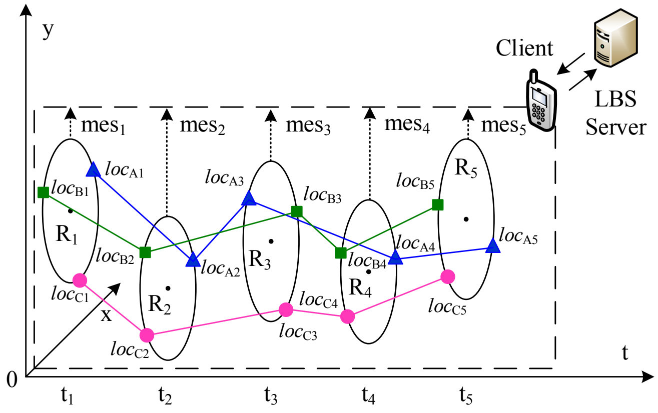

Figure 3: The leakage amount *vs.* $p_{l_{A_1}}$ and $p_{l_{B_1}}$

The idea of DUMMY-T is shown in Fig. 3. Specifically, to protect users' true trajectories from the LBS server, DUMMY-T need generate k-1 dummy trajectories based on k-anonymity method. The trajectory in blue (triangle) is the user's real route, and the routes in green (square) and pink (circle) are dummy paths. Each trajectory can be divided into 5 snapshots with 5 timestamp t_1 to t_5. The obfuscation region of each snapshot is denoted as R_1 to R_5, the message from LBS server at each timestamp can be denoted as mes_1 to mes_5. For each snapshot, DUMMY-T generates a set of dummy locations which cannot be distinguished from others easily, and connects the dummy locations into *k*-1 dummy paths with considering the reachability. Finally, users get several dummy trajectories which the adversary cannot be guessed the real one from them.

Using δ-calculus, we can describe DUMMY-T, as follows:

$$\begin{aligned}
&\mathbf{DUMMY-T} = \mathbf{C} \,|\, \mathbf{SERVER}\\
&\mathbf{C} = c[\overline{s}(mes_1)](p_{A1}loc_{A1} + p_{B1}loc_{B1} + p_{C1}loc_{C1},\ rad_u).c[MV\ m_1].\mathbf{T_2},\\
&\mathbf{T_2} = c[\overline{s}(mes_2)](p_{A2}loc_{A2} + p_{B2}loc_{B2} + p_{C2}loc_{C2}, rad_u).c[MV\ m_2].\mathbf{T_3},\\
&\mathbf{T_3} = c[\overline{s}(mes_3)](p_{A3}loc_{A3} + p_{B3}loc_{B3} + p_{C3}loc_{C3}, rad_u).c[MV\ m_3].\mathbf{T_4},\\
&\mathbf{T_4} = c[\overline{s}(mes_4)](p_{A4}loc_{A4} + p_{B4}loc_{B4} + p_{C4}loc_{C4}, rad_u).c[MV\ m_4].\mathbf{T_5},\\
&\mathbf{T_5} = c[\overline{s}(mes_5)](p_{A5}loc_{A5} + p_{B5}loc_{B5} + p_{C5}loc_{C5}, rad_u).c[MV\ m_5].\mathbf{0},\\
&\mathbf{SERVER} = sr[!\overline{s}(x)](loc,\ rad_s).
\end{aligned} \tag{14}$$

Where the moving functions $m_1 \sim m_5$ are defined as follows:

$$\begin{aligned}
&m_1(loc_{A1}) = m_1(loc_{B1}) = m_1(loc_{C1}) = \{\{loc_{A2}\},\{loc_{B2}\},\ \{loc_{C2}\}\};\\
&m_2(loc_{A2}) = m_2(loc_{B2}) = m_2(loc_{C2}) = \{\{loc_{A3}\},\{loc_{B3}\},\ \{loc_{C3}\}\};\\
&m_3(loc_{A3}) = m_3(loc_{B3}) = \ m_3(loc_{C3}) = \{\{loc_{A4}\},\{loc_{B4}\},\ \{loc_{C4}\}\};\\
&m_4(loc_{A4}) = \ m_4(loc_{B4}) = m_4(loc_{C4}) = \{\{loc_{A5}\},\{loc_{B5}\},\ \{loc_{C5}\}\};\\
&m_5(loc_{A5}) = loc_{A5},\ m_5(loc_{B5}) = loc_{B5},\ m_5(loc_{C5}) = loc_{C5}.
\end{aligned} \tag{15}$$

The definition of functions m_1 shows that, from loc_{A1}, loc_{B1} or loc_{C1}, user can move any one of points loc_{A2}, loc_{B2}, and loc_{C2}. Functions m_2 - m_4 are similar with m_1. According to operational semantics shown in Section 4.2, by monitoring traces (including the true traces and dummy traces) of user U, **SERVER** can obtain 81 traces and their probability distribution, i.e., { $p_{A1}p_{A2}p_{A3}p_{A4}p_{A5} : loc_{A1} \to loc_{A2} \to loc_{A3} \to loc_{A4} \to loc_{A5}$, $p_{A1}p_{A2}p_{A3}p_{A4}p_{B5} : loc_{A1} \to loc_{A2} \to loc_{A3} \to loc_{A4} \to loc_{B5}$, …, $p_{C1}p_{C2}p_{C3}p_{C4}p_{C5} : loc_{C1} \to loc_{C2} \to loc_{C3} \to loc_{C4} \to loc_{C5}$ }, where $p_{A1}p_{A2}p_{A3}p_{A4}p_{A5} : loc_{A1} \to loc_{A2} \to loc_{A3} \to loc_{A4} \to loc_{A5}$ denotes that trace $loc_{A1} \to loc_{A2} \to loc_{A3} \to loc_{A4} \to loc_{A5}$ is observed with probability $p_{A1}p_{A2}p_{A3}p_{A4}p_{A5}$ by **SERVER**.

Next, we define permutation function f. According to Section 5.2, there are 32 permutation functions $F = \{f_1, \ldots, f_{32}\}$ that satisfy the following two conditions: (1) $f(loc_{x_i}) \neq loc_{x_i}$ and (2) $loc_{x_i} \neq loc_{x_j}$ implies $f(loc_{x_i}) \neq f(loc_{x_j})$, where $loc_{x_i} \in X_i$,

$loc_{x_j} \in X_j$, $X_i = \{loc_{A_i}, loc_{B_i}, loc_{C_i}\}$, $X_j = \{loc_{A_j}, loc_{B_j}, loc_{C_j}\}$, $1 \le i, j \le 3$. Next, we discuss the leakage amount of privacy location in the following cases:

Case 1: In this case, the moving functions $m_1 \sim m_5$ are assumed to be rational (i.e., the mobile client moves with a reasonable velocity between the time intervals that the client sends messages $m_1 \sim m_5$, in the other word, **SERVER** cannot perceive any abnormality). Given permutation function f and its inverse function f^{-1}, we can evaluate the leakage amount of privacy location, as follows.

$$LA = \sum_{x_1 \in X_1} \sum_{x_2 \in X_2} \sum_{x_3 \in X_3} \sum_{x_4 \in X_4} \sum_{x_5 \in X_5} \left(\left(\prod_{i=1}^{5} p_{x_i} \right) \log \frac{\prod_{i=1}^{5} p_{x_i}}{\prod_{i=1}^{5} p_{f^{-1}(x_i)}} \right) \tag{16}$$

The minimum amount is evaluated as $\min_{p_{x_1} \cdots p_{x_5}} LA$ that can be easily obtained by solving the first and second derivatives. Next, we give an example of its solution. In this example, we assume that, the **SERVER** has known the true trace point in R_2 *to* R_5 and it does not know the true point in R_1. In this case, the leakage amount relies on $p_{loc_{A_1}}$, $p_{loc_{B_1}}$ and $p_{loc_{C_1}}$, where $p_{loc_{A_1}} + p_{loc_{B_1}} + p_{loc_{C_1}} = 1$. We can evaluate the information leakage with regard to $p_{loc_{A_1}}$ and $p_{loc_{B_1}}$, shown in Fig. 4. When one of the three parameters equals 0, then the leakage amount of location privacy reaches the maximum; if $p_{loc_{A_1}} = p_{loc_{B_1}} = p_{loc_{C_1}}$, then leakage amount of location privacy is zero. This is consistent in our intuition.

Case 2: In this case, at least one of the moving functions $m_1 \sim m_5$ is assumed to be irrational (i.e., the distance between two dummy points is great than a reasonable distance and **SERVER** can perceive this abnormality). For example, we assume that $m_1(loc_{A1}) = m_1(loc_{B1}) = m_1(loc_{C1}) = \{\{loc_{A2}\}, \{loc_{B2}\}, \{loc_{C2}\}\}$, but the distance between point loc_{A1} and point loc_{A2} are greater than a reasonable value. In this case, once **SERVER** observes three traces $loc_{A1} \rightarrow loc_{A2}$, $loc_{B1} \rightarrow loc_{B2}$ and $loc_{C1} \rightarrow loc_{C2}$, then it can infer that at least one point between loc_{A1} and loc_{A2} is dummy. Thus, we can evaluate the leakage amount of privacy location, as follows.

$$\tilde{L}\tilde{A} = \sum_{x_1 \in \tilde{X}_1} \sum_{x_2 \in \tilde{X}_2} \sum_{x_3 \in X_3} \sum_{x_4 \in X_4} \sum_{x_5 \in X_5} \left(\left(\prod_{i=1}^{5} p_{x_i} \right) \log \frac{\prod_{i=1}^{5} p_{x_i}}{\prod_{i=1}^{5} p_{f^{-1}(x_i)}} \right) \tag{17}$$

where $\tilde{X}_1 = \{loc_{B1}, loc_{C1}\}$, $\tilde{X}_2 = \{loc_{B2}, loc_{C2}\}$. Given p_{x_i} , we can directly obtain $\tilde{L}\tilde{A}$.

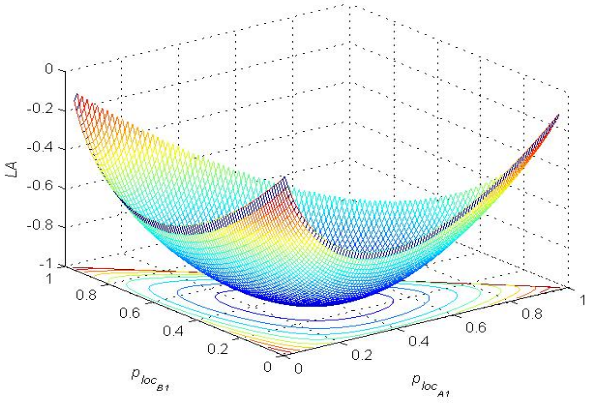

Figure 4: The leakage amount *vs.* $p_{loc_{A_1}}$ and $p_{loc_{B_1}}$

7 Conclusion

In this paper, we propose δ-calculus to formalize obfuscation-based schemes and measure location privacy. Probabilistic automata is adopted to formally characterize the semantics of δ-calculus. Specially, two calculus moving and probabilistic choices are proposed to model nodes' mobility and compute its probability distribution of nodes' locations. Further, the renaming function is proposed to model privacy leakage. By formally defining the attacker's ability and extending relative entropy, we propose an evaluation algorithm to quantify the leakage of location privacy. Experimental results demonstrate that our scheme can accurately quantify the location leakage. Through the proposed δ-calculus, the gap between the obfuscation-based scheme and its measurement is decreased. In the future, the following work should be conducted. (1) In this paper, we only integrate privacy measurement into obfuscation-based schemes. Obviously, it is necessary to design a formal language to describe both elimination and anonymization schemes and synergize them into the quantitative measurement framework. Through this approach, more LPPMs are verified and measured. (2) Although we develop a measurement algorithm to evaluate the privacy leakage, this algorithm is not integrated into the existing tool (such as PRISM [Pramanik, Lau, Zhang et al. (2016)]). It is of great importance to integrate them to automatically calculate the privacy level.

Funding Statement: This research is supported in part by the National Key Research and Development Program of China (Grant No. 2017YFB0803001), in part by the Key research and Development Program for Guangdong Province under grant (Grant No. 2019B010136001), in part by the National Natural Science Foundation of China (Grant No. 61872100) and Guangxi Natural Science Foundation (No. 2017GXNSFAA198372).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Abul, O.; Bonchi, F.; Nanni, M. (2008): Never walk alone: uncertainty for anonymity in moving objects databases. *Proceedings of IEEE 24th International Conference on Data Engineering*, pp. 376-385.

Arapinis, M.; Chothia, T.; Ritter, E.; Ryan, M. (2010): Analysing unlinkability and anonymity using the applied pi calculus. *Proceedings of IEEE Computer Security Foundations Symposium*, pp. 107-121.

Beresford, A. R.; Stajano, F. (2003): Location privacy in pervasive computing. *IEEE Pervasive Computing*, vol. 2, no. 1, pp. 46-55.

Brusó, M.; Chatzikokolakis, K.; Den Hartog, J. (2010): Formal verification of privacy for RFID systems. *Proceedings of IEEE Computer Security Foundations Symposium*, pp. 75-88.

Dahl, M.; Delaune, S.; Steel, G. (2010): Formal analysis of privacy for vehicular mix-zones. *Proceedings of Springer European Symposium on Research in Computer Security*, pp. 55-70.

Deng, Y.; Pang, J.; Wu, P. (2006): Measuring anonymity with relative entropy. *Proceedings of Springer International Workshop on Formal Aspects in Security and Trust*, pp. 65-79.

Ding, J.; Li, X.; Guo, Y.; Yin, L; Zhang, H. (2018): Process calculus for modeling and quantifying location privacy. *Proceedings of International Conference on Identification, Information and Knowledge in the Internet of Things*, pp. 407-415.

Dong, Y.; Pi, D. (2018): Novel privacy-preserving algorithm based on frequent path for trajectory data publishing. *Knowledge-Based Systems*, vol. 148, pp. 55-65.

Emara, K.; Woerndl, W.; Schlichter, J. (2015): On evaluation of location privacy preserving schemes for VANET safety applications. *Computer Communications*, vol. 63, pp. 11-23.

Fischer, L.; Katzenbeisser, S.; Eckert, C. (2008): Measuring unlinkability revisited. *Proceedings of ACM Workshop on Privacy in the Electronic Society*, pp. 105-110.

Freudiger, J.; Manshaei, M. H.; Hubaux, J. P.; Parkes, D. C. (2013): Non-cooperative location privacy. *IEEE Transactions on Dependable and Secure Computing*, vol. 10, no. 2, pp. 84-98.

Guo, Y.; Zhang, H.; Zhang, L.; Fang, L.; Li, F. (2018): Incentive mechanism for cooperative intrusion detection: an evolutionary game approach. *Proceedings of Springer International Conference on Computational Science*, pp. 83-97.

Herescu, O. M.; Palamidessi, C. (2000): Probabilistic asynchronous π-calculus. *Proceedings of Springer International Conference on Foundations of Software Science and Computation Structures*, pp. 146-160.

Kapus, T. (2017): Using PRISM model checker as a validation tool for an analytical model of IEEE 802.15.4 networks. *Simulation Modeling Practice and Theory*, vol. 77, pp. 367-378.

Li, M.; Sun, Y.; Lu, H.; Maharjan, S.; Tian, J. (2020): Deep reinforcement learning for partially observable data poisoning attack in crowdsensing systems. *IEEE Internet of Things Journal.*

Liao, D.; Sun, G.; Zhang, M.; Chang, V.; Li, H. (2017): Towards location and trajectory privacy preservation in 5G vehicular social network. *Proceedings of IEEE International Conference on Computational Science and Engineering*, pp. 63-69.

Liu, X.; Zhao, H.; Pan, M.; Yue, H.; Li, X. (2012): Traffic-aware multiple mix zone placement for protecting location privacy. *Proceedings of IEEE Conference on Computer Communications*, pp. 972-980.

Niu, B.; Gao, S.; Li, F.; Lu, Z. (2016): Protection of location privacy in continuous LBSs against adversaries with background information. *Proceedings of IEEE International Conference on Computing, Networking and Communications*, pp. 1-6.

Niu, B.; Li, Q.; Zhu, X.; Gao, G.; Li, H. (2014): Achieving k-anonymity in privacy-aware location-based services. *Proceedings of IEEE Conference on Computer Communications*, pp. 754-762.

Niu, B.; Li, Q.; Zhu, X.; Gao, G.; Li, H. (2015): Enhancing privacy through caching in location-based services. *Proceedings of IEEE Conference on Computer Communications*, pp. 1017-1025.

Olteanu, A. M.; Huguenin, K.; Shokri, R.; Humbert, M.; Hubaux, J. P. (2017): Quantifying interdependent privacy risks with location data. *IEEE Transactions on Mobile Computing*, vol. 16, no. 3, pp. 829-842.

Pandit, A.; Polina, P.; Kumar, A. (2014): CLOPRO: a framework for context cloaking privacy protection. *Proceedings of IEEE International Conference on Communication Systems and Network Technologies*, pp. 782-787.

Pramanik, M. I.; Lau, R. Y. K.; Zhang, W. (2016): K-anonymity through the enhanced clustering method. *Proceedings of IEEE International Conference on e-Business Engineering*, pp. 85-91.

Roberto, J. B.; Rakesh, A. (2005): Data privacy through optimal k-anonymization. *Proceedings of IEEE International Conference on Data Engineering*, pp. 217-228.

Sweeney, L. (2002): A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557-570.

Tian, Z.; Gao, X.; Su, S.; Qiu, J. (2020): Vcash: a novel reputation framework for identifying denial of traffic service in internet of connected vehicles. *IEEE Internet of Things Journal.*

Tian, Z.; Luo, C.; Qiu, J.; Du, X.; Guizani, M. (2019): A distributed deep learning system for web attack detection on edge devices. *IEEE Transactions on Industrial Informatics.*

Tian, Z.; Shi, W.; Wang, Y.; Zhu, C.; Du, X. et al. (2019): Real time lateral movement detection based on evidence reasoning network for edge computing environment. *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4285-4294.

Wang, L.; Yang, D.; Han, X.; Wang, T.; Zhang, D. (2017): Location privacy-preserving task allocation for mobile crowdsensing with differential geo-obfuscation. *Proceedings of IEEE International Conference on World Wide Web*, pp. 627-636.

Xiao, Y.; Xiong, L. (2015): Protecting locations with differential privacy under temporal correlations. *Proceedings of ACM SIGSAC Conference on Computer and Communications Security*, pp. 1298-1309.

Ye, A.; Li, Y.; Xu, L.; Li, Q.; Lin, H. (2017): A trajectory privacy-preserving algorithm based on road networks in continuous location-based services. *Proceedings of IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 510-516.

Yin, L.; Guo, Y.; Zhang, H.; Huang, W.; Fang, B. (2019): Threat-based declassification and endorsement for mobile computing. *Chinese Journal of Electronics*, vol. 28, no. 5, pp. 1041-1052.

Yin, L.; Liu, H. (2019): Searching activity trajectories with semantics. *Journal of Computer Science and Technology*, vol. 34, no. 4, pp. 775-794.

Yin, L.; Luo, X.; Zhu, C.; Wang, L.; Xu, Z. et al. (2020): ConnSpoiler: disrupting C&C communication of IoT-based botnet through fast detection of anomalous domain queries. *IEEE Transactions on Industrial Informatics*.

Yin, L.; Sun, Y.; Wang, Z.; Guo, Y.; Li, F. et al. (2018): Security measurement for unknown threats based on attack preferences. *Security and Communication Networks*, vol. 2018, pp. 1-13.

Yu, R.; Kang, J.; Huang, X.; Xie, S.; Zhang, Y. et al. (2016): MixGroup: accumulative pseudonym exchanging for location privacy enhancement in vehicular social networks. *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 1, pp. 93-105.

Zhang, Y.; Tong, W.; Zhong, S. (2016): On designing satisfaction-ratio-aware truthful incentive mechanisms for k-anonymity location privacy. *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2528-2541.

Zheng, X.; Cai, Z.; Li, J.; Gao, H. (2017): Location-privacy-aware review publication mechanism for local business service systems. *Proceedings of IEEE Conference on Computer Communications*, pp. 1-9.