applied sciences

MDPI

Article

An Efficient Approach for LBS Privacy Preservation in Mobile Social Networks

Guangcan Yang *,†, Shoushan Luo †, Hongliang Zhu †, Yang Xin †, Mingzhen Li † and Yunfeng Wang †

National Engineering Laboratory for Disaster Backup and Recovery, Information Security Center, School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China; buptlou@263.net (S.L.); zhuhongliang@bupt.edu.cn (H.Z.); yangxin@bupt.edu.cn (Y.X.); MzLi@bupt.edu.cn (M.L.); wangyunfeng@bupt.edu.cn (Y.W.)

* Correspondence: yangguangcan@bupt.edu.cn; Tel.: +86-137-1856-3988

† These authors contributed equally to this work.

Received: 27 November 2018; Accepted: 11 January 2019; Published: 16 January 2019

Abstract: With the rapid development of smart handheld devices, wireless communication, and positioning technologies, location-based service (LBS) has been gaining tremendous popularity in mobile social networks (MSN). Users' daily lives are facilitated by the applications of LBS, but users' privacy leaking hinders the further development of LBS. In order to solve this problem, techniques such as k-anonymity and l-diversity have been widely adopted. However, most papers that combine with k-anonymity and l-diversity focus on the security of users' privacy with little consideration of service efficiency. In this paper, we firstly treat the relationship between k-anonymity and l-diversity in the clustering process from a dynamic and global perspective. Then a service category table based algorithm (SCTB) is designed to identify and calculate l-diversity securely and efficiently, which promotes the cooperative efficiency of users in LBS query, especially when the preference privacy that users request in the clustering process have similarities. Finally, theoretical performance analysis and extensive experimental studies are performed to validate the effectiveness of our SCTB algorithm.

Keywords: location-based service (LBS); k-anonymity; l-diversity; preference privacy

1. Introduction

Due to the rapid advance of mobile communication technologies and modern smart mobiles, location-based service (LBS) has become ubiquitous in recent years. With positioning technologies and applications loaded on smart mobiles, users could query location-based service providers (LBSP) for personalized services according to their locations, such as finding the nearest restaurant, getting a travel guide, or searching for a nearby market, etc.

Although location-based service brings convenience to users and plays an increasingly important role in daily life, people are also aware that their private information may be revealed along with the services provided by LBSP. Hence, the problem of privacy leakage has become a main factor that hinders the further development of LBS. Therefore, how to protect users' privacy information while providing conventional services to users has become a topic of major interest.

Generally speaking, current popular applications on smart phones can be roughly divided into two categories. One is to hide the "who" aspect completely for LBSP, and just uses the service with "what" and "where", such as AMAP. Another is not to hide the "who" for LBSP, and uses the service with "who", "what", and "where", such as Dianping. Here, "who" means the identity of a user, "where" means the location of a user, and "what" means the preference of a user. For the wide application of LBS, users' sensitive information (i.e., "who", "where", "what") should be protected

well while interacting with LBSP. For the aspect of "who", security techniques such as pseudo-ID [1] and pseudonym are usually used to make users' real identity indistinguishable, but it is insufficient to protect users' sensitive information [2]. For the aspect of "what", exposure of a user's preference may result in the leakage of his sensitive information. For example, if Bob often sends some requests about the knowledge of coronary disease to LBSP, then LBSP or adversary could infer that Bob or his family members may have a health problem of coronary disease. For the aspect of "where", if Bob often reveals his locations close to a hospital when sending his requests to LBSP, adversary or LBSP may also infer that Bob may have some health problems. If this sensitive information is leaked, some commercial organizations or adversary may send drug ads to Bob frequently. To address the problem of leaking users' privacy caused by location information, k-anonymity as a classic scheme was proposed. k-anonymity is an anonymity set which is composed of users with different locations, and makes one user's location indistinguishable from the locations of at least k-1 other users at some point [3]. For example, there are k users simultaneously requesting LBSP.

But k-anonymity is insufficient to guarantee users' preference privacy when they request the same preference service in LBS query. There are at least two attacks, the Homogeneity Attack and Background Knowledge Attack, that can be used to compromise a anonymity set [4]. In order to overcome this deficiency, the concept of l-diversity, which ensures at least l well-represented values in each equivalence class, was proposed to protect users' privacy. Figure 1 shows the way to protect users' privacy by k-anonymity and l-diversity. There are 6 users (i.e., k = 6) who are distributed in different locations requesting 4 different services (i.e., l = 4), and LBSP cannot link a specific service to a specific user. There are many studies adopt both k-anonymity and l-diversity to protect personal privacy in LBS query. For example, the authors [5] proposed a Dummy-Location-Selection (DLS) algorithm to protect location privacy and preference privacy. The paper [6] used Snet-Hierarchy to protect personal privacy by k-anonymity and l-diversity. A personalized spatial cloaking scheme named TTcloak was proposed in paper [7], users' location privacy and preference privacy were protected by k-anonymity and l-diversity respectively in Mobile Social Networks.

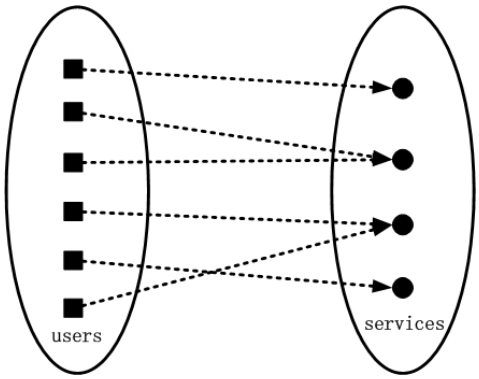


Figure 1. Privacy preservation by k-anonymity and l-diversity.

Currently, most of the existing works concerning the LBS query mainly focus on the security of users' privacy with little consideration of service efficiency. Based on k-anonymity and l-diversity, some current research work, such as paper [8], solved the problem that how to calculate the number of l-diversity without a trusted anonymizer server under a distributed architecture. But the algorithm, proposed to protect preference privacy in the clustering process, has a high degree of complexity. The authors [9] mentioned that users' service should be classified to different service categories, but did not consider how to identify the classified service categories, and it is inevitable to incur a huge time cost if using the way of semantic recognition. Therefore, under the system of distributed architecture, how to identify and calculate l-diversity securely and efficiently in the clustering process is still a challenge. As far as we know, there is no research on this problem. Furthermore, in real applications, it is unreasonable to judge the attribute of a user's location in LBS query from a traditional and static

viewpoint. For example, restaurants and schools are often seen as sensitive or dense locations because of the dense population from the traditional point of view [5,9]. However, as we all know, if the time is not for meal or school day, it is unreasonable for restaurants and schools to be classified as sensitive or dense locations. Thus, the schemes based on the static viewpoint are not very practical.

In this paper, aiming at addressing the above challenges, we propose service category table based (SCTB) algorithm to improve clustering efficiency while protecting users' preference privacy. Therefore, under the system of distributed architecture, we rigorously explore the calculation procedure of l-diversity in LBS query, and introduce the service category table into our SCTB algorithm to make SCTB algorithm not only suitable for a single request but also for continuous request. Different from previous solutions based on the straightforward static scenario (e.g., a restaurant is unified as sensitive or dense locations), the scheme in this paper focuses on the process of security and efficient clustering from a dynamic and global perspective. The main contributions of this paper are as follows.

1. To overcome the vulnerabilities of preference privacy in LBS query, we introduce the service category table to represent different kinds of preference privacy so that the process of identifying the kinds of l-diversity can be fast, secure, and computationally inexpensive.

2. Unlike previous works that focus on location privacy and preference privacy independently, we firstly study the relationships between k and l in the clustering process. Then based on the relationships, the concepts of relatively dense scenario and relatively sparse scenario from a dynamic perspective are proposed. Finally, according to different scenarios mentioned above, we give the corresponding algorithm to improve the success rate of clustering from a global perspective.

3. We propose service category table based (SCTB) algorithm which includes request aggregation protocol, variety judgment protocol, filtering aggregation protocol, and representative aggregation protocol to reduce the time cost of clustering process and improve the success rate of clustering groups while protecting preference privacy of users. We also study the circumstance when users' preferences in the clustering process have similarities.

4. We carry out extensive simulations to evaluate the performance of our SCTB algorithm.

2. Related Work

In this section, we mainly focus on the current situation of LBS privacy protection. There have been many kinds of research focus on how to deal with LBS privacy protection, and we interpret them from three aspects: Technology, architecture, and continuity.

2.1. Technologies of LBS Privacy Preservation

From the perspective of technologies that most recent works proposed for protecting privacy in LBS, they can be roughly classified into two main categories: Obfuscation-based approach and cryptographic-based approach. Obfuscation-based approach is mainly used to confuse LBSP. For this purpose, one way is to distort information based on the original and real users themselves [10–12], such as adding noise in the contents of users' requests or expanding the area of users' real locations. Another way is to use some methods to confuse LBSP through a combination with other users, such as Mix zone [13], spatial cloaking [3], temporal cloaking [14,15], and Dummy [16,17]. Cryptographic-based approach is mainly used to make users' information (include location, preference, etc.) invisible for adversaries when users send requests to LBSP. Recently, Homomorphic Encryption for privacy protection is a research hotspot, which can prevent the disclosure of users' information and achieve a more stringent privacy protection. For example, in paper [18], the POI (i.e., the point of interest) could be calculated by LBSP without actually knowing users' locations. The BV Homomorphic Encryption (BVHE) was used in paper [8] to calculate users' request service vector in aggregation process. However, though a stringent privacy protection can be provide by BVHE, service efficiency cannot be ignored. Thus, how to improve service efficiency while protecting users' privacy (i.e., location privacy, preference privacy) is still an important problem that needs to be addressed.

2.2. Architectures of LBS Privacy Preservation

There are two main categories proposed by most recent works for protecting privacy in LBS from the perspective of the system architecture: TTP-based approach [19] and TTP-free approach [20]. TTP-based approach takes an anonymizer as a trusted third part (TTP), and the anonymizer usually plays a role to confuse LBSP. The anonymizer often adopts some technologies, such as Cloaking and Dummy, to protect users' information before sending the requests to LBSP [21,22]. TTP-free approach does not rely on a trusted third part, users usually adopt some methods, such as P2P [23,24] or encounter scheme [5,25], to achieve k-anonymity, and users' information is often exchanged to confuse LBSP. It is really hard to say which of the two approaches mentioned above is better. TTP-based approach can be applied to almost all scenarios no matter users' distribution. However, the disadvantage is also obvious, that is the third part can be the single point of failure and the bottleneck of the whole system. The amount of private information exposed to a third part can be decreased by adopting TTP-free approach. However, due to the limitation of communication range of users' mobile devices, TTP-free approach may not satisfy the requirement for every mobile user. But with the development of smart mobile and communication technology, the application of TTP-free approach would be more extensive due to the advantages of distributed architecture.

2.3. Continuity of LBS Privacy Preservation

Sporadic privacy protection [8] and continuous privacy protection [26,27] are two important research respects. Sporadic privacy protection is often used for the case that users do not frequently use LBS services for a period of time. The schemes of sporadic privacy protection mainly focus on users' single process of privacy protection with LBSP, and apply to the situations when the degree of association between the two requests is low. Such as paper [28], a scheme named precision-based LPPM was proposed to reduce the precision of users' locations sent to LBSP. Continuous privacy protection is represented by trajectory privacy protection, which treats the problem of privacy protection from the view of users' trajectory [29]. The schemes of continuous privacy protection usually take a user's locations of a certain period as the object of protection, thus an adversary could use the spatial and temporal correlations on users' locations to infer their private information.

3. Preliminaries

In this section, we give the basic concepts and meaning of symbols.

3.1. BV Homomorphic Encryption

BV homomorphic encryption (BVHE) is based on the Ring Learning With Errors (RLWE) problem [30], and supports both additive homomorphisms and multiplicative homomorphisms. The main algorithms of BVHE include key generation (KEYGEN), encryption (ENC), evaluate (EVAL), and decryption (DEC). The details are as follows.

- KEYGEN: For some given parameters n, prime q, error parameter δ, and $f(x) = x^n + 1$. n is the power of 2, and $q \equiv 1 \mod 2n$. The Ring $R = Z[X]/\langle f(x)\rangle$ and the Ring $R_q = Z_q[x]/\langle f(x)\rangle$ can be selected by using n, q and $f(x)$. A discrete Gaussian error distribution $\chi = D_{Z^n,\delta}$ can be defined by taking δ as the variance, where the integer $D > 0$. The message space $R_t = Z_t[x]/\langle f(x)\rangle$ can also be defined by choosing prime $t \in Z_q^*$. Under the setting above, a public-private key pair (pk, sk) can be generated by randomly choosing (s_i, e_i) from χ and a_1 from R_q, where $sk_i = s_i$, $pk_i = (a_0, a_1)$, and $a_0 = -(a_1 s_i + te_i)$.
- ENC: For a plaintext $m \in R_q$ that needs to be encrypted, the encryption process (i.e., $C = E_p k(m)$) is as follows: The ciphertext $C = (c_0, c_1)$ can be calculated by choosing samples $u \leftarrow \chi, and\ f, g \leftarrow \chi$, where $c_0 = a_0 u + tg + m$ and $c_1 = a_1 u + tf$.

- EVAL: Taking additive homomorphisms as an example. For some given plaintexts m_1, m_2, and m_3. If $m_3 = m_1 + m_2$, then the Formula (1) is hold.

$$DEC(ENC(m_3)) = DEC(ENC(m_1) + ENC(m_2)). \quad (1)$$

- DEC: For a ciphertext m that needs to be decrypted, the decryption process (i.e., $m = D_{sk}(C)$) is as follows: The corresponding plaintext of m can be calculated by $m = (c_0 + c_1 \times s_i) \mod t$.

3.2. Service Category

In this paper, the service requests of users are classified into different kinds of service category. These different kinds of service category are distinct and have no semantic relationship, which means they are not semantically similar [31]. Such as banking, restaurant, bus station, post office, and supermarket, etc. But different from the paper [8,9] and others, we use the service category table to denote the various kinds of service category. The details of the service category table are as follows.

One kind of service category, which is symbolized by L_i in this paper, is represented by a binary number of M bits. According to the kinds of service category, the binary number of M bits are divided into different units, which is symbolized by CU. Each CU is composed of B bits, in which the last bit is 1, and the other bits are 0. Under the setting above, the total kinds of service category, which is symbolized by L, can be represented in a service category table, and L also can be computed as $L = M/B$. B must be divisible by M, which means L must be an integer, and the relation between L and B should meet the condition that $2^B > L$. Note that L is usually used to represent a unified threshold of l-diversity (i.e., th_l) in a region. For example, when the total kinds of service category $L = 7$, it can be represented by a service category table as shown in Figure 2.

L_1	000	000	000	000	000	000	**001**
L_2	000	000	000	000	000	**001**	000
L_3	000	000	000	000	**001**	000	000
L_4	000	000	000	**001**	000	000	000
L_5	000	000	**001**	000	000	000	000
L_6	000	**001**	000	000	000	000	000
L_7	**001**	000	000	000	000	000	000

Figure 2. Service Category Table when $L = 7$.

The CU in the service category table above is 001, which is composed of 3-bit binary numbers. The maximum value of the CU is 111, which means the redundancy of the CU is 7, and it allows 7 of the same kinds of service category to add. Of course, when the total kinds of service category that need to be represented are bigger than 7, which means the unified threshold $th_l > 7$ in a region. We only need to redefine the table structure according to the nature of binary numbers. For example, if $L = 10$, then the corresponding CU of this service category table is 0001. Obviously, the CU of this service category table could be 00001 or 000001 and so on, but we follow the principle of minimization.

3.3. Calculation of Service Categories

Under the setting above, we explain how to figure out the different kinds of service category. Here, we also take the total kinds of service category $L = 7$ as an example.

When all bits of a CU are 1, it represents the arithmetic element of a L_i. The arithmetic element is symbolized by AE_i in this paper. Figure 3 shows the corresponding AE_1 of L_1 when $L = 7$.

000	000	000	000	000	000	**111**

Figure 3. AE_1 of L_1 when $L = 7$.

Assume that a random sum $SL_i = \sum_{i=1}^{k} L_i$, which is obtained by adding some different L_i together when $L = 7$. As shown in Figure 4.

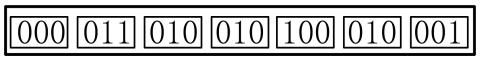


Figure 4. A random sum of L_i when $L = 7$.

Then, we can know whether there is an ith kind of service category by $SC_i = SL_i \bigcap AE_i$. If the bits in the SC_i are all 0, it means there is no ith kind of service category in SL_i. Otherwise, the count number, which is symbolized by NSC and used for counting the kinds of service category, should be added by 1. Of course, the initial value of NSC is 0. Though the final result of NSC, we can figure out how many different kinds of service category in SL_i. From the Figure 4, we can know the final result of NSC is 6, which means there are 6 kinds of service category (i.e., $l = 6$). Note that, the number of different L_i can also be figured out from Figure 4, which is 14. Therefore, in sporadic privacy protection, under the premise that one user sends one kind of service category in a clustering process, we could compare the number of L_i with the number of users in a clustering group, then whether there is an occurrence of overflow can be determined (i.e., the number of L_i equals the number of users normally).

3.4. The Relationship of k and l

As we all know, k-anonymity and l-diversity are classical techniques used to protect privacy in LBS. Many studies and privacy approaches were proposed based on k-anonymity and l-diversity in sporadic privacy protection. The combination of k-anonymity and l-diversity is to ensure that k users request at least l services in a clustering process of LBS query, which is shown as Figure 5. But the relationships between k and l are needed to be rigorously explored.

Theorem 1. *In a clustering process, the number of diversity is not more than the number of anonymity: $l \leq k$.*

Proof of Theorem 1. In sporadic privacy protection, with the premise that one user sends one kind of service category in a clustering process. As shown in Figure 5. One user corresponds to one kind of service category, and some users may have the same kind of service category. Therefore, the number of diversity is not more than the number of anonymity (i.e., the number of users) in a clustering group. □

Theorem 2. *In a clustering process, k-anonymity is easier to be satisfied than l-diversity.*

Proof of Theorem 2. In sporadic privacy protection, with the premise that one user sends one kind of service category in a clustering process. Under the system architecture of TTP-free, assume that there already has a clustering group which does not yet achieve the conditions of clustering success (i.e., $k < th_k$ and $l < th_l$). The condition th_k is easier to be satisfied than the condition th_l, the reason is that finding a new user is easier than finding a new user who must be with a different kind of service category (i.e., different l) from that clustering group. □

Theorem 3. *Under the same condition th_k, bigger th_l can provide better protection in LBS query.*

Proof of Theorem 3. By comparing Figures 5 and 6, we can see that the condition th_k is both 5, the condition th_l of Figure 5 is 3, and the condition th_l of Figure 6 is 4. Assume that the kinds of service category l_a, l_b, l_c are the same in Figures 5 and 6, if an attacker owns some background knowledge, and he can ensure Alice absolutely not request l_a and l_b. Then the attacker may deduce which is Alice from Figure 5 by using his background knowledge. But under the same setting, he cannot ensure which is Alice from Figure 6. □

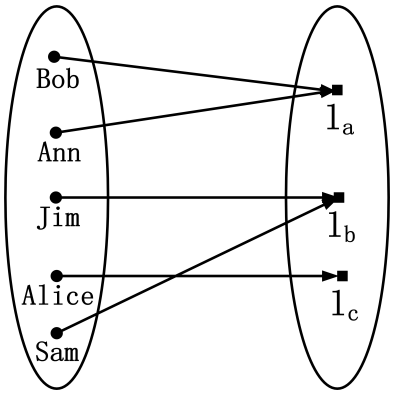


Figure 5. $th_k = 5$ and $th_l = 3$.

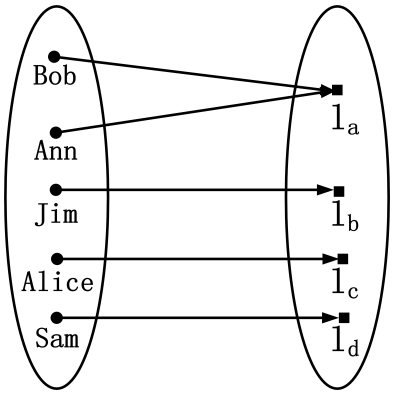


Figure 6. $th_k = 5$ and $th_l = 4$.

4. Motivation and System Model

4.1. Motivation

As we all know, according to the characteristic of homomorphic encryption, if there are too many multiplications on ciphertext, the computational complexity will be high and the accuracy of the original message will be affected by the noise generated during multiplications process. Different from paper [8], which adopted multiplicative homomorphism to calculate the kinds of service category in the clustering process. In this paper, we adopt additive homomorphism to calculate the kinds of service category, the noise as well as the computational complexity can be reduced.

In real applications, under the condition of specific time and space, users usually have common characteristics about interests or demands, so the kinds of service category that users request for LBS may be similar or even identical. For example, suppose that the site is an office building and the period is just off work time, the kinds of service category that users request are more likely as a restaurant or a supermarket rather than a post office or a bank (i.e., these departments or places are no longer within normal working hours). In addition, there is almost not only one clustering group in a region under normal circumstances. From a global perspective, the clustering process should be selective to improve the whole success rate of clustering in this region. The followings are two different scenarios that often appear.

Scenario 1: As shown in Figure 7, assume that there are many clustering groups in region LA_0, the th_k is 6 and the th_l is 4. If the clustering group CP_1 already contains 6 users and the kinds of service category are 3, which means $k \geq th_k$ but $l < th_l$. When the new user U_a wants to participate in the clustering process of CP_1, U_a's kind of service category is the same as one of those kinds of service category in CP_1. We can know that the participating of U_a cannot help CP_1 to complete clustering, on the contrary, it just increases the load of CP_1 and influences the success rate of other clustering

groups in region LA_0. This scenario is what we define as a relatively dense scenario, which is more likely happen in a region with a dense population of specific time and space, rather than in an area with dense population inevitably.

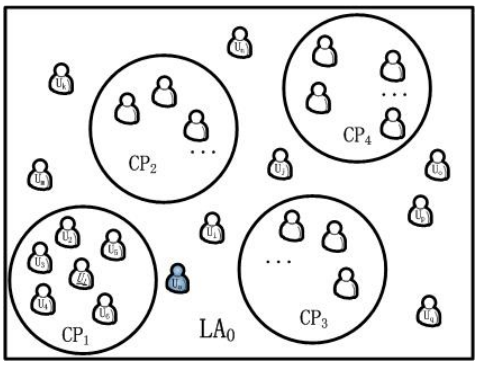


Figure 7. Scenario 1.

Scenario 2: As shown in Figure 8. Assume that there are two clustering groups in region LA_1, the th_k is 4 and the th_l is 3. We also assume that the user U_a is the initiator of the clustering group CP_1, which contains 3 users, and the kinds of service category are 2. The user U_d is the initiator of the clustering group CP_2, which contains 2 users, and the kinds of service category are 2. If there are no other users who want to join in the two clustering groups above (i.e., CP_1 and CP_2) within a certain period of time (i.e., the waiting time before the termination time), both the two clustering process cannot complete clustering, and only wait for the termination time of the two clustering groups. This scenario is what we define as a relatively sparse scenario, which is more likely happen in an area with a sparse population of specific time and space, rather than in an area with sparse population inevitably.

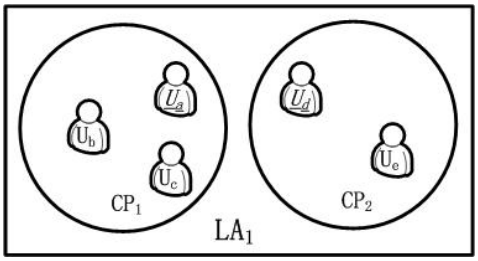


Figure 8. Scenario 2.

4.2. System Model

The system model of this paper is mainly composed of three parts: Service Category Server, Mobile Users, and Location-Based Service Provider. The framework is shown as Figure 9. The system architecture we adopt in this paper is TTP-free, which means there is no trusted anonymizer as an intermediary in our system model, and users can communicate with each other within a distance of communication. The details of our model are as follows.

4.2.1. Service Category Server (SCS)

It has four missions in this paper. Firstly, SCS should compute and prepare pseudo-IDs $PID_i = \{PID_1, PID_2, ...\}$ for users registration, the details can be referred to the paper [8]. Secondly, SCS generates the parameters (n, q, δ, R, R_q) of BVHE, which enable users to generate a public-private key pair (pk, sk). Thirdly, according to the different th_l in different regions, SCS generates the corresponding service category table. Fourthly, SCS regularly updates the corresponding relationship between the kind of service category and binary number in the table, and users' preference privacy can

be well protected in LBS query by this way. Note that how to make reasonable value th_l in different regions is out of our discussion, the reasonable value th_l in a region may be obtained from the analysis of big data or other approaches, but in this paper, we should make sure that there are only one pair thresholds (i.e., th_k, th_l) in one region.

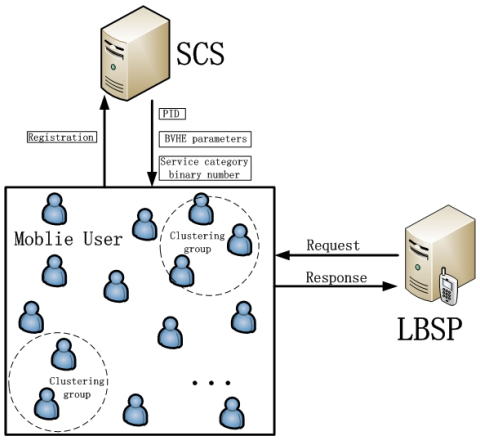


Figure 9. System model.

4.2.2. Mobile Users

If a user wants to obtain service from the LBSP in a region, he must register himself to the SCS, and there are also three missions for the user. Firstly, he gets his pseudo-ID PID_i from SCS, and the pseudo-ID PID_i is unique and different from other users. Secondly, according to the BVHE parameters (n, q, δ, R, R_q), the user generates his own public-private key pair (pk_i, sk_i) as explained in Section 3.1. Thirdly, the user should ask for the corresponding kind of service category according to his preference from SCS at this moment. Note that, in this paper, there is not an anonymizer as a trusted intermediary who passes information between users and LBSP, and no dummy in the clustering process. Therefore, if a user wants to obtain service from the LBSP, he should firstly aggregate other users' requests to meet the conditions of k-anonymity and l-diversity. Then these requests would be sent together to the LBSP by their representative. In the clustering process of these users, when the clustering group finally achieves the condition of success (i.e., $k \geq th_k$ and $l \geq th_l$), the representative of the clustering group firstly will inform these users to prepare their request packets. Then he will repackage their packets as an aggregated package. Finally, he will send the aggregated package to the LBSP of this region.

4.2.3. Location Based Service Provider (LBSP)

Without loss of generality, the main role of LBSP is to process users' request packets and return a list of results to users. Note that there are two main ways to return the results back to these users. One way is to return the results to the representative user, then he distributes the results to his member users. The other way is to return the results to his member users directly by their pseudo-IDs, which is adopted in this paper to reduce the load of the representative user.

In our system model, all roles must follow the relevant regulations and agreements in the LBS system, which means they are all $honest - but - curious$. For LBSP, it is curious and wants to deduce users' locations and preferences (i.e., their kinds of service category). For a user, he also has curiosity and hopes to deduce other users' request contents including their locations and preferences. But the collusion among users and the collusion between users and LBSP are out of the scope of discussion in this paper.

5. Algorithm Design

In this section, we firstly describe the scheme of SCTB. Then SCTB proposed in this paper is explained, which includes request aggregation protocol, variety judgment protocol, filtering

aggregation protocol, and representative aggregation protocol. Lastly, we make a summary of our service category table based (SCTB) algorithm.

5.1. The Algorithm Scheme

For the security parameters th_k and th_l in a region, the service category server bootstraps the LBS system and initializes it. If a user wants to obtain service from the LBS system, he must register himself to the LBS system. Before sending a request to the LBSP, he should firstly unite with other $k-1$ users to become a clustering group by using the request aggregation protocol. Then the kinds of service category l in the clustering group can be calculated by the variety judgment protocol. The representative user of the clustering group firstly should judge the relationship between l and th_l. If $l \geq th_l$, the representative user will judge the relationship between k and th_k. Then there will be two results: If $k \geq th_k$, which means the clustering group has met the conditions of clustering success, the clustering process is ended. If $k < th_k$, which means the clustering group has not met the condition of clustering success, the clustering group will wait for new users to join to meet the condition (i.e., $k \geq th_k$) until timeout. If $l < th_l$, which means the clustering group has not met the condition of clustering success, the representative user will choose filtering aggregation protocol or representative aggregation protocol by judging the relationship between k and th_k for improving the success rate of clustering. Note that, if a clustering group does not meet the conditions (i.e., $k \geq th_k$ and $l \geq th_l$) and runs into a timeout, the clustering process of this group is a failure, and no service request will be sent to LBSP. The scheme of our SCTB algorithm is shown as Figure 10.

5.2. Request Aggregation Protocol

Generally speaking, assume that a user U_1 is not in a clustering group of other users and wants to launch a request to the LBSP in his region. U_1 should unite with other $k-1$ users to form a clustering group. Algorithm 1 describes the pseudo code of the request aggregation protocol. The detailed explanation is as follows.

User U_1 firstly broadcasts the request aggregating message and sets the termination time T, and T starts counting down. If there are some users who want to join in the clustering process of U_1, these users should return their pseudo-IDs to U_1. Then U_1 will count the number k of users who have responded. If the number (include U_1 himself) $k \geq 2$, U_1 takes himself as the representative of the clustering group, which is symbolized by PID_1^R, he names the clustering group as PID_1^{CP1}. When the number of sums SL_i is obtained by one clustering group (i.e., this sum is obtained only from PID_1^{CP1}), the number of groups $NCG_{PID_1^R} = 1$. After the setting above, PID_1^R firstly uses his public key pk_1 to encrypt his kind of service category L_i^1. Then PID_1^R gives his pk_1 and $pk_1(L_i^1)$ to members of this clustering group. Finally, the members use pk_1 to encrypt their kinds of service category and return the result $pk_1(SL_i) = pk_1(L_i^1) + pk_1(L_i^2) + ... + pk_1(L_i^k) = pk_1(L_i^1 + L_i^2 + ... + L_i^k)$ to U_1, where $SL_i = \sum_{n=1}^{k} L_i^n$, $1 \leq i \leq th_l$. PID_1^R names this result of the clustering group as $pk_1(SL_i)^{CP1}$, which is the encrypted sum of kinds of service category in PID_1^{CP1}. PID_1^R firstly uses his private key sk_1 to decrypt $pk_1(SL_i)$ and gets SL_i. Then PID_1^R uses variety judgment protocol to calculate the kinds of service category l in PID_1^{CP1}. Note that here the kinds of service category $l \geq 2$. Otherwise, U_1 would know other users' kinds of service category in the current clustering group. There will be four cases as follows, and here assume that $T > 0$.

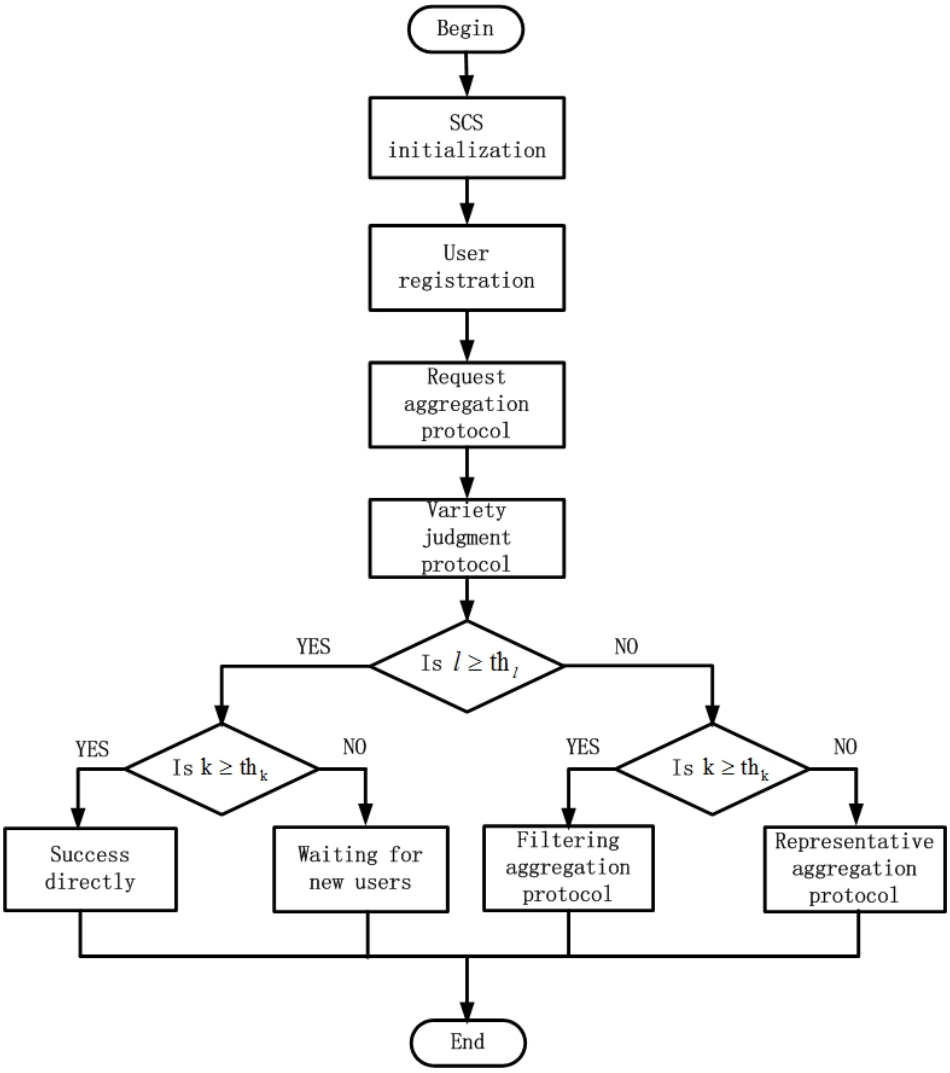


Figure 10. The scheme of service category table based algorithm (SCTB).

Case 1: $k \geq th_k$ and $l \geq th_l$. U_1 firstly notifies the members in PID_1^{CP1} to prepare their inputs based on their kinds of service category. Then he repackages their packets and gets an aggregated package, which is symbolized by Ag. As the respective, U_1 sends Ag to the LBSP.

Case 2: $k < th_k$ and $l \geq th_l$. As explained in Section 3.4, k-anonymity is easier to be satisfied than l-diversity. So waiting for new users to join is a better choice for PID_1^{CP1}. Under this situation, U_1 does not need to judge new users' kinds of service category, and just wait for new users' joining to meet the condition th_k.

Case 3: $k \geq th_k$ and $l < th_l$. We know this case is the same as scenario 1, the filtering aggregation protocol will be carried out. The reasons why we choose it will be explained when we describe it.

Case 4: $k < th_k$ and $l < th_l$. We know this case is the same as scenario 2, the representative aggregation protocol will be carried out. The reasons why we choose it will be also explained when we describe it.

Algorithm 1 Requests aggregation

Input: Initiator, Service category table, Users
Output: Representative, l, k

Initiator U_1 broadcasts the request aggregating message, and sets the termination time T;
if $T > 0$ **then**
  $\{PID_1, PID_2, ..., PID_k\} \rightarrow U_1$;
  U_1 gets the number of users k;
  if $k \geq 2$ **then**
    The representative = PID_1^R, the group name = PID_1^{CP1}, $NCG_{PID_1^R} = 1$;
  else
    The clustering process is a failure;
  end if
  pk_1 and $pk_1(L_i^1) \rightarrow PID_2$
  pk_1 and $pk_1(L_i^1) + pk_1(L_i^2) \rightarrow PID_3$
  ...
  finally $pk_1(SL_i) \rightarrow PID_1^R$;
  PID_1^R names $pk_1(SL_i)$ as $pk_1(SL_i)^{CP1}$, and uses sk_1 to get SL_i;
  PID_1^R uses variety judgment protocol, and gets l;
  if $l \geq 2$ **then**
    if $k \geq th_k$ and $l \geq th_l$ **then**
      The clustering process is a success, the remaining work will be performed;
    end if
    if $k < th_k$ and $l \geq th_l$ **then**
      Waiting for new users to join;
    end if
    if $k \geq th_k$ and $l < th_l$ **then**
      Filtering aggregation protocol is adopted;
    end if
    if $k < th_k$ and $l < th_l$ **then**
      Representative aggregation protocol is adopted;
    end if
  else
    The clustering process is a failure;
  end if
end if

5.3. Variety Judgment Protocol

The purpose of this variety judgment protocol is to figure out the kinds of service category l in the clustering process. When there is only one sum SL_i in a clustering group, we know the $NCG_{PID_1^R} = 1$ just as the illustration in request aggregation protocol. After PID_1^R gets SL_i from PID_1^{CP1}, he would use the way as explained in Section 3.3 to figure out the result of NSC, which represents the kinds of service category l in the clustering process of PID_1^{CP1} at this moment. When there is not only one sum SL_i in this clustering group, such as two sums, then the representative sets $NCG_{PID_1^R} = 2$, and this case will be explained in representative aggregation protocol. After PID_1^R gets the combined sum SL_i from two clustering groups, he also uses the way as explained in Section 3.3 to figure out the result of NSC. Algorithm 2 describes the pseudo code of the variety judgment protocol.

Algorithm 2 Variety judgment

Input: PID_1^R,PID_1^{CP1}, SL_i, $NCG_{PID_1^R}$,
Output: l
1: **if** $NCG_{PID_1^R} = 1$ **then**
2: PID_1^R figures out the result of NSC from SL_i in PID_1^{CP1};
3: **end if**
4: **if** $NCG_{PID_1^R} > 1$ **then**
5: PID_1^R figures out the result of NSC from combined sum SL_i;
6: **end if**

5.4. Filtering Aggregation Protocol

According to the given security parameters th_k and th_l in a region, when a clustering group has already met the condition th_k but not met th_l. If the success of this clustering group (i.e., meet the conditions th_k and th_l) needs to be promoted, the clustering group should make the new users who want to join in the clustering group have different kinds of service category rather than those kinds of service category that the clustering group already has. Therefore, the purpose of our filtering aggregation protocol is to choose new users selectively. There are two reasons for adopting this method: Firstly, it is no use for increasing the success probability of the clustering group when the new users have the same kinds of service category as the clustering group already has, on the contrary, it just increases the load of the clustering group. For example, suppose that the $th_k = 8$ and $th_l = 5$ in a region, and a clustering group now under the conditions that k = 8 and l = 3. Assume that the clustering group meets the conditions th_k and th_l finally, k = 15 and l = 5 may be the final result of the clustering group, which obviously generates more traffic and computation load than the result k = 10 and l = 5 by using our filtering aggregation protocol. And it is more effective especially when users' kinds of service category in LBS query are similar or even identical under a certain time and space, which will be proved in the Section 6. Secondly, it will influence the global success rate of clustering groups in this region. In the condition of the same th_k and th_l, if a clustering group contains too many users, obviously, other clustering groups cannot cluster successfully due to lack of users.

From the variety judgment protocol, we can know the number of l and k in PID_1^{CP1}. Assume that the case is $k \geq th_k$ and $l < th_l$, when a new user PID_{new} wants to join in PID_1^{CP1}. The representative PID_1^R firstly gives his pk_1 and $pk_1(SL_i)$ to PID_{new}. Then PID_{new} uses pk_1 to encrypt his kind of service category L_i^{new} and obtains the intermediate result $pk_1^{inter}(SL_i) = pk_1(SL_i) + pk_1(L_i^{new})$. Finally, PID_{new} returns $pk_1^{inter}(SL_i)$ to PID_1^R, and PID_1^R recalculates the current value of NSC, which is symbolized by NSC^{inter}. If $NSC^{inter} > NSC$, we know the new user's kind of service category is different from the kinds of service category that PID_1^{CP1} already has, and it is helpful to promote the success of PID_1^{CP1}. Therefore, the representative PID_1^R allows the new user PID_{new} to join. Otherwise, PID_1^R refuses PID_{new}. Finally, if the current $l \geq th_l$, the clustering process is a success, or else PID_1^{CP1} waits for more new users to join. Algorithm 3 describes the pseudo code of the filtering aggregation protocol.

Security analysis: It will be explained from two sides in the clustering process.

1. From the perspective of the single newly-joining user who wants to join in PID_1^{CP1}: Because PID_1^R sends the encrypted result $pk_1(SL_i)$ to the single newly-joining user PID_{new}, PID_{new} cannot infer the existing kinds of service category contained in PID_1^R unless he can decrypt $pk_1(SL_i)$.
2. From the perspective of the clustering group: If the user PID_{new} submits the kind of service category which is different from those kinds of service category that PID_1^{CP1} already has (i.e., $NSC^{inter} > NSC$), the representative U_1 cannot know the exact kind of service category because of the different L_i.

If the user PID_{new} submits the kind of service category which is the same as one of these kinds of service category that the PID_1^{CP1} already has (i.e., $NSC^{inter} = NSC$), there will be two cases as follows.

(a) If the kind of service category of user PID_{new} is same as the representative U_1, U_1 will know the kind of service category of user PID_{new} because of the same L_i;
(b) If the kind of service category of user PID_{new} is not the same as the representative U_1, which means it is the same as one of other users in PID_1^{CP1}, the representative U_1 cannot know the exact kind of service category of user PID_{new} because of the different L_i.

But due to the regulation of the filtering aggregation protocol, if $NSC^{inter} = NSC$, the representative U_1 has to refuse PID_{new}. Therefore, he cannot do further exploration for user PID_{new}. Because the corresponding relationship between the kind of service category and binary number is not unchangeable, although the representative U_1 has deduced the meaning of L_i of PID_{new} in this clustering process, he cannot make sure that the meaning of L_i is the same as previous or subsequent clustering process.

Algorithm 3 Filtering aggregation

Input: New users, $pk_1(SL_i)$
Output: New l
1: **while** $T > 0$ and $l < th_l$ **do**
2: PID_1^R continue broadcasting the request aggregating message, the new user PID_{new} wants to join;
3: pk_1 and $pk_1(SL_i) \rightarrow PID_{new}$
$pk_1^{inter}(SL_i) \rightarrow PID_1^R$;
4: **if** $NSC^{inter} > NSC$ **then**
5: The join of PID_{new} is agreed;
6: **else**
7: The join of PID_{new} is refused;
8: **end if**
9: **end while**

5.5. Representative Aggregation Protocol

According to the given security parameters th_k and th_l in a region, both th_k and th_l are not met in an original clustering group. If the success of this clustering group needs to be promoted, the method is to combine with other clustering groups that have the same conditions (i.e., $k < th_k$, $l < th_l$). The reason is that the cooperation among clustering groups could have a higher success probability to promote the success of the original clustering group than just waiting for new users to join. Besides, if other clustering groups nearby are found finally, the time cost of clustering will also be reduced. When the users' kinds of service category are similar or even identical in a certain time and space, for the original clustering group, newly-joining users are more likely to contribute to the value of k rather than the value of l. Note that our method is not to prevent new users from joining, but to find other clustering groups as the first choice.

From the variety judgment protocol, we can know the number of l and k in PID_1^{CP1}. Assume that the case is $k < th_k$ and $l < th_l$. PID_1^R begins to broadcast the representative aggregating message, but different from request aggregating message, the representative aggregating message can only be responded by representatives. Suppose that there is a clustering group PID_b^{CP2} with the same conditions (i.e., $k < th_k$, $l < th_l$) as PID_1^{CP1}, and its representative PID_b^R responds to PID_1^R. Firstly, PID_1^R will send his pk_1 to PID_b^R, then PID_b^R sends his current set of pseudo-IDs and encrypted sum $pk_1(SL_i)$ back to PID_1^R. Note that this SL_i is from decrypted $pk_b(SL_i)^{CP2}$. Then, PID_1^R will

set $NCG_{PID_1^R} = 2$ and recalculate the value of NSC. Finally, the current NSC is figured out by the combined sum which is obtained by adding SL_i of PID_1^{CP1} and SL_i of PID_b^{CP2}. According to the relationship of current k and l between th_k and th_l, PID_1^R chooses the different protocol mentioned above to finish the subsequent process. Algorithm 4 describes the pseudo code of the representative aggregation protocol.

Algorithm 4 Representative aggregation

Input: New clustering groups, $pk_1(SL_i)$
Output: New l, New k
1: **while** $T > 0$, $k < th_k$ and $l < th_l$ **do**
2: PID_1^R broadcasts the representative aggregating message, then the PID_b^R responds;
3: $pk_1 \rightarrow PID_b^R$
4: The pseudo-IDs set of PID_b^R and $pk_1(SL_i) \rightarrow PID_1^R$;
5: $NCG_{PID_1^R} = 2$, and PID_1^R gets current k and l from combined sum SL_i;
6: **if** $k \geq th_k$ and $l \geq th_l$ **then**
7: The clustering process is a success, the remaining work will be performed;
8: **end if**
9: **if** $k < th_k$ and $l \geq th_l$ **then**
10: Waiting for new users to join;
11: **end if**
12: **if** $k \geq th_k$ and $l < th_l$ **then**
13: Filtering aggregation protocol is adopted;
14: **end if**
15: **if** $k < th_k$ and $l < th_l$ **then**
16: Representative aggregation protocol is adopted;
17: **end if**
18: **end while**

5.6. Service Category Table Based Algorithm

Algorithm 5 summarizes the strategies of the service category table based (SCTB) algorithm in this paper. If the user U_i wants to send a request to the LBSP. Firstly, he should aggregate other $k-1$ users by using the request aggregation protocol. If there are some users who agree to form a clustering group together with the user U_i, he will become the representative. Secondly, the representative uses variety judgment protocol to figure out the kinds of service category in the clustering group. Finally, from the comparison between (k,l) and (th_k,th_l), U_i chooses different strategies according to the following four kinds of situations:

(i) $k \geq th_k$ and $l \geq th_l$, the clustering process is a success;
(ii) $k < th_k$ and $l \geq th_l$, waiting for new users to join;
(iii) $k \geq th_k$ and $l < th_l$; using filtering aggregation protocol for the subsequent processing;
(iv) $k < th_k$ and $l < th_l$, using representative aggregation protocol for the subsequent processing.

Algorithm 5 Service Category Table Based (SCTB) algorithm

Broadcast the aggregating message;
Aggregate uers' requests by using request aggregation protocol;
Figure out the current k and l by using variety judgment protocol;
if $k \geq th_k$ and $l \geq th_l$ **then**
    The clustering process is a a success, the remaining work will be performed;
end if
while $T > 0$ **do**
    if $k < th_k$ and $l \geq th_l$ **then**
        Waiting for new users to join;
    end if
    if $k \geq th_k$ and $l < th_l$ **then**
        Filtering aggregation protocol is adopted;
    end if
    if $k < th_k$ and $l < th_l$ **then**
        Representative aggregation protocol is adopted;
    end if
end while

6. Evaluation

In this section, the efficiency and effectiveness of our SCTB algorithm are experimentally evaluated. Firstly, we analyze the computational cost of the PLAM [8] and our SCTB on figuring out the kinds of service category in the clustering process. Then we describe the simulation environment and give the simulation results.

6.1. Performance Analysis

Assume that there is a clustering process which contains k_c users and a data-pool which contains l_c different kinds of service category, k_c and l_c are both random constant. Because PLAM, LLB, and our SCTB adopt the system architecture of TTP-free, and the kinds of service category in a clustering group are both figured out by the representative of the clustering group, the two algorithms (i.e., PLAM and LLB algorithm) are used for comparison. We assume that all conditions are the same and meet the premise that one user sends one kind of service category in a clustering process. For example, user Bob only sends one request about finding a restaurant in one clustering process, but he cannot send the request that contains both finding a restaurant and a hospital at the same time. We now give the computational cost of our SCTB and PLAM about the process of identifying and calculating l-diversity.

Claim 1. *In a clustering process, ignoring the computational complexity of decryption, the computational complexity of SCTB on figuring out l-diversity is* $O(k_cE) + O(l_c)$.

Proof of Claim 1. Firstly, the k_c users should add together to form $pk_1(SL_i)$, and pk_1 is the public key of the representative. We can know the computational complexity is $O(k_cE)$ [32]. Then we decrypt $pk_1(SL_i)$ and get SL_i, the NSC can be figured out by $SC_i = SL_i \bigcap AE_i$. Because the number of AE_i is equal to l_c, and l_c is the total kinds of service category that can be represented. Therefore, the computational complexity is $O(l_c)$. Ignoring the computational complexity of decryption, the computational complexity of this process is $O(k_cE) + O(l_c)$. □

Claim 2. *In a clustering process, ignoring the computational complexity of decryption, the computational complexity of PLAM on figuring out l-diversity is* $k_cO(l_c) + l_cO(k_cE^{k_c})$.

Proof of Claim 2. Under the same settings as above. PLAM adopts a set $a_i = (a_{i1}, a_{i2}, ..., a_{in})$ to denote different kinds of service category that available to a $user_i$, and the set also means the total kinds of service category that can be represented is n, where the number n is equivalent to l_c. Firstly, the representative should calculate $pk(\bar{a}_i) = (pk(1 - a_{i1}), pk(1 - a_{i2}), ..., pk(1 - a_{in}))$, and we know the computational complexity is $k_c O(l_c)$. Then users cooperatively exchange $pk(\bar{a}_i)$ and calculate $\prod_{i=1}^{k_c} pk(\bar{a}_i) = (\prod_{i=1}^{k_c} pk(1 - a_{i1}), ..., \prod_{i=1}^{k_c} pk(1 - a_{in}))$, and we know the computational complexity of $\prod_{i=1}^{k_c} pk(1 - a_{i1})$is $O(k_c E^{k_c})$. Therefore, the computational complexity of $\prod_{i=1}^{k_c} pk(\bar{a}_i)$ is $l_c O(k_c E^{k_c})$. Ignoring the computational complexity of decryption, the computational complexity of this process is $k_c O(l_c) + l_c O(k_c E^{k_c})$. □

According to above-mentioned analysis, we can know that the computational complexity of our SCTB is smaller than the PLAM. Therefore, it is helpful to finish the clustering process as well as reduce the communication overhead among users.

6.2. Performance Metrics

In order to evaluate the performance and effectiveness of the proposed algorithm, some metrics of this paper are explained in this subsection.

Clustering efficiency is a comprehensive evaluation for clustering groups in a region, and it is mainly composed of *response time* and *success rate* in this paper.

Response time is used to measure the time taken for a clustering group from initiating clustering to meet the conditions of k-anonymity and l-diversity (i.e., th_k and th_l) in a region. The smaller the response time, the more efficiency of a clustering group. It also means the faster a clustering group can send the aggregated package to LBSP.

Success rate is used to measure the ratio of the number of clustering groups that meet the conditions th_k and th_l to the number of all clustering groups in a region [3]. The higher the success rate, the more efficiency of cluster groups. It also means more users' requests can be sent to LBSP.

6.3. Simulation and Results

The essence of the SCTB algorithm is to improve clustering efficiency while protecting privacy. Therefore, the extensive simulations are conducted for evaluating the performance of our algorithm. In this section, we describe the details of our simulation environment, then give the simulation results and analysis.

6.3.1. Simulation Setup

We use Matlab R2018a to conduct our simulations, and run all algorithms on a local machine with an Intel Core-i5 2.5 GHz, 6 GB RAM, and Microsoft Windows 7 OS. Assume that there is a region A of size 10 km × 10 km with 100 × 100 locations. We construct a full-mesh network consisting of 100 × 100 nodes to simulate the locations in region A. To evaluate the effectiveness of our SCTB algorithm, we introduce the concept of service category similarity, which is symbolized by L_S in this paper. Suppose that under normal circumstances, the kind of service category of user A is not related to the kind of service category of user B, then the service category similarity between user A and user B is $L_S = 0$. Assume that under certain space-time conditions, the probability of user A's kind of service category is half the same as user B's, then we denote the service category similarity between user A and user B is $L_S = 0.5$. We perform the simulation in the following four different scenarios and average performance results were obtained for running each scenario for 30 min.

Scenario-1: There are 50 × 100 users with transmission radius of tr = 50 m randomly distributed in region A. Assume that the total kinds of service category in the data-pool are 16. All users should join in the clustering process, if a user U_a receives a request aggregating massage when he wants a LBS

query, he will response for it and join in that clustering group. Otherwise, U_a will broadcast the request aggregating message as an initiator, and wait for other users to join. A certain number of users who send requests at a time interval. The ratio of the certain number of users to the total number of users is $1/10$, and the time interval is $0.1s$. The rate of th_l to th_k is $th_l = th_k/2$. We set the service category similarity $L_S = 0$, which means the probability of service category similarity among users is 0%.

Scenario-2: We set the service category similarity $L_S = 0.5$, which means the probability of service category similarity among users is 50%. The rest of the conditions are the same as Scenario-1.

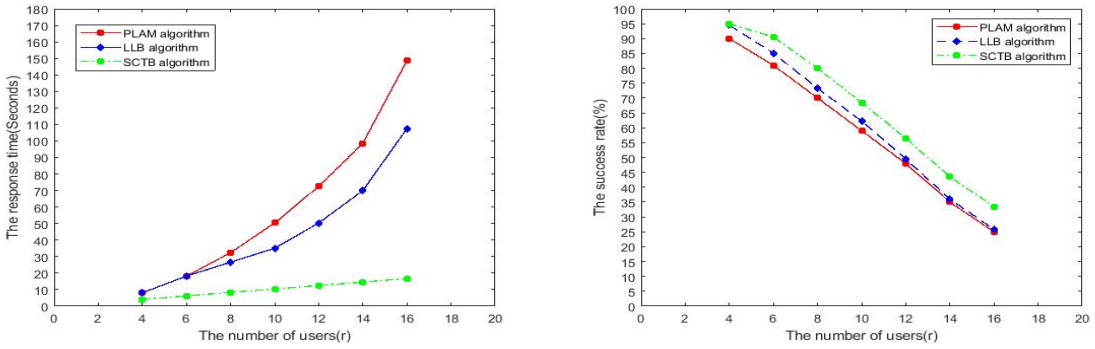
Scenario-3: We set the service category similarity $L_S = 0.75$, which means the probability of service category similarity among users is 75%. The rest of the conditions are the same as Scenario-1.

Scenario-4: Users move independently with the same velocity $v = 1m/s$, and a representative U_a sends clustering requests continuously in the $60s$ under different L_S. The rest of the conditions are the same as Scenario-1.

6.3.2. Simulation Results

We compare the other two algorithms (i.e., PLAM algorithm and LLB algorithm) with our SCTB algorithm. Figure 11a shows the simulation results of response time under Scenario-1. It can be seen that with the number of users in a clustering group increases, the response time increases no matter which kind of algorithm is used. It is because more user leads more time to be spent. It can also be known that the response time of our SCTB is less than PLAM and LLB. That is because the computational complexity of our SCTB is smaller than PLAM and LLB. The reason why the response time of LLB is less than PLAM is the combination of some clustering groups when some certain condition is satisfied. And the condition is as follows: When the number of users in clustering group A is less than $th_k/2$, the clustering group A can join in another clustering group B if and only if the clustering group B is within the communication range of A and the number of users in B is bigger than $th_k/2$ [9].

Figure 11b presents the relationship between the number of users and the success rate under Scenario-1. It can be seen that the success rate of our SCTB algorithm is higher than PLAM and LLB. That is because our SCTB algorithm leads the clustering process to choose new users selectively, and makes the clustering groups more reasonable by using the filtering aggregation protocol as analyzed in Section 5.4. Besides, the representative aggregation protocol promotes the success of clustering groups as well. The reason why the success rate of LLB is higher than PLAM is also the combination of some clustering groups. From Figure 11b, it also can be known that no matter which kind of algorithm mentioned above is used in the clustering process, the success rate decreases as the number of users in a clustering group increases. The reason is that with the number of users in a clustering group increases (i.e., th_k), the clustering group has to spend more time to find users to meet the condition th_k, which leads to the decrease of the success rate.

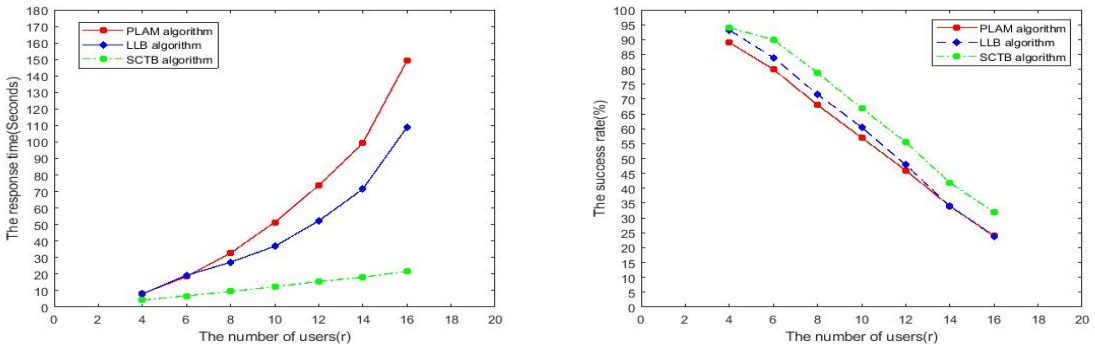


(a) The response time (b) The success rate

Figure 11. The simulation results for Scenario-1.

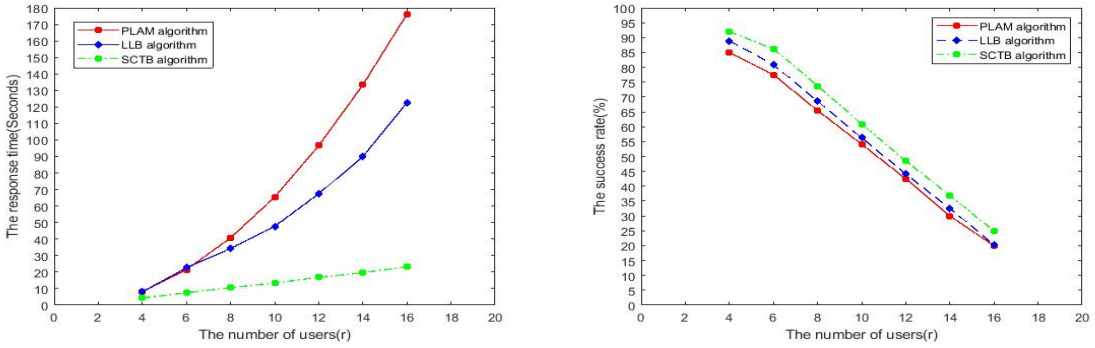
Figures 12 and 13 show the simulation results under Scenario-2 and Scenario-3 respectively. Figures 12a and 13a present the response time under Scenario-2 and Scenario-3 respectively. It can be seen that Figures 12a and 13a reflect a trend similar to Figure 11a, that is, the response time increases with the number of users in a clustering group increases. The response time of PLAM algorithm and LLB algorithm have almost the same performance in Scenario-1 and Scenario-2, which is influenced by the set ratio $th_l = th_k/2$. However, in Figure 13a, with the number of users in a clustering group increases, the response time of PLAM and LLB increases faster than Figures 11a and 12a. The reason is, for a clustering group, the higher service category similarity of users leads a clustering group has to find more users to meet the condition th_l, which will lead more time to be spent.

Figures 12b and 13b present the success rate under Scenario-2 and Scenario-3 respectively. It can be seen that Figures 12b and 13b show a trend similar to Figure 11b, that is, the success rate decreases as the number of users in a clustering group increases. The success rate of PLAM algorithm and LLB algorithm nearly has the same performance in Scenario-1 and Scenario-2 due to the ratio $th_l = th_k/2$. However, as can be seen from Figure 13b, with the higher service category similarity of users, our SCTB still has a better performance than PLAM and LLB. This is because our SCTB chooses new users selectively during the clustering process, and makes the clustering groups more reasonable in a region.



(a) The response time (b) The success rate

Figure 12. The simulation results for Scenario-2.

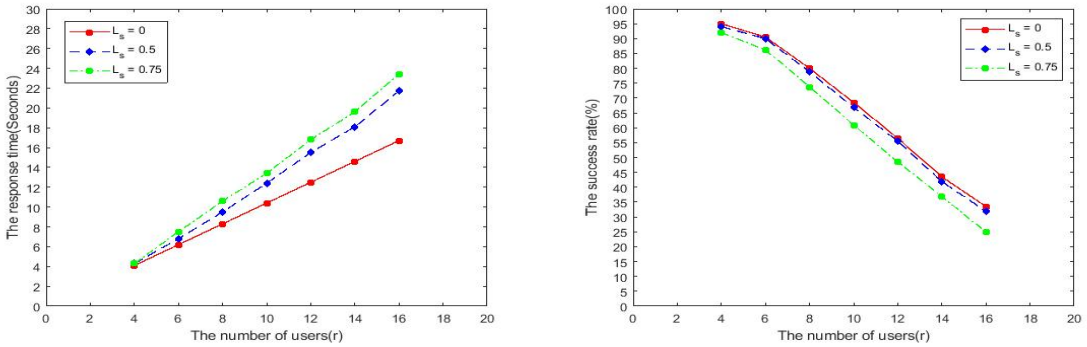


(a) The response time (b) The success rate

Figure 13. The simulation results for Scenario-3.

Figure 14 shows the performance of the SCTB under Scenario-1, Scenario-2, and Scenario-3. Figure 14a shows the response time of SCTB under different scenarios. It can be seen that the response time increases with the number of users in a clustering group increases. In the case of the same th_k, the higher the service category similarity, the longer the response time. Figure 14b presents the success rate of SCTB algorithm under different scenarios. It can be seen that the success rate decreases with

the number of users in a clustering group increases. In the case of the same th_k, the higher the service category similarity, the lower the success rate.



(a) The response time (b) The success rate

Figure 14. The simulation results of SCTB algorithm under different scenarios.

Figure 15 shows the number of clustering times when $th_k = 10$ under Scenario-4. It can be seen that the representative U_a could complete more number of clustering times by using our SCTB than PLAM and LLB. The reason is that the computational complexity of our SCTB is smaller than PLAM and LLB when U_a calculates the kinds of service category (i.e., l) in a clustering group. Therefore, it can allow U_a to spend less time to complete more clustering times. Figure 15 also shows that with the increase of service category similarity, no matter which algorithm mentioned above is used, the number of clustering times decreases. This is because the higher service category similarity means more users are needed to meet the condition th_l. Therefore, more time will be spent. However, we can see that our SCTB algorithm is still has a better performance than other two algorithms, so it is also useful in the scenario of a continuous request.

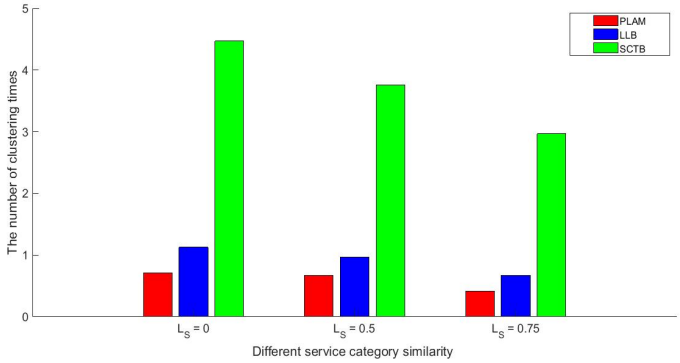


Figure 15. The simulation results for Scenario-4.

6.4. Limitations of Current Work

There are some limitations of our current work. Firstly, limitations of our simulation. We assume that the mobile devices of users have sufficient computing resources to complete calculations and communications, and the factors such as communication delays among devices are not considered as well. Secondly, limitations of our SCTB algorithm. In our paper, we assume that the kinds of service category are distinct and semantically dissimilar, which means they have no semantic relationship. Therefore, how to apply our approach to semantically similar scenarios more practically is still a challenge. However, our simulations prove that our algorithm is useful for the clustering process. Besides, the results match to theoretical development which was presented between Sections 3 and 6. Testing those algorithms (i.e., PLAM, LLB, and our SCTB) in a real environment is still a challenge.

That is because there are many uncertain factors, such as the strength of the network, different user habits, and different social conventions may cause different results.

7. Conclusions and Future Work

In this paper, we use k-anonymity and l-diversity in the distributed architecture of LBS system to study the problem of preference privacy and service efficiency. Different from previous studies, we take the relationship between k and l in the clustering process as the basis for judging users in a relatively dense scenario or relatively spare scenario. To promote the cooperative efficiency of users in the clustering process, especially when users' kinds of service category (i.e., users' preferences) have similarity in certain space-time conditions, we proposed a service category table based (SCTB) algorithm which contains four key protocols: request aggregation protocol, variety judgment protocol, filtering aggregation protocol, and representative aggregation protocol. The extensive simulation experiments are performed to evaluate the effectiveness of our SCTB algorithm and existing algorithms, and theoretical analysis and the simulation results show that our work is useful to improve the efficiency of clustering process while protecting users' privacy in LBS query.

In future work, we plan to study how to efficiently identify and calculate the kinds of l-diversity when users' kinds of service category are semantically similar in the clustering process, so as to expand the scope of application of our SCTB algorithm. Besides, we will test our approach in the real world environment to further increase our contributions.

Author Contributions: G.Y. proposed the idea and conceptualization. G.Y. performed experiments, data analysis and scientific discussions and writed the article. S.L., H.Z. and Y.X. revised the clarity of the work as well as helping to write and organize the paper. Finally, M.L. and Y.W. helped in the proper preparations, English corrections and submission.

Funding: This work was partially supported by the National Key R&D Program of China under Grant 2017YFB0802300, BUPT Excellent Ph.D. Students Foundation, Applied Sci-Tech R&D Special Fund Program of Guangdong Province (No.2015B010131007), National statistical scientific research project of China (2018LY61).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Gentry, C.; Ramzan, Z. Identity-Based aggregate signatures. In Proceedings of the International Conference on Theory and Practice of Public-Key Cryptography, New York, NY, USA, 24–26 April 2006; pp. 257–273.
2. Bettini, C.; Mascetti, S.; Wang, X.; Jajodia, S. Anonymity in Location-Based Services: Towards a General Framework. In Proceedings of the International Conference on Mobile Data Management, Mannheim, Germany, 1 May 2007; pp. 69–76.
3. Gedik, B.; Liu, L. Location Privacy in Mobile Systems: A Personalized Anonymization Model. In Proceedings of the IEEE International Conference on Distributed Computing Systems, Marina del Rey, CA, USA, 30 June–1 July 2005; pp. 620–629.
4. Machanavajjhala, A.; Gehrke, J.; Kifer, D.; Venkitasubramaniam, M. L-diversity: Privacy beyond k-anonymity. In Proceedings of the 22nd International Conference on Data Engineering (ICDE'06)(ICDE), Atlanta, GA, USA, 3–7 April 2006; p. 24.
5. Niu, B.; Li, Q.; Zhu, X.; Cao, G.; Li, H. Achieving k-anonymity in privacy-aware location-based services. In Proceedings of the IEEE INFOCOM, Toronto, ON, Canada, 27 April–2 May 2014; pp. 754–762.
6. Wang, Y.; Xia, Y.; Hou, J.; Gao, S.; Wang, Q. A fast privacy-preserving framework for continuous location-based queries in road networks. *J. Netw. Comput. Appl.* **2015**, *53*, 57–73. [CrossRef]
7. Niu, B.; Zhu, X.; Li, W.; Li, H.; Wang, Y.; Lu, Z. A personalized two-tier cloaking scheme for privacy-aware location-based services. In Proceedings of the International Conference on Computing, NETWORKING and Communications, Garden Grove, CA, USA, 16–19 February 2015; pp. 94–98.
8. Lu, R.; Lin, X.; Shi, Z.; Shao, J. PLAM: A privacy-preserving framework for local-area mobile social networks. In Proceedings of the IEEE INFOCOM, Toronto, ON, Canada, 27 April–2 May 2014; pp. 763–771.
9. Sun, G.; Liao, D.; Li, H.; Yu, H.; Chang, V. L2p2: A location-label based approach for privacy preserving in LBS. *Future Gener. Comput. Syst.* **2017**, *74*, 375–384. [CrossRef]

10. Quercia, D.; Leontiadis, I.; Mcnamara, L.; Mascolo, C.; Crowcroft, J. SpotME If You Can: Randomized Responses for Location Obfuscation on Mobile Phones. In Proceedings of the International Conference on Distributed Computing Systems, Minneapolis, MN, USA, 20–24 June 2011; pp. 363–372.
11. Ardagna, C.; Cremonini, M.; Damiani, E.; de Capitani di Vimercati, S.; Samarati, P. Location privacy protection through obfuscation-based techniques. In Proceedings of the Ifip Wg 11.3 Working Conference on Data and Applications Security, Redondo Beach, CA, USA, 8–11 July 2007; pp. 47–60.
12. Hwang, R.; Hsueh, Y.; Chung, H. A Novel Time-Obfuscated Algorithm for Trajectory Privacy Protection. *IEEE Trans. Serv. Comput.* **2014**, *7*, 126–139. [CrossRef]
13. Beresford, A.; Stajano, F. Location Privacy in Pervasive Computing. *IEEE Pervasive Comput.* **2003**, *2*, 46–55. [CrossRef]
14. Zhang, C.; Huang, Y. Cloaking locations for anonymous location based services: A hybrid approach. *Geoinformatica* **2009**, *13*, 159–182. [CrossRef]
15. Shokri, R.; Troncoso, C.; Diaz, C.; Freudiger, J.; Hubaux, J. Unraveling an Old Cloak: k-anonymity for Location Privacy. In Proceedings of the ACM Conference on Computer and Communications Security, Chicago, IL, USA, 4 October 2010; pp. 115–118.
16. Liao, D.; Huang, X.; Anand, V.; Sun, G.; Yu, H. k-DLCA: An efficient approach for location privacy preservation in location-based services. In Proceedings of the IEEE International Conference on Communications, Kuala Lumpur, Malaysiam, 22–27 May 2016; pp. 1–6.
17. Sun, Y.; Chen, M.; Hu, L.; Qian, Y.; Hassan, M. ASA: Against statistical attacks for privacy-aware users in Location Based Service. *Future Gener. Comput. Syst.* **2017**, *70*, 48–58. [CrossRef]
18. Chen, D.; Zhang, P.; Hu, C.; Wang, H.; Wu, S.; Xing, N. PAPERS: Private and Precise Range Search for Location Based Services. In Proceedings of the IEEE International Conference on Communications, London, UK, 8–12 June 2015; pp. 7347–7352.
19. Peng, T.; Liu, Q.; Wang, G. Enhanced Location Privacy Preserving Scheme in Location-Based Services. *IEEE Syst. J.* **2017**, *11*, 219–230. [CrossRef]
20. Solanas, A.; Martínez-Ballesté, A. A TTP-free protocol for location privacy in location-based services. *Comput. Commun.* **2008**, *31*, 1181–1191. [CrossRef]
21. Vu, K.; Zheng, R.; Gao, J. Efficient algorithms for K-anonymous location privacy in participatory sensing. In Proceedings of the IEEE INFOCOM, Orlando, FL, USA, 25–30 March 2012; pp. 2399–2407.
22. Chow, C.; Mokbel, M.; Aref, W. Casper*:Query processing for location services without compromising privacy. *ACM Trans. Database Syst.* **2009**, *34*, 1–48. [CrossRef]
23. Chow, C.; Mokbel, M.; Liu, X. A Peer-to-Peer Spatial Cloaking Algorithm for Anonymous Location-Based Service. In Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems (GIS '06), Arlington, VA, USA, 10–11 November 2006; pp. 171–178.
24. Chow, C.; Mokbel, M.F.; Liu, X. Spatial Cloaking for Anonymous Location-based Services in Mobile Peer-to-Peer Environments. *GeoInformatica* **2011**, *15*, 351–380. [CrossRef]
25. Niu, B.; Zhu, X.; Lei, X.; Zhang, W.; Li, H. EPS: Encounter-Based Privacy-Preserving Scheme for Location-Based Services. In Proceedings of the Global Communications Conference, Atlanta, GA, USA, 9–13 December 2013; pp. 2139–2144.
26. Li, X.; Wang, E.; Yang, W.; Ma, J. DALP: A demand-aware location privacy protection scheme in continuous location-based services. *Concurr. Comput. Pract. Exp.* **2016**, *28*, 1219–1236. [CrossRef]
27. Memon, I.; Arain, Q. Dynamic path privacy protection framework for continuous query service over road networks. *World Wide Web-Internet Web Inf. Syst.* 2016, 20, 1–33.
28. Herrmann, M.; Troncoso, C.; Diaz, C.; Preneel, B. Optimal sporadic location privacy preserving systems in presence of bandwidth constraints. In Proceedings of the 12th ACM Workshop on Privacy in the Electronic Society (WPES'13), Berlin, Germany, 4 November 2013; pp. 167–178.
29. Peng, T.; Liu, Q.; Meng, D.; Wang, G. Collaborative trajectory privacy preserving scheme in location-based services. *Inf. Sci.* **2017**, *387*, 165–179. [CrossRef]
30. Brakerski, Z.; Vaikuntanathan, V. Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages. In Proceedings of the Cryptology Conference, Santa Barbara, CA, USA, 14–18 August 2011; pp. 505–524.

31. Li, N.; Li, T.; Venkatasubramanian, S. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. In Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering, Istanbul, Turkey, 15–20 April 2007; pp. 106–115.
32. Brakerski, Z.; Vaikuntanathan, V. Efficient Fully Homomorphic Encryption from (Standard) LWE. In Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, Palm Springs, CA, USA, 22–25 October 2011; pp. 97–106.