A Hybrid Approach for Privacy Preservation in Location Based Queries

Zhengang Wu, Liangwen Yu, Jiawei Zhu, Huiping Sun*, Zhi Guan, and Zhong Chen

Institute of Software, EECS, Peking University, Beijing, China
MoE Key Lab of High Confidence Software Technologies (PKU)
MoE Key Lab of Network and Software Security Assurance (PKU)
{wuzg,yulw,zhujw,sunhp,guanzhi,chen}@infosec.pku.edu.cn

Abstract. With rapidly popular location-aware applications, location privacy becomes an emerging issue. This paper studies how to protect the two-fold privacy for both client-side and server-side in location-based queries. This technique is a significant component in privacy-friendly Location Based Services (LBS). Participants protect their own privacy. The LBS server protects against excessive disclose of location records in its Points of Interest (POIs) database while the mobile user protects his exact location by the cloaking technique. The proposed hybrid approach can achieve the challenging goal. Our solution integrates the cloaking technique with a cryptographic protocol, Private Set Intersection (PSI). In addition, this solution is secure in malicious model and also practical.

Keywords: Location privacy, Location Based Services, privacy-preserving protocols, Private Set Intersection, Homomorphic Encryption.

1 Introduction

Popular mobile applications have location-aware capability with the rapid de-velopment of mobile devices with built-in positioning modules. The LBS server holds a database of Points of Interest (POIs). Users can obtain POIs by sending a location-based query. The queries involve extensive applications. e.g. A user can receive personal recommendation information relevant to his location in a location-aware recommender system. However, untrusted participants may learn more sensitive information of the victim by collecting his location information. Thus protection for location privacy becomes an emerging technology.

There are extensive approaches for location privacy preservation. They fall into two major types, cloaking and cryptography. First, the cloaking is succinct and efficient for large data. However, it usually achieves a limited privacy guarantee. Second, the cryptography based approach can provide a stronger privacy guarantee but its computations are extremely expensive. Therefore a hybrid approach is necessary in a practical setting.

* Corresponding author.

Y. Gao et al. (Eds.): WAIM 2013 Workshops, LNCS 7901, pp. 315–326, 2013.

Contributions. This paper proposes a novel hybrid approach to guarantee the client-side location privacy and the server-side content privacy in location-based queries. Only a few existing schemes [1] can achieve the strong privacy while other existing schemes only protect client-side privacy. To the best of our knowledge, our scheme is the first hybrid approach based on Private Set Intersection (PSI) for this goal.

1. Our framework focuses on a range query anchored by an exact location and integrates the query in the k-anonymous cloaking region and the PSI protocol using the grid-based partition. This solution achieves the 2-fold privacy protection that involves the location privacy of the mobile user and the content privacy of the LBS server.
2. We design the PSI protocol secure in malicious model. The LBS server can limit the size of the near range during the PSI protocol's execution. And the PSI protocol is high-efficient due to linear complexity with the size of the POI candidate results for a fixed size of the near range in the experiment.

Outline. The rest of the article is structured as follow. We summarize related works on the topic in Section 2. Preliminary is in its next section. The proposed solution is described in Section 4. Security and privacy is analyzed in Section 5. And Section 6 discusses its efficiency. In the last section we sum the solution up.

2 Related Works

Using private location-based queries, the user can query information about his location in a privacy-preserving manner. Location privacy protection is an emerging technology in several fields [2][3][4][5]. Thus private location-based queries involve a family of location privacy problems that lead to various concerns. The nearest neighbor search problem [6] concerns how to query POIs in a spatial database. e.g. A user queries his nearest hospitals. The nearby friend problem [7] is that a user queries his nearest friends privately in location-aware SNS. Friends are POIs in this scenario. Private proximity test can privately check whether the distance of two locations exceeds a threshold. Private location-based queries involve a wealth of methods which roughly fall into two catalogs.

Cryptography Based Approaches. There are several cryptographic schemes to achieve a private location-based query [1]. Private Information Retrieve (PIR) protocol is a popular building block for strong location privacy. Papadopoulos et al. [6] use secure hardware-aided PIR [8]. Ghinita et al.'s scheme [9] is a two-stage protocol that involves a private-preserving protocol for point-rectangle enclosure which is based on the Paillier Homomorphic Encryption [10] and a PIR protocol. Paulet et al.[1] improve Ghinita et al.'s scheme for performance and integrate an Obvious Transfer (OT) Protocol and a PIR protocol. In addition, zhong et al.[7] construct privacy-preserving protocols for testing location proximity of two users for the nearby friend problem. Their protocols depend on distance computation and the Paillier cryptosystem [10].

Cloaking Based Approaches. The approaches extend the exact location to a coarse-grained region in the spatial domain. i.e. the coarse-grained region covers the exact

location. Cloaking includes abundant research works because of its high efficiency. The Spatial K-anonymity techniques [11][12][13] employ the k-anonymity model for protecting location privacy. K-anonymity [14] requires that a record is anonymous or indistinguishable if at least its same k records appear in the data release. A trusted third party usually adds fake data or dummies to hide actual locations. The amount of the noise must be pondered carefully since too many dummies reduce Quality of Service (QoS). The approaches fail to guard against access pattern attack [15] since the LBS server can learn partial location information of users.

3 Preliminary

Differential Privacy

Definition 1. *The mechanism M maintains ε−Differential Privacy, iff it satisfies* $\Pr(\mathrm{M(D)}) \leq \Pr(\mathrm{M}(D_1)) \cdot e^{\varepsilon}$ *where D and* D_1 *are two neighboring databases that have a distinct row at most.*

Dwork[16][17] proposed Differential Privacy (DP) in 2006. Laplace mechanism [17] is a common DP algorithm (Equation 1) by adding random noise to protect counting query privacy. Histogram involves counting queries. Thus DP constructs a histogram in a privacy-friendly manner [18] [19].

$$\mathrm{M(x)} = \mathrm{q(x)} + \mathrm{Lap}(1/\varepsilon) \tag{1}$$

where q(x) is the query function for counting and the Laplace function Lap(1/ε) is the destiny of Laplace distribution that indicates the noise's amount.

Paillier Homomorphic Encryption

The underlying encryption for our scheme is additive Homomorphic Encryption (HE) over the integers. The Paillier cryptosystem, (Keygen,Enc,Dec), is a public-key probabilistic encryption scheme. Keygen generates a pair of the public key pk and the private key sk by a RSA modulus N (the product of two large primes). Enc is the encryption function form the additive group Z_N to the multiplicative group $\mathrm{Z}^*_{N^2}$ and Dec is the corresponding decryption function. Paillier is semantically secure against Chosen Plaintext Attacks (CPA).

Paillier holds two homomorphic properties based on modular addition. The two following equations are respectively the homomorphic addition of two messages and the homomorphic multiplication by a constant c:

$$\mathrm{Enc}(\mathrm{x}_1 + \mathrm{x}_2) = \mathrm{Enc}(\mathrm{x}_1) \cdot \mathrm{Enc}(\mathrm{x}_2)$$

$$\mathrm{Enc}(\mathrm{c} \cdot \mathrm{x}) = \mathrm{Enc}\left(\sum_{i=1}^{c} x\right) = \prod_{i=1}^{c} \mathrm{Enc(x)} = (\mathrm{Enc(x)})^c$$

Private Set Intersection

Definition 2. *Private Set Intersection is a privacy-preserving protocol for set intersection. Two participants, Alice and Bob, hold their own input sets, A and B respectively. It maintains three conditions as follow:*

1. The correctness: Alice obtains the intersection, $A \cap B$.
2. Alice's privacy: Bob fails to learn Alice's input data A.
3. Bob's privacy: Alice fails to learn Bob's additional data $B - A$.

Private Set Intersection (PSI) can compute the intersection in a privacy-preserving manner. PSI is also a cryptographic primitive to build other privacy-preserving protocols. Freedman et al. [20] proposed the first PSI protocol than depends on Obviously Polynomial Evaluation and Paillier Homomorphic Encryption.

4 Our Solution

Location-based applications need an enhanced privacy protection for both client-side and server-side. e.g. Shopular(www.shopularapp.com), a location-aware coupons issue service, can push related electronic coupons when a mobile user arrives a shop. Obviously, besides the user's location privacy, this application needs to protect its location-related coupons as a valuable asset. Otherwise rivals easily obtain excessive e-coupons by camouflaging a user. In an express delivery service scenario, a mobile customer can query his nearby couriers in the LBS server.

In these scenarios, a query issuer requests a range query anchored by his exact location. Obviously mobile users and LBS servers have fairly their own privacy requirements. First, a mobile user preserves his exact location. Second, a LBS server maintains content privacy to safeguard against excessive disclose of locations of POIs. The LBS server provides POI query services in a limited manner because its POI database is a vital fee-related asset. Therefore we define the Location-based Range Query with Strong Privacy by reference to existing works [1][6] as follow:

Definition 3. *Location-based Range Query with Strong Privacy is a privacy-preserving query processing. It involves two participants, the mobile user and the LBS server. The mobile user queries location-based data in his near range to the LBS server. It maintains three conditions as follow:*

1. The correctness: the mobile user obtains his desired data.
2. The mobile user's location privacy: his exact location is not leaked.
3. The LBS server's content privacy: locations of his additional POIs are not leaked.

4.1 Solution Summary

This scheme includes two major subroutines, a query process in a cloaking region and a private set intersection protocol, as shown in Figure 1.

In the first subroutine, a mobile user hides his exact location in a customizable Cloaking Region. A LBS server can obtain candidate results according to Cloaking Region and non-sensitive information from the user.

The second subroutine implements a private set intersection protocol in a two-party setting. The user's near range and the server's candidate results are two input sets for intersection. The user ultimately obtains his nearby POIs.

In addition, we need a conversion layer between the above two subroutines. Data of both the user and the server change into an acceptable form for the PSI protocol. Simply put, these data become positive integers.

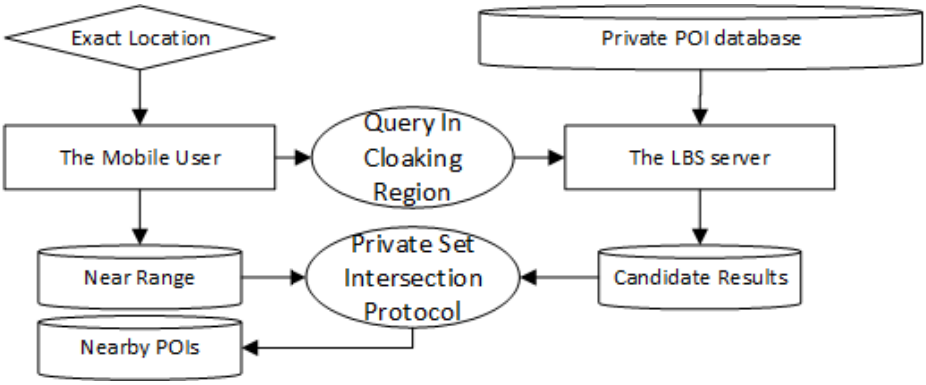

Fig. 1. Solution Summary

4.2 Computations for The Cloaking Region

A mobile user generates a cloaking region which contains the queried range anchored by his exact location. The cloaking region hides the exact location and indicates a coarse location. i.e. The coarse location suppresses the exact one sensitive. The LBS server holds a POI database. The user sends nonsensitive properties and keywords which include the coarse location to the server. Note that the queried range is a small part of the Cloaking Region.

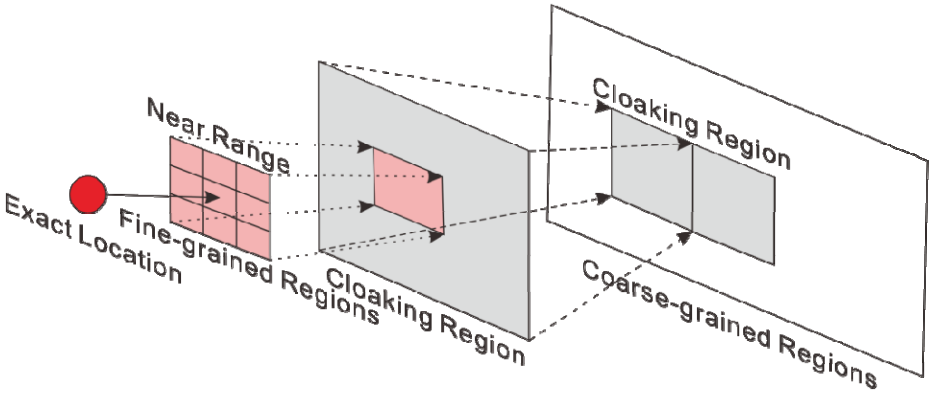

Fig. 2. The Cloaking Region and the Hierarchical Structure

GIS usually manages spatial data and geographical map hierarchically with an adjustable granularity. We can divide the whole spatial data into a two-level structure, coarse-grained regions and fine-grained regions like Figure 2 where the mobile user's near range is generalized to a cloaking region. Note that for convenience a region can be express as a set of little grids. Grids are the smallest units whose size depends on the accuracy of positioning modules.

According to Figure 1, a query in the cloaking region involves computations in both sides. Algorithm 1 and Algorithm 2 depend on k-anonymity and differential privacy respectively. Let Count() be a counting function.

The Mobile User's Computation. In this stage, the mobile user creates a cloaking region to hide his exact location. K-anonymity requires that the user number in the cloaking region exceeds k. If the user number in the local region fails to satisfy k-anonymity, Algorithm 1 constructs the cloaking region by putting together the local region and other coarse-grained regions. In non-trivial situations, Algorithm 1 has a constant time complexity. Note that Count() may need a Trusted Third Party (TTP) to log the statistics but we can estimate the Count() value without a TTP by background knowledge since the cloaking region is not precise.

Algorithm 1: Cloaking-k: Creating The Cloaking Region for Users

Input: The parameter k for k-anonymity;
The current location of the mobile user;
The size limit of the Cloaking Region;
Output: The Cloaking Region CR

```
The user builds his Near Range (NR) anchored by his current exact location.;
CR ⟵ the coarse-grained region which includes NR;
n ⟵ Count(Users in CR)/* It may be an estimated value*/;
while n < k do
    Check whether the current CR satisfies k-anonymity;
    Choosing a coarse-grained region CR_1 randomly;
    CR ⟵ CR ⋃ CR_1;
    n ⟵ n+Count(Users in CR_1);
return CR;
```

The LBS Server's Computation. In this stage, the LBS server needs to protect against excessive disclose of the size of POIs in the received cloaking region. Because a user can draw a histogram according to the size of POIs in the received cloaking region no matter what PSI. The histogram leaks statistical distribution following distinct regions. And PSI fails to protect the size of the POI set. Algorithm 2 is based on Differential Privacy. It adds a Laplace noise over the actual data. It has also a constant time complexity and the dummies will be blinded through the PSI protocol.

Algorithm 2: NoisyQuery: Retrieving Candidate Results for Servers

Input: The parameter ε for ε-differential privacy;
The Cloaking Region CR from the mobile users;
Output: The candidate results relevant to the Cloaking Region.

```
Querying actual locations (a set L) of POIs in the Cloaking Region;
n ⟵ Count(actual locations of the POIs);
Computing the amount Lap(1/ε) of the noise;
Generating dummies (a set D) whose number is the noise's ceiling, ⌈Lap(1/ε)⌉;
Candidate Results is the union L ⋃ D of actual locations and dummies;
return Candidate Results;
```

4.3 Transformation: Preparing for Intersection

We transfer this location-based query problem into the set intersection problem as Fig.3 shows. A geographical map is overlaid with a grid network. A unique integer, Grid ID labels a little grid. Each location falls into the corresponding grid. Some close locations may become a same Grid ID due to precision. The user U's near range can be expressed as a set of Grid IDs like Subfigure(A) where a cross-shaped symbol means a Grid ID in the near range. In a similar way, the LBS server holds also a Grid ID set of locations of POIs (hearts) as Subfigure(B). Thus it is straightforward to obtain the nearby POIs, by the intersection of two Grid ID sets held by the user and the server respectively.

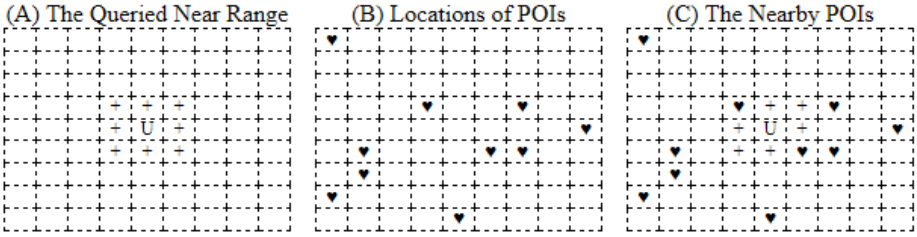

Fig. 3. Computation to Obtain Nearby POIs

The transformation way from locations (coordinates or other similar things) to Grid IDs is flexible. The representation based on the Grid ID set is independent of the shape of the spatial region but obviously the mobile and the LBS server employs the same transformation to ensure the semantic consistency.

4.4 Private Set Intersection for POIs

We propose a PSI variant based on Freedman's PSI [20] to protect the enhanced privacy for both the user and the server in the process of calculating intersection. Our PSI is secure in the presence of malicious users. Greed users cannot obtain excessive POIs. First, users fail to send an excessive near range without the server's permission. The PSI supports a security policy to effectively check the number of the received ciphertexts. Second, the server cannot compute a malicious polynomial whose coefficients are all-zero. The attack using the malicious polynomial, $F(x) \equiv 0$, are discussed in [20][21][22]. We propose an effective method to achieve immunity against the attack.

Algorithm 3 demonstrates our PSI. The mobile user holds a location set A on behalf of his near range. Similarly, the LBS server holds another location set B of POIs in the queried cloaking region. Elements of Both A and B are Grid IDs each of which labels a small piece of location.

Let $|A| = s$ and in the Paillier cryptosystem, Keygen(), Enc() and Dec() are the secret-key generation, the encryption and the decryption respectively.

Algorithm 3: PSI-Location:Private Set Intersection for Locations of POIs

Input: The mobile user holds the NR set A of size s, $\{a_1, a_2, \cdots, a_s\}$.
The LBS server holds the POI set B of size n, $\{b_1, b_2, \cdots, b_n\}$.
The Near Range is not allowed to exceed the maximum size L.

Output: Alice obtains its nearby POIs which belong to the intersection $A \bigcap B$.

1: (Setup phase) Client:$(pk, sk) \leftarrow Keygen(1^k)$ and Client $\longrightarrow$ Server:pk
2: Client encodes A into a polynomial $F(x) = x^s + \sum_{j=0}^{s-1} c_j \cdot x^j$.
Client $\longrightarrow$ Server:$Enc(c_0), Enc(c_1), \cdots, Enc(c_{s-1})$
3: Server receives $Enc(c_0), Enc(c_1), \cdots, Enc(c_{s-1})$.
Server breaks the request if $s > L$, otherwise continues the protocol.
Server computes $Enc(r(F(x)) + x)$ for each $x \in B$ where r is a random integer.
Server obtains the encrypted set $Enc(B') = \{Enc(r(F(x)) + x) | x \in B\}$
Server permutes randomly elements of $Enc(B')$ and sends them to Client.
4: Client decrypts the received ciphertexts $Enc(B')$ to obtains $B^{'}, Dec(Enc(B'))$.
Client computes the set intersection $A \bigcap B^{'}$ that is exactly $A \bigcap B$.

We have some vital points to explain this protocol involving the following three equations.

$$F(x) = \prod_{i \in A} (x - i) = 1 \cdot x^s + \sum_{j=0}^{s-1} c_j \cdot x^j \quad (2)$$

$$Enc(F(x)) = Enc(x^s) \cdot Enc(\sum_{j=0}^{s-1} c_j \cdot x^j) = Enc(x^s) \cdot \prod_{j=0}^{s-1} Enc(c_j \cdot x^j) \quad (3)$$

$$Enc(rF(x) + x) = (Enc(F(x)))^r \cdot Enc(x) \quad (4)$$

First of all, our PSI protects against the malicious polynomial $\mathrm{F(x)} \equiv 0$. In Step 2, the coefficient of the highest item x^s in Equation 2 is 1 invariably. So the user encrypts only the rest of the F(x) coefficients, i.e. $c_0, c_1, \cdots, c_{s-1}$. Above all, $\mathrm{F(x)} \equiv x^s$ if $c_0, c_1, \cdots, c_{s-1}$ are all-zero. Obviously, $\mathrm{rF(x)} + \mathrm{x} \equiv \mathrm{x}$ for all integers in B when $\mathrm{F(x)} \equiv 0$ and thus the whole B will be sent.

Next, the server computes $\mathrm{rF(x)} + \mathrm{x}$ inputting all values in B according to homomorphic properties in Step 3. Equation 3 and 4 show the secure computation for $\mathrm{rF(x)} + \mathrm{x}$. $\mathrm{rF(x)} + \mathrm{x} = \mathrm{x}$ since x is roots of F(x) if $\mathrm{x} \in \mathrm{A} \cap \mathrm{B}$. For another, if $\mathrm{x} \notin \mathrm{A} \cap \mathrm{B}$, $\mathrm{rF(x)} + \mathrm{x}$ is also random because of a random integer r. Thus $\mathrm{rF(x)} + \mathrm{x}$ blinds elements in $\mathrm{B} - \mathrm{A}$.

5 Security and Privacy

Our solution guarantees the two-fold privacy which satisfies privacy requirements of both the client and the server. This scheme is secure in malicious model.

The Correctness. The whole scheme's correctness is easily proofed by the PSI's correctness. Equation 4 ensures that for all $\mathrm{x} \in \mathrm{A} \cap \mathrm{B}$, $\mathrm{rF(x)} + \mathrm{x} = \mathrm{x}$ since x is the F(x)'s roots. Actually, $\mathrm{A} \cap \mathrm{B} \equiv \mathrm{A} \cap \mathrm{B}'$ although $\mathrm{B} \neq \mathrm{B}'$.

The Client's Location Privacy. Two techniques, Cloaking and PSI, protect the client's exact location. First, the exact location is generalized to the cloaking region using Algorithm 1 based on k-anonymity. Second, in our PSI, the Paillier homomorphic encryption's semantical security ensures that the LBS server learns nothing on the user's near range except its size that equals the degree of the polynomial F(x). Note that the server checks the query permission through the knowledge of the near range's size.

The Server's Content Privacy. The LBS server has to protect against excessive disclose of its POI database. DP and PSI preserve its content privacy that actual locations of POIs are sensitive.

First, our PSI protocol randomizes these POI locations in the Cloaking Region (CR) but not in the Near Range (NR). i.e. these integer values in the additional region CR-NR are meaningless through PSI. We can proof it using Equation 4 and the Paillier's homomorphic properties.

Second, the size of the actual POI sub database in the Cloaking Region is still sensitive obviously. We add the appropriate noise to the actual size by Algorithm 2 since our PSI fails to hide the actual size. Laplace mechanism satisfying differential privacy indicates the amount of the additional noisy data.

Security Against the Malicious User. Usually, semi-honest adversaries (passive attackers) only eavesdrops data. However, malicious adversaries (active attackers) can disturb the protocol's execution by elaborating input data. Our PSI protocol can protect against two malicious behaviours as follow.

1. **The Overlarge Queried Region:** The malicious user may submit dishonestly an overlarge region to obtain excessive POIs. Step 3 of Algorithm 3 protects against this attack. Because the length of the received $\mathrm{Enc}(c_j)$ sequence equals the size of the user's near range according to Equation 2 and Step 2 of Algorithm 3. The server only responses approved requests by this binding.
2. **The All-zero Polynomial:** The malicious user may submit dishonestly the degenerated polynomial [20][22],$F(x) \equiv 0$, whose coefficients are all-zero. In this way the server discloses the whole set B computing $\mathrm{Enc}(B')$ by $rF(x) + x \equiv x$. However, to prevent it, our PSI uses a succinct new technique that the server can avoid employing the encrypted coefficient $\mathrm{Enc}(c_s)$ of the highest item x^s according to Equation 3. The user fails to send $\mathrm{Enc}(c_s)$ since $c_s \equiv 1$ in Equation 2. In the worst situation, the polynomial F(x) in the server can degenerate into $F(x) \equiv x^s$ and thus the malicious user fails to receive any meaningful values through the randomness of $rx^s + x$.

6 Efficiency

Our scheme's efficiency depends on the PSI protocol that outputs the majority of both time complexity and communication cost as the following table. Let the size of the server set (candidate results) be β and the size of the client set (near range) be a.

Note that usually $a \leq \beta$ and $a < L$ is usually a small integer. The modulus N of Paillier is at least 1024 bits. The Paillier's ciphertext is twice as long as the corresponding plaintext.

Subroutine	Time Complexity		Communication Cost	
	Client	Server	Client→Server	Server→Client
Query	O(1)	O(1)	O(1)	0
PSI	$O(a+\beta)$	$O((a+1)\beta)$	Na	$2N\beta$

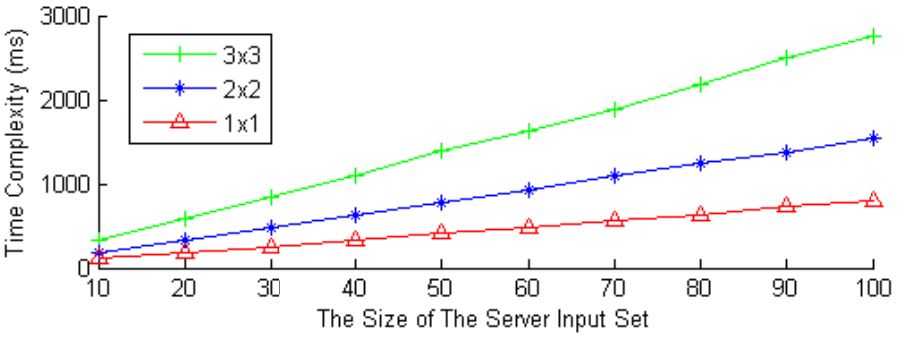

Fig. 4. Performance

Fig. 4, where the unit of time complexity is millisecond (ms), shows the performance of our PSI protocol in single thread mode. We implement it using JAVA. The execution can be within seconds in a laptop with an Intel Core i7 2.4GHz CPU and 8G RAM. It has approximately a linear complexity for a given NR size (3×3, 2×2 or 1×1) regardless of the NR's shape. However, a larger NR size leads to a more complex polynomial that increases the amount of calculation.

In this experiment, each Grid ID has 6 decimal numbers and the server of input set has a variable size because of the addition of the DP noise. We log the total time of the protocol as Fig.4 shows. The client executes the majority of encryption and all of decryption computations and the server calculates wholly the encrypted polynomial $\mathrm{Enc}(rF(x) + x)$ for all $x \in B$ over ciphertexts.

7 Conclusion

This paper discusses private location-based queries where our solution can guarantee the two-fold privacy: the mobile user needs not to leak his exact location; the LBS server protects against excessive disclose of its POI database. The proposed hybrid approach integrates the cloaking technique and the privacy-preserving protocol. We transform queries on the near range located on the cloaking region into secure computations for set intersection. Our approach is secure in malicious model to protect against the user's malicious data. In addition, it is practical and high-efficient according to discussion on its efficiency.

Acknowledgment. This work is partially supported by the HGJ National Significant Science and Technology Projects under Grant No. 2012ZX01039-004-009, Key Lab of Information Network Security, Ministry of Public Security under Grant No.C11606, the National Natural Science Foundation of China under Grant No. 61170263

References

1. Paulet, R., Kaosar, M.G., Yi, X., Bertino, E.: Privacy-preserving and content-protecting location based queries. In: ICDE, pp. 44–53 (2012)
2. Jian, Y., Chen, S., Zhang, Z., Zhang, L.: Protecting receiver-location privacy in wireless sensor networks. In: IEEE INFOCOM, pp. 1955–1963 (2007)
3. Huang, Y., Vishwanathan, R.: Privacy preserving group nearest neighbour queries in location-based services using cryptographic techniques. In: IEEE GLOBECOM, pp. 1–5 (2010)
4. Li, Y., Ren, J.: Source-location privacy through dynamic routing in wireless sensor networks. In: IEEE INFOCOM, pp. 2660–2668 (2010)
5. Pingley, A., Zhang, N., Fu, X., Choi, H.A., Subramaniam, S., Zhao, W.: Protection of query privacy for continuous location based services. In: IEEE INFOCOM, pp. 1710–1718 (2011)
6. Papadopoulos, S., Bakiras, S., Papadias, D.: Nearest neighbor search with strong location privacy. PVLDB 3(1), 619–629 (2010)
7. Zhong, G., Goldberg, I., Hengartner, U.: Louis, lester and pierre: Three protocols for location privacy. In: Borisov, N., Golle, P. (eds.) PET 2007. LNCS, vol. 4776, pp. 62–76. Springer, Heidelberg (2007)
8. Williams, P., Sion, R.: Usable pir. In: NDSS. The Internet Society (2008)
9. Ghinita, G., Kalnis, P., Kantarcioglu, M., Bertino, E.: Approximate and exact hybrid algorithms for private nearest-neighbor queries with database protection. GeoInformatica 15(4), 699–726 (2011)
10. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
11. Mokbel, M.F., Chow, C.Y., Aref, W.G.: The new casper: A privacy-aware location based database server. In: Chirkova, R., Dogac, A., Özsu, M.T., Sellis, T.K. (eds.) IEEE ICDE, pp. 1499–1500 (2007)
12. Mokbel, M.F., Chow, C.Y., Aref, W.G.: The new casper: Query processing for location services without compromising privacy. In: Dayal, U., Whang, K.Y., Lomet, D.B., Alonso, G., Lohman, G.M., Kersten, M.L., Cha, S.K., Kim, Y.K. (eds.) VLDB, pp. 763–774. ACM (2006)
13. Kalnis, P., Ghinita, G., Mouratidis, K., Papadias, D.: Preventing location-based identity inference in anonymous spatial queries. IEEE Trans. Knowl. Data Eng, 1719–1733 (2007)
14. Samarati, P., Sweeney, L.: Generalizing data to provide anonymity when disclosing information (abstract). In: Mendelzon, A.O., Paredaens, J. (eds.) PODS, p. 188. ACM Press (1998)
15. Williams, P., Sion, R., Carbunar, B.: Building castles out of mud: practical access pattern privacy and correctness on untrusted storage. In: Ning, P., Syverson, P.F., Jha, S. (eds.) ACM Conference on Computer and Communications Security, pp. 139–148. ACM (2008)

16. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006)
17. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006)
18. Xu, J., Zhang, Z., Xiao, X., Yang, Y., Yu, G.: Differentially private histogram publication. In: ICDE, pp. 32–43 (2012)
19. Hay, M., Rastogi, V., Miklau, G., Suciu, D.: Boosting the accuracy of differentially private histograms through consistency. PVLDB 3(1), 1021–1032 (2010)
20. Freedman, M.J., Nissim, K., Pinkas, B.: Efficient private matching and set intersection. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 1–19. Springer, Heidelberg (2004)
21. Jarecki, S., Liu, X.: Efficient oblivious pseudorandom function with applications to adaptive ot and secure computation of set intersection. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 577–594. Springer, Heidelberg (2009)
22. Hazay, C., Nissim, K.: Efficient set operations in the presence of malicious adversaries. J. Cryptology 25(3), 383–433 (2012)