A Dynamic Location Privacy Protection Scheme Based on Cloud Storage

Li Li[1,2], Zhengjuan Lv[1], Xiaohong Tong[1], and Runhua Shi[2]
(Corresponding author: Runhua Shi)
Hefei Technology College, Chaohu 238000, China[1]
School of Computer Science and Technology, Anhui University[2]
Hefei 230601, China
(Email: shirh@ahu.edu.cn)
(Received June 11, 2018; Revised and Accepted Nov. 2, 2018; First Online June 15, 2019)

Abstract

In this paper, we introduce the Cloud serve and build a new location-based service model. By employing the primitive protocols of data encryption and oblivious transfer and following the space anonymous ideas, we further present a dynamic location privacy protection scheme based on Cloud storage. This scheme can ensure both the location privacy of the user and the data privacy of the data service provider in location-based service. Especially, it can greatly reduce the storage, computation and communication costs of the data service provider.

Keywords: Cloud Storage; Location-based Services; Location Privacy; Oblivious Transfer

1 Introduction

With the advent of pervasive computing and ubiquitous networking, it can generate large volumes of data anytime and anywhere, so we enter a big data era. In order to deal with big data, Cloud computing arises accordingly [15]. As an important part of Cloud computing services, Cloud storage [7,9,23] provides a relatively efficient, reliable and low-cost storage platform for people or companies in the era of big data [19].

Furthermore, with the rapid development of mobile Internet and the widespread use of various terminal devices (*e.g.*, sensor and phone), it is possible to obtain the exact location of the person at any time and any place. This leads to a new location-based service (LBS). Informally, location-based services essentially provide a query service which is relevant to the user's location [16]. For example, in emergency medical conditions, it can query the nearest hospital; moreover, when the users travel outside, it can query the nearest hotel and theater, or other interesting places of entertainments.

Obviously, location-based service (LBS) brings the convenience to our lives, but it also brings the threats to the privacy of the person [3, 14, 21], *e.g.*, location privacy of the query user. Furthermore, if a user's location information is compromised in location-based services, it may lead to more disclosure of sensitive personal information, such as health, habits, old, etc. Especially, the disclosure of personal location may allow the competitor to track and locate the person, and even to carry out personal attacks.

In the past decade, the researchers had done a great deal of work on location privacy protection, and proposed a series of location privacy protection methods, but different location privacy protection methods have different protection objects. For example, the literatures [1, 22] protect the user's identity information, and the literatures [4,5] protect the user's spatial location information, while the literatures [6, 12] focus on protecting the user's the query privacy, *i.e.*, the user's service type. These existing schemes can be divided into the following categories by different methods:

1) The location privacy protection schemes based on the space generalization method [17, 20];
2) The location privacy protection schemes based on the cloak method [10, 18];
3) The location privacy protection schemes based on the data cutting method [2, 13];
4) The location privacy protection schemes based on privacy information retrieval [8, 24].

However, there are still some deficiencies in the current privacy protection schemes, such as the excessive communication or computation costs, the leakage of part privacy, and the requirement of the trusted key management center or the trusted third party. In addition, in the privacy protection schemes mentioned above, the huge amount of data is stored on the data service provider, and in turn, only the small amount of query result related to location information is returned to the query user. In the age of big data, the excessive storage costs of the data

service provider may be the bottleneck of the development of location-based services. The current popularity of cloud storage has brought huge changes to data storage, and accordingly a lot of data can be delivered to the cloud server to reduce the local storage cost. However, at the same time the convenient services also bring the risk of the leakage of the data privacy. So the important data are first encrypted and then stored on the cloud server. Later, the authorized users can directly access to the cloud server and download the required data.

In this paper, to reduce the local storage cost of the data service provider, we introduce the Cloud server and design a new location-based service model. Furthermore, employing the primitive protocols of data encryption [11] and oblivious transfer [8] and following the space anonymous ideas, we present a dynamic location privacy protection scheme based on Cloud storage. This scheme can effectively solve the problems of data privacy and location privacy in location-based services. In addition, it can reduce the system overheads while it ensures the location privacy of the user. Especially, it can reduce the storage, computation and communication costs of the data service provider.

2 Proposed Scheme

2.1 System Model

Here we first introduce a new system model for location-based services. In our new system model, suppose that there is a mobile user (U), a data service provider (DSP), and a cloud server (CS), as shown in Figure 1. The above models mainly include three processes: initialization phase, private query phase, key update phase.

Initialization phase: Firstly, the data service provider generates and publishes the system parameters. Secondly, the data service provider divides all location-related data into different blocks, and uses their respective public keys to encrypt the data of each block. Then, the encrypted data and the partitioned block map are uploaded to the cloud server for storage. Please note that each block owns a different key pair, *i.e.*, the private key and its corresponding public key.

Private query phase: The mobile user U gets the ciphertext of the querying data from the cloud server according to its current actual location and the block map, and uses the oblivious transfer protocol to request the data service provider for the private key of the ciphertext, such that he/she can decrypt the ciphertext and obtain the corresponding plaintext, which includes the querying result.

Key update phase: The data service provider regularly updates the key pairs of all partitioned blocks and renews the ciphertexts of all blocks periodically with the help of the cloud server.

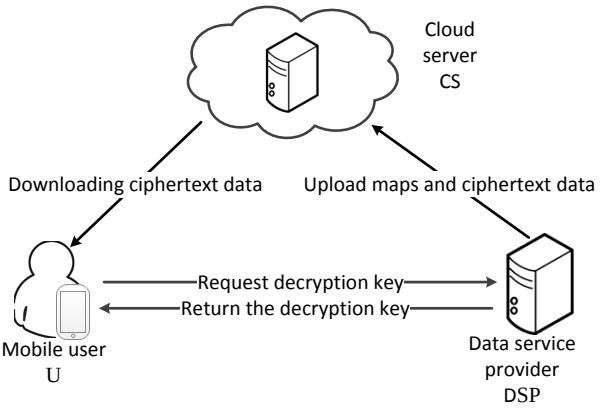

Figure 1: A location privacy protection model based on cloud storage

2.2 Protocol

For the model presented above, we further design a novel protocol by introducing the EIGamal encryption algorithm [11] to protect the data privacy of the data service provider and the oblivious transfer protocol [8] to ensure the user's privacy, which is described as follows:

Initialization phase:

Step 1. The data service provider (DSP) generates and publishes the system parameters.

1) The DSP generates a big prime number p, where $p-1$ has a big prime factor q, and then selects a multiplicative cycle group G on the finite field F_p, such that the order of the cyclic group G is q;
2) The DSP randomly selects two q-order generators of the multiplicative cycle group, which are marked as g and h;
3) The DSP publishes the system parameters $\{F_p, G, q, g, h\}$.

Step 2. The DSP divides the map into different blocks, where each block owns a different key pair, *i.e.*, the private key and its corresponding public key, and uses the corresponding public key to encrypt the data of each block. Finally, the encrypted data and the partitioned block map are uploaded to the cloud server for storage.

1) The DSP establishes a coordinate system according to the external rectangle of the map area (as shown in Figure 2), and divides the map area into $s \times t$ uniform blocks in the coordinate system, where any one of the blocks is denoted as D_{ij}, for $1 \le i \le s$ and $1 \le j \le t$.
Note. The size of $s \times t$ is related to the service accuracy and the computation and communication costs. The larger $s \times t$ is, the less data will

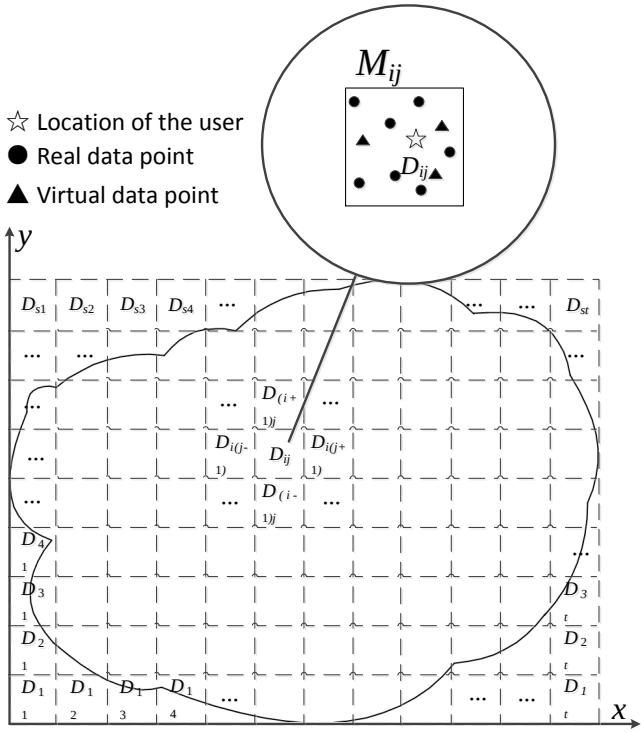

Figure 2: The partitioned diagram of the data blocks

be returned to the user. Conversely, the smaller $s \times t$ is, the more data will be returned to the user. Accordingly, the higher the service accuracy is, in turn, the smaller the computation and communication costs will be.

2) By introducing a small amount of virtual data (*i.e.*, false data), the DSP standardizes all location-related data, so that the data in each block is consistent in the format and size aspects. And the DSP marks the standardized data in the arbitrary block D_{ij} as M_{ij}.

3) The DSP generates the public and private key pair (pk_{ij}, sk_{ij}) for the arbitrary block D_{ij}, where the private key sk_{ij} is randomly generated (*i.e.*, $sk_{ij} = x_{ij}$, $x_{ij} \in Z_q^*$) and its public key $pk_{ij} = g^{x_{ij}}$.

4) The DSP uses the public key pk_{ij} to encrypt the data M_{ij} as follows:

$$\begin{aligned} C_{ij}^1 &= g^{r_{ij}} (mod\ p), \\ C_{ij}^2 &= M_{ij} \cdot {pk_{ij}}^{r_{ij}} (mod\ p). \end{aligned}$$

In the above equations, $r_{ij} \in Z_q^*$, the ciphertext $E_{pk_{ij}}(M_{ij}) = (C_{ij}^1, C_{ij}^1)$, $1 \le i \le s$, $1 \le j \le t$. Here, we assume that M_{ij} is just a plaintext block. Here, we assume that M_{ij} is just a plaintext block. Finally, the DSP sends all ciphertexts (*i.e.*, $E_{pk_{ij}}(M_{ij})s$ for $1 \le i \le s$, $1 \le j \le t$) to the CS and stores them in the CS.

The mobile user (U) gets the ciphertext of the querying data from the cloud server according to its current actual location and the block map, and uses the oblivious transfer protocol to request the data service provider for the private key of the ciphertext, such that he/she can decrypt the ciphertext and obtain the corresponding plaintext, which includes the querying result.

Private query phase:

Step 1. The U locates the block D_{ab} according to its current actual location and the public block map, where $1 \le a \le s$, $1 \le b \le t$.

Step 2. The U privately gets the private key sk_{ab} of the block D_{ab} by the following oblivious transfer protocol.

1) According to the block D_{ab}, the U calculates $v = b + (a-1) \times t$, selects a random number $r \in Z_q^*$, calculates $z = g^r h^v$, and sends z to the DSP.

2) After receiving the information z, the DSP selects a random number $k_{ij} \in Z_q^*$ for each block, and calculates $K_{ij}^1 = g^{k_{ij}}$, $K_{ij}^2 = sk_{ij}(z/h^{j+(i-1)\times t})^{k_{ij}}$ $(1 \le i \le s,\ 1 \le j \le t)$. Then the DSP sends all (K_{ij}^1, K_{ij}^2)s to the U.

3) After receiving all (K_{ij}^1, K_{ij}^2)s, the U calculates $sk_{ab} = K_{ab}^2/(K_{ab}^1)^r$, and further gets the private key sk_{ab}.

Step 3. According to the current block D_{ab}, the U downloads the corresponding ciphertext $E_{pk_{ab}}(M_{ab})$ from the CS and decrypts it with the private key sk_{ab} to get the plaintext data M_{ab}.

$$M_{ab} = C_{ab}^2/(C_{ab}^1)^{sk_{ab}} (mod\ p).$$

Key update phase: The DSP regularly updates the key pairs of all partitioned blocks and renews the ciphertexts of all blocks periodically with the help of the cloud server.

Step 1. For $1 \le i \le s$, $1 \le j \le t$, the DSP randomly generates a private key $sk'_{ij}(i.e., x'_{ij})$ as the new private key of the block D_{ij}. Furthermore, the new public key pk'_{ij} is calculated according to the new private key sk'_{ij}, where $pk'_{ij} = g^{x'_{ij}}$. Similarly, the new private key sk'_{ij} is kept in secret and the new public key pk'_{ij} is published.

Step 2. According to the new private key sk'_{ij} and the new public key pk'_{ij} of the block D_{ij}, the DSP generates an auxiliary message F_{ij} and sends it to CS.

1) According to the new private key sk'_{ij} and the original private key sk_{ij} stored in secret, the DSP calculates:

$$\begin{aligned} \Delta x_{ij} &= sk'_{ij} - sk_{ij} (mod\ q), \\ &\quad (i.e., sk'_{ij} = sk_{ij} + \Delta x_{ij} (mod\ q)), \\ \Delta pk_{ij} &= g^{\Delta x_{ij}} (mod\ p), \end{aligned}$$

$$pk'_{ij} = pk_{ij} \cdot \Delta pk_{ij}.$$

2) The DSP calculates $C'_{ij} = (C^1_{ij})^{\Delta x_{ij}}$, where C^1_{ij} is obtained by querying the CS. Then the auxiliary message $F_{ij} = (C'_{ij}, \Delta pk_{ij})$ is sent to CS.

Step 3. According to the auxiliary message F_{ij} and the new public key pk'_{ij}, the CS updates the ciphertext of the corresponding block D_{ij}. Finally, the CS gets the new ciphertext $E_{pk'_{ij}}(M_{ij})$ and covers the old ciphertext with the new ciphertext.

1) After receiving the auxiliary message $(C'_{ij}, \Delta pk_{ij})$, CS selects a random number $r'_{ij} \in Z^*_q$ calculates the updated ciphertext (C'^1_{ij}, C'^2_{ij}) as follows:

$$\begin{aligned} C'^1_{ij} &= C^1_{ij} \cdot g^{r'_{ij}}, \\ C'^2_{ij} &= C^2_{ij} \cdot C'_{ij} \cdot (pk'_{ij})^{r'_{ij}}, \end{aligned}$$

2) The CS updates the corresponding ciphertext $E_{pk'_{ij}}(M_{ij}) = (C'^1_{ij}, C'^2_{ij})$, which is stored in CS.

3 Analysis

We will analyze the protocol designed above in terms of Correctness, Security and Performance.

3.1 Correctness

In the above protocol, the correctness of data encryption and decryption is guaranteed by EIGamal encryption algorithms. In addition, the correctness of the ciphertext updating is proved as Equations (1) and (2):

$$C'^1_{ij} = C^1_{ij} \cdot g^{r'_{ij}} = g^{r_{ij}} \cdot g^{r'_{ij}} = g^{r_{ij}+r'_{ij}}, \quad (1)$$

$$\begin{aligned} & C^2_{ij} \cdot C'_{ij} \cdot (pk'_{ij})^{r'_{ij}} \\ =& M_{ij} \cdot (g^{x_{ij}})^{r_{ij}} \cdot (g^{r_{ij}})^{\Delta x_{ij}} \cdot (g^{x_{ij}+\Delta x_{ij}})^{r'_{ij}} \\ =& M_{ij} \cdot (g^{x_{ij}+\Delta x_{ij}})^{r_{ij}} \cdot (g^{x_{ij}+\Delta x_{ij}})^{r'_{ij}} \\ =& M_{ij} \cdot (g^{x_{ij}+\Delta x_{ij}})^{r_{ij}+r'_{ij}} \\ =& M_{ij} \cdot (pk'_{ij})^{r_{ij}+r'_{ij}} \end{aligned} \quad (2)$$

3.2 Security

Furthermore, we analyze the security mainly from the following aspects.

1) The user's location privacy. In our proposed protocol, the mobile user U only interacts with the data service provider to obtain the required private key by the oblivious transfer (OT^n_1) protocol. Furthermore, the OT^n_1 protocol protects the input privacy of the mobile user U. According to the OT^n_1 protocol (see Step 2 of Private query phase), the data service provider gets only the message z from the mobile user U. Here $z = g^r h^v$, $v = b + (a-1) \times t$ and r is a number that the mobile user U selects randomly. Obviously, the location information of the mobile user U is hidden in z. But, one equation cannot solve multiple unknown variables, *i.e.*, r and v (a and b), so the DSP cannot obtain any location information of the mobile user U only from the message z, that is, the location privacy of the mobile user U can be guaranteed by OT^n_1.

2) The data privacy of the data service provider. Though the data service provider stores all ciphertexts in the cloud server, the cloud server cannot get any plaintext without the private keys.

   For each block D_{ij} $(1 \le i \le s,\ 1 \le j \le t)$, the data service provider encrypts the corresponding data M_{ij} by the EIGamal encryption algorithm, to form a ciphertext $E_{pk_{ij}}(M_{ij})$, and then sends it to the cloud server for storage. The specific process of the encryption is as follows: randomly select $r_{ij} \in Z^*_q$, and calculate $C^1_{ij} = g^{r_{ij}} (mod\ p)$ and $C^2_{ij} = M_{ij} \cdot pk^{r_{ij}}_{ij} (mod\ p)$. The final ciphertext $E_{pk_{ij}}(M_{ij})$ is (C^1_{ij}, C^2_{ij}). According to EIGamal encryption, if the cloud server or other attackers want to get the plaintext M_{ij}, he/she must know r_{ij} or sk_{ij}. But both the random number r_{ij} and the private key sk_{ij} are generated secretly by the data service provider. Unless one can solve the discrete logarithm problem (i.e., given an element $b \in G$, to solve a, such that $b = g^a$), he/she cannot get r_{ij} and sk_{ij}. Accordingly, the attacker cannot get the plaintext only from the ciphertext without the key. Therefore, the data privacy of the data service provider is guaranteed by the difficulty of the discrete logarithm problem.

3) The key privacy of the data service provider. On the one hand, the mobile user U only gets one private key associated?with his/her location from the data service provider by executing one OT^n_1 protocol, but not any other private key (it implies that the U cannot get other service information except his/her own area).

According to the OT^n_1 protocol, the mobile user U can only obtain one unique private key from the data service provider, and its security is guaranteed by the OT^n_1 protocol. The detailed analysis is as follows:

Suppose that the mobile user U is privately located in the area D_{ab}. The U calculates $v = b + (a-1) \times t$, selects a random number, and calculates $z = g^r h^v$. Then the U transmits z to the data service provider. After the data service provider receives z, the DSP selects a random number $k_{ij} \in Z^*_q$ for each block in the map, and calculates $K^1_{ij} = g^{k_{ij}}$ and $K^2_{ij} = sk_{ij}(z/h^{j+(i-1)\times t})^{k_{ij}}$ $(1 \le i \le s,\ 1 \le j \le t)$. Finally, the data service provider sends all (K^1_{ij}, K^2_{ij})s to the U. According to his/her location information (*i.e.*, the values of both a and b), the mobile user U

can get exactly $sk_{ab} = K_{ab}^2/(K_{ab}^1)^r$, but no other private key (see Equation (3)) based on the difficult assumption of the computational Diffie-Hellman problem (*i.e.*, given (g, g^a, g^b) for a randomly chosen generator g and random $a, b \in \{0, ..., q-1\}$, it is computationally intractable to compute the value g^{ab}).

$$\begin{aligned} K_{ij}^2/(K_{ij}^1)^r &= sk_{ij}(z/h^{j+(i-1)\times t})^{k_{ij}}/(g^{k_{ij}})^r \\ &= sk_{ij}(g^r h^v/h^{j+(i-1)\times t})^{k_{ij}}/(g^{k_{ij}})^r \\ &= sk_{ij}(g^r h^v/h^{j+(i-1)\times t})^{k_{ij}}/(g^{k_{ij}})^r \\ &= sk_{ij}(g^r h^{v-(j+(i-1)\times t)})^{k_{ij}}/(g^{k_{ij}})^r \\ &= sk_{ij}(h^{v-(j+(i-1)\times t)})^{k_{ij}} \\ &\neq sk_{ij} \end{aligned} \tag{3}$$

On the other hand, the updated private key is random, *i.e.*, $sk'_{ij} = x'_{ij}$, where the public key is $pk'_{ij} = g^{x'_{ij}}$. Similarly, based on the difficult assumption of the discrete logarithm problem, the new private key sk'_{ij} is secure while its public key is open. Furthermore, $\Delta x_{ij} = sk'_{ij} - sk_{ij}(mod\ p) = x'_{ij} - x_{ij}(mod\ p)$, and the updated ciphertext $E_{pk'_{ij}}(M_{ij}) = (C'^1_{ij}, C'^2_{ij})$ where $C'^1_{ij} = C^1_{ij} \cdot g^{r'_{ij}} = g^{r_{ij}+r'_{ij}}(mod\ p)$, $C'^2_{ij} = C^2_{ij} \cdot C'_{ij} \cdot (pk'_{ij})^{r'_{ij}} = M_{ij} \cdot (pk'_{ij})^{r_{ij}+r'_{ij}}(mod\ p)$, and $C'_{ij} = (C^1_{ij})^{\Delta x_{ij}}$. It is also known from EIGamal encryption algorithms that the mobile user U and the cloud server cannot obtain M_{ij} without the updated private key sk'_{ij}, that is, its security is guaranteed by EIGamal encryption algorithm. After the key is updated, any one cannot decrypt the plaintext data without the updated private key. Similarly, the cloud server can't get the plaintext of each data block. In fact, only the authorized U can get the private key sk'_{ij} from the data service provider by executing the OT_1^n protocol.

In addition, when the key is updated, the data service provider only sends the auxiliary message $F_{ij} = (C'_{ij}, \Delta pk_{ij})$ to the cloud serve, where $C'_{ij} = (C^1_{ij})^{\Delta x_{ij}}$ and $\Delta pk_{ij} = g^{\Delta x_{ij}}(mod\ p)$. Similarly, according to the difficulty of solving the discrete logarithm problem, the cloud server cannot get any private information from the auxiliary message $F_{ij} = (C'_{ij}, \Delta pk_{ij})$.

In summary, it can be seen from the above analysis that the required data associated with the user' location is stored in the cloud server by the encrypting method, and the decryption key is managed by the data service provider, where the data and the key are stored and managed separately, so that both the location privacy of the user and the data privacy of the DSP are protected well. In addition, the data service provider may add a small amount of virtual data (*i.e.*, false data) to each block in a moderate amount, so that the format and size of all blocks is completely consistent, which also reduces the risk of information leakage.

3.3 Performance

1) Computation costs. In the proposed scheme, the computation costs involve three parties, the user, the data service provider, the cloud server. The computation cost of the user is to query the private key by the OT protocol. The computation cost of the data service provider is to manage (*e.g.*, generate and update) all key pairs and encrypt all blocks of data. Furthermore, in private query phase, the data service provider still needs to encrypt all private keys, such that only the authorized user can decrypt one of them. The computation cost of the cloud server is to assist the data service provider to update all ciphertexts stored in the cloud server. The detailed computation costs of the proposed scheme are listed in Table 1. Here, D_G, M_G, and E_G denote the costs of one modular division operation, one modular multiplication operation, and one modular exponentiation operation in group G, respectively.

2) Storage costs. In our scheme, the storage costs mainly include two parties, the cloud server and the data service provider. The storage cost of the cloud server is to store all ciphertexts of the data service provider and all public keys of different blocks. The storage cost of the data service provider is to keep all private keys in secret. Here, we assume that the lengths of a plaintext block, a public key and a private key are 512 bits, 512 bits and 160 bits. The detailed storage costs of this scheme are shown in Table 2.

3) Communication costs. In initialization phase and key update phase, the communication costs between the data service provider and the cloud server are to exchange messages, including the ciphertexts of the data and the updated messages of the keys, which are related to the number of the blocks, *i.e.*, $s \times t$. In private query phase, the user only needs to send one message to the data service provider and to receive $s \times t$ messages (i.e, the ciphertexts of all keys) from the data service provider in turn.

According to the above analysis, this scheme has the following advantages compared with the existing methods of location privacy protection:

1) Combining with the cloud storage service, a large number of data, which would be stored originally in the data service provider, is converted to the ciphertext, and then stored in the cloud server, so that it not only effectively protects the data privacy, but also greatly reduce the storage costs of the data service provider;

2) The user can obtain the decryption key of his/her required ciphertext by the *OT* protocol, which can effectively protect the user's location privacy. Furthermore, the data service provider only needs to return the ciphertexts of the keys to the user, instead of

Table 1: The computation costs of the proposed scheme

Participant	Computing cost
User	$D_G + M_G + 3E_G$
Data service provider	$(s \times t)D_G + 2(s \times t)M_G + 9(s \times t)E_G$
Cloud serve	$3(s \times t)M_G + 2(s \times t)E_G$

Table 2: The storage costs of the proposed scheme

Participant	Storage cost
Cloud serve	$3(s \times t)512bits$
Data service provider	$(s \times t)160bits$

the ciphertexts of all actual data, so it can effectively reduce the communication cost between the user and the data service provider;

3) When updating the block keys and ciphertexts, the data service provider only needs to update the key of each block, while the main update operations of the corresponding ciphertext are completed by the cloud server, so it can effectively reduce the computation cost of the data service provider;

4) The key generation, distribution, storage and update are independently managed by the data service provider, without any other key management center or a trusted third party, so it can lower the implementation costs of the system, and meantime improve the performance of the system.

4 Conclusion

In this paper, we present a new location privacy protection model by introducing the cloud server, and then design the corresponding protocol without any trusted third party, in which we employ the technologies of data encryption, oblivious transfer and space anonymous. The analysis results show that our proposed scheme can well ensure both the location privacy of the query user and the data privacy of the data service provider in location-based services, and especially it can greatly reduce the storage, computation and communication costs of the data service provider.

Acknowledgments

This work was supported by Natural Sciences Key Fund of Anhui Province Education Department (No. KJ2018A0823), and Research Project of Hefei Technology College (No. 201814KJB001).

References

[1] M. Ahmadi and B. S. Ghahfarokhi, "Preserving privacy in location based mobile coupon systems using anonymous authentication scheme," in *The 13th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC'16)*, pp. 60–65, 2016.

[2] B. Bamba, L. Liu, P. Pesti, and T. Wang, "Supporting anonymous location queries in mobile environments with privacygrid," in *Proceedings of the 17th International Conference on World Wide Web*, pp. 237–246, 2008.

[3] B. S. Bhati and P. Venkataram, "Performance analysis of location privacy preserving scheme for manets," *Internationl Journal Network Security*, vol. 18, no. 4, pp. 736–749, 2016.

[4] Y. Che, Q. Yang, and X. Hong, "A dual-active spatial cloaking algorithm for location privacy preserving in mobile peer-to-peer networks," in *Wireless Communications and Networking Conference (WCNC'12)*, pp. 2098–2102, 2012.

[5] C. Y. Chow, M. F. Mokbel, H. V. Leong, *et al.*, "On efficient and scalable support of continuous queries in mobile peer-to-peer environments," *IEEE Transactions on Mobile Computing*, vol. 10, no. 10, pp. 1473–1487, 2011.

[6] R. Ghasemi, M. M. A. Aziz, N. Mohammed, M. H. Dehkordi, and X. Jiang, "Private and efficient query processing on outsourced genomic databases," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 5, pp. 1466–1472, 2017.

[7] M. S. Hwang, C. C. Lee, T. H. Sun, "Data error locations reported by public auditing in cloud storage service," *Automated Software Engineering*, vol. 21, no. 3, pp. 373–390, Sep. 2014.

[8] H. Jannati and B. Bahrak, "An oblivious transfer protocol based on elgamal encryption for preserving location privacy," *Wireless Personal Communications*, vol. 97, no. 2, pp. 3113–3123, 2017.

[9] R. Kaur, I. Chana, and J. Bhattacharya, "Data deduplication techniques for efficient cloud storage management: A systematic review," *The Journal of Supercomputing*, vol. 74, no. 5, pp. 2035–2085, 2018.

[10] L. Kuang, Y. Wang, P. Ma, L. Yu, C. Li, L. Huang, and M. Zhu, "An improved privacy-preserving framework for location-based services based on double cloaking regions with supplementary information constraints," *Security and Communication Networks*, vol. 2017, 2017.

[11] C. C. Lee, M. S. Hwang, and S. F. Tzeng, "A new convertible authenticated encryption scheme based on the elgamal cryptosystem," *International Journal of Foundations of Computer Science*, vol. 20, no. 02, pp. 351–359, 2009.

[12] N. Li, T. Li, and S.Venkatasubramanian, "Closeness: A new privacy measure for data publishing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 7, pp. 943–956, 2010.

[13] T. C. Li and W. T. Zhu, "Protecting user anonymity in location-based services with fragmented cloaking region," in *IEEE International Conference on Computer Science and Automation Engineering (CSAE'12)*, vol. 3, pp. 227–231, 2012.

[14] J. Ling, Y. Wang, and W. Chen, "An improved privacy protection security protocol based on NFC," *International Journal of Network Security*, vol. 19, no. 1, pp. 39–46, 2017.

[15] L. Liu, Z. Cao, C. Mao, "A note on one outsourcing scheme for big data access control in cloud," *International Journal of Electronics and Information Engineering*, vol. 9, no. 1, pp. 29–35, 2018.

[16] Z. Y. Luo, R. H. Shi, M. Xu, and S. Zhang, "A novel quantum solution to privacy-preserving nearest neighbor query in location-based services," *International Journal of Theoretical Physics*, vol. 57, no. 4, pp. 1049–1059, 2018.

[17] B. Niu, Z. Zhang, X. Li, and H. Li, "Privacy-area aware dummy generation algorithms for location-based services," in *IEEE International Conference on Communications (ICC'14)*, pp. 957–962, 2014.

[18] X. Pan, J. Xu, and X. Meng, "Protecting location privacy against location-dependent attacks in mobile services," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 8, pp. 1506–1519, 2012.

[19] S. Rezaei, M. Ali Doostari, and M. Bayat, "A lightweight and efficient data sharing scheme for cloud computing," *International Journal of Electronics and Information Engineering*, vol. 9, no. 2, pp. 115–131, 2018.

[20] Q. C.To, T. K. Dang, and J. Küng, "A hilbert-based framework for preserving privacy in location-based services," *International Journal of Intelligent Information and Database Systems*, vol. 7, no. 2, pp. 113–134, 2013.

[21] E. Troja and S. Bakiras, "Leveraging p2p interactions for efficient location privacy in database-driven dynamic spectrum access," in *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 453–456, 2014.

[22] X. Wang, L. Dong, C. Xu, and P. Li, "Location privacy protecting based on anonymous technology in wireless sensor networks," in *The 7th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP'15)*, pp. 229–235, 2015.

[23] Z. Wang, Y. Lu, G. Sun, "A policy-based deduplication mechanism for securing cloud storage," *International Journal of Electronics and Information Engineering*, vol. 2, no. 2, pp. 70–79, 2015.

[24] X. Yi, R. Paulet, E. Bertino, V. Varadharajan, *et al.*, "Practical approximate k nearest neighbor queries with location and query privacy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 6, pp. 1546–1559, 2016.

Biography

Li Li received the MS degree from Anhui University in 2018. She is currently a Lecturer with Hefei Technical College. Her research works mainly include network and information security, cloud computing and privacy protection.

Zhengjuan Lv received the MS degree from Shandong Normal University in 2012. She is a Lecturer with Hefei Technical College. Her research interest is computer application and multimedia technology.

Xiaohong Tong received his MS degree from Hefei University of Technology China in 2006 and he was a visiting scholar at Bloomfield University USA in 2016. Since 2005, he has been an Associate Professor at the Information Center of Hefei Technical College. His research and project works focus on data communication and signal processing.

Run-hua Shi received the Ph.D. degree from University of Science and Technology of China in 2011. He is currently a Professor with Anhui University. His current research interest includes classical and quantum cryptography, in particular, privacy-preserving multiparty computation.