Journal of Theoretical and Applied Information Technology
31st August 2021. Vol.99. No 16

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

PRIVACY PRESERVING LOCATION BASED SERVICES QUERY SCHEME BASED ON FULLY HOMOMORPHIC ENCRYPTION

FIFI FAROUK[1], YASMIN ALKADY[2], RAWYA RIZK[3]

[1]Technology and Information Systems Dept., Port Said University, Port Said, Egypt.

[2]Faculty of Information Technology and Computer Sciences, Sinai University, Egypt

[3]Electrical Engineering Dept.,Port Said University, Port Said, Egypt.

E-mail: [1]bondoka7000@himc.psu.edu.eg, [2]yassmin.hosny@su.edu.eg, [3] r.rizk@eng.psu.edu.eg

ABSTRACT

Location-Based Services (LBSs) are increasingly popular in today's society. LBSs are used in a broad variety of applications such as social networks with location sharing. LBSs can help people enjoying a convenient life and has attracted considerable interest recently. However, the privacy issues of LBS are still challenging today. Usually LBS Providers (LBSPs) offer user privacy protection statement to assure users that their private location information would not be given away. However, many LBSs run-on third-party cloud infrastructures. So, the privacy is still in a big problem. While a large number of privacy-preserving solutions for LBS have been proposed, nowadays most of these solutions do not consider the fact that LBS is typically cloud-based. Aiming at the challenges, in this paper, we present a new efficient Privacy Preserving LBS Query scheme based on Fully Homomorphic Encryption (*PPQ_FHE*). In *PPQ_FHE* scheme, the LBSP's data are outsourced to the cloud server in an encrypted manner, and a registered user can get accurate LBS query results without divulging his/her location information to the LBSP and cloud Server Provider (CSP). Specifically, based on improving FHE over Advanced Encryption Standard (AES). We provide a security analysis to show that *PPQ_FHE* scheme preserves privacy in the presence of different threats. *PPQ_FHE* analysis and evaluation results demonstrate that proposed *PPQ_FHE* scheme can preserve the privacy of the LBS users in an efficient and secure way. The results prove the feasibility and efficiency of *PPQ_FHE* scheme in terms of query's response time, query accuracy, throughput and query overhead.

Keywords: *Advanced Encryption Standard, Fully Homomorphic Encryption, Location-Based Services, Privacy Preserving, and Security.*

1. INTRODUCTION

Nowadays, widely spread computing Location-Based Services (LBS) can allow people to enjoy a better life and have recently attracted significant interest [1–2]. Today, nearly all devices such as mobile wireless phones and tablets have Global Positioning System (GPS) to collect the location information of their users. Although there are some concerns from the users about privacy, security and third-party usage of their private location information, these LBS services reached everywhere. Still, LBS Providers (LBSPs) despite the user's legitimate privacy concerns, are unwilling to build private LBS systems; in which they cannot access location information of users.

With the fast development of cloud computing, more and more LBSPs start to consider outsourcing their location data and services to the cloud server to benefit from the several advantages of cloud computing such as economic costs, huge flexibility, fast deployment, excellent performance of computations, and infinite bandwidth resources [3]. Beside reduction of scaling LBS deployment and data storage cost, LBSPs go to Cloud Service Providers (CSPs) for assistance. As an obvious example, Yelp and Foursquare use the cloud services of Amazon.com. However, the principle of query processing and data outsourcing to the cloud leads to more privacy and confidentiality issues [4] because the identity of the user, query content, and user's location should be protected from both LBSP and CSP, beside protecting LBS data from the CSP.

Therefore, the user requests should also be encrypted prior to being submitted to the cloud server. However, encryption of data makes the available location query service a challenging task, since the ciphertext does not bear the natures of numerical computation and character match in the plaintext field. So, there are two basic problems that should be solved in a cloud-based LBS application over the outsourced encrypted location data: 1) How to find out all target locations over the encrypted location data according to the encrypted user request; 2) How to compute the distances between these target locations and user current location over the encrypted outsourced location data.

In this paper, a new scheme, Privacy Preserving Query using Fully Homomorphic Encryption (PPQ_FHE) is proposed. It shows how to improve privacy preserving LBS system based on cloud computing by using new approach to deal with encrypted data, this approach is called Fully Homomorphic Encryption over Advanced Encryption Standard "FHE over AES" [5], it enables the cloud server to perform LBS query over the encrypted LBS data without divulging users' location information. However, the scheme only can enforce a user location coordinate query according to a user's current location.

The rest of our paper is organized as follows. In Section 2, we review some related literatures. Section 3, the preliminaries are discussed. Then, a system model and a threat model are formalized. Problem statements are depicted in Section 4. The proposed approach is presented in Section 5. What's more, some analyses and performance evaluations are conducted in Sections 6 and 7, respectively. Finally, we draw our conclusions in Section 8.

2. RELATED WORK

The core of privacy-preserving in LBS is to cut off the relevance of the identity information, location information, and query content in the case of continuous querying. The query contents to ensure that the user information is not relevant to each other.

There are two modes for communication related to the privacy preserving query in LBS; the first relies on Trusted Third Party (TTP) and the second is free from trusted third party (TTP-free). Although TTP-based ways can collect sufficient information to achieve maximum protection of privacy [6-7] based on K-*anonymous* region formation [8-9]. The basic principle of this concept is via sending a group of locations not just the real location, so the possibility of guessing the location of the user is always less than 1/K. The majority of techniques that are based on K-*anonymity* [10-11] use a middleware (the anonymizer) as shown in figure 1. The anonymizer represents a third party that is in charge of Cloaking Region (CR) formation, that contains the real location of the user, as well as (K-1) other neighbors; where K represents the total number of users including the origin requester. Using this technique, a typical scenario could be a user trying to locate the nearest bank. The user sends requests (including his/her credentials) to the anonymizer via a wireless network. Then, the anonymizer, that keeps all current users' locations, will authenticate the requester and chooses a set of K-1 neighbors in order to form a CR that could be sent instead of the user's location. This way reduces the risk of breaching the privacy of the user by making it harder to know the position that has triggered the process (since the server answers the whole CR).

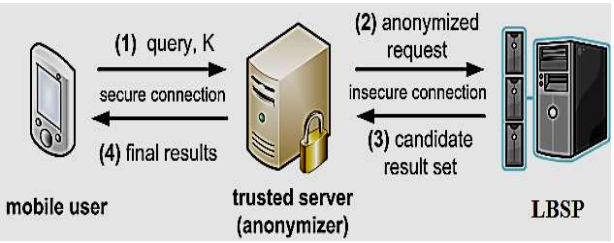

Figure 1: The k-anonymity Model

However, this approach has some drawbacks. At first, the data of users is exposed to the third party (the anonymizer), so the user's privacy preserving issue is not solved yet. Second, the third party (the anonymizer) needs to continuously update the current location of all the subscribed users, this needs a permanent communication and remote users' monitoring, which obviously forms violation for the privacy of the users. Finally, the strength of these approaches depends as a whole on having a huge number of neighbors at the time of requests' receiving. So, relying on a middleware cannot be the best solution to secure location-dependent queries and hence any secure solution requires a direct communication with the LBSP without intermediate parties or TTP; as there are two problems: (1) It is hard to obtain the TTP that fill the bill. (2) Centralized attack on TTP can make it the bottleneck of the query scheme.

TTP-free-based solutions take advantage of the limited information to help maximize privacy

protection. This type of solutions has the advantage of being efficient in all three aspects of query privacy-preserving, efficiency and accuracy over TTP-based solutions.

Lien et al. [12] have proposed a Private Circular Query Protocol (PCQP) without trusted third party usage, which utilizes Paillier and Moore curve cryptosystem to apply the location and query protection of content. The scheme has a huge number of homomorphic addition and multiplication operations, so it needs higher computations and communication cost.

Utsunomiya et al. [13] made some improvements on the basis of PCQP and proposed a Lightweight Private Circular Query Protocol (LPCQP) with divided Place of Interest-table (POI-table) to effectively reduce the number of homomorphic additions and multiplications. The dividing of the POI-table is performed only once in the initialization process and the number of sub-tables observably influence the accuracy of the query. In some extreme cases, the scheme cannot return enough k nearest neighbors of POIs (k POIs) to the user. Moreover, when the number of POIs in a sub-table is much bigger than k, the homomorphic addition and multiplication operations lead to greater number of unnecessary cost of computation. LPCQP uses the homomorphic encryption scheme proposed by Smart and Vercauteren [14] to ensure the security. The scheme requires a big size public key and huge amount of computation.

A Lightweight Privacy-Preserving Authentication protocol, (LPPA) [15] uses a communication model for VANET which achieves the privacy protection requirements, using only hash functions and exclusive-OR operations. This scheme is considered as a weak scheme against attacks because the privacy can't be guaranteed without any powerful type of encryption.

In [16], Tianqi et al. presents Human-in-the-Loop-Aided Privacy-Preserving (HLAPP) scheme which applies a new technique (i.e., the block design and the HitL) to construct a privacy-preserving architecture for smart healthcare without using encryption. HLAPP scheme avoids using homomorphic; because of homomorphic drawbacks such as the huge storage overheads and computation brought by the homomorphic encryption. Thus, HLAPP scheme is different from the previous works. It is applied for privacy protection in smart healthcare without needs of location detection by LBS as previous works.

Privacy-Preserving Implicit Authentication (PPIA) scheme [17] uses cosine similarity and partial homomorphic public key encryption scheme. Partial homomorphic encryption scheme performs only one type of computations (addition or multiplication) not both, thus only one type of computations can be performed on the encrypted data. PPIA scheme uses the Paillier public key encryption scheme which is additive homomorphic. So that PPIA scheme is less efficient than schemes that use fully homomorphic. In addition, homomorphic encryption produces noise that concatenate with cipher text.

Privacy-Protection Based on Sanitizable Signature (PP-SS) scheme [18] uses a special type of digital signature to avoid traditional digital signatures, that cannot meet the privacy and diversity requirements. PP-SS scheme uses unique security attributes of sanitizable signatures to achieve the security and privacy protection of medical data and performs it on smart mobile medical scenarios. PP-SS uses digital signature without using any type of encryption. PP-SS scheme have clear drawback which is depending on sanitizer that is usually a TTP. TTP is centralized attack that forms bottleneck of the query scheme.

In [19], an Intelligent Terminal Based Privacy-Preserving (IT-PP) scheme is introduced. It uses the Paillier public key encryption scheme which refers to additive homomorphic. IT-PP is based on additive homomorphic which is partially homomorphic that performs only (addition and subtraction) computations on encrypted data and doesn't perform multiplication computations on encrypted data like fully homomorphic.

Considering the advantages and disadvantages of the PCQP and LPCQP, this paper proposes an efficient private preserving protocol using FHE over AES to mitigate the drawbacks of LPCQP without damaging the security. The proposed scheme utilizes the FHE scheme to address the problem of secure querying over encrypted data in LBS. To omit the noise of homomorphic additions and multiplications, the proposed scheme dynamically produces ciphertext without any noise by using AES symmetric encryption algorithm. The data security depends on FHE over AES [5].

3. PRELIMINARIES

In this section, the cryptographic tools that are utilized in the proposed solution are briefly presented.

3.1. A Brief Overview of Advanced Encryption Standard (AES)

In 2000, the Rijndael algorithm was announced by the National Institute of Standard and Technology (NIST) as the new AES. It was selected because it has a strong performance on about all platforms and it is easy to be implemented in hardware. AES is strong symmetric key encryption algorithm. It has a basic security objective, which the best attack against it should be key exhaustion (trying every possible key until you find one that works). If key exhaustion represents the best attack, then key size will be a determinant for the strength of symmetric key algorithm. To find an n-bit key, it is mandatory to try 2^{n-1} keys, but if you make n sufficiently big, this will be too impractical. AES uses 128, 192, or 256-bit keys.

The proposed *PPQ_FHE* scheme referred to use AES-128. An AES encryption process consists of a number of encryption rounds (*Nr*) that depends on the size of the key. The standard calls for 10 rounds for AES-128. The round function operates on a 4×4 matrix of bytes. The basic operations that are performed during the round function are *AddKey*, *SubBytes*, *ShiftRows*, *MixColumns* [20]. AES has the ability to be implemented homomorphically. It is widely used in security-aware applications [21-22]. Moreover, the AES circuit has a regular (and quite "algebraic") structure, which is amenable to be implemented in hardware. For these same reasons AES is often used as a benchmark for implementations of protocols for secure multi-party computation (MPC) [23-25].

3.2. Homomorphic Cryptography

The homomorphic cryptography objective is to assure the data privacy in communication, storage, or in use through processes with mechanisms and techniques similar to conventional cryptography, but with adding the abilities to perform computation operations over encrypted data, searching an encrypted data, etc. Homomorphic encryption is currently used for several applications such as electronic voting, spam filtration, management of data, and processing queries in clouds, and also multiparty computations. FHE is used to perform any arbitrary computation over encrypted data as shown in figure 2.

Homomorphic encryption scheme has three main categories which are: Partially Homomorphic Encryption (PHE), Somewhat Homomorphic Encryption (SHE), and Fully Homomorphic Encryption (FHE) that will be discussed as following:

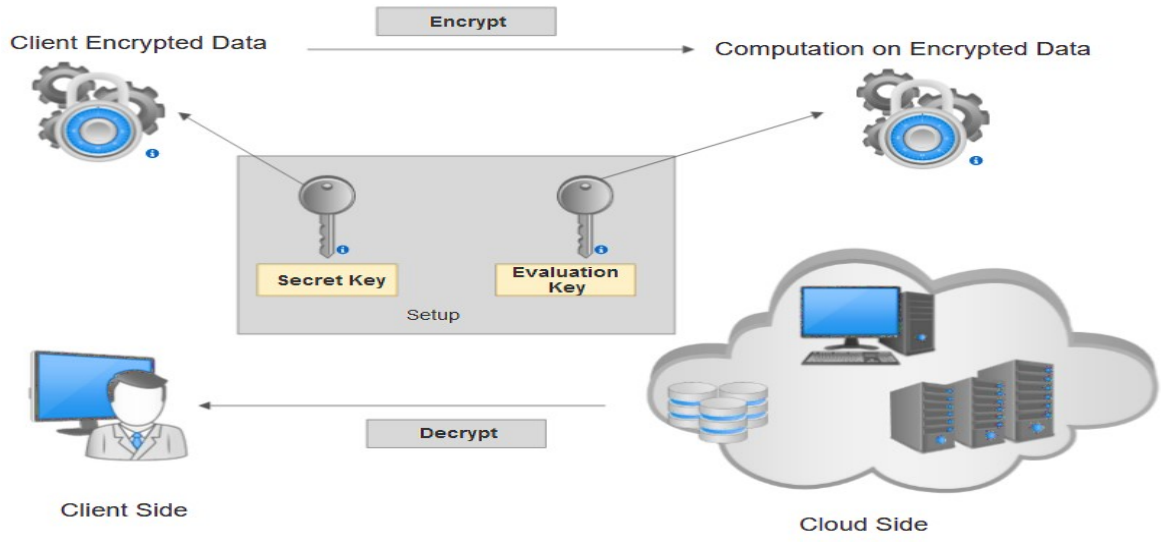

Figure 2: Concept of Homomorphic

- PHE is related to only one of two operations (addition and multiplication). It can be performed on the encrypted data, thus only one of them can be performed on the encrypted data but not both. PHE schemes are useful in certain applications where specific computations are required. Helios

Voting scheme [26] make use of PHE. Using this approach, the encrypted voted ballots can be publicly stored in the cloud and the public can count the votes as well as verify their votes for each candidate without using TTP. PHE according to the type of operations include; additive and multiplicative homomorphic. Additive homomorphic deals with addition of encrypted data. Let us consider two data (a) and (b) and they encrypted to $Enc(a)$ and $Enc(b)$ respectively. On applying additive homomorphic encryption, we get as given in Eq. (1):

$$Enc(a) + Enc(b) = Enc(a + b) \quad (1)$$

So, the sum of the encrypted messages is equivalent to the encrypted sum of those messages. Multiplicative homomorphic is related to multiplication of the encrypted data. Let us assume two data (a) and (b) and they encrypted to Enc(a) and Enc(b). While applying multiplicative homomorphic encryption leads to Eq. (2).

$$Enc(a)xEnc(b) = Enc(axb) \quad (2)$$

- SHE use more than one operation performed on the encrypted data. But the limitation of SHE is that not all operations apply to all types of data. SHE is suitable for a several real time applications like financial, medical and recommender systems. Since SHE supports limited number of operations, it will be much faster than FHE.

- FHE supports any number of operations on any encrypted data. The circuit designed for FHE is evaluated homomorphically. This makes it suitable for any sort of applications working with encrypted data. Compared to PHE and SHE, FHE is little bit less efficient due to the noise overhead.

Homomorphic encryption consists mainly of four tuples represented as, H = {Key Generation (*KG*), Encryption (*Enc*), Decryption (*Dec*), Evaluation (*Eval*)}; where the functions performed as the following:

1- Key Generation: Generating the private-public pair keys for performing encryption and decryption by considering the security parameter as the input.
2- Encryption: The encryption algorithm takes the plaintext and the public key as the input, and generates the ciphertext as the output.
3- Decryption: The decryption algorithm takes the ciphertext and the private key as the input, and produces the plaintext as the output.
4- Evaluation: The evaluation function performs homomorphic evaluation for the ciphertext. Evaluation function takes the public key, a function and the ciphertext as inputs and return another ciphertext as output, as given in Eq. (3),

$$C^* = Eval\,(f, c) \quad (3)$$

Where C^* is the ciphertext after performing homomorphic encryption, c is a ciphertext generated using for homomorphic encryption, f is the function used for the computation of the ciphertext, and pk is the public key.

3.3. Fully Homomorphic Evaluation Over AES

FHE-AES is based on matrix operations which have light computations. It uses symmetric keys of small size thus making it suitable for many data centric applications. It gains its security from being too hard to factorize a large integer [25], which is the base of many public key cryptosystems.

As shown in figure 3, FHE over AES scheme is used for designing an efficient and practically affordable FHE that uses AES symmetric algorithm [27]. It can handle arbitrary size of computations without the need of noise management and has scope of parallelization [28]. AES gives a good design space for investigating FHE techniques because it supports parallel nature as well as algebraic nature of computations.

The main concept is translating operations on integers in a ring $M_4(Z_n^*)$ where M_4 means that all operations are on square matrices of size 4 and Z_n^* refers to a set of integer numbers in algebra theory, ring $M_4(Z_n^*)$ is sufficiently small to be used practically. FHE over AES is used for optimization of communication with the cloud in a bootstrapping-free manner. In the context of making FHE scheme useful enough, we propose a scheme with the following set of operations: *KeyGen*, *Enc*, *Eval,* and *Dec* which are explained in details in [29].

4. SYSTEM MODEL

In this section, the proposed system model is introduced as well as security requirements and then we present the threats.

4.1. System Entity

In our system model, we mainly focus on how the LBSP offers accurate and efficient LBS to users based on the requested privacy-preserving location data. Specifically, the system consists of three entities: LBS user, LBSP, and CSP, as shown in figure 4.

4.1.1. LBS user

LBS users are the location data users, who enjoy convenient LBSs by submitting LBS query requests to the LBSP anywhere and anytime. In order to hide query requests of LBS users for protecting privacy: Firstly, LBS users encrypt their query requests and submit the encrypted query requests to LBSP to detect available distention ways over encrypted data and then send these data to the CSP for computing the shortest way also over encrypted data. Note that the LBS users are usually referred to as the legal registered users, unregistered users and revoked users from the provider cannot enjoy LBSs. figure 5 shows different types of LBS users with their relation with LBSP.

4.1.2. LBSP

LBSP is an owner for location data that outsources the location data to CSP. To assure the confidentiality achievement for location data, all location data is sent to LBSP in an encrypted manner and then is being uploaded to the cloud server in an encrypted manner also. Besides that, as LBS user tries to join the system, the LBSP affords authentication and registration services for him. Once the LBS user passes authentication, the LBSP will send certain security parameters of interest to the user through well secured channels of communication. So, the LBSP is also able to revoke expired LBS users, who no longer have the capabilities of query for the outsourced location data when revoked by the LBSP.

4.1.3. CSP

Encrypted LBS query request submitted by a legal LBS user to LBSP; which outsources large-scale location data to the cloud server to be managed by CSP; for enjoying the low-cost storage services and powerful computation services. CSP is responsible for computing the shortest way to the desired destination. In the whole query processes, the CSP does not know any contents about outsourced location data, the user's query request, and the current location of the LBS user.

4.2. Security Requirement

LBS data *confidentiality* and LBS user's query privacy are vital for secure LBS applications success. In our security model, the CSP, the LBSP, and the LBS user are considered as honest but curious. Specifically, the LBSP affords the LBS data accurately, but it is curious and want to know more about the query location of LBS user's; the CSP honestly performs the operations for searching the required LBS data for the user, but also tries to analyze the encrypted LBS data for obtaining the LBS data in the real world and so guessing the LBS user query location according to the request of LBS user. Moreover, the LBS user may try to access LBS data that are not in his/her privileges of access or to access LBS data without registering. So, to assure LBS data confidentiality in the cloud server and the LBS user's query privacy, the next security requirements should be fulfilled in Privacy.

Preventing collision attacks: On one side, the sensitive data assets of LBSP are held secret from the CSP, i.e., even if the CSP stores all the data from the LBSP and queries from the LBS users, it does not have the ability to determine any LBS data item. On the other side, the query location of LBS user is protected from the CSP, the LBSP, and other users, i.e., even if the CSP can obtain all LBS users' queries and all the responses to the LBS users, it cannot identify the LBS users' query location precisely. Under such circumstances, the resource of LBS data and location information of user can assure the privacy-preserving requirements. Moreover, the requirements of privacy include the CSP's responses as well, i.e., just the legal LBS user only can decrypt them. Note that, in the proposed current model, it is even taken into consideration that any two parties may collude to disclose the privacy of third party and such situation is solved because each party is not able to reveal any other party's data, thus preventing collision attacks probability.

Authentication: An encrypted LBS query, which is sent by a legal LBS user and has not been altered while being transmitted, should be authenticated, i.e., if an illegal LBS user fakes an LBS query, this harmful operation should be detected. Moreover, only the correct queries can be

received by the LBSP. Meanwhile, the LBSP responses should as well be authenticated so that the LBS users can receive reliable and authentic query results at LBS system that is secured.

4.3. Threats

We assume that the mobile cloud is "honest but curious". This threat model is mostly consistent with other works in this area with one important difference; we specifically consider the following threats in proposed system:

1) *Tracking threat:* As the LBS user may continuously send location queries, the CSP and LBSP are able to record queries. Based on the analysis of recorded queries, it may infer approximate locations if CSP and LBSP are able to learn any significant information from recorded queries.
2) *Linkage threat:* The CSP stores the encrypted index and performs computations to detect the shortest way of destination. In addition, the LBSP can execute search operation to link the query with the retrieved result. Furthermore, it is also possible for the CSP and LBSP inferring the query based on the index.
3) *Frequency analysis threat:* Since the CSP and LBSP are able to track and record access frequencies, it might be possible for the CSP and LBSP to infer some locations by analyzing the home and work more often compared to other access pattern, for example, a user tends to visit place.

5. IMPLEMENTING PPQ_FHE SCHEME

Here, we present privacy-preserving query using FHE (PPQ_FHE) scheme in LBS and cloud outsourced, which mainly consists of five phases: system initialization, new user registration, data creation system, user revocation, and location data encryption.

5.1. System Initialization

The operation of system initialization is executed by the LBS user, u_i , to setup the system running environment. The parameters used in the system are presented in table 1.

First u_i chooses λ as input, then running keyGen(λ) to compute *SK* to encrypt its query. The key arranged in the form of a matrix of 4×4 bytes in Z_n^* hence does not involve any computation theoretically. keyGen(λ) is described in key generation function.

```
/*Key_generation Function*/
   KeyGen(){
    int λ ;
/*λ is a security parameter*/
   1.   get a  matrix k of size4, k €M4(ZN);
   2.   SK=KeygenAES(λ);
   3.   output (SK);
        4. }
```

In addition, u_i also implements a secure AES symmetric encryption algorithm *Enc( )*. u_i selects a random value $X \in Z_n^*$ to perform *MD5* [30] hashing function *H ( )*, where $H : \{0, 1\}^* \rightarrow Z_n^*$ in the system, which maps a message of arbitrary length to an element in Z_n^* to check integrity. From discrete assumption problem there is cyclic group *G* which generates the parameter of bilinear group called *g* [31].

Then u_i keeps < *SK*, *X*> as the master key secretly; then sends <*X*> to LBSP through secure channel and publishes the public parameter <*g*>. In the next side LBSP selects a random value $X_{ui} \in Z_n^*$ for u_i and computes $g^{X/X_{ui}}$. Then X_{u_i} and $g^{X/X_{ui}}$ are sent to u_i by secure communication channel.

5.2. New User Registration

The LBS user u_i needs to register with the LBSP in order to be able to get a service and send queries. In the initial phase, u_i gets the AS_i from the LBSP. u_i receives X_{ui} and $g^{X/X_{ui}}$ based on AS_i and randomly selects $r_{ui} \in Z_n^*$. $regK_{ui}$ is computed as follows:

$$regK_{ui} = g^{X/X_{ui}} \times g^{r_{ui}} \quad (4)$$

Then, (u_i, $regK_{ui}$) are sent to LBSP through secure channel and stored in LBSP at *KList*. As a registered user of the LBSP; u_i is authorized with <$regK_{ui}$> which will be utilized for retrieving LBS resources in a privacy preserving way later.

During the registration, the client negotiates permissions to get access to groups of data records and *ASP*. *ASP* is generated by LBSP. It is used to determine whether the client has permissions to access a record.

Then, u_i sends $regK_{ui}$ to the LBSP in the same secure channel as a registration request. LBSP

verifies $regK_{ui}$ as element of *KList*. After that, LBSP selects *ASP* based on the input of attribute set represented by $regK_{ui}$ which is described in the following authorization function.

```
/*Authorization Function*/
Authorize(){
    1. int regK_i ;
    2. list KList;
/*regK_ui is a registration key*/
/*klist is a list of keys that saved on LBSP*/
    3. if (regK_ui ∈ Klist );
/* verifying regK_ui is the element of KList*/
    4. then (u_i ← ASP(regK_ui ));
/* Selecting ASP according to detected user*/
    5. Else
    6. Return ( Not Authorized);}
```

5.3. Data Creation System

Generally, the LBSP has many LBS resources, and most of resources' information. The construction of LBS data is organized as ***the Category Set*** and ***the Location Data Set***, a CATEGORY refers to the location data general name, that contains multiple concrete location data. Each concrete location data is a four-tuple {*ID, TITLE, COORDINATE, DESCRIPTION*} format of {ID_d, T_d, (X_d,Y_d), D_d}, where *d* means any location data belonging to C_d in a Category Set.

All these attributes describe a certain location detailed information as shown in the following example:

Category Set {***CATEGOR***}	**Location Data Set** {***ID, TITLE, COORDINATE, DESCRIPTION***}
Hotel:	*{*
	{150, TULIP Hotel, (x_{250}, y_{250}), (5 stars)},
	{151, Westin Hotel, (x_{251}, y_{251}), (4 stars)},
	....
	{200, Four Seasons Hotel, (x_{500}, y_{500}), (5 stars)}
	}

In this paper, FHE over AES is adopted for encrypting the LBS data prior to being outsourced to CSP. The proposed scheme gives the CSP the ability for providing exactly the same query service over encrypted location data; as the plaintext environment mentioned before just like if location data and user's query request information are exposed to the CSP.

In *PPQ_FHE* scheme, u_i needs to encrypt the CATEGORY of interest and his/her location coordinates by AES. FHE over AES encrypted data is performed by LBSP. CSP then performs the computations of LBS query over outsourced encrypted location data; in order to determine the shortest way to the destination of interest. Surely, the necessary decryption operations will be involved for u_i once the encrypted result of LBS query is received.

Moreover, *PPQ_FHE* scheme designs flexible and efficient mechanisms for user registration and user revocation, to assure that registered users only have the ability for using LBS system and unregistered users or revoked users cannot access it. Figure 6 illustrates the whole architecture of PPQ_FHE.

5.4. User Revocation

User revocation is a necessary and yet challenging task in practical implementation of LBS system. In this paper, we propose an efficient user revocation mechanism while still having the ability to prevent the revoked user from having the query response effectively. More concretely, for a given user who is going to be revoked by LBSP, the LBSP will scan the information of user in the *KList* to find out the u_i information and deletes (u_i, $regK_{ui}$). Once (u_i, $regK_{ui}$) is deleted from *KList*, so u_i is no longer has the capability of getting a response to his request and search location data because this user has been revoked already.

5.5. Location Data Encryption

- ***Implementing AES at User Side:***

When u_i searches a location of interest, it submits the specified C_d, his/her current location coordinates, and coordinates of destination location to the LBS system. The destination location coordinates are determined by GPS [32]. The LBSP first searches over category set according to the submitted C_d to get all target locations; and then sorts target locations in an ascending order according to the distances between the user's current location and desired target destination locations before outsourcing them to CSP. They are easily computed by CSP according to the user's coordinate and each target location coordinate, and finally the nearest location is returned to the user using Euclidean distance.

In order to achieve location data security, u_i needs to encrypt all location information with AES before sending them to LBSP. LBSP performs FHE over AES encrypted data then outsources them to the CSP to perform computations over encrypted data to detect the nearest location to u_i.

To enable an efficient and privacy-preserving LBS query, *PPQ_FHE* scheme will be used to encrypt the different location data attributes. *PPQ_FHE* scheme takes the following steps to encrypt the location data sets. These steps are shown in the Encryption and Hashing function at the user side.

First, u_i uses secret value X to perform MD_5 hashing function to C_d. Increasing the sample space of location information can resist the exhaustive attack. It can be implicitly formed by:

$$E_1 = H(C_d)^X \quad (5)$$

Second, u_i adopts AES symmetric encryption scheme to encrypt *TITLE* and *DESCRIPTION* attributes. It can be implicitly formed as follows:

$$E_2 = ENC_{AES}(T_d, D_d, SK) \quad (6)$$

Third, u_i uses the secretly preserved invertible matrix M_4 (arranged in the form of a matrix of 4×4 bytes in Z_n*) to encrypt destination coordinate (X_d, Y_d). It can be formed as follows:

$$E_3 = ENC_{AES}(M_4(X_d, Y_d), SK) \quad (7)$$

Fourth, for the remaining item, the current location coordinate (X_{ui}, Y_{ui}), u_i adopts AES symmetric encryption scheme to encrypt coordinate (X_{ui}, Y_{ui}) arranged in the form of a matrix of 4×4 bytes in Z_n*. It can be implicitly formed as:

$$E_4 = ENC_{AES}(M_4(X_{ui}, Y_{ui}), SK) \quad (8)$$

The output encrypted items are then grouped in the form of encrypted query EQ_1. It can be implicitly formed by:

$$EQ_1 = \{E_1, E_2, E_3, E_4\} \quad (9)$$

```
/*Encryption and Hashing function at user side*/
Enc_User(){
String Cd;
/* Cd is a category item*/
int X;
/*X is a secret value generated by the user and
sends to LBSP*/
E1= H( Cd )X;
/*Performing Hashing function of the category*/
E2= ENCAES (Td , Dd ,SK);
/*Performing AES Encryption function of the
TITLE and DESCRIPTION attributes*/
E3= ENCAES (M4 (Xd,Yd), SK);
/*Performing AES Encryption function of the
Coordination of the destination attribute.*/
E4= ENCAES (M4 (Xi,Yi), SK );
/*Performing AES Encryption function of the
Coordination of the user.*/
output (EQ1= { E1 , E2, E3, E4});
/*Concatenate the results in Encrypted Query in
the first stage.*/
}
```

- ***Implementing FHE at LBSP side:***

To preserve user's query privacy at LBSP as well as enabling correct search over encrypted location data, LBSP performs evaluation function of fully homomorphic over AES encrypted data. There is only one general evaluation function (f) defined for computations. It is expected that f can be translated into basic operations on integers.

In order to perform addition, subtraction, multiplication and division of two numbers in a homomorphic manner, we add/subtract/multiply/divide their encrypted query simply as two matrices. Our evaluation function doesn't require any evaluation key. All operations on matrices are also performed within the ring $M_4(Z_n^*)$. The importance of evaluation function, which is represented in Eq. (10), is detecting all the ways that lead to required destination which refer to required POIs.

$$EQ_2 = Eval(f, EQ_1) \quad (10)$$

```
/*Evaluation function at LBSP side*/

Eval_LBSP(EQ1){
    1. E2' =Eval (f, E2);
    2. E3' =Eval (f, E3);
    3. E4' =Eval (f, E4);
    4. EQ2= { E2, E3, E4};
    5. List_EQ2=List (EQ2);
    6. output (List_EQ2);}
```

Then LBSP sends $List\_EQ_2$ of the ways that lead to a certain destination to CSP; to achieve privacy preserving. The powerful CSP searches over encrypted outsourced location data on behalf of the user query. encrypted target location (closest POIs to the user location) is returned back by CSP which sends it to LBSP.

The distance between target location *T* and *VU* is calculated via *haversine* formula [33]; which is also known as great circle distance. It is one of the example methods that bare used to resolve the problem of distance calculation and this method is also used for many researches. *Haversine* formula is the method used for calculating the distance between two coordinates on a two-dimensional map. The distance is the real distance by which the earth's spherical trigonometry. This formula performs calculation from current point to destination point with trigonometric function by using latitude and longitude. In the calculation steps, *Haversine* will first change the value of the latitude and longitude integer number into radians by dividing the values of latitude and longitude by 180/pi. It can be implicitly formed by Eqs.(11-12). The value of 180/pi is approximately 57.29577951.

$$lat = Latitude\ /(180/pi) \quad (11)$$

$$long = Longitude\ /(180/pi) \quad (12)$$

Then these numbers are calculated in the algorithm *Haversine*. The formula of *Haversine* is:

$$d = 3963.0 \times cos\ [(sin\ (lat\ VU) \times sin\ (lat\ T)) + cos\ (lat\ VU) \times cos\ (lat\ T) \times cos\ (long\ T - long\ VU)] \quad (13)$$

Where *d* is the distance between two coordinates on two-dimensional map, *d* is in miles. It can be multiplied by 1.609344 in order to be converted into kilometers.

Then CSP selects the closest POI and sends it to LBSP; who translates it into LBS data form, and sends the result attached with hashing value E_1 to *VU*. Then LBSP gets the closest POI and puts it in a form of LBS data then sends the result to u_i. This operation is done using Minimum POI formalization function.

```
/*Minimum POI formalization at LBSP side*/
        Formal_Q( ){
1-      EQ3= Q(closest POI);
        /*Q is a function that puts closest POI in a
        query form*/
2-      Output (EQ3);
        }
```

Finally, u_i performs decryption process described in the following decryption function.

```
/*Decryption function at user side*/
    Dec_User(EQ3){

1. ( Cd )^X= decode(E1');
/* decode is tools to decrypt MD5 by comparing
hashed value with database*/
2. (Td , Dd)=DEC_AES(E2');
/*Performing AES Decryption function to return
TITLE and DESCRIPTION attributes*/
3. (Xd,Yd)=DEC_AES(E3');
/*Performing AES Decryption function to destination
Coordination attribute.*/
4. (Xi,Yi)=DEC_AES(E4');
5. }
/*Performing AES Decryption function to user
Coordination attribute*/.
```

6. SECURITY ANALYSIS

In this section, the security properties of *PPQ_FHE* scheme are analyzed. Specifically, following the security requirements, this analysis will focus on how the proposed *PPQ_FHE* scheme can achieve the user's query location privacy, source authentication of the query request and response, and the LBS data confidentiality.

- The u_i's query location privacy preserving is achieved in the *PPQ_FHE* scheme. In *PPQ_FHE* scheme, u_i's query is encrypted in the form of (E_1, E_2, E_3, E_4) by using AES symmetric encryption cryptography before being sent to LBSP. To avoid the exhaustive attack against (E_1, E_2, E_3, E_4); the LBSP performs FHE over encrypted data with AES to detect all the ways leading to the required destination, then forwards them to CSP to search over encrypted outsourced location data and computes nearest location and then returns encrypted result to LBSP. After that LBSP delivers result to u_i for decryption. Since *SK* is

only known by registered user, the cloud server cannot deal with u_i and his/her query.

- The authentication of the query request and response are achieved in *PPQ_FHE* scheme. Each registered user is signed by secure mechanism with the help of *ASP* to guarantee the source authentication. Moreover, for any unregistered user, since he/she does not have the secret $regK_{ui}$, he/she also cannot submit valid query request to the LBSP.
- *PPQ_FHE* scheme can achieve confidential user query. LBSP obtains encrypted data with a secure symmetric encryption algorithm AES and also *PPQ_FHE* scheme can achieve confidential LBS data. Specifically, the cloud sever cannot obtain the actual location information of the resource, although it can get all the outsourced data items and users' query information. In *PPQ_FHE* scheme, before the LBSP outsources its encrypted data items to the CSP, each item is evaluated homomorphically by using FHE over AES encrypted items.

From the given analysis, we can conclude that *PPQ_FHE* scheme is secure, privacy preserving, and can achieve the desirable security goals in the security model under consideration.

7. PERFORMANCE EVALUATION

7.1. Simulation Tools And Topology Setup

In this section, the performance of *PPQ_FHE* scheme is evaluated from the perspectives of LBSP, LBS user, and CSP respectively. The software and hardware configurations of the LBSP and LBS user side were performed on a 64-bit Ubuntu12.04 LTS system with an Intel Core i7 processor and 16GB RAM. The CSP side has been a virtual machine with Intel Xeon processor E5-4600, 16 GB memory on the Dell blade server M830, and VMware vSphere ESXiOS to create a private cloud. The open source Charm library [34] was applied for implementing the pairing group operations that are supported by the standard PBC library [35] and FLINT [36], for the finite filed arithmetic in Z_n*. We used Github library [37] to obtain the JAVA source code and adopted FHE over AES scheme released in HELib. A real-life dataset Open-StreetMap [38] was used, which contains 62556 real world locations.

The simulators of LBSP and LBS user were deployed in one workstation using NS2 [39], and the simulator of CSP has been deployed in one workstation using CloudSim [40] to conduct the simulations, since NS2 can be used to construct complex network topologies and simulate the query sending/responding.

Available NS2 satellite models were extended to obtain specific instrument for *PPQ_FHE* scheme simulation. Every time the request from LBS user and response from LBSP had to route through CSP only. In NS2, LBSP was considered as the routing point. At this router, an authentication activity was performed with every request and therefore causes some delay. Also, anonymity techniques cause some fraction of processing time at LBSP to make the user anonymous.

7.2. Performance Analysis And Results

In simulation, the real values measured and taken from real prototype were plugged into various parameters of the topology setup. The scalability of four different schemes (K-*anonymity*, PCQP, LPCQP, and PPQ_FHE) had been tested with increasing the number of requests generated per second from LBS user with variable transaction query. Every dataset in simulation was run and tested 10^2 times. The results of these scenarios are presented in figures 7-11. Also, table 2 shows a comparison between privacy preserving schemes in terms of Query Encryption, Database Outsourcing, Search Efficiency, and Per-Query Privacy.

The comparisons between schemes show that:

a) Regarding Query Encryption

K-anonymity does not include query encryption technique for query privacy preserving. *PCQP, LPCQP* uses FHE to preserve query privacy, and *PPQ_FHE* involves FHE-AES to preserve query privacy.

b) Regarding Database Outsourcing

K-anonymity, *PCQP, LPCQP* do not involve database outsourcing technique to cloud server.

c) Regarding Complexity of Search Efficiency

K-anonymity has $O(N^2)$, which is the highest complexity value of search efficiency because this concept uses k users to establishes and sends query which is updated for each request; therefore the response will reach to the same k users and it cannot be applied to real-time services. *PCQP* have high complexity values of search efficiency $O(log(N^2))$ but less than *K-anonymity* because this concept depends on homomorphic scheme and the query reply is sent to specific requester not for k

users. *LPCQP* and *PPQ_FHE* have the same complexity value of search efficiency *O (log(N))*; because they improved search efficiency with divided (POI-table) into sub-tables to effectively reduce the number of homomorphic additions and multiplications. So that *LPCQP* and *PPQ_FHE* have the least complexity value of search efficiency.

d) Regarding Per-Query Privacy

All schemes satisfy per-query privacy approach with different ways (anonymizer or homomorphic cryptography).

The simulation results of different scenarios show the following:

- *Response Time with Variable Number of Users*

Response time refers to the time that elapses between an inquiry end or demand on a system, and the beginning of a response. The architectures have been simulated for varying number of requests generated (10, 50, 200, and 500 queries) per second and response time is shown in figure 7 and table 3.

In figure 7, the request and response size were kept fixed. X axis represents the increasing number of query requests, and Y axis represents the response time in millisecond 'ms'. It can be clearly seen that with increasing the number of users (i.e., the number of requests generated /sec), response time is increased with a rapid rate in all schemes. It is shown that, *K-anonymity* has the largest response time because it needs to publish the response to CR and reaches the query response to each neighbor in CR. Then, the response time of *PCQP* is less than K-*anonymity* followed by *LPCQP*; because of the large amount of computations. *PPQ_FHE* scheme has the least time for responding; because it avoids noise overhead by using AES over FHE. Therefore, it has a highest performance in query response time.

- *Query Accuracy*

The corresponding accuracy rates of query are very important in identifying the performance of schemes. It is expressed by: *the query accuracy = the number of requested queries / no.of received queries*. The best performance is when accuracy ≅ 1. We assumed that, the worst case happens in the simulation which is formed of 500 requests, when all users send query requests at the same time that leads to formation of a bottleneck. Accuracy rate is shown in table 4 and figure 8.

As shown in figure 8, although the accuracy performance of the proposed approach has a little variation when the bottleneck happens, the performance drop of proposed *PPQ_FHE* scheme is less than 10% and those variations are acceptable. We assume that the worst case happens in LBS, when there are 500 queries requested at the same time; which leads to bottleneck formation. As shown, the *PPQ_FHE* has the highest accuracy followed by *LPCQP* then *PCQP*.

- *Time Cost Comparison*

The comparison of the total time cost for the four operations; (KeyGeneration, Encryption, Evaluation and Decryption) is shown in table 5 and figure 9. This measure had not been performed on *K-anonymity* scheme because it doesn't support cryptosystem.

Figure 9 shows that the proposed *PPQ_FHE* scheme is much more efficient than the others because the time cost in Key Generation, Encryption, and Decryption operations in *PPQ_FHE* scheme is the least among the other schemes. However, verifying operation takes more time in *PPQ_FHE* scheme.

- *Location Overhead Comparison*

The overhead of location-based services has been studied also. The overhead was measured as the number of packets transmitted and sent during location updates, queries and replies. All compared schemes have a large update mechanism because the update in them is global and crucial; however *PPQ_FHE* scheme uses a real-life dataset Open-StreetMap. So, LBSP imports the updated data directly from dataset. As a result of the overhead comparison, *PPQ_FHE* scheme has the lowest location overhead in general as shown in figure 10.

- *Complexity Comparison*

In the initialization process, user has to download the lookup-table and the Moore curve's setting parameters at *PCQP* and *LPCQP*. PrivacyGrid [41] was used for *K-anonymity*. The size of this information is about 200 KB under the construction of Open-Street Map real world dataset, and it can be considered as a negligible setup cost for every registered user.

A comparison between *PPQ_FHE* scheme and existing schemes in terms of LBS user, LBSP, and CSP computation complexity and computation complexity is introduced in table 6. Computation complexity refers to the amount of resources required for running algorithm. In addition,

communication complexity studies the amount of communication required to solve a problem when the input to the problem is distributed among two or more parties.

In *k-anonymity* based on cloaking region with size of inputs *n,* and types of points *m.* It is assumed that the LBS user wishes to retrieve a type of *k* nearest POIs at his or her location, while the computation complexities of the LBS user and the LBSP are $O(m+n)$ and $O(mn^2)$, respectively. The total communication complexity is $(2n+m)\ logN$ bits.

In *PCQP* scheme, the LBS user needs to compute *n* Paillier encryptions (about *n* exp.) and 1 Paillier decryption (about 2 exp.). The total computation complexity of the LBS user is $O(n)$ exp. The LBSP needs to compute n^2 exp., so the total computation complexity of the LBSP is $O(n^2)$ exp.; in addition, the communication complexity is $n^2 log2N$ bits.

In *LPCQP* scheme, the LBS user needs to generate a group *G,* a generator *g*, and a prime *q* for each query and compute a discrete logarithm. This process takes more time than computing *n* exp. The total computation complexity of LBS user is $O(n)+$ *generate G,g,q and solve discrete log*. The total computation complexity of the LBSP is $O(2n^2+1)$. In addition, the communication complexity is $2n\ log2N$ bits.

In *PPQ_FHE*, the total computation complexity of the LBS user is $O(1)$ because AES is normally working on a fixed block size, and takes approximately the same time independently of input. $O(1)$ tells that it doesn't matter how much the input grows, the algorithm has fixed query size, thus the algorithm will always be just as fast. The total computation complexity of LBSP is $O(\lambda \log n)$ where λ is a security parameter. In addition, the communication complexity is $2n\ log2N$ bits.

From table 6, it can be noticed that the performance of *PPQ_FHE* scheme is better than the others in terms of user and LBSP computations and communication complexities. In addition, there are no CSP computation complexities for K-*anonymity*, *PCQP* and *LPCQP* because there is no outsourcing data to cloud. As noticed, *LPCQP* needs to generate a group *G*, a generator *g*, and a prime *q* for each query and compute a discrete logarithm. This process takes more time than computing *n* exp.

- *Privacy Preserving Metric*

It can be observed from table 6 that the performance of *PPQ_FHE* scheme is better than other schemes in terms of user and LBSP computations, and communication complexities. Also, there are no CSP computation complexities for *K-anonymity*, *PCQP* and *LPCQP* because there is no data outsourcing to cloud. As noticed, *LPCQP* needs to generate a group *G*, a generator *g*, and a prime *q* for each query and compute a discrete logarithm. This process takes more time than computing *n* exp.

Privacy preserving metrics means how to measure the level of privacy preservation. In the proposed *PPQ_FHE* scheme, tracking success ratio is the used metric. Tracking success ratio represents the possibility that the attacker can succeed in tracking the user query information with variant number of users' query requests. Tracking success ratio is referred as *P*, and can be formed as follows:

$$P = \frac{Num\_Q}{Total\_Q} \tag{14}$$

Where *Num_Q* is the number of query requests that are guessed correctly, and *Total_Q* is the total number of query requests. *P* can be intuitive to indicate the threatening degree from the network attackers. From Eq. (14) it can be concluded that, the value of tracking success ratio is a real number between 0 and 1. *P* value increases when it gets close to 1; this means that the attacker chance for successful tracking is high (the worst case).

Figure 11 and table 7 illustrate the comparison of tracking success ratio of the proposed *PPQ_FHE* scheme with the existing schemes. The proposed *PPQ_FHE* scheme has tiny value of *P* at high number of query requests (500 requests), as shown in figure 11. It means that, this scheme has the highest and most effective defense against tracking attack. So, the attacker does not have the ability to track the query request; because *PPQ_FHE* scheme encrypts the query before being uploaded to LBSP. This prevents the query from being tracked. It is clear from the results that, the proposed *PPQ_FHE* has an obvious superiority of preventing location tracking followed by *LPCQP* then *PCQP*. The scheme that has the highest *P* for location tracking is *K-anonymity* scheme. Table 7 clarifies the results in more details.

8. CONCLUSION AND FUTURE WORK

In this paper, PPQ_FHE scheme is proposed based on FHE technique over AES symmetric cryptography to prevent noise with data then outsourcing LBS data to the cloud in a privacy-preserving fashion. The LBSP is allowed to perform the search while protecting the privacy of users' queries and identities. The CSP is also allowed to perform computations over encrypted data to detect the shortest way to destination. So, the service data are kept confidential from CSP and LBSP. Specifically, for an LBS query request from a registered user, the LBS query execution is directly performed over ciphertext on the cloud server without decryption, and the result of LBS query can only be decrypted by the registered user. Thus, the user can get accurate LBS query result without divulging his/her location information. Detailed security analysis shows its security strength and privacy preserving ability. Extensive experiments are conducted to demonstrate its efficiency in terms of query's response time, query accuracy, time cost, query overhead, and complexity. In the future, LBS will be taken into consideration in Vehicular Ad Hoc Network (VANET) system, which plays an important role in Vehicle to Vehicle (V2V) communication systems. VANET system with the help of LBS offers a novel solution to reduce overhead and congestion between vehicles on the roads. Furthermore, it is used to send warning messages to vehicles in case of accidents to avoid predictable collisions. Instead of traditional ways of traffic light system, VANET with LBS supports Intelligent Traffic Light Systems (ITLS) to save power consumption by using RSU. RSU is responsible for collecting the orders of vehicles that will arrive at the intersection and controls the traffic lights at the intersections as requested.

REFERENCES:

[1] R. Rizk, and H. Nashaat, "Smart prediction for seamless mobility in F-HMIPv6 based on location based services," *China Communications*, vol. 15, no. 4, pp. 192-209, April 2018.

[2] H. Zhu, T. Liu, G. Wei, and H. Li, "PPAS: Privacy protection authentication scheme for vanet," *Cluster Comput.*, vol. 16, no. 4, pp. 873–886, Dec. 2013.

[3] Rabab F. Abdel-Kader, Samar El-sherif, and Rawya Y. Rizk, "Two-stage cryptography scheme for secure distributed data storage in cloud computing" International Journal of Electrical and Computer Engineering (IJECE), vol. 10, no. 3, pp. 3295-3306, June 2020.

[4] F. Farouk, Y. Alkady and R. Rizk, " Efficient Privacy-Preserving Scheme for Location Based Services in VANET System," *in IEEE Access*, vol. 8, pp. 60101-60116, 2020.

[5] Y. Alkady, F. Farouk and R. Rizk, "Fully Homomorphic Encryption with AES in Cloud Computing Security," *in Proceedings of the International Conference on Advanced Intelligent Systems and Informatics*,Springer, Berlin, Heidelberg, pp.270–283, 2018.

[6] M. Gruteser and D. Grunwald, "Anonymous usage of locationbased services through spatial and temporal cloaking," *in Proceedings of the 1st International Conference on Mobile Systems,Applications and Services*, ACM, San Francisco, Calif, USA, pp. 31–42, 2003.

[7] C. Y. Chow, M. F. Mokbel and X. Liu, "A Peer-to-Peer Spatial Cloaking Algorithm for Anonymous Location- Based Services," *in Proceedings of the* 14*th Annual ACM International Symposium on Advances in Geographic In-formation Systems*, Arlington, pp. 171–178, 10-11 Nov. 2006.

[8] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing location-based identity inference in anonymous spatial queries," *IEEE Trans. on Knowledge and Data Engineering*,vol. 19,no.12, pp.1719–1733, 2007.

[9] C. Chow, M. Mokbel, and W. Aref, "Casper:query processing for location services without compromising privacy,"*ACM Transactions on Database Systems (TODS)*,vol.34,no.4, 2009.

[10] H. Kido, Y. Yanagisawa and T. Satoh, "An Anonymous Communication Technique Using Dummies for Location- Based Services," *in Proceedings of the International Conference on Pervasive Services of the IEEE ICPS* 05, San-torini, pp. 88–97, July 2005.

[11] J. Um, H. Kim, and J. Chang, "An advanced cloaking algorithm using Hilbert curves for anonymous location based service," *in Proceedings of the 2nd International Conference on Social Computing*, Minneapolis, MN, USA, pp. 1093–1098, 2010.

[12] I. Lien, Y. Lin, J. Shieh, and J. Wu, "A novel privacypreserving location-based service protocol with secret circular shift for k-NN search," *IEEE Trans. on Information Forensics and Security*,vol.8, no.6, pp.863–873, 2013.

[13] Y. Utsunomiya, K. Toyoda, and I. Sasase, "LPCQP: Lightweightprivate circular query

protocol with divided POI-table and somewhat homomorphic encryption for privacy-preserving kNN search," Journal of Information Processing, vol.24, no.1, pp. 109–122, 2016.

[14] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled)fully homomorphic encryption without bootstrapping," *in Proceedings ofthe 3rd Conference on Innovations in Theoretical Computer Science,ITCS* 2012, USA, pp. 309–325, Jan. 2012.

[15] X. Li, T. Liu, M. S. Obaidat, "A Lightweight Privacy-Preserving Authentication Protocol for VANETs," *IEEE SYSTEMS JOURNAL*, vol. 14, no. 4, pp. 3547 - 3557, Sep. 2020.

[16] T. Zhou, J. Shen, D. He, "Human-in-the-Loop-Aided Privacy-Preserving Scheme for Smart Healthcare," *IEEE Trans. on Emerging Topics In Computational Intelligence*, pp. 1 - 10, June. 2020.

[17] F. Wei, P. Vijayakumar, N. Kumar, "Privacy-Preserving Implicit Authentication Protocol Using Cosine Similarity for Internet of Things," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5599-5606, April. 2021.

[18] Z. Xu, M. Luo, N. Kumar, "Privacy-Protection Scheme Based on Sanitizable Signature for Smart Mobile Medical Scenarios*," Wireless Communications and Mobile Computing*, vol. 2020, pp. 1-10, 2020.

[19] S. Zeadally, F. Wei, N. Kumar, "An Intelligent Terminal Based Privacy-Preserving Multi-Modal Implicit Authentication Protocol for Internet of Connected Vehicles*," IEEE Transactions On Intelligent Transportation Systems*, pp. 1-13, 2020.

[20] S. Sahmoud, W. Elmasry, S. Abudalfa,"Enhancement the Security of AES Against Modern Attacks by Using Variable Key Block Cipher,"*International Arab Journal of e-Technology*, vol. 3, pp. 17-26, Jan. 2013.

[21] R. Rizk, and Y. Alkady, "Two-phase hybrid cryptography algorithm for wireless sensor networks," *Journal of Electrical Systems and information Technology, ELSEVIER,* vol. 2, no. 3, pp. 296–313, Dec. 2015.

[22] Yasmin Alkady, M. I. Habib, and Rawya Rizk, "A new security protocol using hybrid cryptography algorithms," in Proc. of *9th International Computer Engineering Conference (ICENCO)*, pp. 109-115, Cairo, Egypt, 2013.

[23]B. Pinkas, T. Schneider, N. P. Smart, and S. C. Williams," Secure two-party computation is practical,"*in Proceedings ofASIACRYPT 2009*,*Lecture Notes in Computer Science (LNCS)*, *Springer, Berlin, Heidelberg*, vol. 5912, pp. 250–267, 2009.

[24]I. Damgard and M. Keller," Secure multiparty aes,"*in Proceedings of Financial Cryptography 2010*, *Lecture Notes in Computer Science (LNCS)*, *Springer, Berlin, Heidelberg*, vol. 6052, pp. 367–374, 2010.

[25] C. Orlandi J. Nielsen, P. Nordholt and S. Sheshank. A new approach to practical active-secure two-party computation. Manuscript, 2011.

[26] R. Kusters, T. Truderung and A. Vogt, "Clash Attacks on the Veri_ability of E-Voting Systems", *IEEE Symposium on Security and Privacy*, vol. 6345, pp. 395–409, 2012.

[27]Z. Brakerski and V. Vaikuntanathan, "Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages, "*in Proceedings of CRYPTO'11, Lecture Notes in Computer Science (LNCS)*, *Springer-Verlag*, vol 6841, pp. 505–524, 2011.

[28]N. P. Smart and F. Vercauteren, "Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes, In Public Key Cryptography,"*in Proceedings of the 13th International Conference on Practice and Theory in Public Key Cryptography (PKC'10), Lecture Notes in Computer Science (LNCS)*, *Springer –Verlag*, vol. 6056, pp. 420–443, 2010.

[29] Y. Alkady, F. Farouk, and R. Rizk, "Fully homomorphic encryption with AES in cloud computing security," *in Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2018*, Advances in Intelligent Systems and Computing, Springer, vol. 845, pp. 370-382, Cairo, Egypt, September 2018.

[30] Z. Yong-Xia and Z. Ge, "MD5 Research," *in Proceedings ofthe 2010 Second International Conference on Multimedia and Information Technology*, Kaifeng, pp. 271–273, 2010.

[31] A. Menezes, "An Introduction to Pairing-Based Cryptography," in *1991 Mathematics Subject Classification*, Primary 94A60, 1991.

[32] I.Skog and P. Handel, "In-Car Positioning and Navigation Technologies—A Survey,"*IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 4-21, March 2009.

[33] Nitin R. Chopde, Mangesh K. Nichat, "Landmark Based Shortest Path Detection by Using A* and Haversine Formula," *International Journal of Innovative Research in Computer and Communication*

Engineering (IJIRCCE), vol. 1, no. 2, pp. 2320–9801, April 2013.

[34] J. A. Akinyele, C. Garman, I. Miers, M. W. Pagano, M. Rushanan, M. Green, and A. D. Rubin, "Charm: a framework for rapidly prototyping cryptosystems," *Journal of Cryptographic Engineering*, vol. 3, no. 2, pp. 111–128, 2013.

[35] B. Lynn et al., "The pairing-based cryptography library. internet:crypto," 2006.

[36] W. Hart, F. Johansson, and S. Pancratz, "FLINT: Fast Libraryfor Number Theory," 2013, version 2.4.0. [Online]. Available: http://flintlib.org

[37] M. Varia, S. Yakoubov, and Y. Yang, "HEtest: A Homomorphic Encryption Testing Framework," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8976, pp. 213–230, 2015.

[38] OpenStreetMap contributors. (2017) Planet dump retrieved from https://planet.osm.org. [Online]. Available: https://www.openstreetmap.org

[39] "The Network Simulator – ns 2, *http://www.isi.edu/nsnam/n*

[40] R. N. Calheiros, R. Ranjan, A. Beloglazov, and C. De Rose, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software-Practice and Experience*, vol. 41, no. 1, pp. 23–50, Jan. 2011.

[41] B. Bamba, L. Liu, P. Pesti, and T. Wang, "Supporting anonymous location queries in mobile environments with PrivacyGrid,"*in Proceedings of 17th Int'l Conference on World Wide Web, ACM,* pp. 237–246, April 2008.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

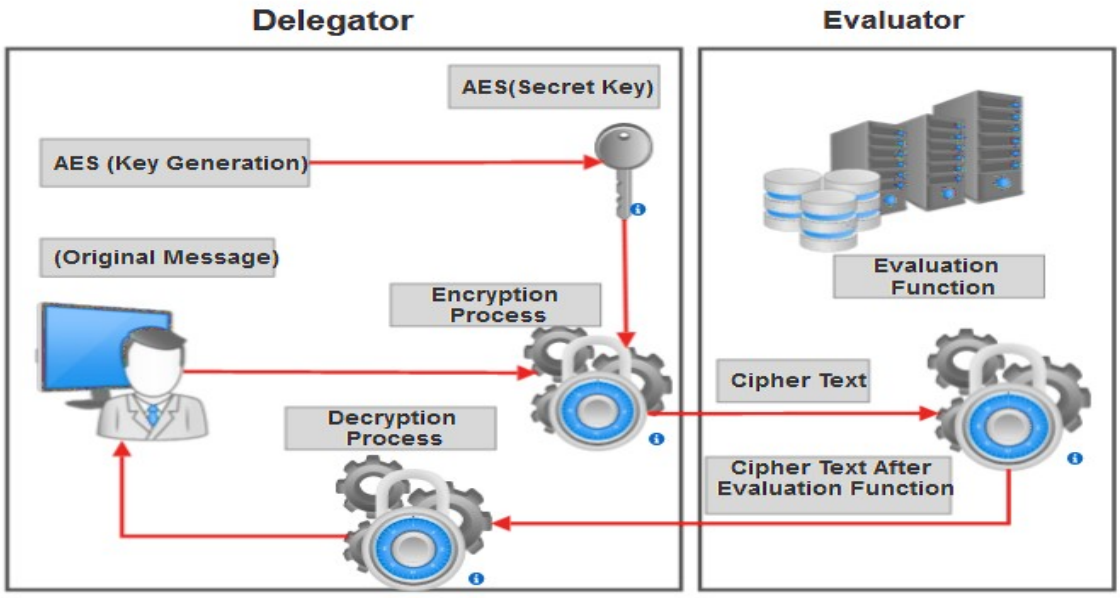

Figure 3: FHE Over AES Symmetric Key

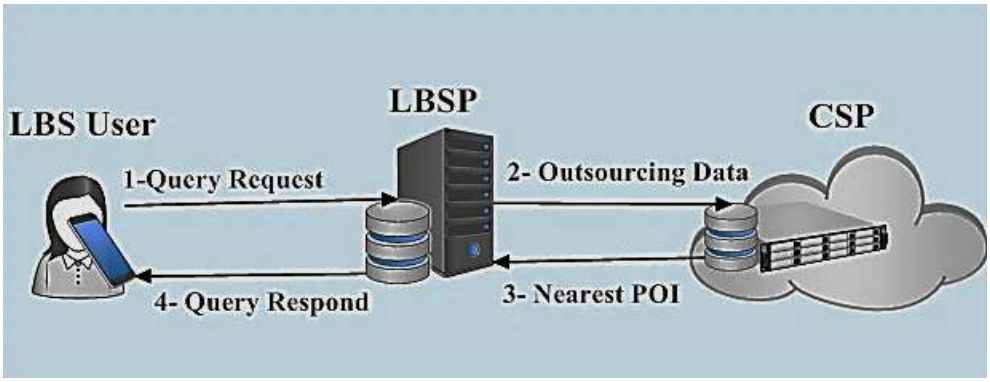

Figure 4: System Model

Journal of Theoretical and Applied Information Technology
31st August 2021. Vol.99. No 16

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

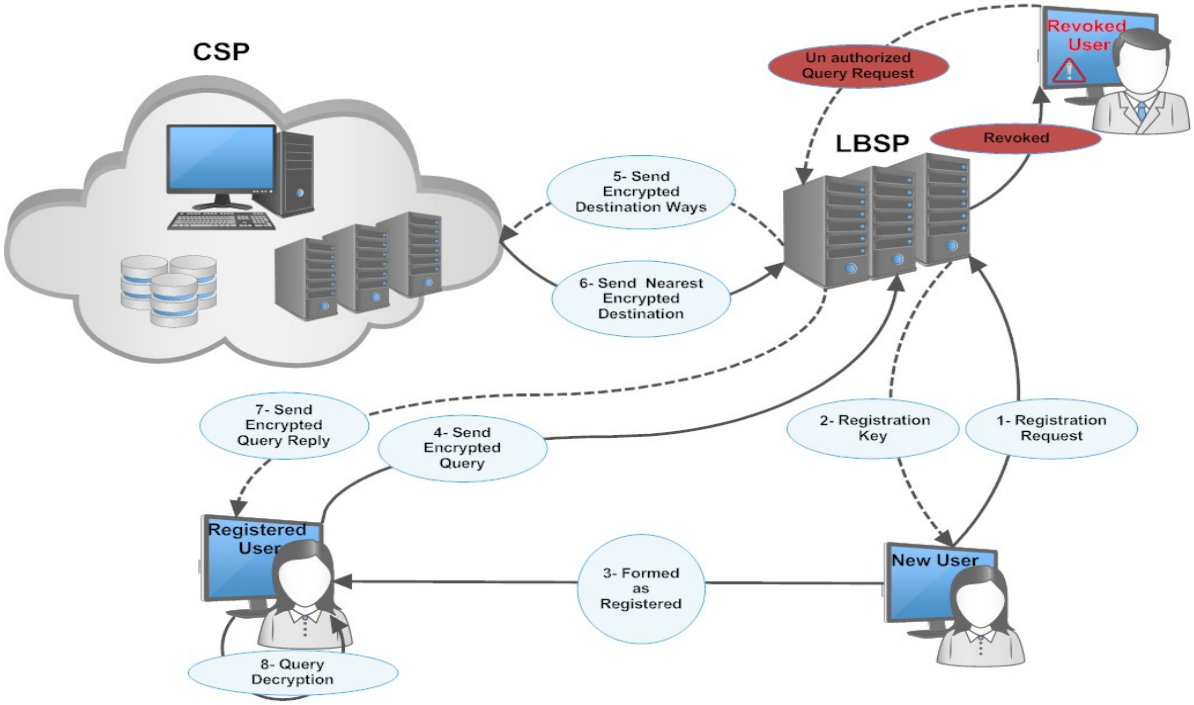

Figure 5: LBS Users' Relation With LBSP

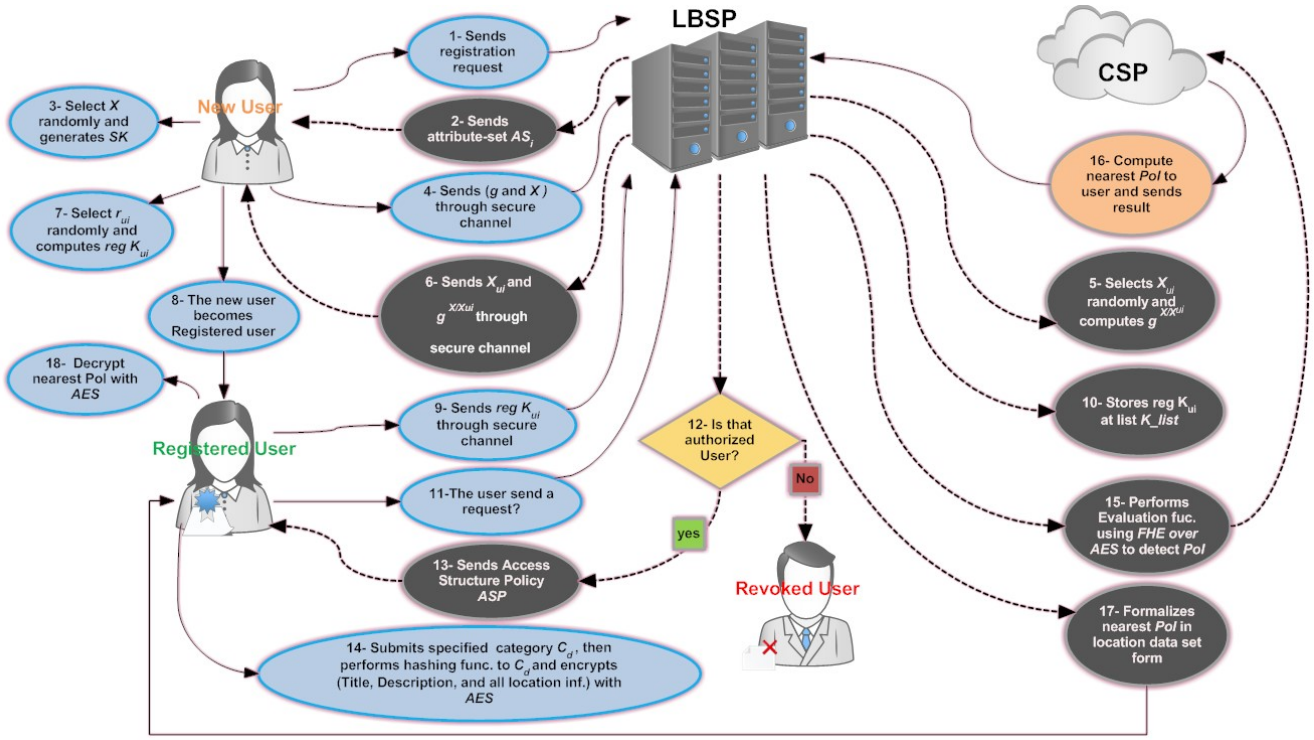

Figure 6: Architecture of PPQ_FHE

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

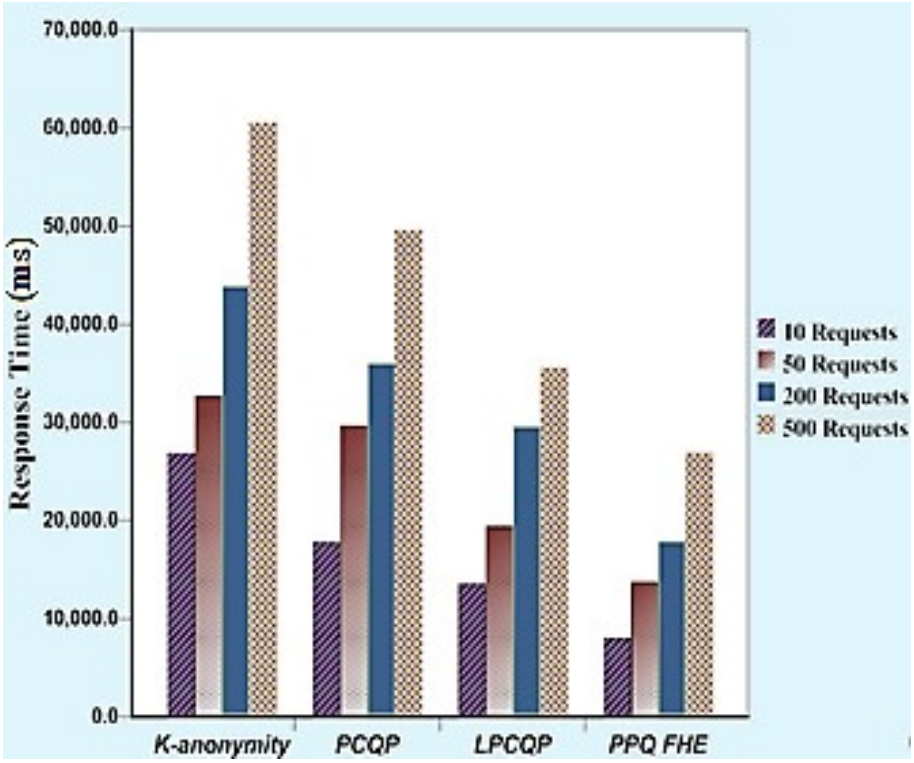

Figure 7: Response Time

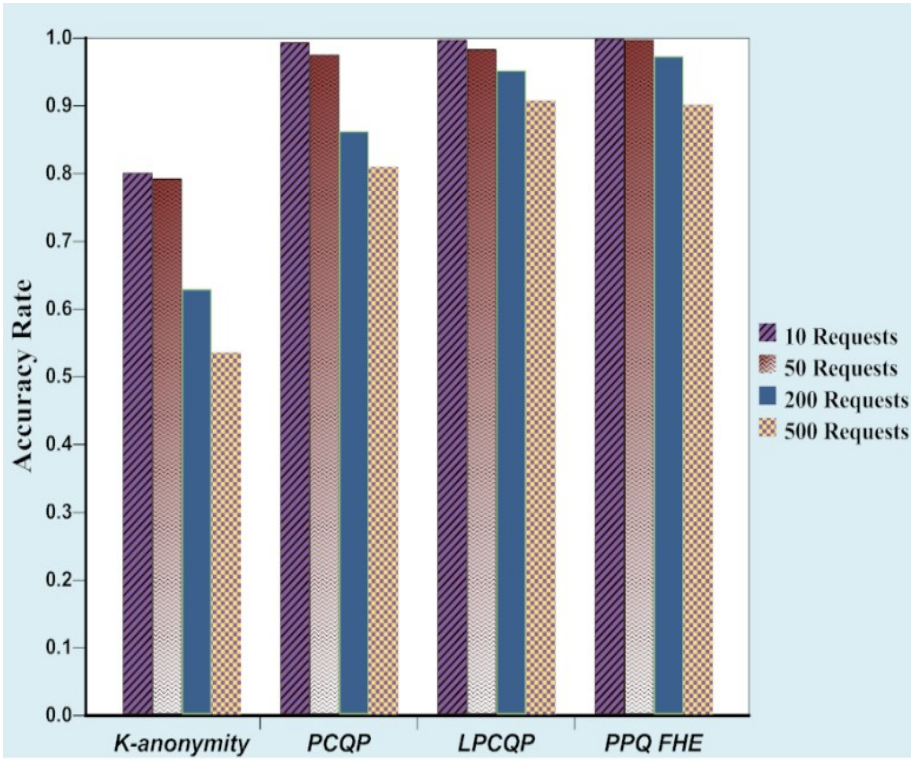

Figure 8: Accuracy Rate

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

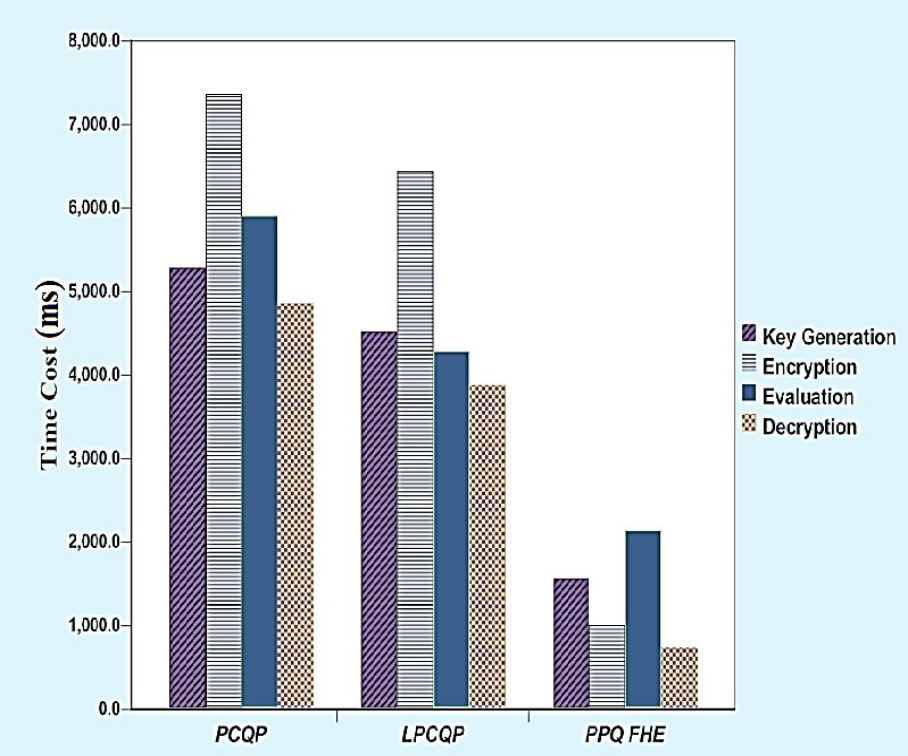

Figure 9: Time Cost

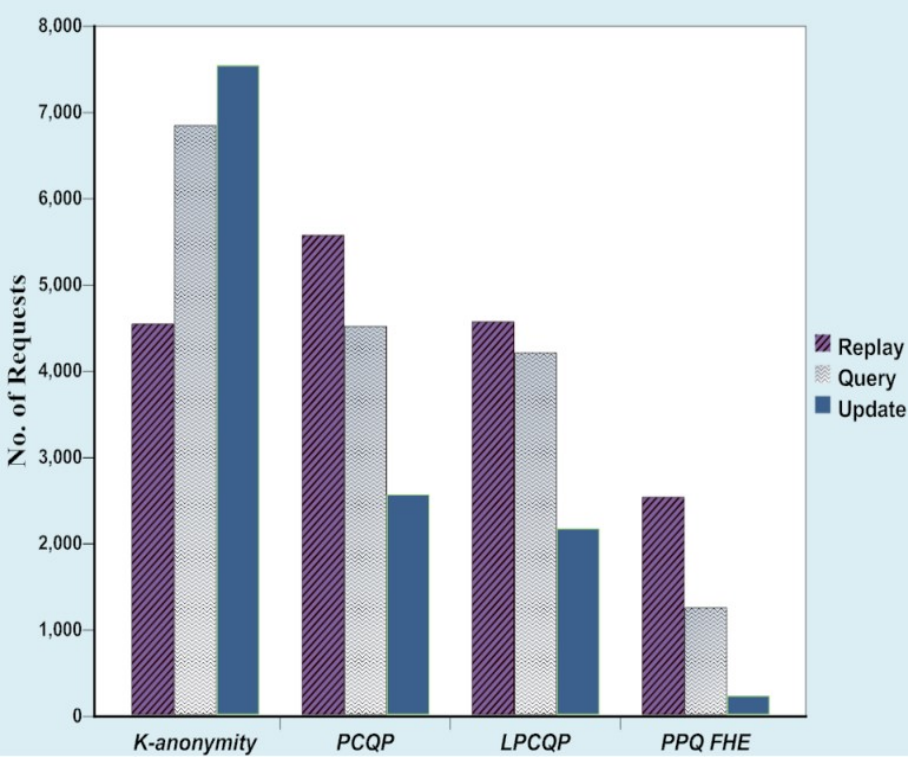

Figure 10: Overhead

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

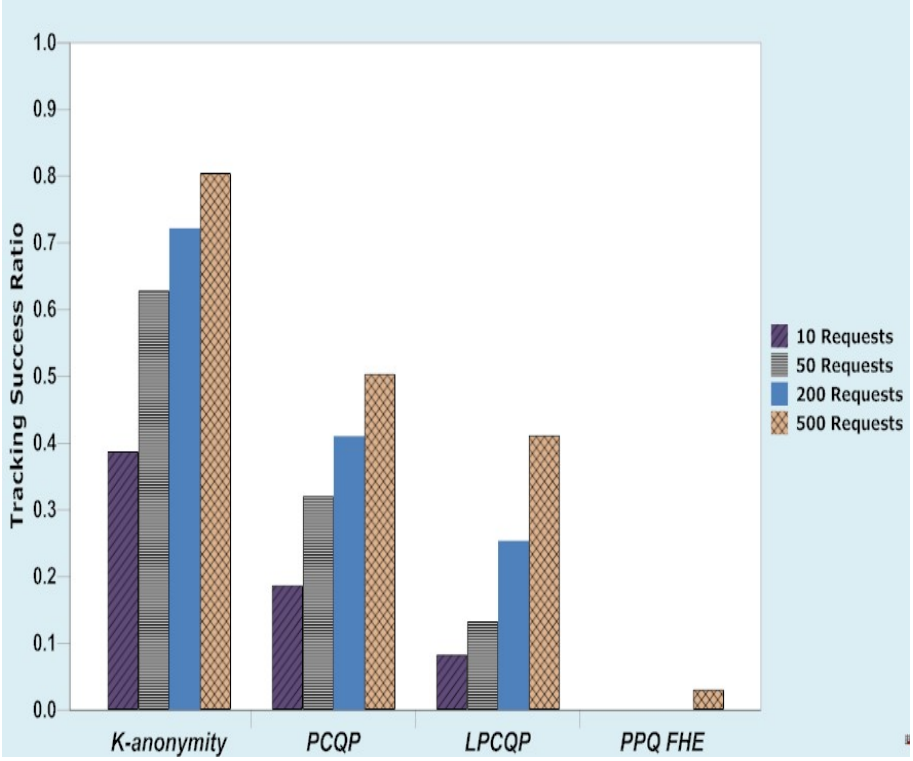

Figure 11: Tracking Success Ratio

Table 1: List of Notations

Symbol	Definition
λ	The security parameter
AS_i	The Attribute- Set
ASP	The Access Structure Policy
C_d	The destination category
D_d	The description of destination
DEC_{AES}	The AES decryption function
ENC_{AES}	The AES encryption function
Eval	The evaluation function of fully homomorphic
H ( )	The hashing function using *MD5*
ID	The Identification of user in query
keyGen( )	The key Generation function
KList	The list of registration keys
lat	The value of Latitude in Radians
long	The value of Longitude in Radians
M_4	The matrix of size 4
MD5	The Message Digest Algorithm 5
Pi	The value of pi is 22/7
POI	Point Of Interest
$regK_i$	The Registration-Key
SK	The secret key of AES
T_d	The destination title
u_i	The LBS user, i refer to the user number
X_d , Y_d	The destination coordination
X_i, Y_i	The current user coordination
Z_n*	Denote the set of all integers

Table 2: Comparison of Privacy Preserving Location Based Works

Approach	Comparison Metrics			
	Query homomorphic encryption	**DB outsourcing**	**Complexity of Search efficiency**	**Per-query privacy**
K-anonymity	NO	NO	*O* (N^2)	YES
PCQP	YES	NO	*Olog* (N^2)	YES
LPCQP	YES	NO	*Olog* (*N*)	YES
PPQ-FHE	YES	YES	*Olog*(*N*)	YES

Table 3: Comparison of Response Time (ms)

Approach	Number of Query Requests			
	10	50	200	500
K-anonymity	26845.32	32641.52	43875.58	60532.4
PCQP	17781.35	29652.16	35987.35	49547.8
LPCQP	13520.87	19364.84	29574.94	35536.9
PPQ_FHE	8014.75	13745.14	17845.17	26879.3

Table 4: Accuracy Rate

Approach	Number of Query Requests			
	10	50	200	500
K-anonymity	0.801	0.792	0.629	0.535
PCQP	0.993	0.975	0.862	0.809
LPCQP	0.997	0.983	0.952	0.907
PPQ_FHE	0.999	0.997	0.973	0.901

Table 5: Comparison of Time Cost (ms)

Approach	Operation Type			
	Key Generation	Encryption	Evaluation	Decryption
PCQP	5285.63	7361.26	5904.51	4852.6
LPCQP	4521.94	6441.08	4286.01	3879.7
PPQ_FHE	1563.22	1003.54	2138.82	732.1

Table 6: Comparison of Computation Complexity and Communication Complexity

Approach	Computation Complexity			Communication complexity
	User comp.	LBSP comp.	CSP comp.	Comm.
K-anonymity	*O(m+n)*	*O(mn^2)*	Not Independent in CSP	*(2n+m) logN*
PCQP	*O(n)*	*O(n^2)*		*n^2log2N*
LPCQP	*O(1)+generate G,g,q and solve discrete log*	*O($2n^2$+1)*		*2n log2N*
PPQ_FHE	*O(1)*	*O(λ log n)*	*O(log n)*	*2nlogN*

Table 7: Tracing Success Ratio

Approach	Number of Query Requests			
	10	50	200	500
K-anonymity	0.387	0.628	0.721	0.804
PCQP	0.186	0.320	0.410	0.503
LPCQP	0.082	0.132	0.253	0.411
PPQ_FHE	≅ 0	≅ 0	≅ 0	0.032