doi: 10.20532/cit.2019.1004606

Privacy Preserving Location-Based Client-Server Service Using Standard Cryptosystem

Adebukola Onashoga, Adesina Sodiya and Idowu Osinuga

Federal University of Agriculture, Abeokuta, Nigeria

Location-Based Mobile Services (LBMS) is rapidly gaining ground and becoming increasingly popular, because of the variety of efficient and personalized services it offers. However, if users are not guaranteed their privacy and there is no assurance of genuineness of server's response, the use of these services would be rendered useless and could deter its growth in mobile computing. This paper aims to provide confidentiality and integrity for communication that occurs between users and location service providers. A practical system that guarantees a user's privacy and integrity of server's response, using a cryptographic scheme with no trusted intermediary, is provided. This scheme also employs the use of symmetric and asymmetric encryption algorithms to ensure secure message and key transfer. In order to overcome the problem of computational complexities with these algorithms, AES-256 is used to encrypt the message and user's location. Several researches have been done in this category but there is still no system that checks the integrity of server's response. The proposed scheme is resistant to a range of susceptible attacks, because it provides a detailed security analysis and, when compared with related work, shows that it can actually guarantee privacy and integrity with faster average response time and higher throughput in LBMS.

ACM CCS (2012) Classification: Security and privacy → Security services → Privacy-preserving protocols

Keywords: cryptography, message authentication, integrity

1. Introduction

Location-Based Mobile Service (LBMS) has become one of the most widely used mobile applications because of the high demand in the use of mobile phones (Yoon *et al.* [1], Kasamani and Gikundi [2]). The future gets closer to us with location-based services (LBS) taking into account location information of the user, available anywhere and anytime. Location is a basic factor that determines the means by which people interact and get things done in their environment.

Location based services, as defined by Schiller and Voisard [3], are "services that integrate a mobile device's location or position with other information so as to provide added value to a user". In a broad perspective, Xu and Gupta [4], described LBMS as network-based services with the aim of providing added value to the user by using mobile user's location with other information. The delivery mechanisms used for LBMS include mobile internet, mobile applications, Short Message Service (SMS) text messaging, Multimedia Messaging Service (MMS), services using GPS, indoor location services, digital out-of-home, digital signage, print media and television (Khan and Light [5]). Location privacy is of outmost importance since location service providers use clients' location information to offer them convenient and useful services. However, if users of such services are not assured that their privacy is guaranteed and will not be breached, they may opt-out of such service or even oppose its implementation (Popa *et al.* [6], Eckhoff and Wagner [7]). A lot of research has been conducted concerning how to enjoy location-based services while protecting the location privacy of the mobile users (Memon [8], Kim [9], Sun *et al.* [10], Shen *et al.* [11], Gardner *et al.* [12]). For example, for a given LBS user who wants to find the nearest bank to him, sending his present location may put his privacy at risk. To secure his privacy, he hides his present location and his identity as the person sending query to the service provider. The simplest way to achieve this is to remove his identity in exchange for a pseudonym which is sent to the service provider, but it is not enough to preserve the user's privacy, since his/her identity can be unveiled through a quasi-identifier (Samarati and Sweeney [13]).

In most instances, many mobile device users will refuse to use devices that are GPS enabled or will switch it off, even if it is installed on their mobile devices, for fear of a breach of location privacy. Another instance is when drivers switch off the transponders in their vehicles for fear of attack on online databases that are usually tampered with or misused by passive attackers who have detailed movement of users and can attempt criminal attacks such as burglarizing homes when they are sure residents are away Weber [14].

Location privacy concerns are the issues that should be tackled on daily basis because continual advancement in location services has opened the possibility of breach in users privacy, causing them to opt out of such services or even oppose to their use (Olumofin *et al.* [15], Olumofin and Goldberg [16], Popa *et al.* [17], Arain *et al.* [18]).

Introducing a combined concept of location privacy, message authentication and integrity of a server results in using cryptographic approaches to develop a Location-Based Client-Server service satisfying the aforementioned requirements, which is the main motivation for our research.

In this paper, the design of Location-Based Service for enhancing Privacy and Integrity (LBS_PI) is presented. The designed location-based application for mobile users does not require a trusted third party and guarantees location privacy protection of users while maintaining practical functionality and benefits of such services. This paper also provides self-verifiable information by the LBS server.

2. Literature Review

2.1. Overview of Cryptographic Methods

Cryptography is a very strong tool for protecting data and information transfer. Cryptography lays a foundation of many security frameworks and forms the backbone of encryption and decryption processes. Cryptography is divided into three branches; asymmetric, symmetric and hashing schemes. All these branches entail encryption and decryption processes. Encryption is a cryptographic method that converts information from a plain text (readable) into unintelligible (non-readable) form, to avoid access by unauthorized entities. Encryption assures integrity, authenticity, privacy, access control and so on (Pradhan and Sharma [19]).

Symmetric cryptographic schemes are also called single-key, secret-key, symmetric key schemes because a single key is used for both encryption and decryption processes. Symmetric schemes are generally faster, compared to asymmetric keys, and are used to establish session keys since it involves only one key (Paar and Pelzl, [20]). The only challenge posed by symmetric schemes is the difficulty involved in having a secure key management that involves a large number of users. Examples of some symmetric schemes are DES, AES, MARS, Serpent, Twofish, Blowfish and RC6 *etc.*

On the other hand, asymmetric schemes make use of two separate keys, inverse of each other; they are called: the public key (used for encryption and known to everyone) and the private key (used for decryption and known only to the receiver). Asymmetric scheme has been designed to solve the key management problem encountered with symmetric schemes (Paar and Pelzl, [20]). It entails a lot of mathematical complexities and, as such, is slower and not ideal for a large volume of data. Types of asymmetric schemes of practical importance are RSA, El-Gamal and Elliptic Curve cryptosystems.

2.2. RSA Cryptosystems

RSA is one of the first practical public-key cryptosystems and it is widely used to secure data transmission. RSA stands for Ron Rivest, Adi Shamir and Leonard Adleman (Rivest-Shamir-Adleman) who were the first researchers to propose the scheme in 1977 and it is based on a one-way function of integer factorization scheme.

RSA consists of three fundamental phases: key generation phase, encryption phase and decryption phase. The key generation phase is where the public/private key pair (e, d) of each user is generated, with (e, d) computed in modulus n and used to encrypt and decrypt data as required. The security of RSA relies on the difficulty of solving the integer factorization problem, *i.e.*, given n and e, it is difficult to compute d if prime integers P and Q are extremely large. RSA uses the longest key length, as compared to discrete logarithm and Elliptic curve cryptosystems. Since they require arithmetics with very long operands and keys, this implies that the longer the operands and keys, the more secure the algorithms become.

Thus, the main use of the encryption feature in a public-key algorithm like the RSA is to securely perform key exchange for a symmetric cipher. In a practical sense, RSA is often used together with a symmetric cipher like AES, where the latter does the key transport, while the symmetric cipher does the actual bulk data encryption. It is important to note that the Euler's phi function plays an important role in the RSA cryptosystem. Just like with the RSA scheme, the security of RSA Digital Signature Scheme also relies on integer factorization.

2.3. Euler's Phi Function

The Euler's phi function is a very useful tool in asymmetric cryptosystems, specifically for RSA. Given a set of integers as in (1):

$$Z_n = \{0, 1, ..., n-1\}, \tag{1}$$

Euler's phi function is defined as "the number of integers in Z_n relatively prime to n is denoted by $\phi(n)$". However, computing Euler's phi function for a large number is not a thorough method for large numbers in asymmetric cryptosystems. A better approach is factoring n, given in (2). Let k have the following canonical factorization:

$$k = P_1^{e_1} \cdot P_2^{e_2} \cdot ... \cdot P_n^{e_n}, \tag{2}$$

where the P_i are distinct prime numbers and e_i are positive integers, then,

$$\phi(k) = \prod_{i=1}^{n} (P_i^{e_i} - P_i^{e_i - 1}). \tag{3}$$

2.4. RSA Signature Scheme

The RSA signature scheme is based on RSA encryption and has become the most popular digital signature scheme amidst its contemporaries. Its security is based on integer factorization. Just as the handwritten signature is used to authenticate the sender of a message, the digital signature is used for the similar purpose, but offers more functions as an encryption method. It offers security protection for digital transactions. The private key is used to append signature on a document by the sender of the message, while the corresponding public key is used by the recipient of the message to verify the signature in order to check if the message is from the right source.

2.5. Advanced Encryption Standard (AES-256)

AES-256, a standard secret-key encryption algorithm, was employed in this paper to encrypt the actual message and location coordinates before transmission to the receiver, because it is generally designed to be highly resistant to a cryptanalysis attack. AES-256 uses the largest key size out of the three different versions available (*i.e.*, AES-128, AES-192 and AES-256) and has been chosen for implementation since it is more secure than the other two versions. However, any other symmetric encryption or hashing algorithm can be used as security of the proposed scheme, as it does not totally depend on the secret-key algorithm employed.

2.6. Related Work

Sweeney [21] introduced k-anonymity as a property by which a user is indistinguishable from other $(k-1)$ users, if attempts are made to identify a user. This forms the basis on which popular cloaking techniques are built. The major limitation is that k-anonymity is not efficient in a sparse populated area because it can reveal a user's location. Queries from multiple users are aggregated at the anonymity server which serves as an intermediary between the client and the server (often referred to as trusted third party).

Gruseter & Grunwald [22] developed a system model that protects identity and privacy of users. It consists of a trusted party server called anonymizer, placed as an intermediary between the client and the server.

The anonymizer expands the exact user's location into a cloaked region so that it contains the exact user location and other $(k-1)$ users. This way, the server cannot distinguish the exact user from other $(k-1)$ users. The anonymizer refines the candidate set and sends the actual result to the user. This procedure incurs a low communication cost between client and anonymizer, but shows several disadvantages. First, the anonymizer becomes a performance bottleneck because it needs to serve all its subscribed users and also maintain accurate records of their location. Secondly, the anonymizer is prone to collusion from malicious users since it represents the central point of attack. Thirdly, there are no integrity checks if the server has actually returned the correct result to client queries. Lastly, it is limited to privacy guarantees in areas of dense population and distribution of users.

Ghinita *et al.* [23] developed a location privacy model which runs on a client-server architecture. The client uses PIR to encode the plaintext message into an "incomprehensible" query. Then the server computes the encoded result blindly. The client derives the actual result from the encoded result sent by the server. This work is built on the computational PIR (cPIR) protocol introduced in Kushilevitz and Ostrovsky [24], where clients make use of PIR to query an LBS provider for nearby points of interest (POI) and which allows the client to retrieve a small fraction of the LBS database that is cost effective. The system ensures message confidentiality, but no identity and location privacy guarantees. Also, the scheme is not self-verifiable (no integrity guarantees).

This approach [*ibid.*] was extended in Ghinita [25], where a user location is hidden inside a cloaked region and the PIR protocol is run between the client and the LBS provider in order to disclose an optimally small number of POI for database protection and also to reduce storage requirements on the user's mobile device. Such a method provides strong location and identity privacy guarantees since the server is blinded and the weakest trust assumptions. Additionally, a message is kept confidential from the passive attacker but it is very complex applying PIR protocols to LBS privacy because it entails finding effective methods of transforming LBS queries that are content-based to PIR protocols that are index-based.

Similar to Ghinita's [25] research is the work by Olumofin *et al.* [15] who developed a hybrid scheme that can achieve efficient query privacy for location based services and is a trusted party free, algorithm that achieves a good compromise between user location privacy and computational efficiency. Both cloaking technique and PIR protocol have been combined to complement each other, thus guaranteeing strong location privacy, flexibility and scalability. The technique achieves strong location privacy in the sense that a user determines his or her privacy level, forms a cloaking region around his location and the area of interest, and uses the PIR protocol to query the database. An extension of PIR was used, *i.e.*, the Symmetric PIR (sPIR), which establishes database secrecy by assuring that no information other than what is relevant to the current location is unveiled to the querying user, thereby giving a stronger privacy protection to the server in the presence of a malicious client or attacker. The use of an anonymizing network has not been supported because protecting a user's location is superior to hiding the user identity, since an attacker can identify a user that has made a query from a certain geographical location. The major problem with this approach is a low LBS server efficiency, since the size of the cloaking region and its boundaries are controlled by the user and not by the server. Secondly, the size of the cloaking region should not be too small or too large in the sense that a small cloaking region implies that privacy can be easily breached while the larger region size makes the computation on the client's end more complex which can impair the performance. The same problem encountered in Ghinita [25] when applying the PIR protocol to location-based service content is very challenging. However, Ghinita's [25] work employed the computational PIR protocol, while Olumofin *et al.* [15] used a higher version of the PIR protocol (*i.e.* the symmetric PIR), which ensures database secrecy from abusive clients.

In a different perspective, Popa *et al.* [26] built a practical system called Vehicular Privacy (VPriv) that performs computational functions over paths travelled by drivers while preserving their identity. Its aim is to combine different cryptographic notions with engineering efforts to design new schemes and systems that are secure and practical. The use of cryptographic tools makes it difficult, if not impossible, for an attacker to decode the encoded message. The drawbacks are: modern cryptographic protocols used are computationally intensive and the system is expensive to implement. Secondly, all clients that provide data must be online at the same time which makes it impractical in a very large population setting. Thirdly, the system lacks location privacy, which makes it vulnerable to side information attacks and lastly, the system does not guarantee integrity checks against a malicious server.

To combat location privacy problems such as side information attacks encountered in VPriv, Popa *et al.* [17] have built a practical system called Privacy Statistics (PrivStats) that performs overall statistical computation of paths travelled by drivers while preserving their location and identity privacy and has the application verifying whether drivers provide valid data. The approach used is based on standard cryptographic techniques, *i.e.*, RSA technique and theoretical protocols are transformed and implemented on mobile devices and moving vehicles for the sake of data publishing. It guarantees stronger privacy in the face of side information and provides protection against abusive clients by allowing them to upload not more than the required tuples to the server without a trusted party. Secondly, it is efficient on resource constrained devices and does not require all users to be online at the same time for it to work effectively. The major drawbacks are: the addition of noise and junk records as a means to enhance privacy can impair the performance, consume storage and incur processing cost on the client's mobile device. Additionally, the anonymizing network can degrade the quality of service. Lastly, the system is not self-verifiable (there are no integrity guarantees).

Recently, Pan *et al.* [27], proposed a new incremental clique-based cloaking algorithm called ICliqueCloak, for defence against location dependent attacks. The authors combined two privacy metrics: location *k*-anonymity which protects user identity, but cannot protect location disclosure; and cloaking granularity which protects location disclosure, but cannot protect user identity. Both metrics employed serve as a complement to each other. Also, ICliqueCloak was designed to protect privacy against location-dependent attacks when users' locations are continuously updated as they move (unlinkability), as compared to existing *k*-anonymity location cloaking algorithms (Gong et al [28]) that are concerned with snapshot user location only (location privacy) and cannot combat attacks when users are in continuous movement. Since the system is designed to handle multiple users, a trusted anonymizing proxy is needed to provide spatial and temporal cloaking functions. This single point of vulnerability makes it highly susceptible to attack if the anonymizing proxy is compromised and the anonymization cost is slightly increased. In ICliqueCloak, users cannot issue new request until the previous request has been serviced. In other words, users cannot issue more than a query at a time, which can degrade performance as a tradeoff for privacy.

Li and Jung [29] proposed a fine-grained Privacy-preserving Location Query Protocol (PLQP) that enables users to obtain location information about other users without violating privacy (*e.g.*, searching the proximity of a user's location from a querying user). The protocol comprises of many mobile users, but peers in the system are untrusted. The scheme preserves privacy by means of encrypting location information and it also guarantees access control (*i.e.*, controls the rate at which users can learn about each other's location information). But the latter part is not within the scope of this work. Also, integrity checks are outside the scope of this work since it is not a client-server architecture.

As an improvement on the work of Ghinita [25] and Olumofin *et al.* [15], Paulet *et al.* [30] proposed a protocol for location based queries by introducing two stage approaches based on Oblivious Transfer Protocol and Private Information Retrieval, to ensure privacy guarantees for both server and user. The system comprises a set of users, a mobile service provider and a location server. From the user's viewpoint, the mobile service provider and the location server are seen as a single server. From the server's viewpoint, the location server owns a set of POI records where each record describes the POI giving both GPS coordinates (x, y) and location description, while the service provider establishes the communication between the location server and the user and is not made to collude and reveal information about the user to the location server. The major edge this approach has over the previous work is that it provides stronger privacy guarantees for both clients and the server, and is more effective in terms of computation and communication overhead. There are also a number of disadvantages. Firstly, the addition of dummies to user's records can distort data, thereby impairing performance. Secondly, it is very complex to convert PIR protocols that are index-based to location-based queries that are content based. Since the server is blinded and a user's location is in a cloaked region, the response to user queries may not be accurate (as there are no integrity checks).

Shokri *et al.* [31] proposed a user-collaborative privacy-preserving model (MobiCrowd) for location services that does not rely on a centralized party, but instead trust is distributed among mobile peers that form a network to achieve privacy. So, its performance depends on the network characteristics (*e.g.*, time-dependent mobility), not just on what an individual device does. MobiCrowd employs the technique of hiding users (identity privacy guarantees) from the server while still allowing them to get query results from other peers. In essence, users can only contact the LBS server, but they cannot get the information required from other mobile peers, thereby reducing the risk of location disclosure. Protocols can achieve a higher fault tolerance since trust is evenly distributed among peers. The approach does not rely on a trusted third party but privacy protection is placed with the users themselves. It also combats the Bayesian inference attack that allows attackers to have prior information before launching the attacks. The major drawbacks are: the approach entails a higher communication and computational cost for resource-limited devices such as smartphones, the system is prone to network congestion and it does not guarantee message confidentiality.

Elghazal *et al.* [32] investigated the practicalities of LBS, and suggested a mobile application that had been developed to improve the efficacy of smartphones by reducing the power consumption of Wi-Fi components using GSM cells ID information. The set-up of the proposed application has showed a good saving in smartphone power consumption with the LBS concept. Garzon *et al.* [33] debate the technical and environmental factors that influence the reliability of proactive LBS. The authors introduced a proof-of-concept estimator which evaluated the likelihood that a location-dependent action gets triggered by a proactive LBS. The outcome of comparing the estimator against an exemplary proactive LBS in the real world has showed the validity of their concept. Sun *et al.* [34] introduced location labeling to differentiate locations of mobile users with respect to sensitive and ordinary locations. The authors designed a location-label based (LLB) algorithm for protecting location privacy of users while minimizing the response time for LBS requests. A performance evaluation was conducted to assess the accuracy of the proposed algorithm through extensive simulations. Chen *et al.* [35] proposed an efficient structure to shield user privacy. The designed structure utilizes redundant POI records to safeguard privacy against an LBS provider, but uses a semi-trusted third party, called proxy, to filter out redundant POI records. To protect privacy against the proxy, they designed a new filtering protocol, Blind filter, to allow the proxy to filter out redundant encrypted POI records in a blind way. Based on juxtaposition with similar solutions, their structure was found to be robust to dual identity attack, along with reduced computation overhead.

Memon *et al.* [36] proposed a scheme that can prevent exposure of users' private information and enable secure communication using asymmetric cryptography. The authors claimed that the designed scheme was made robust against eavesdropping attacks by providing mutual authentication using the asymmetric cryptography scheme. The authors' work requires a trusted third party and does not guarantee location privacy protection of the users. Similarly, Kumar and Sunitha [37] designed a public key cryptosystem for privacy sensitive location-based services. The authors expanded previous research to enable symmetric key exchange between connecting parties which can be used to securely share the location coordinates to compute the authentic remoteness of communicating parties. The authors claimed that their work preserved users location privacy, but they did not address a trusted third party scenario. Jannati and Bahrak [38] designed an oblivious transfer protocol based on Elgamal encryption for preserving location privacy. The authors suggested a better protocol for the protection of both client's location privacy and the server's database security, with negligible degradation in the system performance, but failed to address a trusted third party scenario. Also, Solanas and Martínez-Ballesté [39] proposed privacy protection in location-based services through a public-key privacy homomorphism. The authors proposed a novel cryptosystem scheme for privacy of users of LBS assurance using a public-key infrastructure. The authors' scheme, unlike existing approaches, does not need any trusted third party to hide users' location, but no attention was devoted to integrity and authenticity of server responses. Therefore, to solve the aforementioned gaps in literature, a combined concept of location privacy, message authentication and integrity of server results using cryptographic approaches is introduced in the present paper.

3. Methodology

The scheme employed in this paper involves the use of symmetric encryption algorithm (AES-256), RSA cryptosystem, and RSA Digital Signature Scheme as basic tools.

With respect to previously mentioned basic tools, a scheme that preserves privacy while enhancing the integrity of the result is thus proposed to solve the fundamental problems of information confidentiality, integrity and authenticity of server responses. This scheme is thus named Location-Based Service for enhancing Privacy and Integrity (LBS_PI). The architecture of the proposed scheme is depicted in Figure 1.

3.1. Phases Involved in LBS_PI Architecture

The four phases involved in LBS-PI are:

- Client Registration Phase;
- Client Request Generation Phase;
- Server Response Generation Phase;
- Client Response Retrieval Phase.

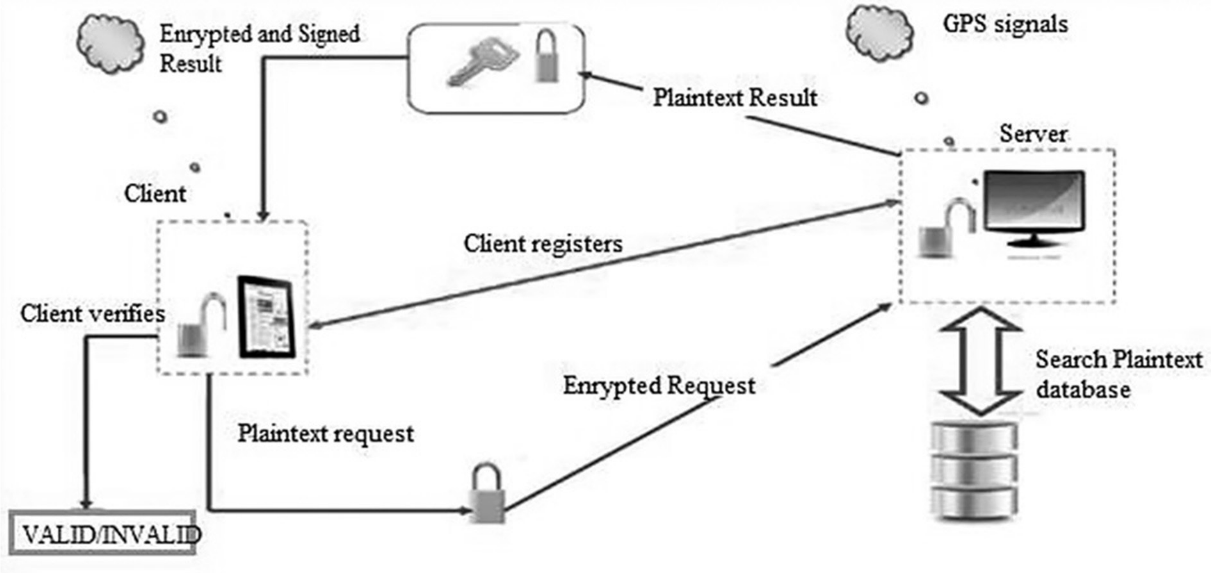


Figure 1. LBS_PI Architecture.

3.1.1. Client Registration Phase

Before the communication commences, the client registers with the server if he/she is a new user u_i, by supplying the username u_{id} on a secure website (https). This phase involves a key generation module. The server generates a public/private key pair for each client.

Key Generation Module (KGen). For a new user u_i, the server S computes the public (e_{ui}, n_{ui}) and the private key pair (d_{ui}, n_{ui}). The server keeps the public key (e_{ui}, n_{ui}) in its database so that when another user u_{i+1} signs up, there will be no duplicate public keys which could lead to redundancy. The private key (d_{ui}, n_{ui}) of each new user u_i generated by the server is discarded following the registration because the public key pair (e_{ui}, n_{ui}) is known to everyone, but (d_{ui}, n_{ui}) is known only to the owner of the key.

By default, the server generates its own public/private key pair (e_s, n_s) and (d_s, n_s) respectively. The server keeps its private key (d_s, n_s) secret, while the public key (e_s, n_s) is made known to everyone.

Client ← Server:

Step 1: Server chooses two large prime numbers p_{ui}, q_{ui}.

Step 2: Server computes the product of p_{ui}, q_{ui} as: $n_{ui} = p_{ui} \cdot q_{ui}$

Step 3: Server computes $\phi(n_{ui}) = (P_{ui} - 1)(q_{ui} - 1)$

Step 4: Server chooses public key e such that $gcd(e, (n_{ui})) = 1$ and $3 \leq e \leq \phi(n_{ui}) - 1$

Step 5: Server computes private key d as: $d_{ui} = e_{ui}^{-1}\ mod\ \phi(n_{ui})$.

Step 6: Server sends public/private key pair (e_{ui}, n_{ui}), (d_{ui}, n_{ui}) to u_i.

3.1.2. Client Request Generation Phase

The querying client generates the query request in relation to the client's location coordinates and, using the encryption module, encripts the location-based query request to the server in the form $(C_{Lui}, C_{mui}, C_{kui})$.

Encryption module. After the user u_i has been registered and the keys $(e_{ui}, n_{ui}/d_{ui}, n_{ui})$ have been generated, he/she can query the LBS server for POIs by sending a message m_{ui}, such that his/her location is not revealed to unauthorized entities. The query content is kept confidential and he/she can verify the authenticity and integrity of server results. To send the message and location coordinates (m_{ui}, L_{ui}), u_i does the following.

(i) The user u_i chooses an integer as the session key k_{ui} to perform message and location coordinate transfer.

(ii) u_i runs the Client Symmetric Encryption (CSE) algorithm. The algorithm allows u_i to supply the message m_{ui}, location coordinates L_{ui}, and session key k_{ui}, that will be used to perform the encryption process.

(iii) u_i inputs m_{ui} and the location coordinates L_{ui} which comprise the latitude as x_{ui} and the longitude as y_{ui} are represented as $L_{ui} = (x_{ui}, y_{ui})$. The longitude and latitude indicate the actual position of the user at the point of querying the location-based server for POIs. Therefore, m_{ui}, L_{ui} are encrypted with k_{ui} as C_{mui}, C_{Lui} (ciphertext of the actual message and location coordinates, respectively) using the AES-256 algorithm.

(iv) u_i then encrypts the session key k_{ui} with the server's public key $PKs = (e_s, n_s)$ by running the Client Asymmetric Encryption (CAE) algorithm. The RSA algorithm is used to encrypt the session key k_{ui}.

The steps are outlined as follows.

Step 1: u_i selects an integer as session Key k_{ui}.

Step 2: u_i encrypts location coordinates $L_{ui} = (x_{ui}, y_{ui})$ with k_{ui} using AES-256 as:

$$E_{kui}(L_{ui}) = C_{Lui}.$$

Step 3: u_i encrypts the plaintext message m_{ui} with k_{ui}, using AES-256 as:

$$E_{kui}(m_{ui}) = C_{mui}.$$

Step 4: u_i encrypts the key k_{ui} using the server's public key (e_s, n_s) as:

$$E_{es}(k_{ui}) \rightarrow k_{ui}^{es}\ mod\ n_s = C_{kui}.$$

The user u_i sends encrypted location coordinates, message and session key as $(C_{Lui}, C_{mui}, C_{kui})$ to the server.

3.1.3. Server Response Generation Phase

In this phase, the server *S* decrypts the encrypted request, searches its database for the POI that matches the request, encrypts the response and digitally signs it with its private key on the leftmost bit of its symmetric key. The server *S* sends (C_{rs}, C_{ks}, *DSig*) to the client.

The server receives the encrypted message from u_i in the form (C_{Lui}, C_{mui}, C_{kui}). The server runs the Server Asymmetric Decryption (SAD) algorithm which accepts (C_{Lui}, C_{mui}, C_{kui}) and its private keys (d_s, n_s) as the input to retrieve the Session key k_{ui}. The server computes the following:

Decryption module:

Step 1: Server *S* decrypts C_{kui} with (d_s, n_s) to give k_{ui} as:

$$D_{ds}(C_{kui}) \rightarrow C_{kui}^{ds} \, mod \, n_s = k_{ui}$$

After the session key has been retrieved in plaintext k_{ui}, it is used to compute the ciphertext location coordinates and message to plaintext L_{ui}, m_{ui}.

Step 2: Server *S* runs Server Symmetric Decryption (SSD) algorithm on C_{Lui} using *AES 256* with the session key k_{ui} as:

$$D_{kui}(C_{Lui}) = L_{ui}.$$

Step 3: Server *S* runs Server Symmetric Decryption (SSD) algorithm on C_{mui} using *AES 256* with the session key k_{ui} as:

$$D_{kui}(C_{mui}) = m_{ui}.$$

S searches its plaintext database for a matching response (in terms of POIs) to m_{ui}. The response r_s must be related to the location coordinates L_{ui} indicating the position of u_i.

Encryption module. Before *S* sends the response r_s to u_i, it does the following:

(i) Server *S* chooses an integer as the session key k_s to perform the response transfer to the user.

(ii) Server *S* runs Server Symmetric Encryption (SSE) algorithm. The algorithm allows *S supply* the response r_s, and session key k_s that will be used to perform the encryption process.

(iii) Server *S encrypts* r_s with k_s as C_{rs} (ciphertext of the server's response) using the AES-256 algorithm.

(iv) Server *S* then encrypts its session key k_s with u_i public key (e_{ui}, n_{ui}) by running the Server Asymmetric Encryption (SAE) algorithm. The RSA algorithm is used to encrypt session key k_s.

The steps are outlined, as follows.

Step 1: Server *S* selects an integer as session Key k_s.

Step 2: Server *S* encrypts r_s to C_{rs} with k_s, using AES-256, as:

$$E_{ks}(r_s) = C_{rs}.$$

Step 3: Server *S* encrypts k_s with (e_{ui}, n_{ui}), using RSA, to produce C_{ks} as:

$$E_{eui}(ks) \rightarrow k_s^{eui} \, mod \, n_{ui} = C_{rs}.$$

Digital signature module. The RSA Digital Signature Scheme (RSA-DSS) is employed by the server *S* to sign the response r_s before sending it to u_i. The aim of employing the digital signature scheme, as mentioned earlier, is to guarantee that the sender of the response r_s is genuine, r_s is correct and the message has not been altered in transit. The steps are outlined, as follows.

Step 1: Server *S* selects the leftmost bit of C_{ks}, where $C_{ks} = \{1....9\}$

Step 2: Server *S* signs the leftmost bit of C_{ks} with its private key (d_s, n_s) using RSA-DSS as:

$$E_{ds}(C_{ks}) \rightarrow C_{ks}^{ds} \, mod \, n_s = DSig.$$

The Server *S* sends to the u_i the signed ciphertext of response and session key in the form (C_{rs}, C_{ks}, *DSig*).

The server *S* signs the leftmost bit of C_{ks} instead of the whole message in a bid to increase the speed of computation and conserve memory space.

3.1.4. Client Response Retrieval Phase

The client verifies if the message is from a genuine server by using the server's public key to verify the encrypted symmetric key. After verification, it decrypts the message.

Verification module. u_i verifies $DSig$ with (e_s, n_s) as:

$$D_{es}\,(DSig) \rightarrow DSig^{es}\ mod\ n_s = C_{ks}.$$

$$C_{ks} = \begin{cases} C_{ks}\ mod\ n_s \rightarrow \text{valid result} \\ C_{ks}'\ mod\ n_s \rightarrow \text{invalid result} \end{cases} \tag{4}$$

Decryption module. After verification is done, the message is decrypted with the client's private key.

Step 1: u_i decrypts C_{ks} with (d_{ui}, n_{ui}) to give k_s as:

$$D_{eui}(C_{ks}) \rightarrow C_{ks}{}^{dui}\ mod\ n_{ui} = k_s$$

Step 2: u_i uses k_s to decrypt C_{rs} to get r_s as:

$$D_{ks}(C_{rs}) = r_s.$$

4. Implementation, Results and Evaluation

4.1. Implementation

The proposed scheme was implemented on the client side using the Java programming language (JDK 1.7) on a 32-bit Windows 7 operating system. XML was used to connect the Android application to the server. The MYSQL database provides storage for the user's credentials and POIs. The Android Development Kit was used to develop the respective Android mobile application. The mobile phone that ran the client during testing is the Techno N7 smartphone with the Android OS version 4.0.4, kernel version 3.0.13 and the baseband version MAUI.11AMD.W12.22.SP.VI.P7.

On the server side, PHP was used to interface the Android application via XML, while Java was used to develop the RSA encryption algorithm. The server configuration is an Intel(R) core i5, 8 GB RAM, CentOS, HP server.

The proposed scheme consists of four fundamental phases: Client registration phase, Client request generation phase, Server response generation phase and Client response retrieval phase.

4.1.1. Client Registration Phase

The main menu of the proposed scheme has three interfaces, namely: interface to Find Place, Register New User and Suggest Places. The new user registers herself/himself after clicking the "register new user" button on the HOME page, which links to the interface Register Friends. This interface requires users to input their Username and Password while the server generates the public/private key pair for each individual user that is being registered.

During the registration at the server side, the server awaits incoming requests, registers the new client details, generates the public and private keys, and then sends the details to the user. Each interface performs functions on the menu list as shown in Figure 2.

4.1.2. Client Request Generation Phase

After the new user has been duly registered, the user can issue requests for POIs to the LBS server. The first thing the user does is to login with his/her credentials in order to query the location-based service. If a user enters incorrect credentials, a prompt message that denies access to that user is displayed.

After access has been granted to the user, an interface will be displayed showing the category of places where user issued requests for POIs services to the LBMS server.

As the user chooses a particular POI, the GPS enabled on the mobile device acquires location coordinates (in terms of longitude and latitude) with reference to the user's position. The POIs span 5 km radius around the user's position. The interface is shown in Figure 3.

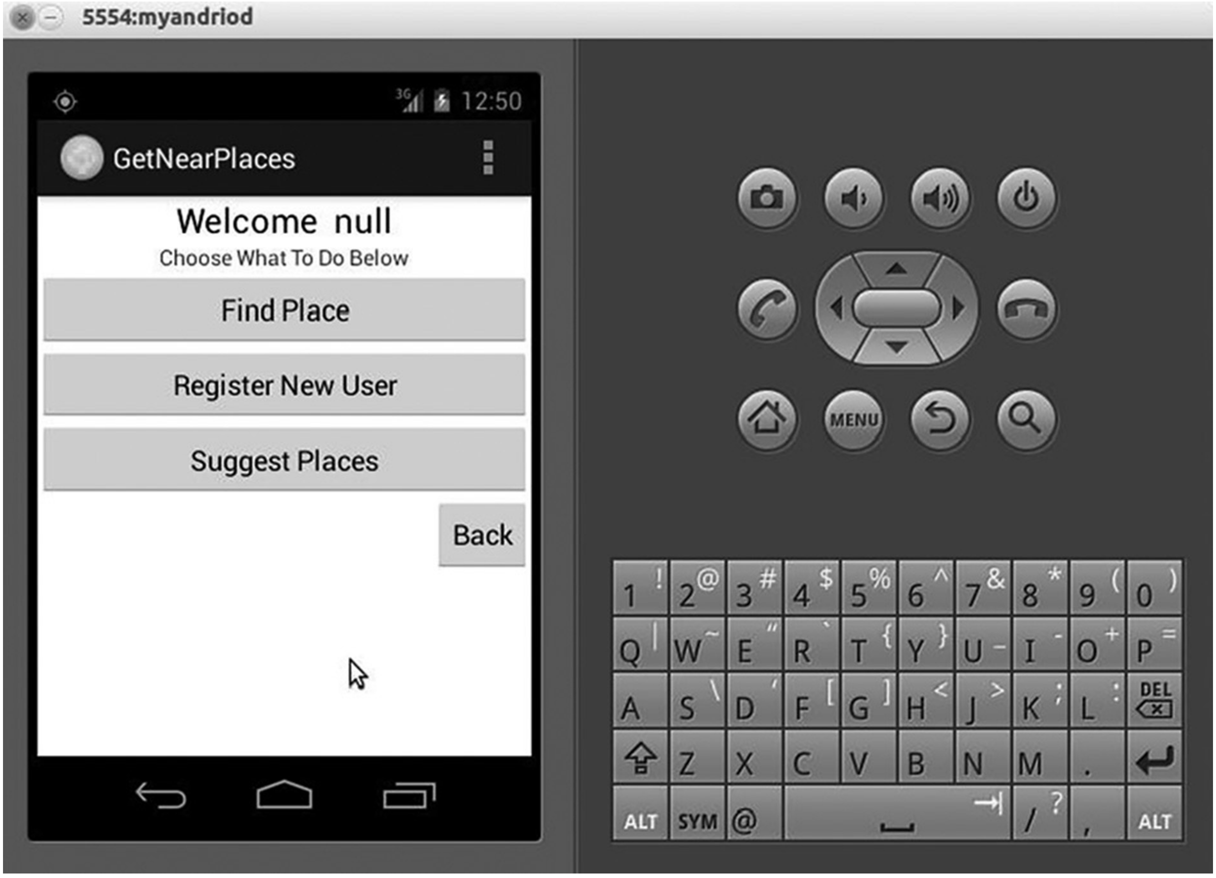
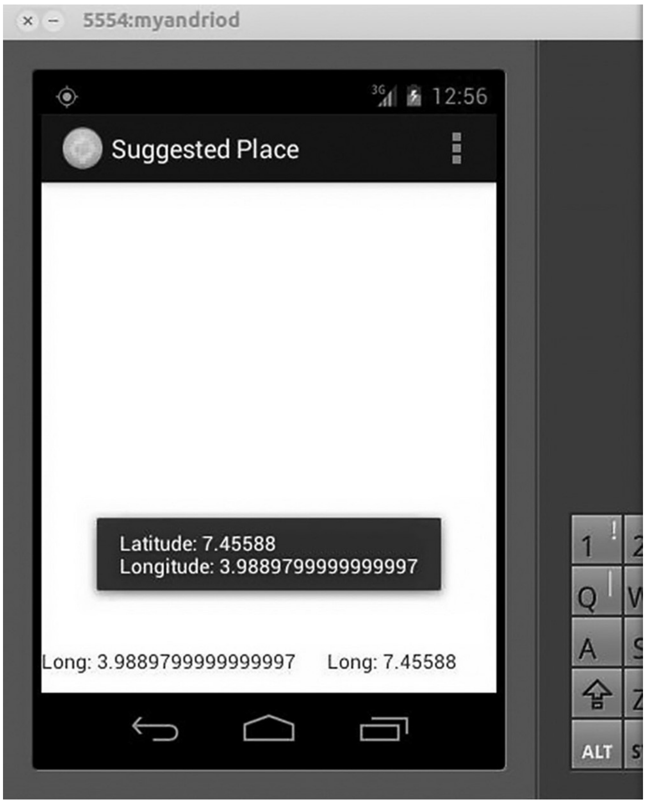


Figure 2. LBMS home page



Latitude: 7.45588
Longitude: 3.9889799999999997

Long: 3.9889799999999997 Long: 7.45588

Figure 3. Location coordinates with reference to user's position

4.1.3. Server Response Generation Phase

The server generates a response to the user's request in relation to the proximity of user's position by providing the "Suggested Places" interface (in terms of longitude and latitude).

During response generation at the server side, the server awaits the incoming request, decrypts the client's request, queries the POI according to the proximity, encrypts and signs the response, and finally sends the response to the user.

By clicking on one of the Suggested Places (*i.e.* POI) buttons, direction is provided on how to locate the Suggested Place.

4.1.4. Client Response Retrieval Phase

If a malicious server sends an incorrect result to the user, the user verifies the authenticity of the result and the origin by using the public key of the server. If not, a dialog box pops up indicating that the sender is not genuine.

4.2. Results and Evaluation

In order to evaluate the performance of LBS_PI, this section provides detailed results of analyses and experiments with existing related researches.

4.2.1. Security Analysis

Security robustness of the scheme relates to the issues of message confidentiality, location and identity privacy, as well as message and origin integrity. Security analysis is evaluated, and computational and communication overhead of the scheme is discussed.

LBS_PI has two aspects of security: message confidentiality and message integrity.

Message confidentiality. Message confidentiality implies the computational infeasibility of an adversary to gain access to any useful information on the content of RSA from which the session key can be recovered. An adversary cannot successfully obtain the session key if the transaction is intercepted because he/she has to derive the private key d through which the session keys (k_{ui}, k_s) can be recovered. The adversary must be able to solve the integer-factorization problem in order to gain access to information and this is achieved by factorizing n into prime numbers (p and q), provided they are not too large. The secret knowledge of d increases the computational hardness of this attack.

User → Server. Assuming that a user $u_i \in U$ (U denotes a set of many users) requests for a POI from the server S, u_i has a public/private key pair e_{ui}, d_{ui}, respectively. By default S generates its public/private key pair as e_s, d_s; both asymmetric keys are used for session key transfer. u_i computes the following.

Step 1: u_i selects a session key k_{ui} to encrypt the request p (we assume p to comprise of location coordinates (x_{ui}, y_{ui}) and the message m_{ui}) as:

$$E_{kui}(p) = C_p. \tag{1}$$

Step 2: u_i encrypts k_{ui} by computing

$$E_{es}(k_{ui}) \rightarrow k_{ui}^{es} \; mod \; n_s = C_{kui}. \tag{2}$$

Note: S decrypts C_{kui} iff $d_s = e_s^{-1} \; mod \; \phi(n_s)$ as:

$$D_{ds}(C_{kui}) \rightarrow C_{kui}^{ds} \; mod \; n_s = k_{ui}. \tag{3}$$

Proof:

$D_{ds}(C_{kui}) \rightarrow C_{kui}^{ds} \; mod \; n_s = k_{ui}$

Substituting equation (2) into (3)

$(k_{ui}^{es})^{ds} \; mod \; n_s = k_{ui}$

Since $(e_s * d_s) \; mod \; \phi(n_s) = 1$

Therefore $k_{ui}^{1} \; mod \; n_s = k_{ui}$.

Assuming an adversary Oscar $u_j \in U'$ (U' denotes a set of attackers) with public/private key pair (e_{uj}, d_{uj}), we show that u_j is not a legitimate receiver of the request sent by u_i and cannot recover the session key k_{ui} to decrypt the request p as follows:

$$D_{duj}(C_{kui}) \rightarrow C_{kui}^{duj} \; mod \; n_s \neq k_{ui}. \tag{4}$$

Proof:

$D_{duj}(C_{kui}) \rightarrow C_{kui}^{duj} \; mod \; n_s$

Substituting equation (2) into (4)

$(k_{ui}^{es})^{duj} \; mod \; n_s$

Since $(e_s * d_{uj}) \; mod \; \phi(n_s) \neq 1$

Therefore $(k_{ui}^{es})^{duj} \; mod \; n_s \neq k_{ui}$.

Server → user. Server S responds to the request from u_i by computing the result r with its session key k_s, which is computed using e_{ui}:

$$E_{ks}(r) = C_r \tag{5}$$

$$E_{eui}(k_s) \rightarrow k_s^{eui} \; mod \; n_{ui} = C_{ks} \tag{6}$$

The user u_i authenticates the result from S (C_r, C_{ks}).

u_i decrypts C_{ks} as:

$$D_{dui}\,(C_{ks}) \rightarrow C_{ks}^{dui} \; mod \; n_{ui} = k_s \tag{7}$$

Note: u_i decrypts C_{ks} iff $d_{ui} = e_{ui}^{-1}$ mod $\phi(n_{ui})$

Proof:

$D_{dui}\,(C_{ks}) \rightarrow C_{ks}^{dui} \; mod \; n_{ui}$

Substituting equation (6) into (7)

$(k_s^{eui})^{dui} \; mod \; n_{ui}$

Since $(e_{ui} * d_{uj}) \; mod \; \phi(n_{ui}) = 1$

$k_s \; mod \; n_{ui} = k_s$

Message integrity. Message integrity means that the message has not been modified in transit, that the sender of the message is genuine (data origin authentication) and, if the origin of the message is genuine, that the message sent from the origin is correct. Integrity could be achieved using the digital signature of the RSA scheme.

After the server S encrypts the result r, S signs the left-most bit of C_{ks}.

Note: Let DS = Digital Signature

S computes DS as:

$$E_{ds}(C_{ks}) \rightarrow C_{ks}^{ds} \; mod \; n_s = DS \tag{1}$$

Upon receiving the result from S, u_i computes the following:

$$D_{es}\,(DS) \rightarrow DS^{es} \; mod \; n_s = C_{ks} \tag{2}$$

Substituting equation (1) into (2)

$(C_{ks}^{ds})^{es} \; mod \; n_s$

Since $(e_s * d_s) \; mod \; \phi(n_s) = 1$

Therefore $C_{ks}^{1} \; mod \; n_s = C_{ks}$

Assume a malicious server S_m (with private/public key pair d_j, e_j) sends a compromised result to u_i. As long as the session key k_s cannot be recovered by S_m, it is implied that k_s cannot be known, which further determines that the result cannot be altered while in transit, and the user verifies the genuineness of server's result.

S_m signs with d_j as:

$$E_{dj}(C_{ks}) \rightarrow C_{ks}^{dj} \; mod \; n_j \neq DS \tag{3}$$

u_i verifies that the result is not genuine:

$$D_{es}(DS) \rightarrow DS^{es} \; mod \; n_j \neq C_{ks}. \tag{4}$$

Proof:

$D_{es}\,(DS) \rightarrow DS^{es} \; mod \; n_j \neq C_{ks}$

Substituting equation (3) into (4)

$D_{es}\,(DS) \rightarrow (C_{ks}^{\;dj})^{es} \; mod \; n_j$

Since $(d_j * e_s) \; mod \; \phi(n_j) \neq 1$

Therefore $C_{ks} \; mod \; n_j \neq C_{ks}$

4.2.2. Performance Comparison

Performance of the developed LBS_PI was measured based on known and benchmarked metrics via response time and throughput. Response time is the measurement of the computation time (in seconds) needed for user and server to encrypt, decrypt, sign and verify requests and responses as required, while throughput is the number of responses the server can process per time unit.

Experimental results show the practical performance of the LBS_PI when benchmarked with Oblivious Transfer and Private Information Retrieval of Paulet *et al.* [24] and Privacy Statistics of Popa *et al.* [27], based on response time and throughput, respectively.

Table 1. Request generation.

Method	Response time
LBS_PI	5.801 sec
Paulet *et al.* (2014)	23.907 sec

In Table 1, at request generation, the average response time when the user issues a request for POIs and when his/her location coordinates appear on the mobile device (client-side) was 5.801 sec as compared to 23.907 sec of Paulet *et al.* [24]. Similarly, in Table 2, at response generation, the average response time when the server receives requests from the user, searches for corresponding response and sends response to the user, was 4.571 sec as compared to 1.75 sec of Paulet *et al.* [24].

Table 2. Response generation.

Method	Response time
LBS_PI	4.571 sec
Paulet *et al.* (2014)	1.75 sec

At response retrieval, Table 3 shows the average response time when the user verifies and decrypts, which was 0.491 sec compared to 0.112 sec obtained by Paulet *et al.* [24], whose work entailed only decryption and not verification.

Table 3. Response retrieval.

Method	Response time
LBS_PI	0.491 sec
Paulet *et al.* (2014)	0.112 sec

Table 4. Throughput comparison.

Method	Throughput		
	One	Two	Three
LBS_PI	1.6 sec	3.2 sec	6.4 sec
Popa *et al.* (2011)	1.55 sec	3.19 sec	6.3 sec

As shown in Table 4, the average time taken by the server to generate the response in LBS_PI for one request is 1.6 sec, while for two requests the time spent is doubled compared to the previous request. As the request increases to three, the change in processing time slightly increases. When benchmarked with Popa *et al.* [27], the processing time is slightly faster in LBS_PI, although the change in time is very close.

On the storage aspect, users only store their user id (*i.e.*, username) and private key as their main credentials. Therefore, much storage space is not consumed by the proposed scheme on the user/client side.

5. Conclusion and Future Work

Providing confidentiality and integrity for communication that occurs between users and location service providers is a crucial issue. In this paper, a practical encryption scheme that secures message, location and key transfer is discussed. The scheme is secure, practical, simple and easy to realize. The scheme achieves efficiency in both computational and communication dimensions while enhancing integrity, confidentiality, non-repudiation and also achieving a good compromise between quality of service and response time on the clients' mobile devices.

The Location-Based Service for enhancing Privacy and Integrity (LBS_PI) was designed to address the problem of privacy and integrity. LBS_PI was tested on 8 POIs spanning a 5 km radius around users' positions. A detailed security analysis demonstrating the resistance of LBS_PI to a range of susceptible attacks was provided.

The performance of LBS_PI, when compared with related work, showed that LBS_PI could actually guarantee privacy and integrity with faster average response time and higher throughput in Location-Based Mobile Services.

This paper addresses two crucial issues in privacy preservation of LBMS: message confidentiality and integrity of server's response. The scheme increases complexity for the common modulus attack, the integer factorization attack as well as the brute force attack. Future work will entail a scheme that can handle multiple users simultaneously, along with congestion caused by starting the scheduling mechanism.

References

[1] S. Yoon *et al.*, "Understanding Motivations and Acceptance of Location-Based Services", *International Journal of Hospitality and Tourism Administration*, vol. 19, no. 2, pp. 187–209, 2018. https://doi.org/10.1080/15256480.2017.1305316

[2] B. Kasamani and D. Gikundi, "A Location-Based Service for Handyman Order Placement", *Journal of Systems Integration*, vol. 8, no. 4, pp. 29–41, 2017.

[3] J. Schiller and A. Voisard "Location-Based Services", Elsevier, 2004.

[4] H. Xu and S. Gupta, "The Effects of Privacy Concerns and Personal Innovativeness on and Experienced Customers' Adoption of Location-Based Services", in *Electronics Markets*, vol. 19, no. 2, pp. 137–149, 2009. https://doi.org/10.1016/j.csi.2015.06.001

[5] Khan and A. Light, "Marketing Goes Local: Location-Based Marketing Provides Solutions to Technology's Disruption of Product Placement, Promotion and Pricing" [Online]. Available: http://www.google.com eg/url

[6] R. A. Popa *et al.*, "Privacy and Accountability for Location-Based Aggregate Statistics", in *Proc. of the 18th ACM Conference on Computer and Communications Security*, 2011, pp. 653–666. https://doi.org/10.1145/2046707.2046781

[7] D. Eckhoff and I. Wagner, "Privacy in the Smart City – Applications, Technologies, Challenges, and Solutions", *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp 489–516, 2017. http://dx.doi.org/10.1109/COMST.2017.2748998

[8] I. Memon, "Authentication User's Privacy: An Integrating Location Privacy Protection Algorithm for Secure Moving Objects in Location Based Services", *Wireless Personal Communications*, vol. 82, no. 3, pp. 1585–1600, 2015. https://doi.org/10.12733/jcis8227

[9] H. S. Kim, "What Drives You to Check in on Facebook? Motivations, Privacy Concerns, and Mobile Phone Involvement for Location-Based Information Sharing", *Computers in Human Behavior*, vol. 54, pp. 397–406, 2016. https://doi.org/10.1016/j.chb.2015.08.016

[10] G. Sun, *et al.*, "L2P2: A Location-Label Based Approach for Privacy Preserving in LBS", *Future Generation Computer Systems*, vol. 74, pp. 375–384, 2017. https://doi.org/10.1016/j.future.2016.08.023

[11] N. Shen, *et al.*, "An Efficient and Privacy-Preserving Location Sharing Mechanism", *Computer Standards and Interfaces*, vol. 44, pp. 102–109, 2016. https://doi.org/10.1016/j.csi.2015.06.001

[12] Z. Gardner, *et al.*, "Trading-off Location Accuracy and Service Quality: Privacy Concerns and User Profiles", in *Proc. of the IEEE International Conference on Localization and GNSS (ICL-GNSS)*, 2017, pp. 1–5. http://dx.doi.org/10.1109/ICL-GNSS.2017.8376244

[13] P. Samarati and L. Sweeney, "Protecting Privacy when Disclosing Information: K-Anonymity and Its Enforcement Through Generalization and Suppression", in *Proc. of the IEEE Symposium on Research in Security and Privacy, SRSP*, 1998, pp. 384–393.

[14] R. Weber, "The Digital Future–A Challenge for Privacy?", *Computer Law & Security Review*, Elsevier, vol. 31, no. 2, pp. 234–242, 2015. https://doi.org/10.1016/j.clsr.2015.01.003

[15] F. Olumofin *et al.*, "Achieving Efficient Query Privacy for Location-Based Services", in *Proc . of the International Symposium on Privacy Enhancing Technologies Symposium*, Springer, Berlin, Heidelberg, 2010, pp. 93–111. https://doi.org/10.1007/978-3-642-14527-8_6

[16] F. Olumofin, and I. Goldberg, "Revisiting the Computational Practicality of Private Information Retrieval", in *Proc. of the International Conference on Financial Cryptography and Data Security*, Springer, Berlin, Heidelberg, 2011, pp. 158–172. https://doi.org/10.1007/978-3-642-27576-0_13

[17] R. A. Popa *et al.*, "Privacy and Accountability for Location-Based Aggregate Statistics", in *Proc. of the 18th ACM Conference on Computer and Communications Security*, pp. 653–666, 2011. https://doi.org/10.1145/2046707.2046781

[18] Q. Arain *et al.*, "Location Monitoring Approach: Multiple Mix-Zones with Location Privacy Protection Based on Traffic Flow over Road Networks", *Multimedia Tools and Applications*, vol. 77, no. 5, pp. 5563–5607, 2018. https://doi.org/10.1007/s11042-017-4469-4

[19] S. Pradhan and B. Sharma, "A New Design to Improve the Security Aspects of RSA Cryptosystem", *International Journal of Computer Science and Business Informatics*, vol. 3, no. 1, pp. 1694–2108, 2013.

[20] C. Paar and J. Pelzl, "Understanding Cryptography: A Textbook for Students and Practitioners", Springer Science & Business Media, 2009.

[21] L. Sweeney, "*k*-anonymity: A Model for Protecting Privacy", *International Journal on Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002. https://doi.org/10.1142/S0218488502001648

[22] M. Gruteser and D. Grunwald, "Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking", in *Proc. of the 1st International Conference on Mobile Systems, Applications and Services*, ACM Press, New York, USA, 2003, pp. 31–42. http://dx.doi.org/10.1145/1066116.1189037

[23] G. Ghinita *et al.*, "Private Queries in Location-Based Services: Anonymizers are not Necessary", in *Proc. of the 2008 ACM SIGMOD International Conference on Management of Data*, 2008, pp. 121–132. http://dx.doi.org/10.1145/1376616.1376631

[24] E. Kushilevitz and R. Ostrovsky, "Replication is not Needed: Single Database, Computationally-Private Information Retrieval", in *Proc. of the*

38th Annual Symposium on Foundations of Computer Science, 1997, pp. 364–373.

[25] G. Ghinita, "Private Queries and Trajectory Anonymization: A Dual Perspective on Location Privacy", *Transactions on Data Privacy*, pp. 3–19, 2009.

[26] R. A. Popa *et al.*, "VPriv: Protecting Privacy in Location-Based Vehicular Services", in *Proc. of the USENIX Security Symposium (USENIX Security)*, 2009.

[27] X. Pan *et al.*, "Protecting Location Privacy Against Location-Dependent Attacks in Mobile Services", *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 8, pp.1506–1519, 2011.
http://dx.doi.org/10.1109/TKDE.2011.105

[28] Z. Gong *et al.* "Protecting Privacy in Location-Based Services Using K-Anonymity without Cloaked Region", in *Proc. of the 11th IEEE International Conference on Mobile Data Management*, 2010, pp. 366–371.
http://dx.doi.org/10.1109/MDM.2010.33

[29] X. Y. Li and T. Jung, "Search Me if You Can: Privacy-Preserving Location Query Service", in *Proc. of the 32nd IEEE International Conference on Computer Communications (IEEE INFOCOM '13)*, 2013, pp. 2760–2768.
https://doi.org/10.1109/INFCOM.2013.6567085

[30] R. Paulet *et al.*, "Privacy-Preserving and Content-Protecting Location Based Queries", *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 5, 2014.
http://dx.doi.org/10.1109/TKDE.2013.87

[31] R. Shokri *et al.*, "Hiding in the Mobile Crowd: Location Privacy Through Collaboration", *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 3, pp. 266–279, 2013.
https://doi.org/10.1109/TDSC.2013.57

[32] M. Elghazal *et al.*, "Applying Location-Based Services for Reducing Mobile Power Consumption", in *Proc. of the IEEE International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–5.
http://dx.doi.org/10.1109/ICEngTechnol.2017.8308156

[33] S.R. Garzon, *et al.*, "Geofence Index: A Performance Estimator for the Reliability of Proactive Location-Based Services", in *Proc. of the 18th IEEE International Conference on Mobile Data Management (MDM)*, 2017, pp. 1–10.
https://doi.org/10.1109/MDM.2017.12

[34] G. Sun *et al.*, "Efficient Location Privacy Algorithm for Internet of Things (IoT) Services and Applications", *Journal of Network and Computer Applications*, vol. 89, pp. 3–13, 2017.
https://doi.org/10.1016/j.jnca.2016.10.011

[35] J. Chen *et al.*, "Blind Filtering at Third Parties: An Efficient Privacy-Preserving Framework for Location-Based Services", *IEEE Transactions on Mobile Computing*, vol. 17, no. 11, pp. 2524–2535, 2018.
http://dx.doi.org/10.1109/TMC.2018.2811481

[36] I. Memon *et al.*, "Enhanced Privacy and Authentication: An Efficient and Secure Anonymous Communication for Location-Based Service Using Asymmetric Cryptography Scheme", *Wireless Personal Communications*, vol. 84, no. 2, pp. 1487–1508, 2015.
https://doi.org/10.1007/s11277-015-2699-1

[37] K. M. Kumar and N. R. Sunitha, "Public Key Cryptosystem for Privacy Sensitive Location-Based Services", in *Proc. of the International Symposium on Sensor Networks, Systems and Security*, Springer, Cham, 2017, pp. 163–172.
https://doi.org/10.1007/978-3-319-75683-7_12

[38] 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.
https://doi.org/10.1155/2018/7823979

[39] Solanas, and A. Martínez-Ballesté, "Privacy Protection in Location-Based Services Through a Public-Key Privacy Homomorphism", in *European Public Key Infrastructure Workshop*, Springer, Berlin, Heidelberg, 2007, pp. 362–368.
https://doi.org/10.1007/978-3-540-73408-6_28

Received: January 2019
Revised: September 2019
Accepted: May 2020

Contact addresses:

Adebukola Onashoga
Federal University of Agriculture
Abeokuta
Nigeria
e-mail: onashogasa@funaab.edu.ng

Adesina Sodiya
Federal University of Agriculture
Abeokuta
Nigeria
e-mail: sodiyaas@funaab.edu.ng

Idowu Osinuga
Federal University of Agriculture
Abeokuta
Nigeria
e-mail: osinuga08@gmail.com

ADEBUKOLA ONASHOGA is an Associate Professor of information security and former head of Department of Computer Science, Federal University of Agriculture, Abeokuta, Ogun State, Nigeria. She has over 50 publications in reputable local and international journals, mostly on computer security. Her current research focuses on cybersecurity analytics, developing privacy preservation algorithms, building access control mechanism for IoT, cryptography and developing secure mobile applications. She is the leading member of a research group on Information Security and Innovative Systems at her university. She is a member of ACM. She equally serves as the chairman of the Board of Trustees to some IT organizations and is presently a council member of CPN and NCS, serving in the capacity of the Southwest Coordinator.

ADESINA SODIYA is a Professor of computer science and information security, currently in the Department of Computer Science, Federal University of Agriculture, Abeokuta, Ogun state, Nigeria. He completed his PhD in computer science in 2004 with a focus on Information Security. His areas of interest include cyber security, attack/intrusion detection, authentication systems, cryptography, systems' security, privacy protection, and security of distributed systems. He has published over eighty scholarly papers in national and international journals, and was the Editor-In-Chief of two scientific journals: Journal of Computer Science and Its Applications, and Journal of Information Security, Privacy and Digital Forensic. In 2010, he won TWAS-AAS-MICROSFT Award as outstanding computer science researcher living and working in Africa. He is member of IEEE, and of two IFIP technical committees: TC-3 (ICT and Education) and TC-11 (Security and Privacy Protection in Information Processing Systems). He is a fellow of Nigeria Computer Society and immediate Vice-President/Vice Chairman of Computer Professional Registration Council of Nigeria (CPN). He is also a member of the Global Commission for the Stability of Cyberspace.

IDOWU OSINUGA is an Associate Professor of optimization theory. He received the BSc degree (with honours) in mathematics from the Ogun State University (now Olabisi Onabanjo University), Ago-Iwoye, Nigeria in 1994; the MSc degree in operations research from the University of Ibadan, Ibadan, Nigeria in 2000; the postgraduate diploma in computer science from the University of Agriculture, Abeokuta, Nigeria in 2002 and the PhD degree in Optimization from the University of Ilorin, Ilorin, Nigeria in 2008. His research interests are nonlinear optimization, applied operations research and computational mathematics. Dr. Osinuga is a member of the Nigerian Mathematical Society (NMS), Society for Industrial and Applied Mathematics (SIAM) and London Mathematical Society (LMS).