# Reducing Dependency on Middleware for Pull Based Active Services in LBS Systems

Saroj Kaushik[1], Shivendra Tiwari[2], and Priti Goplani[3]

[1,2] Dept. of Computer Science and Engineering
[3] Dept. of Mathematics, Indian Institute of Technology,
Delhi, Hauz Khas, New Delhi, India 110016
{saroj,shivendra}@cse.iitd.ac.in,
jca082084@maths.iitd.ac.in

**Abstract.** The middleware is the most commonly used solution to address the location privacy. But it becomes a bottleneck in terms of system performance and availability as the entire client's service transactions are routed through the middleware to the actual Location Based Service Providers (LSP). The proposed architecture mainly targets a variety of applications where the availability of the services is probably more important than the location security. In the new flexible middleware based architecture the client and the LSPs can communicate directly. Autonomy on the client-server communication increases the possibility of communication even in the scenarios where the middleware is not available. But it also introduces authentication and security challenges to be addressed. The trusted middleware is used to generate the authentication certificates containing the Proxy Identity (also called Pseudonyms) to fulfill the authentication requirements at the LSP servers. The rest of transactions among the clients and the LSPs are accomplished independently. Further, the level of anonymity can be tuned by altering pseudonyms generation techniques i.e. "One-to-One", "One-to-Many" and "Many-to-One" depending on the type of the service and security requirements. It also attempts to maintain almost the same level of security for the targeted services.

**Keywords:** Trusted Middleware, Location Based Services (LBS), LB Service providers, Authorization, Pseudonyms, Location Based Service Provider (LSP), Location Privacy.

## 1 Introduction

A Location Based Services (LBS) are entertainment, information and alert type services which are accessible through Over the Air (OTA) network on computers and mobile devices. The LBS services can be divided into various major categories – Pull vs Push services, Person vs Device oriented services, and Active vs Passive services. The push services need real-time location update to the LSP server, so that it can keep track of user's current location, i.e. security alerts, news updates, geo-fencing, and friend finder etc. However, the pull services don't require the continuous update of

O. Senac, M. Ott, and A. Seneviratne (Eds.): ICWCA 2011, LNICST 72, pp. 90–106, 2011.

the user location, and hence user can send the location on-demand to LSP to get the services like point of interest (POI) Searches, Geocoding, and Reverse Geocoding etc. Person-oriented LBS comprises all of those applications where a service is user-based. Thus, the focus of application is to position a person or to use the position to enhance a service. Device-oriented LBS applications are external to the user, where instead of only a person, an object (e.g., a car, a bus) or a group of people (e.g., a fleet) could be located. In device-oriented applications, the person or object located is usually not controlling the service e.g., car tracking for theft recovery. In Active Services, the user initiates the service request; however in the Passive services a third party locates one user (locatee) at the request of another user (the locator). Typical Passive location services are friend finder services, location-based gaming, or fleet management [10].

In LBS systems, there are numerous actors such as content providers (LSP), operators, virtual operators and service administrators etc, all of which can be separate entities. A service provider (LSP) will have automatic access to a customer's location as the location is an essential input to provide the location aware services. Simultaneous observation of the three attributes such as "location" of the user, the "time" at which that location is observed and the "identity" of user creates a threat to user's privacy. The "identity" of the user has the highest importance from privacy point of view. The server has access to learn the location of the customer while the customer is using the service, but it should never know the customer's identity or a combination of it along with "location" and/or "time" attributes. Request trends or query pattern is also crucial for privacy model along with these attributes. Disclosure of the combination of user's identification and query pattern (it implicitly involves user's location) is dangerous. By analyzing the query pattern an adversary can determine the user's location [15, 16, 17, and 18]. Thus there is a need to protect the user's location from being misused. Currently, the middleware architecture is considered to be trusted approach which acts as a three way privacy mediator between the law, the users and the LSPs. The middleware manages a very large number of information providers and high volatility of users' interests (e.g., profile updates, insertion, and deletion etc). The whole subscription database also lies with middleware. The use of the middleware as a single window system ensures greater security as the entire request from the clients to the application service providers are routed through it. So the service providers have no clue of user identity and its location. It has to support high availability despite node failures (e.g., guarantee notification delivery), perform accounting, security and privacy functions etc.

In this paper, we have proposed a system architecture with the least use of trusted middleware and greater autonomy in client-LSP server communication. The proposed architecture is flexible in nature and uses middleware to get the user authentication, but achieve the actual services directly from the service providers (LSP) without middleware's intervention. The architecture is backward compatible and hence the old style of client-LSP communication through the middleware is still possible for the services other than pull based and/or the services requiring a very tight location security. Other than the introduction of the domain in this section, related research work and problem definition are described in section 2 and section 3 separately. The Proposed Solution is explained in sections 4. Section 5 contains possible LBS applications which could be handled by the proposed system. Advantages of the proposed system, limitation and future research possibilities are mentioned in sections 6 and 7. Finally, section 8 concludes the underlying research.

## 2 Related Work

Fig.1 depicts the architecture already in use for middleware based LBS services, showing mobile users, network operator, third-party service providers, and several of the aforesaid subsystems [10]. The client has to go through the middleware to reach the content providers (LSP) asking for any service.

Yingying Chen, et al [7] proposed both; a centralized architecture as well as a fully decentralized enforcement mechanism. They proposed a trusted middleware for facilitating the access control of the location information by enforcing that the mobile devices are only able to access the location information in a manner that conforms to their privileges. Apurva Mohan et al [5] proposed an interesting idea of access control by user profile. They proposed an implementation which can change the policies dynamically. A LBAC (Location Based Access Control) system is integrated with privacy-enhanced techniques based on location obfuscation [2]. Pseudonyms are used by Christian Hauser [6] for handling Identity Privacy. With pseudonyms there arise problems like non associate-ability, non-repudiation and accountability. As the disclosure of personal information in the context of a pseudonym is a monotonic process, the users should be enabled to use different pseudonyms. By this mechanism, users can tune their level of anonymity.

The ticket based service access scheme for the mobile users proposed by Hua Wang et al talks about the mobile databases accessed across multiple service domains anonymously [12]. However this research only talks about the anonymity while mobile roams among the multiple service providers. It doesn’t consider the level of anonymity, and have to contact the credential center for the ticket clearance all the time. It also doesn’t consider the ticket clearance scenarios where the Credential Center is not available due to any unforeseen reason. In the emergency applications a mobile client needs to access the services where user might not be much worried about the security of the data.

Pseudonyms are another useful research done by Christian Hauser [14] for handling Identity Privacy. As the disclosure of personal information in the context of a pseudonym is a monotonic process, the users should be enabled to use different pseudonyms. By this mechanism, users can tune their level of anonymity [13, 14]. They presented both a centralized architecture as well as a fully decentralized enforcement mechanism. But this work is related to location sharing and access control management in order to reveal location to different entities.

Fig-2 shows dependency of the Middleware in the whole architecture. The middleware is the entity which interacts with the content providers (LSP) to get various services for client. The logical working of the architecture depicted in the figure goes as– client forwards every request to middleware; middleware interacts with LSP for the requested service and passes back the response to client.

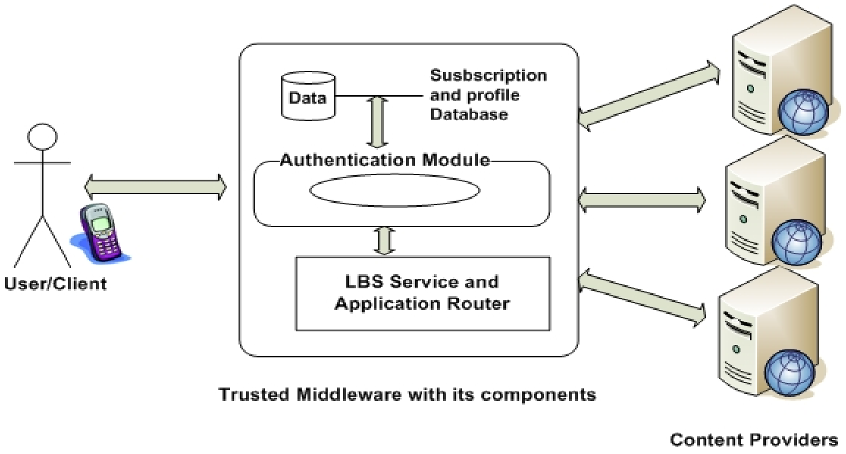
Data
Susbscription and profile Database
Authentication Module
LBS Service and Application Router
User/Client
Trusted Middleware with its components
Content Providers

**Fig. 1.** Architecture of middleware centric Location based Services

## 3 Problem Definition

The key aspect of the problem is being middleware a bottleneck. The client-LSP communication is completely dependent on the middleware. It raises availability, speed, reliability, and response time etc issues in the location services. A user in a human-less jungle would need important services like Food Search, Pedestrian & Car Navigation services etc. At this point the user might not be worried about its location privacy or it might be willing to compromise with the privacy up to some extent for survival-critical services. Following are the key problematic points which will be addressed in the subsequent sections:

- ***Minimize the Dependency on Middleware:*** Availing the direct communication among client and content providers (LSP); minimizing the use of middleware are the major aspect we want to handle. Obviously, it creates other new issues which we have to handle in order to support this point.

- ***Location Revelation to Un-trusted LBS Providers (LSP):*** Due to direct communication among the clients and un-trusted LSPs, they could be potential privacy threats of misusing the location information of the user. Making the user’s location queries and current location, time of the location observation (along with the identity) known to service providers could be dangerous. The transaction queries have to be anonymous.

- ***Minimize Traceability of the Transaction Pattern:*** Even after making the transactions anonymous using proxy identity (pseudonyms), getting service by the same identity every time gives a fair chance to the eavesdroppers to predict user’s nature and day to day routine. This data can be misused in a variety of ways by adversaries.

- ***Ensuring Authorization and Accountability in Location Servers:*** Surely use of single or multiple Pseudonyms could be a potential solution to the problem discussed above. However the challenge to the LBS service providers is to keep the accountability of the usage of the services. They also need to filter out the users according to the authorization and access control they have allowed. In this way we need a standard mechanism to ensure the accountability of the users along with the access control even if they don't reveal their real identity while getting the services.

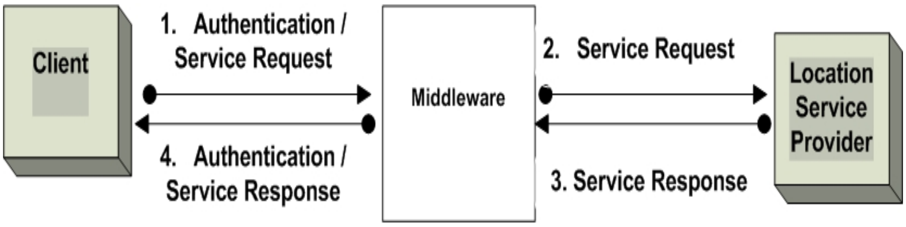
Client
1. Authentication / Service Request
4. Authentication / Service Response
Middleware
2. Service Request
3. Service Response
Location Service Provider

**Fig. 2.** Transaction Flow in Currently Prevalent Architecture

The main focus is to minimize the dependency on the middleware but the user should get the service without letting know its location to the LSPs – on the contrary, location is essential in order to get the Location Based Service. This problem can be tackled by disassociating user identity and its location. But the LSPs need identity to maintain access control and accountability. Following subsections describe a solution which is flexible in nature which allows using the old architecture for the high security requirement services; at the other hand it also allows to bypass the middleware and the clients directly communicate with the LSPs. There are assumptions for the proposed solution such as client is already registered with middleware and has taken the subscription of services of its choice; client has already determined its location and the subscription and profile database lies with the middleware only.

## 4 Proposed Solution

The block diagram of the proposed system is shown in Fig. 3. It mainly consists of three modules:

***a. The Client or User Application:*** A client module refers to the end user application for the location services. It generally resides on mobile and PDA devices.

***b. Un-trusted Location Based Service Providers:*** The un-trusted LSP servers are the real service providers. The key responsibilities of the LSPs are providing service to clients, credential verification, determining access control information.

***c. The Trusted Middleware:*** It is a trusted component of the whole system for direct communication between client and LSP. The user Authentication and Subscription Management, pseudonym generation, service request routing, location determination and billing are the key responsibilities of the middleware.

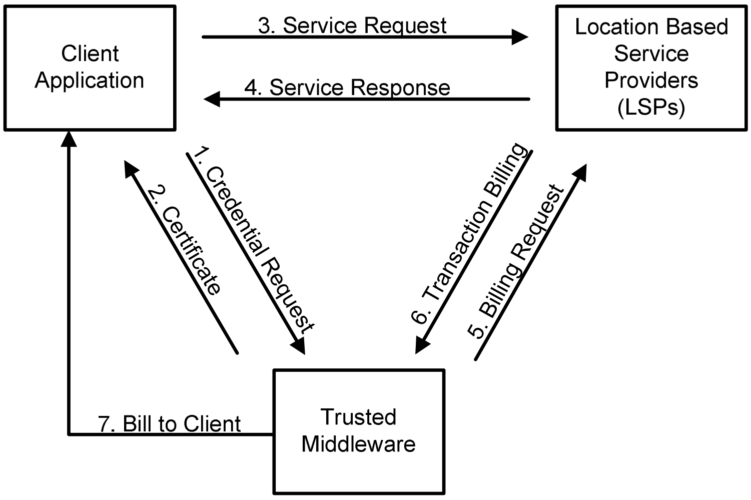

**Fig. 3.** Block diagram of the proposed architecture

### 4.1 Flexible Middleware Architecture

The middleware is mainly divided into three parts – Authentication, Service Routing, and the Billing Management module. The Authentication module takes care of the verification of the client's subscription and authenticity. The service request module routes the request to the desired LSP which is as per the old architecture. The Billing management module handles the transaction billing activities. The middleware generates a Pseudonym against a user to hide user's real identity. The applicability of the pseudonyms enables the user to make it anonymous. But on the other hand service provider needs to know if the client is an authenticated; client is authorized for the requested service and the period of service availability. The functioning of middleware modules are given as below:

**The User Authentication:** When the client wants to initiates the service session with the location server (LSP), it has to first contact middleware to get the Authentication Certificate; which will be used for the subsequent request to the location server (LSP). The middleware has to generate the pseudonym(s) according the user's requested level of anonymity (discussed in the later sections). The certificate should contain the pseudonym generated by the middleware. The certificate is digitally signed by middleware so as to ensure that the certificate is actually generated by middleware. It should also be encrypted by the LSP's public key so that the certificate is used by the targeted LSP only.

**Access Control to the Services (authorization):** We can easily anticipate the problem in the above discussion – authorization. How the LSPs decide if the underlying client is authorized for the requested service even if it is a registered user? The middleware should include all the required access control information into the certificate like services subscribed and temporal validity of the certificate. The access control information should be stored in the LSP server for the running session with the pseudonym for verifying subsequent requests.

The detailed architecture of the proposed system is shown in fig.4. It is flexible as it can also work as old system where each service request and response is routed through middleware which works as a transaction router. Alternatively, the proposed system can disassociate middleware after generating the certificate(s). Now actual service transactions take place between the client and LSP based on the certificate(s). In this mode middleware works as a certifying authority.

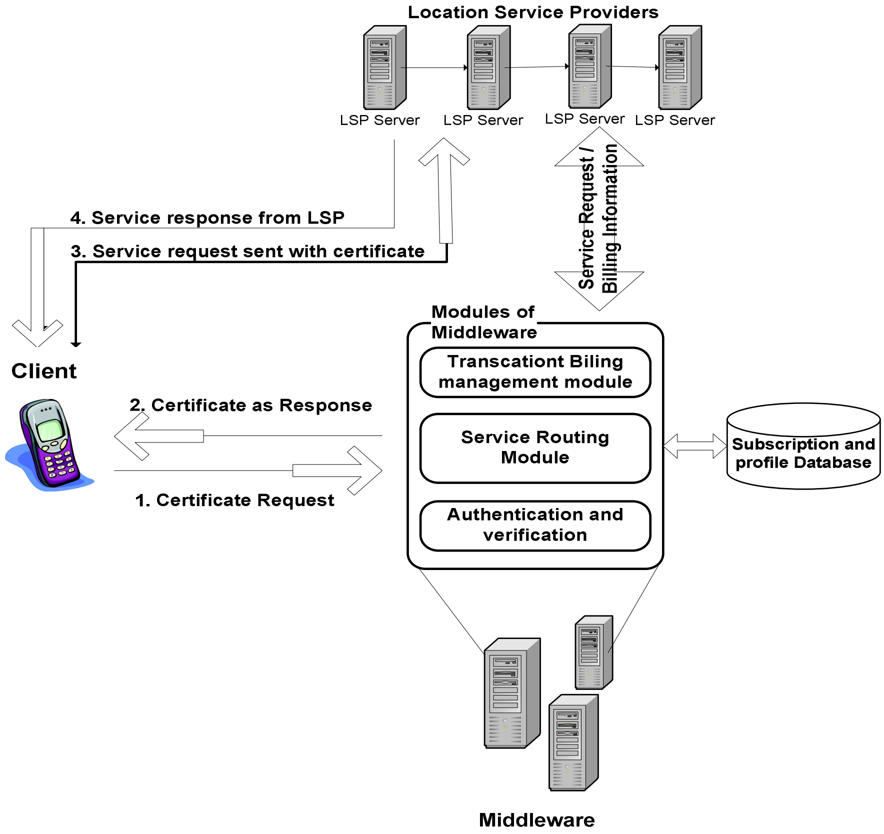
Location Service Providers
LSP Server
LSP Server
LSP Server
LSP Server
Service Request / Billing Information
4. Service response from LSP
3. Service request sent with certificate
Client
Modules of Middleware
Transcationt Biling management module
Service Routing Module
Authentication and verification
2. Certificate as Response
1. Certificate Request
Subscription and profile Database
Middleware

**Fig. 4.** Proposed Flexible Middleware Architecture with transaction flow

### 4.2 Transactional Flow

Following are the steps and the transaction flow of the proposed system architecture:

1. Clients send request to Middleware
   - Middleware generates a Pseudonym against the user's requested level of anonymity.
   - It creates a complete package with the Pseudonym and Access control information as the part of certificate content.
   - It uses LSP's public key to encrypt the certificate content which can be further decrypted by its private key only.
   - The certificate will also be digitally signed by the middleware.

2. Client application prepares and sends the service request to the LSP server
   - The session should be created into the LSP server so that the subsequent requests for the current session don't require further authentication.
   - The request should contain the certificate received from the middleware.

3. LSP Server's Authentication and Session Creation
   - The location server decrypts the certificate by using its private key. It checks for temporal validity of the service and the certificate.
   - It saves the Pseudonym and the corresponding access control information in the newly created session database for the future use.
   - The service response is communicated back to the client.

4. After completion of transaction and/ or expiry of temporal validity, middleware sends transaction accountability request to LSP. In response to this request LSP sends account and whole record of transactions for the users specified.

5. Middleware then generates the bill for the specified pseudonym and sends it to the corresponding user.

### 4.3 Pseudonym Generation

Fig-5 shows the pseudonym generation and recovering the real identity from the given pseudonym. The pseudonym is generated corresponding to a user id with the help of a key available with middleware. The key is selected randomly by the "Key Selector" block on the basis of the "Encryption Key Index" that is generated by the "Random Key Generator" bolck. The key is further passed to the encryption module to perform encryption operation. The pseudonym can be directly generated by using standard symmetric encryption method i.e. Data Encryption Standard (DES)/ Advanced Encryption Standard (AES) given in [8]. The pseudonyms should be recoverable, efficiently computable and hard to decrypt without the key. The key index is suffixed with pseudonym in order to identify the corresponding key later at the time of recovering the user id by middleware. When recovering the user-id from a given pseudonym, the decryption method in the Fig 5 is used. The "Key Index Extract" block extracts the key index from the pseudonym that is passed to the "Key Selector" block. The "Key Selector" blocks fetches the appropriate key which is further used for the decryption by the "Decryption" block.

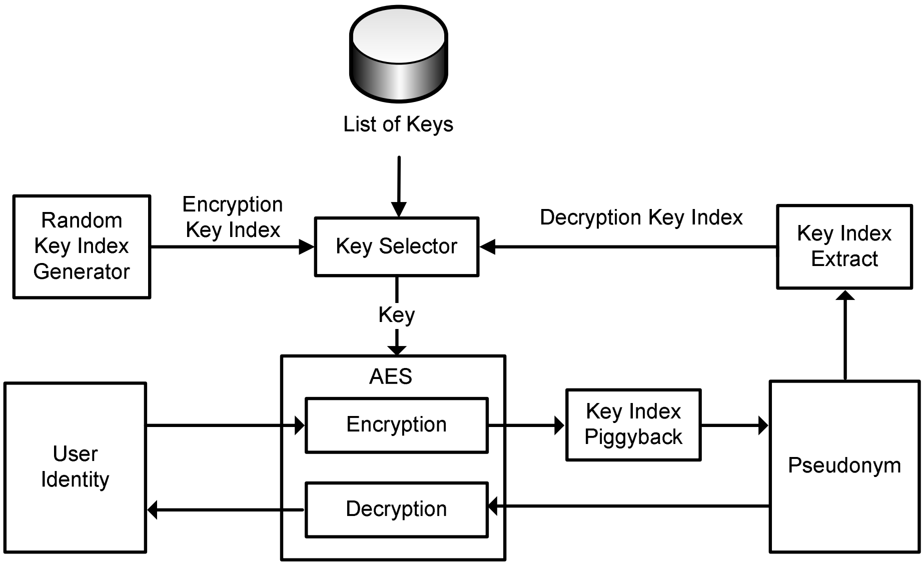

**Fig. 5.** Pseudonym Generation and Recovering User-id – Flow Diagram

In the proposed architecture, user can tune its level of anonymity. Different levels of anonymity can be achieved and their usage scenarios are described in the following subsections. In the algorithms mentioned in the following subsections the AES() procedure has been used as a standard procedure that encrypts the given input using AES cryptography algorithm. Also the notations $E_{ki}$ and $D_{ki}$ are used in the equations for an encryption and decryption operators respectively with the key $k_i$. .

### 4.3.1 One-to-One Pseudonym

It is simplest form of the anonymity denoting a user-id's mapping to a single pseudonym. Apply AES Encryption method [8] to generate Pseudonym by using the User-Id:ID and a key:k, such as $P= E_k(ID)$. The pseudonym can be reversed by using the same key by middleware for billing purpose as $ID = D_k(P)$. A user can opt for a single pseudonym for the entire session if his privacy requirements are low.

Creation of the pseudonym:

$$P= E_k(ID) \tag{4.3.1.1}$$

Recovering user-id from the pseudonym:

$$ID = D_k(P) \tag{4.3.1.2}$$

The procedure *GenPseud-OnetoOne(User-Id:ID, key:k)* generates the pseudonym corresponding to the given user-id. Here k is the key; ID is the user identity to be encrypted; and P is the pseudonym to be generated as an output.

```
GenPseud-OnetoOne(User-Id:ID,key:k)
 {
    P = AES(User-Id:ID,key:k);
    Return P;
 }
```

### 4.3.2 One-to-Many Group Pseudonyms

In this method the user itself requests multiple pseudonyms for a single user-id. The middleware generates ‘n’ number of pseudonyms and separate ‘n’ corresponding certificates but with the same access control information. The same ID is supplied to the encryption procedure along with ‘n’ different randomly selected keys ($k_1 \ldots k_n$). As a result ‘n’ different pseudonyms ($P_1 \ldots. P_n$) are generated. The pseudonyms can be generated and recovered as given below:

Creation of the pseudonym(s):

$$\begin{aligned} P_1 &= E_{k1}(ID) \\ P_2 &= E_{k2}(ID) \\ &\cdots \\ P_n &= E_{kn}(ID) \end{aligned} \tag{4.3.2.2}$$

Recovering user-id from the pseudonym(s):

$$\begin{aligned} ID &= D_{k1}(P_1) \\ ID &= D_{k2}(P_2) \\ &\cdots \\ ID &= D_{kn}(P_n) \end{aligned} \tag{4.3.2.2}$$

The procedure *GenPseud-OnetoMany(* `User-Id:ID`, Keys:($k_1 \ldots k_n$), Anonymity_Degree:n*)* given below also calls the standard AES encryption method ‘n’ times separately to generate the pseudonyms ($P_1 \ldots P_n$) corresponding to the given user-id name and a list of different keys ($k_1 \ldots k_n$).

```
GenPseud-OnetoMany(User-Id:ID,Keys:($k_1 \ldots k_n$),Anonymity_Degree:n)
{
    Initialize i = 0;
    /*loop through all the keys to generate n pseudonyms*/
    While (i < n)
    {
        /*create pseudonyms for each key*/
        $P_i$ = GenPseud_One-to-One(User-Id, $k_i$);
        i++;
    }
    /*a list of n pseudonyms*/
    Return ($P_1 \ldots P_n$);
}
```

### 4.3.3 Many-to-One Group Pseudonyms

The "Many-to-One" pseudonym generation method assigns the same pseudonym to 'n' number of users. This scheme can be used for group subscription where a group of people are agreed to get billed with the same subscription account – a group ID i.e. GID. If we would like to know which user exactly used the service in the group, it can be achieved by piggybacking group user index information with the pseudonym itself; however the rest of the Pseudonym generation method remains the same.

Creation of the pseudonym:

$$P = E_k(GID) \tag{4.3.3.3}$$

Recovering global user-id from the pseudonym:

$$GID = D_k(P) \tag{4.3.3.2}$$

The procedure defined below generates a single pseudonym corresponding to the given user-id and a group key. It uses two functions namely *getGroupUserID()* that maps the User-Id to the GID and the *getGroupPseudonym()* that fetches already created group pseudonym to avoid the rework. We have not defined these procedures explicitly.

```
GenPseud-manytoOne(User-Id:ID, Key:k, Anonymity_Degree:n)
{
    /*a group of users are recognized by a Global ID*/
    GID = getGroupUserID(ID);
    /*Fetch the pre-generated pseudonym for the GID*/
    P =  getGroupPseudonym(GID);
    if (P == NULL)
    {
           /*generate the new pseudonym if it doesn't exist*/
           P = Gen Pseud_One-to-One(GID,  k);
    }
    Return P;
 }
```

For a scenario with rigorous privacy requirements, one can opt for "Many-to-One" or "One-to-Many" pseudonym schemes. Use of these two schemes makes the chances of a user being tracked, very low. One can argue on the reliability of the different anonymity levels. As a drawback, while using the "One-to-Many" pseudonyms, samples of multiple transactions having the same control information and other attributes may lead to identify the pattern of the requests, in the worst case. Even if the multiple transactions are mapped to one pseudonym, the scenario boils down to a "One-to-One" pseudonym. It still doesn't reveal the user's real identity.

### 4.4 Certificate Generation

The authentication certificates are used to validate a client's subscriptions communicating directly to the LSP. The certificate is a digitally signed document by the middleware with its private key – it uses public key cryptography to encrypt the certificate. The LSP or the client can decrypt the certificate with the middleware's public key.

The authentication certificate contains access control attributes. The LSP can blindly serve the authorized services to the client as the certificate is generated by the middleware with the information available at the subscription database. The access control information along with the pseudonym, are stored in the LSPs so that during the further service requests, it can use the existing credentials. The certificate contains – Pseudonym, Access Control information (a list of services which are allowed to the user on service request) and certificate validity (the service should be provided to the user till the valid period only). Procedure for certificate generation is given as follows.

Generate_certificate() procedure produces is a high-level list of instructions in simple English. The *PrepareContent()* procedure only creates a well formatted attribute document using XML or any other document format, hence it has not been defined separately.

```
Generate_certificate(User-Id:ID, Pseudonyms:P, Access
Control:authm, LSP Public Key:LSPk, Middleware Private Key :
MDWk)
{
    /* prepare the certificate content for the given
    pseudonym P */
    C = PrepareContent(User-Id:ID, Pseudonyms:P, Access
    Control:authm);


    /*encrypt the certificate with the public key of the LSP
    i.e. LSPk*/
    C1 = AES(LSP Public Key:LSPk, Certificate Content:C);


    /*Certificate Signing: Encrypt the certificate using with
    the private key of the middleware MDWk */
    Cert = AES(Middleware Private Key:MDWk, Encrypted
    Certificate:C1);


    /*return the encrypted certificate*/
    Return Cert;
}
```

## 5 Applications with the Flexible Middleware

The applications requiring the user profile information for decision making need to follow middleware path to get the services. Following are some of the applications which can be handled efficiently by the proposed Flexible Middleware architecture with direct communication between client and LSP. By using the proposed architecture for these applications we can achieve high availability and scalability due to the direct communication to the LSPs. Also proposed architecture expects less response time even in the high number of transactions per minutes scenarios.

*1. Proximity Search:* It is one of the most popular LBS applications. It is a technology that let people search for information like nearest cinema/gas station in their area using mobile equipments.

*2. Turn by Turn Navigation:* This application is about getting the navigation instructions to reach a particular place. For this application, when client sends the request along with the certificate to the LSP, LSP sends the response to the device / user at the location from which the request has arrived as LSP anyways knows the location but identity is hidden.

*3. Geocoding & Reverse Geocoding* **:** Geocoding is the process of finding associated geographic coordinates from other geographic data, such as street addresses, or zip codes (postal codes) and reverse Geocoding is opposite of this. This method makes use of data from street geographic information system.

*4. M-Commerce and Advertisement*: It is a new form of advertising and commerce that uses location-tracking technology in mobile networks to target consumers with location-specific advertising on their mobile devices. The proposed architecture works well for pull based advertisements in which user subscribes and then asks for the incoming ad alerts and LSP sends the advertisements to a specific pseudonym.

*5. Near-me Area Network (NAN)*: It is a logical communication network that focuses on two way communication among wireless devices in close proximity [11]. For example, a user lost its child in the street, and wants to locate him/her with the help of passers-by in the proximity. This application can be handled by the proposed system as follows – it is assumed that a user has already subscribed for this type of service (or this can be accessed on an emergency basis also). Users sends a request with child's picture to LSP who broadcasts the picture to all those devices with whom the LSP is communicating and are in proximity of the area from which request has come. If any device wishes to respond this type of query, it can directly ping the LSP. The LSP server further informs to the victim user(s).

## 6 Advantages of the Proposed Solution

In this section, we highlight advantages of the proposed system. Table 1 has an intutive comparison of the proposed architecture against the prevelent one and is based on the metrics taken from software architecture theory. The attribute values for the columns is based on the theratical knowledge of the corresponding architectures [19]. The "Good" means the sufficient attribute strength of the corresponding architecture; the "Average" value denotes that it is not sufficient and it can be further improved; however the "Poor" value indicates that the attribute is unaccepable for the real deployments, hence it must be improved further. The attribute value "NA" indicates that the value is either not available or insignificant.

*1. Threat Handling:* It doesn't require disclosing the real user identity to the LSP as the pseudonyms are used instead of user-id. The authentication and authorization are taken care by the use of authentication certificates. Randomized use of Pseudonym minimizes the risk of disclosure of request pattern analysis.

*2. Configurable Levels of Anonymity:* Through the use of One-to-One, "One-to-Many" and "Many-to-One" pseudonym(s) user is having control on its degree of anonymity. Depending upon its security requirements user can opt for one or group pseudonyms

*3. Distributed Architecture*: Middleware is no more a bottlenecked component. Apart from distributed transaction architecture, proposed solution also provides distributed transaction accountability.

*4. Recoverable Pseudonyms:* Pseudonyms are recovered easily with the help of keys lying with middleware. They can be recovered by the direct decryption methods using the corresponding key. The keys are stored in the middleware that can be identified with the associated key index in the pseudonyms. This is required for billing purposes or tracking the user incase of using the system for criminal activities.

*5. Dynamic and Easy Access Control:* The access control information is included in the certificates itself; hence LSPs don't have to look for authentication information anywhere else.

**Table 1.** Feature Comparisons

| **Middleware Type(→), Middleware Key Features (↓)** | **Conventional Middleware Architecture [10]** | **Token Based Authentication System [12]** | **Flexible Middleware Architecture (Proposed)** |
|---|---|---|---|
| **Flexibility to LSP Selection** | NA | Good | Good |
| **Distributed Services** | NA | Good | Good |
| **Access Control Configurability** | Good | Average | Good |
| **Emergency Services in case of Middleware Failure** | NA | NA | Good |
| **Distributed Transaction Accountability** | NA | Good | Good |
| **Response Time** | Poor | Good | Good |
| **Service Transactions Autonomy** | NA | Poor | Good |
| **High Availability of services** | Average | Average | Good |
| **Transaction Accountability** | Good | Good | Good |
| **Location Privacy** | Good | Average | Good |
| **Extendability (adding more LSPs/features)/scalability** | Poor | Good | Good |
| **Flexibility (bypass middleware)** | Poor | Poor | Good |
| **Robustness** | Average | Average | Good |
| **Reliability** | Good | Good | Good |

6. *Non Link-ability of Pseudonyms:* As the pseudonyms can be picked and changed randomly by the user for each of the service request, therefore it is difficult to link them to the user's real profile on the basis of footprint trace.

*7. Less Transaction Time:* Overall transaction time is reduced because of direct communication between client and LSP after certificate generation. Time of encryption and certificate generation is amortized as the certificates are valid for iterative service transactions.

*8. High Availability*: In the proposed architecture services are available all the time despite the possibility of middleware failure. After Authentication Certificate(s) are generated for a session, there is no dependency on the middleware for further transaction.

## 7 Limitations and Future Work

The proposed solution handles the access control and authorization issues efficiently in theory, but this is yet to be proved by the real statistics. The work is in progress to implement the complete system architecture so that we can evaluate the system with the real performance figures. This architecture is more suitable for the Pull-based Active Services. The new architecture is less suitable for the applications which need continuous tracking of the user such as child fencing. So an open issue is still to minimize the middleware usage for other kinds of location services (including push based services). In case of a session loss, certificate has to be sent again to location server and thus creates another computation overhead at LSP while decrypting the certificate. To prevent the misuse of certificate by the client itself by sharing them to other users, we can associate the generated certificate to the device id. Or the device foot prints can be used to ensure that the authentication certificates are used by only the targeted devices. But again this will raise an issue of only one device being used by a user for all transactions which is another constraint.

## 8 Conclusion

The proposed solution is a methodological way of minimizing the dependency on the middleware and enabling the direct client-LSP communication. It also handles the issue of access control based, temporal based, and transaction based accountability. The authentication happens through the middleware's trusted certificate without revealing the identity to the LSPs. The authorization and access control information get included into the certificate itself therefore the location servers don't have to bother about accountability of the users due to anonymity. Also, the clients can directly talk to the content providers, rather than sending each and every service request through the middleware as in prevalent architectures. The certificate encrypted with the location server's public key makes easy to handle the decryption by the location server using its private key. The anonymity again became blurred due by implementing anonymity in two ways i.e. "One-to-Many" and "Many-to-One" anonymity.

## References

1. Lioudakis, G.V., et al.: A Middleware architecture for privacy protection. The International Journal of Computer and Telecommunications Networking 51(16), 4679–4696 (2007)
2. Ardagna, C., Cremonini, M., Damiani, E., De Capitani di Vimercati, S., Samarati, P.: New Approaches for Security, Privacy and Trust in Complex Environments. In: Venter, H., Eloff, M., Lahuschagne, L., Eloff, J., von Solms, R. (eds.) IFIP International Federation for Information Processing, vol. 232, pp. 313–324. Springer, Boston (2007)
3. The European Opinion Research Group. European Union citizens' views about privacy: Special Eurobarometer 196 (December 2003)

4. Ardagna, C.A., Cremonini, M., De Capitani di Vimercati, S., Samarati, P.: Access Control in Location-Based Services. In: Bettini, C., Jajodia, S., Samarati, P., Wang, X.S. (eds.) Privacy in Location-Based Applications. LNCS, vol. 5599, pp. 106–126. Springer, Heidelberg (2009)
5. Mohan, A., Blough, D.M.: An attribute-based authorization policy framework with dynamic conflict resolution. In: Proceedings of the 9th Symposium on Identity and Trust on the Internet. ACM International Conference Proceeding Series, pp. 37–50 (2010)
6. Hauser, C., et al.: Privacy and Security in Location-Based Systems With Spatial Models. Institute of Communication Networks and Computer Engineering University of Stuttgart, Germany (2002)
7. Chen, Y., Yang, J., He, F.: A Trusted Infrastructure for Facilitating Access Control. IEEE, Los Alamitos (2008), 978-1-4244-2677-5/08/ 2008
8. Hohenberger, S., Weis, S.A.: Honest-verifier private disjointness testing without random oracles. In: Proceedings of the 6th Workshop on Privacy Enhancing Technologies, June 2006, pp. 265–284 (2006)
9. Hauser, C., Kabatnik, M.: Towards Privacy Support in a Global Location Service. In: Proceedings of the WATM/EUNICE (2001)
10. Schiller, J., et al.: Location-Based Services, pp. 16, 91–96. Morgan Kaufmann Publishers, San Francisco (2005), ISBN: 1-55860-929-6
11. Kin, Y.W.: NAN: Near-me Area Network. In: IEEE Internet Computing. IEEE computer Society Digital Library. IEEE Computer Society, Los Alamitos (2010)
12. Hua, W., et al.: Ticket-based Service Access scheme for Mobile Users. In: ACSC 2002 Proceedings of the Twenty-fifth Australasian Conference on Computer science, vol. 4 (2002), ISBN:0-909925-82-8
13. John, B., et al.: Method for Generating Digital Fingerprint Using Pseudo Random Number Code. International Patent WO 2008/094725 A1
14. Hauser, C., et al.: Privacy and Security in Location-Based Systems With Spatial Models. Institute of Communication Networks and Computer Engineering University of Stuttgart, Germany
15. Duckham, M., Kulik, L.: A Formal Model of Obfuscation and Negotiation for Location Privacy. In: Gellersen, H.-W., Want, R., Schmidt, A. (eds.) PERVASIVE 2005. LNCS, vol. 3468, pp. 152–170. Springer, Heidelberg (2005)
16. Gedik, B., Liu, L.: A Customizable k-Anonymity Model for Protecting Location Privacy. In: ICDCS 2005 (2005)
17. Magkos, E., et al.: A Distributed Privacy-Preserving Scheme for Location-Based Queries. In: Proceedings of the 2010 IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks, WoWMoM (2010)
18. Hengartner, U.: Hiding Location Information from Location-Based Services. IEEE, Los Alamitos (2007), 1-4244-1241-2/07
19. Quality Attributes, `http://www.softwarearchitectures.com/go/Discipline/DesigningArchitecture/QualityAttributes/tabid/64/Default.aspx` (last visited on April 27, 2011)