Enabling Secure Location-based Services in Mobile Cloud Computing

Yan Zhu
Computer and Communication Engineering
University of Science and Technology Beijing
30 Xueyuan Rd, Beijing 100083, China
zhuyan@ustb.edu.cn

Di Ma
Computer and Information Science
University of Michigan-Dearborn
4901 Evergreen Rd, Dearborn, MI 48128
dmadma@umd.umich.edu

Dijiang Huang
Computing Informatics & Decision Systems Eng.
Arizona State University
699 S. Mill Avenue, Tempe, AZ 85281
dijiang.huang@asu.cn

Changjun Hu
Computer and Communication Engineering
University of Science and Technology Beijing
30 Xueyuan Rd, Beijing 100083, China
huchangjun@ies.ustb.edu.cn

ABSTRACT

The increasing spread of location-based services (LBSs) has led to a renewed research interest in the security of services. To ensure the credibility and availability of LBSs, there is a pressing requirement for addressing access control, authentication and privacy issues of LBSs in a synergistic way. In this paper, we propose an innovative location-based fine-grained access control mechanism for LBSs, enabling effective fine-grained access control, location-based authentication and privacy protection. Our proposed approach is based on the construction of a spatio-temporal predicate-based encryption by means of efficient secure integer comparison. Our experimental results not only validate the effectiveness of our scheme, but also demonstrate that the proposed integer comparison scheme performs better than previous bitwise comparison scheme.

Categories and Subject Descriptors

D.4.6 [**Security and Protection**]: Cryptographic controls; E.3 [**Data Encryption**]: Public key cryptosystems

General Terms

Security

Keywords

Location-based Service; Mobile Cloud; Access Control; Comparison Mechanism; Attribute-based encryption

1. INTRODUCTION

Today, many mobile applications are constructed as client/server applications, and make use of the 3G connection to store data and perform computation in the cloud. Examples of such applications include document sharing, media players and map browsers. In this paper we particularly focus on location-based services (LBSs) in mobile cloud, which have experienced explosive growth in recent years, particularly leveraging fast development of mobile technology and the wide use of mobile devices. In LBSs, the location of a device, representing one of most important contextual information about the device and its owner, is exploited to develop innovative and value-added services to the user's personal context. Many individual, commercial and enterprise-oriented LBSs are already available and have gained popularity. Analysts project revenues for LBSs to grow from $2.8 billion in 2010 to hit a $10.3 billion by 2015 [1].

The increasing popularity of LBSs has led to a renewed research interest in location-based security, where one important problem is to enforce fine-grained spatio-temporal access control on a large number of users to prevent the unauthorized access of services and the disclosure of valuable LBS data [2, 3]. Moreover, the possibility to identify the user who requests a given service and her/his location information at the time of the request has raised much concern on potential privacy violation [4]. Therefore, we are facing the secure challenge of utilizing LBSs: on one hand, the user needs to be identified by LBS server to get personalized location service as precise as possible; on the other hand, s/he wants to maintain the privacy of her/his location information from the LBS server. Hence, it is critical to explore a systematic mechanism to address above challenging issues.

Using location information for access control, i.e., location-based access control (LBAC), is not a new concept [5]. However, one major challenge in geo-spatial computing is identified as "fine-grained access control mechanisms permitting the precise release of location information to just the right parties under the right circumstances" [6]. To better illustrate the requirement of fine-grained spatio-temporal access control in LBSs, we present a typical example, which employs a payment-based subscription model where a 3-dimensional spatio-temporal authorization is defined as follows [7]: A paying user u subscribes to a LBS for a $s$patial region (x_f, y_f, x_e, y_e) and a $t$ime interval (t_a, t_b); the user u is allowed to read a broadcast from the LBS about a spatial

MCC'13, August 12, 2013, Hong Kong, China.

Table 1: Attribute lists for data user's service certificate (access privileges).

ServiceName	Period-of-Validity	ServiceArea	Category	QoS
Wireless	2010/01/10 -2012/06/15	[(33°08',111°36'),(33°50', 112°25')] (Phoenix)	$\geq$ 3(IndividualPlan)	$\leq$ 3(HighBandwidth)
RealPlay	2011/04/01-2011/12/31	[(39°12',71°11'),(40°12', 73°58')] (New York), [(38°44', 74'15'),(39°06',75°30')] (Vineland)	= 2(FamilyPlan)	$\geq$ 2(LowRate)
IFriends	2011/04/25-2012/06/13	[(38°44', 74°15'), (39°06',75°30')] (Vineland)	$\geq$ 10(MobileClient)	= 4(CertifiedMembers)
Transport	2011/01/08-2011/12/22	[(33°22',111°45'),(33°27', 111°54')] (Tempe), [(33°22', 111°38'),(33°28',111°45')] (Misa)	= 1(PrepaidClient)	= 1(Category)
RealTraffic	2010/08/28-2011/12/03	[(31°20', 108°25'),(37°00',115°00')] (Arizona)	$\leq$ 5(VIPmember)	= 4(Frequency)

coordinate (x, y) at time t if and only if $x_f \leq x \leq x_e$ and $y_f \leq y \leq y_e$ and $t_a \leq t \leq t_b$. In reality, spatio-temporal authorization can be more complex and subject to multi-dimensional restrictions. Table 1 lists several example service certificates stored on a user's mobile device. Each certificate consists of five attributes: *ServiceName* is a string attribute used to distinguish different services; *Period-of-Validity* is a time attribute on year/month/day basis; *ServiceArea* is a location attribute on geometric and symbolic representations, respectively; and *Category* and *QoS* are two integers which can be specified by the service providers.

Attribute based encryption (ABE) schemes [8, 9] have been recently introduced for fine-grained access control. However, there has been little work on studying integer comparison mechanisms to support spatio-temporal control in the context of ABE. Even though Bethencourt *et al.* [10] presented a *bitwise comparison* method to implement integer comparison based on CP-ABE scheme, it is not efficient enough for practical applications. For example,

Suppose a global-oriented LBS needs to make approximate 1 mile precision to express geographical coordinates. In accordance with general latitude and longitude expressions $(degrees, minutes)$*, we must implement the integer representation in the range [1; 21,600] and the efficient integer comparison function in it, where 21,600=360*60.*

Hence, lacking an efficient secure comparison mechanism makes existing ABE schemes difficult to realize various predicates required by fine-grained access control. Furthermore, since a user's service certificate could contain complicated relationships among various attributes (see Table 1), it is necessary to explore a comprehensive cryptographic technique to express those relationships.

In addition to access control in LBSs, potential privacy violation of LBS users has also raised a lot of concern. Although LBS services introduce problematic issues for privacy leakage due to the nature of the service, some location-privacy preserving approaches have been proposed in recent years, such as,

Unlinkability of LBS Transactions: different transactions of the same customer are unlinkable [11];

Anonymity Set of User Identity: many users hold the same identifying information [12];

Obfuscation of Location Query: the coarse-grained spatio-temporal information is used to query LBS service [13].

In general, it is an effective way for developing a practical framework covering spatio-temporal access control model, various cryptographic methods, and location-privacy approaches to realize the secure LBS. Therefore, further research is needed to effectively defend against the risk of privacy leakage and unauthorized LBS access.

Contributions. In this paper, we address the afore-mentioned security and privacy issues in LBSs by constructing a location-based fine-grained access control (LFAC) framework to provide spatio-temporal access control as well as user privacy protection. Our primary goal is to design a new cryptosystem which can support flexible access control over various types of comparison-based constraints, including:

- *Control over independent values*, such as, service name, device ID. For example, (ServiceName = "*RealTraffic*") and (Category = "*FamilyPlan*");
- *One dimensional attribute control*, such as, time, level, or salary. For example, $(3 \leq \text{Category} \leq 5)$ and $(2,000 \leq \text{Salary} \leq 5,000)$);
- *Complex control on multi-attributes*, such as, two dimension coordinate, periodic control on Week and Hour. For example, $((3 \leq \text{Week} \leq 5)$ AND (8:00PM $\leq$ Hour $\leq$ 10:00PM)) and $((33°08' \leq X \leq 111°36')$ AND $(33°50' \leq Y \leq 112°25'))$.

To achieve our goal, we make the following contributions:

1. Our LFAC framework is built upon a novel construction of a spatio-temporal predicate-based encryption scheme (ST-PBE). In LFAC, access policies are enforced entirely by spatio-temporal attribute matches between ciphertexts and private keys on the user side, and no user identification information is required.
2. To achieve fine-grained access control under the framework of ST-PBE, we propose a secure cryptographic integer comparison scheme to support various spatio-temporal comparison-based predicts required by fine-grained access control. It is a new cryptographic scheme which can realize the capability of supporting range attributes allows a user to employ coarse-grained spatio-temporal information in service query to achieve the obfuscation feature.

Organization. This paper is organized as follows. Section 2 discusses the system and security models of our approach. Section 3 overviews our proposed scheme. In Section 4, we analyze the performance of our scheme. Finally, we conclude this paper in Section 5.

2. SYSTEM AND SECURITY MODELS

2.1 System Model

We consider a LBS system on mobile cloud involving three different entities [14] as illustrated in Figure 1. We assume that Certification Center is a trusted third party (TTP), in which the user ID is omitted from the location query and the network address of the query message is anonymized through sender anonymity mechanisms such as Crowds or Onion Routing.

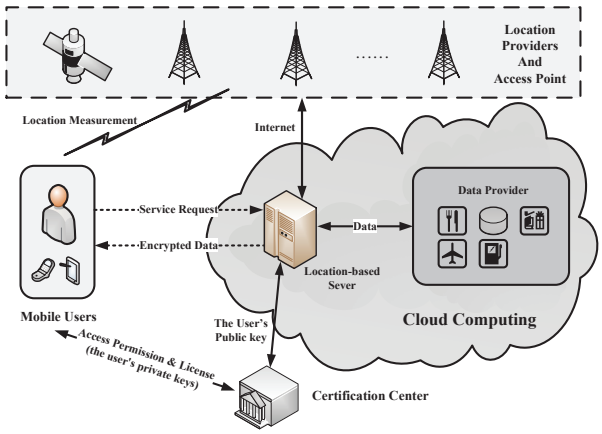

Figure 1: Location-based service architecture.

Mobile User is an entity who wants to access LBS data. It is issued with a service certificate with certain access privileges. A user can access its current location information either by a equipped GPS device or through a location information provider.

Location-based Service Provider is an entity that provides customized location-based services according to the user's request with her/his location information.

Certification Center is a trusted third party (TTP) which issues user certificates (containing user private keys) and provides necessary public parameter information to realize secure LBSs.

To ensure data access compliant with the assigned policy, the fine-grained access control are introduced into LBSs, in which we make use of the ABE's "*privilege-constraint*" attribute matching mechanism to transfer the service authentication into the client side, such that this minimum guarantee that the user's information cannot be obtained by the LBS providers. The system operates as follows:

- A user obtains her/his service certificate (the private key with access privilege $\mathcal{P}$) from the certificate center;
- Upon receiving a request, which contains the objective requested and location information L_c, from an authorized user, the LBS provider
  - Authenticates the user's access privilege by means of anonymous authentication, and receives the user's LBS query in a secure way;
  - Processes the related data from content/data provider with respect to location L_c; and
  - Employs an encryption method to convert the result data into a ciphertext C that embeds the necessary access constraint $\mathcal{L}$, and sends the ciphertext to the user.
- The authorized user can decrypt C and obtain the requested data by using her/his certificate (private-key) with access privilege $\mathcal{P}$.

Through interactions with other entities in the system, an authorized user can enjoy the services provided by LBSs only if her/his access privilege $\mathcal{P}$ satisfies the location-based access constraint $\mathcal{L}$ and L_c belongs to the area defined in $\mathcal{P}$.

2.2 Security Model

Within the three entities involved, the certificate center is a trusted entity. Hence, we are concerned with security risks with respect to LBS servers and data users:

- LBS servers: Similar to [15, 16], we consider "Honest but Curious" servers. That is, servers are assumed to follow the proposed protocol in general, but try to find out as much "privacy" information as possible based on the users' inputs. In particular, we assume the servers are more interested in learning user's private information, such as access habits, history of past movements, and access privileges, rather than user's secret information such as private keys.
- Data users: A dishonest user would try to access data outside the scope of her/his access privileges. To do so, unauthorized users may attempt to change the spatio-temporal constraints in her/his service certificate independently or cooperatively (called collusion attacks). Each party is preloaded with a private key and the public key can be easily obtained when necessary.

3. OVERVIEW OF PROPOSED SCHEME

3.1 Notations

Let $\mathcal{A}$ denote a set of attributes. A user's access privilege $\mathcal{P}$ is defined as an arbitrary Boolean function on AND/OR logical gates and various comparison predicates, such as $Equal, Contain, Cross$, over $\mathcal{A}$. For example,

$$\mathcal{P} = (Equal(A, x) \text{ AND } (Contain(B, [v_1, v_2]) \text{ OR } Smaller(C, y))),$$

where $[v_1, v_2]$ denotes the integer range between v_1 and v_2. Similarly, the access constraint $\mathcal{L}$ can also be defined as the combination of a set of attribute constraints over these comparison predicates. For example, $Contain(A, [x_1, x_2])$, $Equal(B, v)$, $Greater(C, y)$). Other notations are summarized in Table 2.

Table 2: Notations

Name	Description
$PK_{\mathcal{A}}$	the public key over $\mathcal{A}$;
$SK_{\mathcal{P}}$	the private key over $\mathcal{P}$: $SK_{\mathcal{P}} = (SK_1, SK_2)$;
MK	the master key held by the system manager;
$\mathcal{H}_{\mathcal{L}}$	the header of ciphertext over $\mathcal{L}$;
E_D	the body of ciphertext, or the encrypted service data;
C	the ciphertext consists of $\mathcal{H}_{\mathcal{L}}$ and E_D;
ek	the session key used to encrypt the data.

3.2 Spatio-Temporal Predicate-based Encryption

A spatio-temporal predicate-based encryption (ST-PBE) scheme, constructed on Key-Policy ABE model, consists of four algorithms as follows:

- Setup$(1^{\kappa}, \mathcal{A})$: Takes a security parameter κ and a set of attributes $\mathcal{A}$ as input, outputs the master key MK and the public-key $PK_{\mathcal{A}}$;
- GenKey$(MK, u_k, \mathcal{P})$: Takes a user's ID number u_k, the user's associated access privilege $\mathcal{P}$ and MK as input, outputs the user's private key $SK_{\mathcal{P}}$ over $\mathcal{P}$;

- Encrypt($PK_{\mathcal{A}}, \mathcal{L}$): Takes an access constraint $\mathcal{L}$ and PK as input, outputs the ciphertext header $\mathcal{H}_{\mathcal{L}}$ and a random session key ek; and
- Decrypt($SK_{\mathcal{P}}, \mathcal{H}_{\mathcal{L}}$): Takes a user's private key $SK_{\mathcal{P}}$ and a ciphertext header $\mathcal{H}_{\mathcal{L}}$ as input, outputs a session key ek.

Given a cryptographic system based on our ST-PBE definition, we must guarantee that this cryptosystem can follow the principle in range-based access control: Let $A_k \in \mathcal{A}$ be a range-based attribute and $(\mathcal{P}, \mathcal{L})$ be a privilege-constraint pair with A_k, where $Contain(A_k, [x_i, x_j]) \in \mathcal{P}$ and $Contain(A_k, [x_a, x_b]) \in \mathcal{L}$. Secure comparison problem requires that the access is granted if and only if $[x_i, x_j] \cap [x_a, x_b] \neq \emptyset$. That is, given the above-mentioned $(\mathcal{P}, \mathcal{L})$, we can compute $(MK, PK_{\mathcal{A}}) \leftarrow Setup(1^{\kappa}, \mathcal{A})$ and $SK_{\mathcal{P}} \leftarrow GenKey(MK, u_k, \mathcal{P})$, such that the following equation holds if and only if the access is granted over $(\mathcal{P}, \mathcal{L})$ according to a fine-grained access control model:

$$\Pr\left[\begin{array}{c} (\mathcal{H}_{\mathcal{L}}, ek) \leftarrow Encrypt(PK_{\mathcal{A}}, \mathcal{L}), \\ Decrypt(SK_{\mathcal{P}}, \mathcal{H}_{\mathcal{L}}) = ek \\ \forall A_k, Contain(A_k, [x_i, x_j]) \in \mathcal{P}, \\ Contain(A_k, [x_a, x_b]) \in \mathcal{L}, [x_i, x_j] \cap [x_a, x_b] \neq \emptyset \end{array} : \right] = 1$$

The practical ST-PBE scheme can be constructed on the comparison-based encryption (CBE) [17] by using forward/ backward derivation functions (FDF/BDF).

The session key ek generated by Decrypt($SK_{\mathcal{P}}, \mathcal{H}_{\mathcal{L}}$) is used for data encryption. Operations on data are not shown in the framework since the data sender could easily employ a symmetric key cipher ($E_D \leftarrow \mathcal{E}_{ek}(D), D \leftarrow \mathcal{D}_{ek}(E_D)$) for data encryption/decryption with ek. Note that the key ek can only be obtained through the decryption algorithm.

3.3 Location-based Fine-grained Access Control

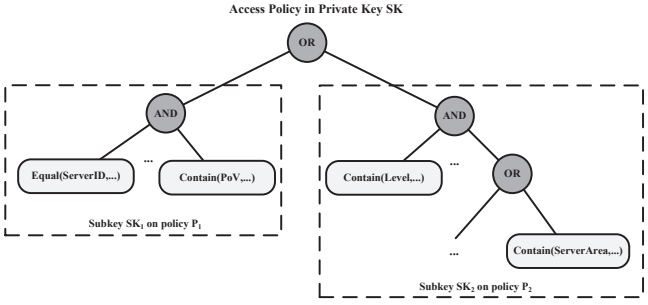

Figure 2: Illustration of Access Policy for Location-based Service.

In this section, we describe how to build LFAC system for LBSs from ST-PBE presented in the previous section. Note that the user's private-key $SK_{\mathcal{P}}$ is constructed over a spatio-temporal predicate-based privilege $\mathcal{P}$ with AND, OR and comparison operations. In LFAC, we use $SK_{\mathcal{P}}$ for two different purposes, service authentication [1] and service data access. Due to the dual purposes, we represent $SK_{\mathcal{P}}$ as $SK_{\mathcal{P}} = (SK_1, SK_2)$ where SK_1 is used for service authentication while SK_2 is used to regulate service data access.

As shown in Figure 2, SK_1 is used to implement a simple service-aware authentication protocol. SK_1 is generated over partial of the user's access privileges denoted as $\mathcal{P}_1$: $SK_1 = GenKey(MK, u_k, \mathcal{P}_1)$. $\mathcal{P}_1$ only consists of a determination of "ServiceID" (SrvID) and "period-of-valid" (PoV), that is,

$$\mathcal{P}_1 = (Equal(\text{SrvID}, \text{“}s\text{”}) \text{ AND } Contain(\text{PoV}, [v_1, v_2])).$$

Let c be the current time. Given an access constraint $\mathcal{L}_1 = (Equal(\text{ServiceID}, \text{“}s'\text{”}), Equal(\text{PoV}, c))$, the LBS server sends a ciphertext header $\mathcal{H}_{\mathcal{L}_1}$ which hides a random session key ek_1: $(\mathcal{H}_{\mathcal{L}_1}, ek_1) \leftarrow Encrypt(PK_{\mathcal{A}}, \mathcal{L}_1)$. The user can be authenticated if s/he can retrieve ek_1 by using the operation $ek_1 = Decrypt(SK_1, \mathcal{H}_{\mathcal{L}_1})$, which implies "$s$" = "$s'$" and $c \in [v_1, v_2]$. SK_2 also makes use of the same way to support the predicate-based privilege $\mathcal{P}_2$. We integrate the two keys into a private-key by using "OR" operation.

Note that, the user can hold multiple private keys $\{SK_{\mathcal{P}}\}$ which correspond to different services which the user has subscribed to. However, these keys are not inter-operable. This is because that even if these services share the same attribute set $\mathcal{A}$, these attributes may have different values according to different access privileges.

Based on ST-PBE, the workflow of LFAC is described in Figure 3. We illustrate the workflow as follows:

Lightweight Service Authentication: When a mobile user sends a simple "hello" service request, the location-based server invokes the algorithm $(\mathcal{H}_{\mathcal{P}_1}, ek_1) \leftarrow Encrypt$ $(PK_{\mathcal{A}}, \mathcal{P}_1)$ and sends $\mathcal{H}_{\mathcal{P}_1}$ to the mobile user. If the mobile user can obtain the valid $ek_1 \leftarrow Decrypt(SK_1, \mathcal{H}_{\mathcal{P}_1})$ in a given time, s/he can move to the next step. Note that, the temporary session key ek_1 should be changed in each request.

Obfuscation and Query: The mobile user makes use of obfuscation methods to get a coarse-grained spatio-temporal query $\mathcal{L}$, and then employs a symmetrical encryption scheme to encrypt the query by using the key ek_1 obtained in the Lightweight Service Authentication phase, that is, $Q \leftarrow \mathcal{E}_{ek_1}(\mathcal{L})$. After receiving and decrypting the ciphertext $(\mathcal{L} \leftarrow \mathcal{D}_{ek_1}(Q))$, the server checks whether the query is valid. If the query is not valid, it will be stopped. Otherwise, the user queries the desired data from data providers and moves to the next step.

LBS Information Transmission: The server first encrypts the spatio-temporal query $\mathcal{L}$ to generate a header of ciphertext $\mathcal{H}_{\mathcal{L}}$ and a new session key ek_2 by $(\mathcal{H}_{\mathcal{L}}, ek_2) \leftarrow Encrypt(PK_{\mathcal{A}}, \mathcal{L})$. And then it encrypts the data to generate the real ciphertext body E_D by using a symmetrical encryption with the help of ek_2, $E_D \leftarrow \mathcal{E}_{ek_2}(D)$. Next, the ciphertext $C_2 = \mathcal{H}_{\mathcal{L}} || E_D$ is sent to the mobile user. The mobile user can decrypt the header to get $ek_2 \leftarrow Decrypt(SK_2, \mathcal{H}_{\mathcal{L}})$ and then decrypt the body to get the desired data by using the key ek_2, that is, $D \leftarrow \mathcal{D}_{ek_2}(E_D)$.

It is easy to find that the LBS query condition $\mathcal{L}$ is specified by the mobile user and it is a valid query if and only if $\mathcal{L}$

[1] The goal of such a design is to prevent abusive service requests targeting a DoS attack: the LBS provider only responds to users who have subscribed to the requested services and thus are able to pass the authentication process.

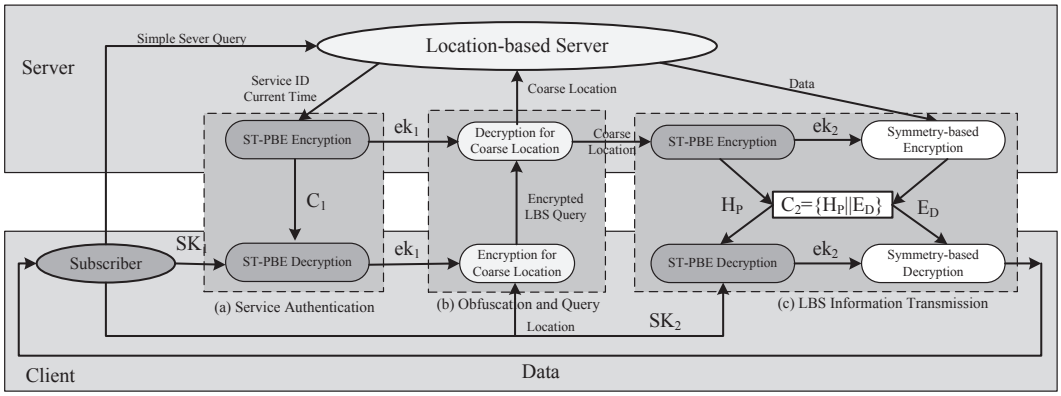

Figure 3: Illustration of LFAC workflow for location-based service.

can satisfy the access control policy $\mathcal{P}$ in the mobile user's private-key $SK_{\mathcal{P}}$.

The storage structure of LBS information transmission is shown in Figure 4. According to different attribute-values and access constraints, the query results can be stored separately in different blocks, e.g., a certain hotel information can be encrypted by using its precise location (e.g., *Equal* $(ServiceArea, Pos)$) and the service level (e.g., *Equal* $(ServiceLevel, 4)$). Of course, we can also encrypt the result data into one block by using a large constraint, e.g., $(Constain(ServiceArea, Range))$, in order to reduce computation cost.

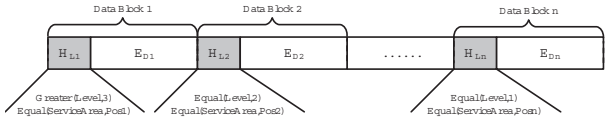

Figure 4: The storage structure of LBS information transmission.

In LFAC, user privacy is protected through several ways. First, access policies are enforced by "*privilege-constraint*" attribute matching between ciphertexts and private-keys on the *client-side* with the help of the KP-ABE model. In the procedures of authorization and encryption, the LBS server does not need any user identification information to enforce access policies. Hence, this ensures that user's information, including user identity and access privileges in the user's private key, will not be disclosed to the LBS servers. Furthermore, this mechanism significantly increases the difficulty of identifying different LBS requests sent by the same mobile user from a mass of user requests, which is called as *unlinkability* of LBS transactions.

Moreover, the proposed LBS architecture provides a privacy-preserving mechanism by separating users' location information from users' identities, in which the LBS server cannot link a certain location with the specific user even if the server obtains user's location information in a LBS service query. Users holding the same license (access privileges) may consist of a large "k-anonymity set" [2]. The larger the size of the anonymity set, the greater the level of anonymity can be offered.

Furthermore, since interval or range predicates are introduced into our location-based access control mechanism, it allows a user to use a coarse-grained spatio-temporal information in location to achieve *obfuscation*. That is, a "location" in our system can be a region as opposed to a point and it serves as a "cloak" to the user's actual position. For example, the user can query the desired information in a relatively large zone instead of his exact position. Cloaking also adds noise to the spatio-temporal information of service queries. In our LBS architecture, obfuscation is achieved through desired query scope or perturbation algorithms [18, 13]. It is true that obfuscation methods may result in inaccuracy or imprecision of the location/time. Thus, some location-based applications may not work well if they get updates only on a scale of hours and kilometers, as opposed to seconds and meters. In this case, we need to carefully select the proper granularity to balance the need for privacy protection and service usability.

[2]The anonymity set is defined as the group of people who hold the same access privileges.

4. PERFORMANCE EVALUATION

We have implemented our ST-PBE scheme in Qt/C++ and experiments were run on an Intel Core 2 processor with 2.16 GHz and 500M of RAM on Windows Server 2003. All disk operations were performed on a 1.82TB RAID 5 disk array. Using GMP and PBC libraries, we have implemented a cryptographic library (called as PKUSMC) upon which temporal attribute systems can be constructed. This C library contains approximately 5,200 lines of code and has been tested on Windows and Linux platforms.

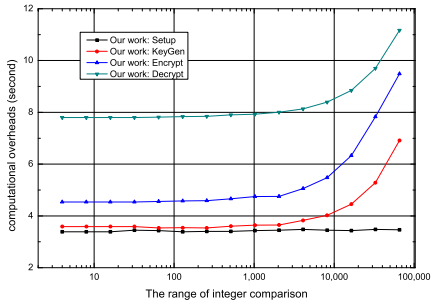

Figure 5: Computational costs of our scheme under different comparison range (the effective calculation length is $L = 2048$-bits).

We show the practical computational costs of algorithms for our scheme in Figure 5 under the effective calculation length is $L = 2048$-bits. In this example, for a certain comparison range $[1, Z]$, we generate a secret-key with licence

$[t_1, t_2]$, where $t_1 \in_R [1, Z/4]$ and $t_2 \in_R [3Z/4, Z]$; and a message is encrypted by the time $t \in_R [Z/4, 3Z/4]$. So, we ensure that $\max(t - t_1, t_2 - t) \geq Z/4$. As the value of Z is changed from 4 to 65,536, the computational costs should keep pace with the growth of comparison ranges. However, we not this growth is not significant by comparing with bilinear operations.

Computational overheads of main cryptographic operations of ST-PBE scheme were shown in Figure 6. The experiments involve 23 independent LBS requests, where we choose randomly some policies and location queries over a set of 7 attributes. These attributes include string, integer, and location expressions. We found that it takes much more time on the additional operations, such as, the policy-tree construction and the access constrain generation. Therefore, the overhead of decryption in service authentication is smaller than those in LBS data transmission because the coordinate expressions are larger in LBS data transmission. Overall, different numbers and scales of attributes and constrains have affected the system performance to some extent, but not very seriously.

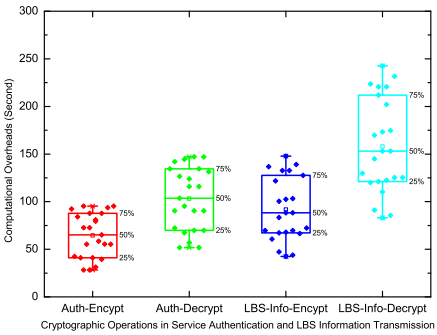

Figure 6: The computational overheads of cryptographic operations in our LBS system.

5. CONCLUSIONS

In this paper, we have proposed a novel cryptographic access control framework for LBSs on mobile cloud. This framework facilitates fine-grained access control, location-based authentication, and privacy protection. Our framework is based on a spatio-temporal predicate-based encryption (ST-PBE) scheme which implemented a novel secure cryptographic integer comparison mechanism to support various predicates required in LBSs. In addition, the implementation of a proof-of-concept prototype and corresponding evaluation demonstrate the feasibility of our methodology.

6. ACKNOWLEDGMENTS

The work of Y. Zhu and C.-J. Hu was supported by the National Natural Science Foundation of China (Project No. 61170264 and No. 10990011) and the National 973 Program (Project No. 2013CB329606).

7. REFERENCES

[1] Jan Ten Sythoff. Location-based services, market forecast, 2011-2015. Technical report, Pyramid Research, 2010.

[2] Heechang Shin and Vijayalakshmi Atluri. Spatio-temporal access control enforcement under uncertain location estimates. In *Proceedings of the 23rd Annual IFIP WG 11.3 Working Conference on Data and Applications Security XXIII*, pages 159–174, Berlin, Heidelberg, 2009. Springer-Verlag.

[3] Maria Luisa Damiani, Elisa Bertino, and Claudio Silvestri. The probe framework for the personalized cloaking of private locations. *Trans. Data Privacy*, 3:123–148, August 2010.

[4] Reza Shokri, George Theodorakopoulos, Jean-Yves Le Boudec, and Jean-Pierre Hubaux. Quantifying location privacy. In *IEEE Symposium on Security and Privacy*, pages 247–262. IEEE Computer Society, 2011.

[5] Mohamed F. Mokbel and Walid G. Aref. Gpac: generic and progressive processing of mobile queries over mobile data. In Panos K. Chrysanthis and George Samaras, editors, *Mobile Data Management*, pages 155–163. ACM, 2005.

[6] Muntz et al. It roadmap to a geospatial future. Technical report, Committee on Intersections Between Geospatial Information and Information Technology, National Research Council, 2003.

[7] Mudhakar Srivatsa, Arun Iyengar, Jian Yin, and Ling Liu. A scalable method for access control in location-based broadcast services. In *INFOCOM*, pages 256–260. IEEE, 2008.

[8] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, pages 457–473, 2005.

[9] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM Conference on Computer and Communications Security, CCS*, pages 89–98, 2006.

[10] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy*, pages 321–334, 2007.

[11] ZP Jin, J. Xu, M. Xu, and N. Zheng. A location privacy preserving algorithm based on linkage protection. In *Information Theory and Information Security (ICITIS), 2010 IEEE International Conference on*, pages 190–194. IEEE, 2010.

[12] Byoungyoung Lee, Jinoh Oh, Hwanjo Yu, and Jong Kim. Protecting location privacy using location semantics. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1289–1297, 2011.

[13] John Krumm. Inference attacks on location tracks. In Anthony LaMarca, Marc Langheinrich, and Khai N. Truong, editors, *5th International Conference of Pervasive Computing, PERVASIVE*, volume 4480 of *Lecture Notes in Computer Science*, pages 127–143. Springer, 2007.

[14] Claudio Agostino Ardagna, Marco Cremonini, Sabrina De Capitani di Vimercati, and Pierangela Samarati. A privacy-aware access control system. *Journal of Computer Security*, 16(4):369–397, 2008.

[15] Sabrina De Capitani di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, and Pierangela Samarati. Over-encryption: Management of access control evolution on outsourced data. In *VLDB*, pages 123–134, 2007.

[16] Shucheng Yu, Cong Wang, Kui Ren, and Wenjing Lou. Achieving secure, scalable, and fine-grained data access control in cloud computing. In *INFOCOM*, pages 534–542. IEEE, 2010.

[17] Yan Zhu, Hongxin Hu, Gail-Joon Ahn, Mengyang Yu, and Hong-Jia Zhao. Comparison-based encryption for fine-grained access control in clouds. In Elisa Bertino and Ravi S. Sandhu, editors, *CODASPY*, pages 105–116. ACM, 2012.

[18] Matt Duckham and Lars Kulik. A formal model of obfuscation and negotiation for location privacy. In *3th International Conference of Pervasive Computing, PERVASIVE*, pages 152–170, 2005.