K-anonymous Location-based Fine-grained Access Control for Mobile Cloud

Yaser Baseri
Department of Computer Science and Operations Research
Universite de Montreal, Canada
Email: yaser.baseri@umontreal.ca

Abdelhakim Hafid
Department of Computer Science and Operations Research
Universite de Montreal, Canada
Email: ahafid@iro.umontreal.ca

Soumaya Cherkaoui
INTERLAB Research Laboratory
Universite de Sherbrooke, Canada
Email: Soumaya.cherkaoui@usherbrooke.ca

***Abstract*—Mobile cloud computing is a revolutionary computing paradigm for mobile application which enables storage and computation migration from mobile users to resources rich and powerful cloud servers, but emerges various privacy concerns. Attribute based encryption is a public key encryption that ensures the security of stored data in the cloud and provides fine grained access control using defined policies and constraints. Location of a device is one of the contextual policies which is used to improve data security, authenticate users and provide access to services and useful information for mobile users. However, unlike other policies and attributes used in attribute based encryption, location of mobile users are dynamic. In this paper, we investigate providing Location Based Services (LBS) for attribute based access control in mobile cloud. More specifically, we propose a multi-authority attribute based access control scheme and protect users privacy against malicious authorities. The proposed scheme uses dynamic location of a mobile user as contextual information about that user, employs coarse location as an attribute in attribute based encryption to achieve $K-$anonymity, and filters the returned results for more accuracy. The attribute based encryption is integrated with proxy re-encryption to outsource the computation to a cloud server with "unlimited" computational power. The proposed scheme achieves efficiency by reducing computational cost on resource-constrained mobile users.**

***Keywords:* Location Based Services, Dynamic Location, Key Revocation, Attribute Based Encryption, Outsourcing.**

I. Introduction

In some applications of mobile cloud computing, Location Based Services (LBS) are popular services provided by mobile devices and remote servers, in which users can use the geographical information for gaining access features (e.g. health, indoor object search, entertainment, work, personal life). LBS adopts Data as a Service (DaaS) model; it is accessible by mobile devices, through the mobile network, and makes use of the geographic positions of these devices.

In location based services, the location of a device represents one of the most important contextual information about a device and its owner; it is exploited to improve data security, and to support access to the services and information provided by the cloud for mobile users. Indeed, by integrating access control mechanisms with conditions based on the physical position of users, we can improve data security and immune it against unauthorized accesses and disclosures. Furthermore, in some applications, we need this information to provide convenient services for mobile users based on their positions (e.g. social networking as an entertainment service which uses information on the geographical position of the mobile device).

The main challenge of location based access control is the release of information only to authorized parties satisfying predefined conditions; this is called fine-grain access control. Attribute-based encryption (ABE) technique is a promising approach to achieve fine-grained access control [10], [3]. ABE provides access control over encrypted data using access policies and assigned set of attributes embedded in ciphertexts and private keys. In particular, ciphertext policy ABE (CP-ABE) provides access such that encrypted data can be decrypted only by a user possessing a set of attributes. Thus, based on policy embedded in ciphertext, different users are able to access different pieces of information based on the attributes they are assigned.

Providing fine grained access control for attribute based encryption requires issuing different attributes for each user. Since each authority issues a bunch of attributes for each user, the employed ABE (CP-ABE) should support coexistence of multi-authorities. Multi-authorities CP-ABE [6], [15], [4] is more appropriate for location-based access control for cloud, as users hold attributes issued by different authorities.

Using CP-ABE in the context of LBS introduces several challenges including (1) location anonymity: mobile users should not be traceable while using LBS; (2) dynamic location: location of mobile users are changing over time; CP-ABE should support dynamic update of location and key related to that location attribute; and (3) computational cost on mobile devices (users): the execution of the scheme should not impose high computational cost on mobile users with limited resources.

In this paper, we propose a new location based service scheme for attribute based access control in mobile cloud to support location privacy, confidentiality of stored data and dynamic location update without imposing significant computational cost on mobile devices.

A. Related Work

Only a few privacy preserving techniques have been proposed for location-based access control. In [1], the authors proposed a scheme based on the traditional access control in

978-1-4673-9292-1/16/$31.00 ©2016 IEEE

which the servers are trusted. The scheme uses onion encryption to increase the security of their scheme and decrease trust level on servers; it also adds an encryption layer to model the time. To provide fine-grained access control, the data owner should encrypt data for each user imposing high computational cost on their scheme. In [11], the authors used ciphertext policy anonymous attribute based encryption [7] to provide location privacy, confidentiality of location based service data and defined access policy. They assumed unlimited computational capacity for cloud server and imposed high computational overhead on the server (exhaustive search on key space corresponding to a given location range). Moreover, in [11], the location of a device is declared by that device, while a malicious user may cheat the location to get more services. In [18], the authors proposed a scheme, based on comparison based encryption[17], to construct a special-temporal predicate based encryption by means of secure integer comparison. Although the authors globally reduced the computational cost, the mobile user on the base comparison based encryption still does some bilinear pairing. Thus, their work still imposes high level of computational load on mobile users. Moreover, if the coarse location is not sufficiently dense, the scheme will not support required level of anonymity. Finally, since the location of a device is declared by that device, a malicious user can cheat the location and get more services. None of the existing schemes supports dynamic location update for mobile users. To conclude, we can summarize the limitations of existing schemes as follows: (1) high computation overhead on mobile users [1], [18]; (2) declaring fake location by malicious user and getting ineligible access to services and information [11], [18]; and (3) breaking the location privacy of user when the coarse location is not sufficiently dense [18].

In this paper, we propose a scheme that supports (1) low computation overhead on mobile users by outsourcing the heavy computations from mobile users (with restricted computational capabilities) to the server (with "unlimited" computational power); (2) efficient dynamic location updating of mobile user in without changing the entire private key of that user; (3) efficient and anonymous location based services for mobile users of cloud storage; and (4) protecting the identity of users against each single authority and even against compromising up to $(\mathcal{K}-2)$ out of $\mathcal{K}$ authorities. Note that, to provide location privacy, we use comparison based encryption [14] as a kind of CP-ABE; it models interval as an estimation of exact location and proposes a way to hide the exact location of users from cloud server. To provide anonymity of users, we also adapt *AnonyControl* [5] in the scheme and propose a way to protect the identity of users against each authority.

The rest of this paper is organized as follows. Section II discusses the system and security models of the proposed scheme. Section III, presents some preliminaries. Section IV describes the proposed scheme. Section V presents the analysis and evaluation of the proposed scheme. Finally, Section VI concludes the paper and presents future work.

II. System and Security Models

In this section, we first present the system model and its architecture. Then, we describe the security assumptions about different entities in that architecture.

A. System Model

In this paper, we propose a new system model for mobile cloud computing (see Figure 1), which introduces *Anonymizer* to preserve location privacy in location based access control. We define a system, which has at least three entities in its architecture: *Anonymizer*, *Location Service Provider* (*LSP*) and *Cloud Service Provider* (*CSP*). *Anonymizer* defines grid cells or cloaking areas and broadcasts them in predefined time intervals. *LSP* issues contextual attributes for each user including unforgeable exact location information l for that user, expanded location to a cloaking area (as an estimation of exact location) and time of access. This expansion is performed by mapping location points to intervals and changing the two-dimensional coordinate points to a grid cell using comparison based encryption defined in [14]. User sends its query including the expanded location and time of access provided by *LSP* to *Anonymizer*. Upon receiving *K* requests for a grid cell, *Anonymizer* generates an *Anonymizing Spatial Region* (*ASR*), a grid cell which contains at least *K* users and satisfies anonymity requirement, and performs *K*-anonymity cloaking to protect location privacy of the user. It sends the clustered requests of an *ASR* to *CSP*. *CSP* defines real-time access time. Then, it partially decrypts data based on that time and other access policies to outsource computation cost of decryption. Finally, it generates responses to these *K* queries and sends them back to *Anonymizer*. *Anonymizer* filters and sends the responses back to users. Finally, the authorized users will be able to decrypt the received data. In this way, we provide K-anonymity for users and security for queries.

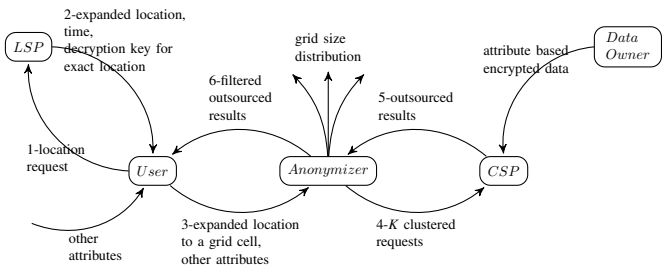

Figure 1. Architecture of the system

B. Threat Model

In the threat model used in LBS, *CSP*s are assumed to be honest but curious in practice [16]. That means that *CSP*s will faithfully follow the proposed scheme, but can launch passive attacks to get as much secret information as possible. Hence, the data stored in the cloud should remain encrypted all the time and any required transformation should

not reveal the plaintext in the process. Users who want to receive LBS, while keeping their location information secret, may be malicious, forge their real locations and collude to escalate access rights to get services not entitled to. Attribute authorities $AA_i (1 \leq i \leq \mathcal{K})$ are assumed to be semi-honest in the sense that they will not collude all together and the system can tolerate compromising at most $(\mathcal{K}-2)$ of them. Each AA_i is in charge of a subset of the whole attribute set and for each attribute that is in charge of, it knows the exact information of the key requester. Hence, by aggregating these information from all authorities, the complete attribute set of the key requester is recovered and thus his identity will be disclosed to all authorities. *LSP*, which provides location access right for each user, knows *ASR* and location information for each user, is assumed to be honest. *Anonymizer* is responsible for defining cloaking areas, collecting all messages as an intermediate tier between user and *CSP* and constructing *ASR*s. Hence, it may become a target for adversary, and may reveal the cloaking procedure. We assume, as in [9], [8], that *Anonymizer* will not collude with other entities. We also assume that the communication channels are secure and packets are untraceable when queries and information are transmitted on these channels. This assumption can be realized using Secure Socket Layer (*SSL*) or some other techniques [13], [12].

III. Preliminaries

In this section, we briefly introduce composite order bilinear group. Then, we present Multi-Dimensional Range Derivation Functions (MDRDF).

A. Composite Order Bilinear Map

Definition 1 (Composite Order Bilinear Groups)**.** Let p and q be two large primes, $N = pq$ be the RSA modulus, s_1, s_2, p', q', p, q be secret large primes, $s = s_1 s_2$, $n' = p'q'$, $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic bilinear groups of composite order $n = sn'$, α and β be two random exponents in $\mathbb{Z}$, and $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be a bilinear map with the following properties:

- Bilinearity: $\forall g_0, g_1 \in \mathbb{G} : e(g_0^{\alpha}, g_1^{\beta}) = e(g_0, g_1)^{\alpha\beta}$.
- Non-degeneracy: $e(g_0, g_1) \neq 1$.
- Computability: $\forall g_0, g_1 \in \mathbb{G}$, there is an efficient algorithm to compute $e(g_0, g_1)$.

where N is public and n, s, p', q' are secret. We refer to the tuple $\mathbb{S} = (N = pq, \mathbb{G}, \mathbb{G}_T, e)$ as a composite order bilinear group. Not that for two subgroups $\mathbb{G}_s$ and $\mathbb{G}_{n'}$ of order s and $n' = p'q'$ in $\mathbb{G}$, if $g \in \mathbb{G}_s$ and $h \in \mathbb{G}_{n'}$, then $e(g, h)$ is the identity element in $\mathbb{G}_T$.

B. Multi-Dimensional Range Derivation Functions

The idea of multi-dimensional derivation functions is using "one-way" property, to represent the total ordering relation of integers; this means that for two upper and lower bound integer values $(t_{i,j}, t_{i,k})$ and $(t'_{i,j}, t'_{i,k})$, if we know the value of $v_{\{t_{i,j}, t_{i,k}\}}$, and if $t_{i,j} \leq t'_{i,j}$ and $t_{i,k} \geq t'_{i,k}$, then, it is easy to compute $v_{\{t'_{i,j}, t'_{i,k}\}}$ from $v_{\{t_{i,j}, t_{i,k}\}}$, while the reverse is hard [14].

Let $\mathbb{G}_{n'}$ be a multiplicative group of composite order $n' = p'q'$, φ be a random generator in group $\mathbb{G}_{n'}$, where $\varphi^{n'} = 1$, $\{\lambda_i, \mu_i\}_{A_i \in \mathbb{A}}$ be the set of large random elements λ_i and μ_i in $\mathbb{Z}^*_{n'}$, which are relatively prime to other elements in $\{\lambda_i, \mu_i\}_{A_i \in \mathbb{A}}$, $U = \{t_{i,j}, t_{i,k}\}_{A_i \in \mathbb{A}}$ be the set of all upper and lower bounds for each attribute $A_i \in \mathbb{A}$, $\psi : U \rightarrow V$ be an order-preserving cryptographic mapping of U to a set of cryptographic values V of the form of $v_{\{t_{i,j}, t_{i,k}\}_{A_i \in \mathbb{A}}}$ (a cryptographical value reflecting the integer values of range bounds over each attribute $A_i \in \mathbb{A}$) and Z be a maximum integer value that an element in U can have. Then, we define the mapping function $\psi(.)$ to map the integer set U into V as follows:

$$\begin{aligned} v_{\{t_{i,j}, t_{i,k}\}_{A_i \in \mathbb{A}}} &\leftarrow \psi(\{t_{i,j}, t_{i,k}\}_{A_i \in \mathbb{A}}) \\ &= \varphi^{\prod_{A_i \in \mathbb{A}} \lambda_i^{t_{i,j}} \mu^{Z - t_{i,k}}} \in \mathbb{G}_{n'} \end{aligned}$$

Accordingly, multi-dimensional range derivation function is defined as follows:

Definition 2 (Multi-Dimensional Range Derivation Function[14])**.** A function $F : V \rightarrow V$ based on set U is defined as a multi-dimension range derivation function if it satisfies the following conditions:

- Easy to compute: the function F can be computed in a polynomial-time, i.e. if $t_{i,j} \leq t'_{i,j}$ and $t_{i,k} \geq t'_{i,k}$, $\forall A_i \in \mathbb{A}$, then $v_{\{t'_{i,j}, t'_{i,k}\} \forall A_i \in \mathbb{A}} = F_{\{t_{i,j} \leq t'_{i,j}, t_{i,k} \geq t'_{i,k}\} \forall A_i \in \mathbb{A}}(v_{\{t_{i,j}, t_{i,k}\} \forall A_i \in \mathbb{A}})$;
- Hard to invert: it is infeasible for any probabilistic polynomial (PPT) algorithm to compute $v_{\{t'_{i,j}, t'_{i,k}\}}$ from $v_{\{t_{i,j}, t_{i,k}\}}$ if $t_{i,j} > t'_{i,j}$ or $t_{i,k} < t'_{i,k}$.

Specifically, $F(.)$ can be expressed as follows:

$$\begin{aligned} v_{\{t'_{i,j}, t'_{i,k}\}} &\leftarrow F_{\{t_{i,j} \leq t'_{i,j}, t_{i,k} \geq t'_{i,k}\}}(v_{\{t_{i,j}, t_{i,k}\}}) \\ &= (v_{\{t_{i,j}, t_{i,k}\}})^{\prod \lambda_i^{t'_{i,j} - t_{i,j}} \mu_i^{t_{i,k} - t'_{i,k}}} \\ &= (\varphi^{\prod \lambda_i^{t_{i,j}} \mu^{Z - t_{i,k}}})^{\prod \lambda_i^{t'_{i,j} - t_{i,j}} \mu_i^{t_{i,k} - t'_{i,k}}} \\ &= \varphi^{\prod \lambda_i^{t'_{i,j}} \mu^{Z - t'_{i,k}}} \in \mathbb{G}_{n'}. \end{aligned}$$

IV. Proposed Scheme

There are five entities in the scheme: $\mathcal{K}$ *Attribute Authorities* (AA_i) including *LSP*, *User* (*U*), *Anonymizer*, *Cloud Service Provider* (*CSP*) and *Data Owner* (*DO*). The scheme consists of five phases: setup, key generation, encryption, access request and cloaking, and decryption.

A. Setup Phase

In the setup phase, which is performed by the central Trust Authority (*TA*), some parameters are fixed. It is assumed that the public keys corresponding to attribute authorities $AA_i (1 \leq i \leq K)$ are certified by *TA*, i.e. each authenticated participant

should be able to provide its digital certificate if asked. The setup algorithm consists of three steps.

Step 1. central trust authority *TA*

- Chooses a bilinear map system $S = (N = pq, \mathbb{G}, \mathbb{G}_T, e(.,.))$ of composite order $n = sn'$.
- Chooses two subgroups $\mathbb{G}_s$ with order s and $\mathbb{G}_{n'}$ with composite order $n' = p'q'$ of $\mathbb{G}$, where p' and q' are two large prime numbers.
- Selects random generators $\omega \in \mathbb{G}$, $g \in \mathbb{G}_s$ and $\varphi \in \mathbb{G}_{n'}$ such that there exists $e(g, \varphi) = 1$ but $e(g, \omega) \neq 1$.
- Selects public hash functions $H : \{0,1\}^* \rightarrow \mathbb{G}$ to map each binary attribute string into a group element in $\mathbb{G}$.
- Chooses $\lambda_i, \mu_i \in \mathbb{Z}_n^*$ for each attribute loc_i (i-th dimension of location $1 \leq i \leq m$) in location attribute set $\mathbb{A}$ and ensures that λ_i, μ_i are relatively prime to all other elements in $\{\lambda_i, \mu_i\}_{loc_i \in \mathbb{A}}$.
- Chooses random exponents $\alpha, \beta \in \mathbb{Z}_n^*$ and generates master key $MK = (p, q, n', \alpha, \beta)$.
- Publishes the global parameters $GP = (S, g, \omega, h = \omega^\beta, \eta = g^{1/\beta}, e(g,\omega)^\alpha, \varphi, \{\lambda_i, \mu_i\}_{l_i \in Loc}, H(.))$.

Step 2. Attribute authority AA_k

- Selects randomly $\mathcal{K} - 1$ integers $s_{kj} \in \mathbb{Z}_n^* (j \in \{1, ..., \mathcal{K}\} \setminus \{k\})$ and computes $g^{s_{kj}}$ to share with each other authority AA_j.
- Receives $\mathcal{K} - 1$ pieces of $g^{s_{jk}}$ generated by AA_j.
- Computes secret parameter $x_k \in \mathbb{Z}_n^*$ as follows:

$$x_k = \Big(\prod_{j \in \{1,...,\mathcal{K}\} \setminus \{k\}} g^{s_{kj}} \Big) / \Big(\prod_{j \in \{1,...,\mathcal{K}\} \setminus \{k\}} g^{s_{jk}} \Big) = g^{\Big(\sum_{j \in \{1,...,\mathcal{K}\} \setminus \{k\}} g^{s_{kj}} - \sum_{j \in \{1,...,\mathcal{K}\} \setminus \{k\}} g^{s_{jk}} \Big)}$$

It is clear that these randomly produced integers satisfy $\prod_{k \in \{1,...,\mathcal{K}\}} x_k = 1 \ mod \ n$.

B. Key Generation Phase

When a new user *U* wants to access the system, he requests from each authority to issue a secret key. This is performed in two sub-phases:

1) Attribute Key Generation: Attribute authorities $AA_i (1 \leq i \leq K$, including ***LSP***) collaborate to issue secret keys for each user, just one time during registration. This process consists of 2 steps.

step 1. Attribute authority AA_k

- Selects a random number $\gamma_k \in \mathbb{Z}_n^*$, computes $x_k g^{\gamma_k}$ and shares it with other authorities.
- Computes $D_r = \Pi x_i g^{\gamma_i} = g^{\Sigma \gamma_i} = g^r$ and $D = (g^\alpha * D_r)^{\frac{1}{\beta}} = g^{\frac{\alpha + r}{\beta}}$ (let $r = \Sigma \gamma_i$).
- Sends D to user U.

Step 2. Attribute authority AA_k ($AA_k \neq LSP$)

- Chooses a random number $r_j \in_r \mathbb{Z}_n^*$ for any attribute j.
- Computes $D_j = g^r (H(att(j)))^{r_j}$ and $D'_j = \omega^{r_j}$ and sends them to user U.

2) Location Key Generation: It is performed by ***LSP*** to issue dynamic contextual attributes including expanded location, time of access and an unforgeable exact location information l, while requesting to access the data stored on the cloud server. This is done in one step.

Step 1. The location service provider *LSP*

- Constructs location range $L_U = \{[l_{i,a}, l_{i,b}]\}_{loc_i \in \mathbb{A}}$ for user U ($loc = (loc_1 \parallel ... \parallel loc_m)$, where loc_i is the i-th dimension of location defined by *Anonymizer* to construct a grid cell in m dimensions.).
- Chooses two random numbers $r_{loc} \in_r \mathbb{Z}_n^*$ and $r_{time} \in_r \mathbb{Z}_n^*$.
- Computes $D_{loc} = g^r (H(loc))^{r_{loc}}$, $D'_{loc} = \omega^{r_{loc}}$, $D_{time} = g^r (H(time))^{r_{time}}$, $D'_{time} = \omega^{r_{time}}$.
- Computes $DK_U = (v_{L_U})^{r_{loc}} = \varphi^{r_{loc} \prod_{loc_i \in \mathbb{A}} \lambda_i^{l_{i,a}} \mu_i^{z - l_{i,a}}}$ as the delegation key of user U, where $v_{L_U} = v_{\{[l_{i,a}, l_{i,b}]\}_{loc_i \in loc}} = \varphi^{\prod_{loc_i \in \mathbb{A}} \lambda_i^{l_{i,a}} \mu_i^{z - l_{i,a}}} \in G_{n'}$.
- Sends D_{loc}, D'_{loc}, D_{time}, D'_{time}, DK_U and exact location information l to user U.

After receiving D, D_j, D'_j ($\forall j \in$ main attributes), D_{loc}, D'_{loc}, D_{time}, and D'_{time}, user U aggregates all as his private key: $SK_U = (D, D_{loc}, D'_{loc}, D_{time}, D'_{time}, \forall j \in main \ attributes \ D_j, D'_j)$.

C. Encryption Phase

During the encryption phase, the data owner *DO* should interact with *CSP* to define dynamic access policy and encrypt data based on that policy. This phase consists of two rounds: The first round is performed while uploading the file to the cloud and the second round is performed while receiving access request by *CSP*.

1) Data Uploading (First Round Encryption): This round is performed by *DO* and *CSP* while uploading information to the server. It consists of 3 steps.

Step 1. The data owner *DO*

- Defines access control policy for all attributes. More especially, DO defines location constraints $L_P = \{[\rho_i, \bar{\rho}_i]\}_{loc_i \in \mathbb{A}}$, where loc_i is the i-th dimension of those constraints. Note that $[\rho_i, \bar{\rho}_i]$ corresponds to attribute constraint $[t_{i,j}, t_{i,k}]$, if the policy does not designate negative attributes or wildcards over loc_i. For negative attributes or wildcards, the reader can refer to [14].
- Computes $v_{L_P} = v_{\{\rho_i, \bar{\rho}_i\}_{loc_i \in \mathbb{A}}} = \varphi^{\prod_{loc_i \in \mathbb{A}} \lambda_i^{\rho_i} \mu_i^{z - \bar{\rho}_i}}$
- Selects random numbers $s_{loc} \in_r \mathbb{Z}_n^*$ and $s_{main} \in_r \mathbb{Z}_n^*$ for location and main attributes (i.e. all attributes, except contextual attributes).
- Computes $ek_{DO} = e(g, \omega)^{\alpha (s_{loc} + s_{main})}$.

Step 2. The cloud service provider *CSP*

- Selects random number $s_{time} \in_r \mathbb{Z}_n^*$, computes $ek_{CSP} = e(g, \omega)^{\alpha (s_{time})}$ and sends it to *DO*.

Step 3. The data owner *DO*

- Computes $ek = ek_{DO} * ek_{CSP} = e(g,\omega)^{\alpha(s_{loc}+s_{main}+s_{time})} = e(g,\omega)^{\alpha s}$, generates a random key ak to encrypt the target file (i.e., the file we want to encrypt and for which we define access control), and uses that session key ek and exact location information l as key to encrypt the random key ak with symmetric encryption $ENC_{ek,l}(.)$.
- Shares the secret s_{main} in the tree access structure T with root R as described in [2]. Indeed, it chooses a polynomial q_x for each node x in T, and sets $q_R(0) = s_{main}$ for the root node R and shares that secret in the tree access structure T. Note that the set of leaf nodes y assigned atomic attribute $att(y)$ in the set of main attributes.
- Uploads the initial ciphertext $CT_{init} = (ENC_{ek,l}(ak), C_{DO} = h^{(s_{main}+s_{loc})}, C_{loc} = (v_{L_P}\omega)^{s_{loc}}, C'_{loc} = (H(loc))^{s_{loc}}, \forall y \in main\ attributes\ C_y = \omega^{q_y(0)}, C'_y = (H(att(y))^{q_y(0)})$ to CSP.

2) Access Time Encryption (Second Round Encryption): This round is performed, upon receipt of access request by *CSP*, to set the current time and solve dynamic location update. This round is performed in one step.

Step 1. The cloud service provider *CSP*

- Computes $C_{time} = \omega^{s_{time}}, C'_{time} = (H(time))^{s_{time}}, C_{CSP} = h^{s_{time}}$ and $C = C_{CSP} * C_{DO} = h^{(s_{main}+s_{loc}+s_{time})} = h^s$.

The final ciphertext would be $CT = (ENC_{ek,l}(ak), C, C_{loc}, C'_{loc}, C_{time}, C'_{time}, \forall y \in main\ attributes\ C_y, C'_y)$.

D. Access Request and Cloaking Phase

During this phase, authorized user U sends his access request for the target file via *Anonymizer* to CSP. This phase consists of two steps.

Step 1. Access request: Upon receiving keys, user *U*

- Chooses a random number $b \in \mathbb{Z}_n^*$, and raises all components of SK_U and DK_U to the power $1/b$ (i.e., $SK_U^{1/b} = (D^{1/b}, D_{loc}^{1/b}, D_{loc}^{'1/b}, D_{time}^{1/b}, D_{time}^{'1/b}, \forall j \in main\ attributes\ D_j^{1/b}, D_j^{'1/b})$ and $DK_U^{1/b} = (v_{L_U})^{r_{loc}/b}$).
- Sends his request to access the target file, $SK_U^{1/b}$, $DK_U^{1/b}$ and his own location range L_U (corresponding to a grid cell) to *Anonymizer*.

Step 2. Cloaking: Upon receiving *K* requests for a grid cell, *Anonymizer*

- Removes identifiers of users, generates an *ASR* corresponding to the location range, clusters requests and performs *K*-anonymity cloaking to protect location privacy of the users.
- Sends the clustered requests of an *ASR* to *CSP*.

E. Decryption Phase

When the access request is received by CSP, the eligibility of user to access the target file should be checked. This process is performed in three sub-phases: decryption delegation, decryption and data access.

1) Decryption Delegation: This sub-phase is done by CSP to compute blind delegation key corresponding to location privilege L_P in one step.

Step 1. Upon receiving a request, *CSP*

- Checks whether user location range L_U satisfies location privilege L_P over all location dimensions.
- Computes $(v_{L_P})^{r_{loc}/b}$ from $(v_{L_U})^{r_{loc}/b}$ as follows:

$$\begin{aligned}(v_{L_P})^{r_{loc}/b} &= (v_{\{\rho_i,\bar{\rho}_i\}_{loc_i\in\mathbb{A}}})^{r_{loc}/b} \\ &= F_{\{t_{i,a}\le\rho_i, t_{i,b}\ge\bar{\rho}_i\}_{loc_i\in\mathbb{A}}}((v_{L_U})^{r_{loc}}) \\ &= F_{\{t_{i,a}\le\rho_i, t_{i,b}\ge\bar{\rho}_i\}_{loc_i\in\mathbb{A}}}((v_{\{t_{i,a},t_{i,b}\}_{loc_i\in\mathbb{A}}})^{r_{loc}}) \\ &= (\varphi^{\prod_{loc_i\in\mathbb{A}}\lambda_i^{\rho_i}\mu_i^{z-\bar{\rho}_i}})^{r_{loc}/b} \in G_{n'}.\end{aligned}$$

2) Decryption: This sub-phase is performed by CSP to compute blind session key $ek^{1/b}$ and transfer it to the user in just one step.

Step 1. The cloud service provider *CSP*

- Computes $Dec_{time} = \frac{e(D_{time}^{1/b}, C_{time})}{e(D_{time}^{'1/b}, C'_{time})} = e(g,\omega)^{(rs_{time})/b}$ for contextual attribute $time$
- Computes $Dec_{loc} = \frac{e(D_{loc}^{1/b}, C_{loc})}{e((v_{L_P})^{r_{loc}/b} D_{loc}^{'1/b}, C'_{loc})} = e(g,\omega)^{(rs_{loc})/b}$ for contextual attribute loc.
- Computes $DecNode_y = \frac{e(D_y^{1/b}, C_y)}{e(D_y^{'1/b}, C'_y)} = e(g,\omega)^{(rq_y(0))/b}$ for each main attribute y. Then, it recursively computes $Dec_{main} = e(g,\omega)^{(rq_R(0))/b} = e(g,\omega)^{(rs_{main})/b}$ as described in [2].
- Computes $A = Dec_{time} * Dec_{loc} * Dec_{main} = e(g,\omega)^{r(s_{time}+s_{loc}+s_{main})/b} = e(g,\omega)^{(rs)/b}$.
- Computes $ek^{1/b} = \frac{e(C, D^{1/b})}{A} = \frac{e(g,\omega)^{(\alpha+r)s/b}}{e(g,\omega)^{(rs)/b}} = e(g,\omega)^{(\alpha s)/b}$ and sends it to *Anonymizer*.

3) Data Access: This sub-phase is performed by *Anonymizer* and user to find the symmetric key ak and access the file encrypted by that symmetric key. It consists of 2 steps.

Step 1. Upon receiving the responses from CSP, *Anonymizer* filters the responses to send them back to their related users.

Step 2. Each user U

- Computes the session key ek by raising $ek^{1/b}$ to the power b received in key generation phase.
- Decrypts $ENC_{ek,l}(ak)$ and computes ak, using its own exact location information l received in key generation phase and computed session key ek.
- Decrypts the file using key ak.

V. Analysis and Evaluation Discussion

The security analysis discusses and provides proofs on how the proposed scheme supports location privacy and is immune against authorities collision attacks, user collision attacks and chosen plaintext attacks. Due to space constraints, the analysis

	$User$		DO		LSP	
	Computation Cost	Communication Cost	Computation Cost	Communication Cost	Computation Cost	Communication Cost
Setup	0	0	0	0	(2K-2)(T_M+T_E)	(2K-2)$l_{\mathbf{G}_{n'}}$
Key Gen. (dynamic)	0	0	0	0	(K+4)T_M+(17 +2m)T_E	(K+7)$l_{\mathbf{G}_{n'}}$
Encryption	0	0	$\mathcal{O}(\mathcal{T})T_P$	$l_{ENC(ak)}$+(3+2$\mathcal{T}$)$l_{\mathbf{G}_{n'}}$	0	0
Acc. Req. & Cloaking	(6+2$\mathcal{T}$)T_E	(6+2$\mathcal{T}$)$l_{\mathbf{G}_{n'}}$	0	0	0	0
Decryption	T_E+$T_{ENC(ak)}$	$l_{\mathbf{G}_{n'}}$ + $l_{ENC(ak)}$	0	0	0	0

Table I

COMPUTATION AND COMMUNICATION COST ON DIFFERENT PARTS IN THE PROPOSED SCHEME ($\mathcal{T}$: NUMBER OF LEAVES IN THE ACCESS TREE, T_M: TIME FOR MULTIPLICATION, T_E: TIME FOR EXPONENTIATION, AND T_P: TIME FOR PAIRING)

is not included in the paper; it will be made available for interested readers and will be included in an extended version of this paper to be submitted soon.

In this section, we analyze the computation and communication cost of the proposed scheme on parties involved in the system. The analysis concerns the most significant computations, in the scheme, namely multiplication (M), exponentiation (E), and pairing (P). Let us remember that CSP has unlimited computational power, while mobile users have limited computation and communication resources. Hence, the aim of the system is to reduce the communication and computation cost on mobile users. Table I shows that the computation and communication cost on mobile users is minimum which makes the proposed scheme suitable for mobile devices. Moreover, each attribute authority AA_i (except LSP) is involved in the system just one time during the registration. Hence, it does not have impact on the scheme efficiency. However, LBS assigns dynamic attributes location and time for each user anytime the user wants to access the system. Hence, we just consider communication and communication cost for LBS in Table I. *Anonymizer* does not have any role except an intermediary which gathers, clusters and delivers K clustered requests to CSP, and returns back the responses to intended users. Hence, it does not have any communication and computation overhead on the system.

VI. CONCLUSION

In this paper, we presented location based service for attribute based access control in mobile cloud. The proposed scheme supports dynamic location for mobile devices, and minimizes the computation and communication overhead on these devices with limited resources. It investigates providing K-anonymous location based services for mobile users and supporting multi-authorities in a way that privacy of each user is protected against authorities and CSP. In our future work, we will design and run experiments to evaluate the performance of our work in real environments.

REFERENCES

[1] E. Androulaki, C. Soriente, L. Malisa, and S. Capkun. Enforcing location and time-based access control on cloud-stored data. In *Distributed Computing Systems (ICDCS), 2014 IEEE 34th International Conference on*, pages 637–648, June 2014.

[2] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *Security and Privacy, 2007. SP'07. IEEE Symposium on*, pages 321–334. IEEE, 2007.

[3] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 89–98. Acm, 2006.

[4] Taeho Jung, Xiang-Yang Li, Zhiguo Wan, and Meng Wan. Control cloud data access privilege and anonymity with fully anonymous attribute-based encryption. *Information Forensics and Security, IEEE Transactions on*, 10(1):190–199, Jan 2015.

[5] Taeho Jung, Xiang-Yang Li, Zhiguo Wan, and Meng Wan. Control cloud data access privilege and anonymity with fully anonymous attribute-based encryption. *Information Forensics and Security, IEEE Transactions on*, 10(1):190–199, 2015.

[6] Allison Lewko and Brent Waters. Decentralizing attribute-based encryption. In *Advances in Cryptology–EUROCRYPT 2011*, pages 568–588. Springer, 2011.

[7] Jin Li, Kui Ren, Bo Zhu, and Zhiguo Wan. Privacy-aware attribute-based encryption with user accountability. In *Information Security*, pages 347–362. Springer, 2009.

[8] Qin Liu, Chiu C Tan, Jie Wu, and Guojun Wang. Cooperative private searching in clouds. *Journal of Parallel and Distributed Computing*, 72(8):1019–1031, 2012.

[9] T. Peng, Q. Liu, and G. Wang. Enhanced location privacy preserving scheme in location-based services. *Systems Journal, IEEE*, PP(99):1–12, 2014.

[10] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *Advances in Cryptology–EUROCRYPT 2005*, pages 457–473. Springer, 2005.

[11] Jun Shao, Rongxing Lu, and Xiaodong Lin. Fine: A fine-grained privacy-preserving location-based service framework for mobile devices. In *INFOCOM, 2014 Proceedings IEEE*, pages 244–252. IEEE, 2014.

[12] Guojun Wang, Qiushuang Du, Wei Zhou, and Qin Liu. A scalable encryption scheme for multi-privileged group communications. *The Journal of Supercomputing*, 64(3):1075–1091, 2013.

[13] Guojun Wang, Fengshun Yue, and Qin Liu. A secure self-destructing scheme for electronic data. *Journal of Computer and System Sciences*, 79(2):279–290, 2013.

[14] Z. Wang, D. Huang, Y. Zhu, B. Li, and C. Chung. Efficient attribute-based comparable data access control. *Computers, IEEE Transactions on*, PP(99):1–1, 2015.

[15] Kan Yang and Xiaohua Jia. Expressive, efficient, and revocable data access control for multi-authority cloud storage. *Parallel and Distributed Systems, IEEE Transactions on*, 25(7):1735–1744, 2014.

[16] Shucheng Yu, Cong Wang, Kui Ren, and Wenjing Lou. Achieving secure, scalable, and fine-grained data access control in cloud computing. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. Ieee, 2010.

[17] Yan Zhu, Hongxin Hu, Gail-Joon Ahn, Mengyang Yu, and Hongjia Zhao. Comparison-based encryption for fine-grained access control in clouds. In *Proceedings of the second ACM conference on Data and Application Security and Privacy*, pages 105–116. ACM, 2012.

[18] Yan Zhu, Di Ma, Dijiang Huang, and Changjun Hu. Enabling secure location-based services in mobile cloud computing. In *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing*, pages 27–32. ACM, 2013.