LSD-ABAC: Lightweight Static and Dynamic Attributes Based Access Control Scheme for Secure Data Access in Mobile Environment

Fei Li, Yogachandran Rahulamathavan, and Muttukrishnan Rajarajan
School of Engineering and Mathematical Science
City University London, London, United Kingdom
Email: {Fei.li.1, Yogachandran.Rahulamathavan.1, and R.Muttukrishnan}@city.ac.uk

***Abstract*—Technology advancements in smart mobile devices empower mobile users by enhancing mobility, customizability and adaptability of computing environments. Mobile devices are now intelligent enough to capture dynamic attributes such as unlock failures, application usage, location and proximity of devices in and around its surrounding environment. Different users will have different set of values for these dynamic attributes. In traditional attribute based access control, users are authenticated to access restricted data using long term static attributes such as password, roles, and physical location. In this paper, in order to allow secure data access in mobile environment, we securely combine both the dynamic and static attributes and develop novel access control technique. Security and performance analyse show that the proposed scheme substantially reduces the computational complexity while enhances the security compare to the conventional schemes.**

***Index Terms*—Access control, dynamic attributes, attribute-based encryption, multi-authority.**

I. Introduction

Bring your own devices (BYOD) is a trend for today's organizations and enterprises. One of the remarkable outcomes of Cisco survey conducted in the US in 2012 says that 95% of the survey participants are allowed to use their mobile devices within their organizations [1]. More and more staffs prefer to bring their own mobile devices such as tablets, smartphones and laptop computers to their work place. It is against the tradition where employees are allocated with company devices embedded with specific softwares and policies to achieve security. As a result, IT department requires more flexible and creative solutions to maintain the security and privacy in the collaborative environments [15]. Mobility enables users to create new ideas – they want to work from home or the office using social networks and cloud services to get the job done with a seamless solution [2].

On the other hand, advances in cloud computing and outsourcing enabled flexible computing capabilities at reduced costs and capital expenditures. Security vulnerabilities associated with this new paradigm cannot be satisfied with traditional access control techniques. Traditionally, we assume that data owners, users, and storage server are in the same domain and also that the server is fully trusted [3]. However, in cloud computing and outsourcing environments, data confidentiality is not guaranteed since the data is stored and processed within the third party environment. Personnel information of the data owners and commercial interests of users can be leaked to third party if the data owners store decrypted data in public servers. Hence, achieving the data confidentiality in a distributed environment is challenging and attribute based encryption (ABE) technique has been proposed as a plausible solution [6]–[8], [11].

ABE is one of the most promising cryptographic techniques. Using ABE, the data owners can enforce fine-grained access policies based on nature of the data. For instance, let us assume, an employer uploads encrypted file to the cloud using ABE, where access policy of that file is defined using the following attributes and functions AND and OR: "Manager" OR "Finance Office" AND "Company A". Hence, an employee who is a "Manager" employed at "Company A" can decrypt the file.

In BYOD case, the conventional ABE schemes are not enough to protect the data due to the mobility of user device. If you consider the previous example, the manager can access the file while he is traveling in train. The risk level associated in the example is high and the manager may not aware of the surrounding environment. Or someone may steal manager's mobile device and try to access the file illegally. However, smart devices have the capability to learn the dynamic attributes in and around the devices, hence, change in user behavior can be detected easily [4]. Hence, it is crucial to include dynamic attributes in ABE in order to enhance security by exploiting features of smart mobile devices, which will leads to seamless solution. However, conventional ABE developed based only on static attributes.

In this paper, we propose new algorithm which support data owner to incorporate the dynamic attributes within ABE scheme. The contribution of this paper is two folds:

C1. First we will develop an algorithm to securely incorporate dynamic and static attributes within ABE (Algorithm 1)

C2. Then we exploit a semi-trusted cloud server to outsource computational and communication costs associated with the user (Algorithm 2)

We demonstrate that the performance of the Algorithm 1 is comparable with the conventional ABE scheme in terms of computational and communication costs. However, the Algorithm 2 improves the computational and communication costs substantially compare to Algorithm 1 since we incorporate the semi-trusted cloud server. It should be noted that both the proposed algorithms add additional layer of security (i.e., inclusion of dynamic attributes) on top of the security of conventional ABE schemes (please note that the construction of conventional ABE is based on well-known identity based encryption scheme [6]).

The reminder of this paper is organized as follows: we discuss the related work in Section II and in Section III we securely incorporate dynamic and static attributes to the conventional ABE scheme. We introduce semi-trusted cloud server in Section IV to outsource communications and computational cost of algorithm developed in Section III. Security and performance analyse of proposed schemes are provided in Section V. Conclusions are drawn in Section VI.

II. Related Work

Let us discuss the pioneering works in attribute based access control literature. ABE was firstly proposed by Sahai and Waters [6], where they constructed an identity based encryption (IBE) of a message under several attributes that compose a fuzzy identity. There are two main types of ABE, the key-policy ABE which was proposed by Goyal et al. [7], and ciphertext-policy ABE which was proposed in [8]. Chase [18] presented a multi-authority ABE (MA-ABE) system, whereby any polynomial number of independent attribute authorities monitor attributes and distribute private-keys. Data owner can decide a number d_k and a set of attributes from an attribute authority, and encrypt a message such that only an user with minimum d_k attributes from the revelent attribute authority can decrypt the message.

Chase and Chow proposed another work [19] which improved the previous scheme [18]. In this work, the central authority was removed and anonymous key issuing protocol was developed to address privacy of the users. As a result, multiple attribute authorities cannot collaborate and pool the attributes by tracing the global identifier of the users. Lewko and Waters [20] proposed a fully decentralized ABE scheme, where user could have zero or more attributes from each attribute authority and do not require a trusted server. In their work, the attribute authority can join and leave freely without re-initializing the system.

Outsourcing the decryption of ABE ciphertexts was designed for a single authority in [5]. Since we consider multiple authorities and dynamic attributes, our work is different from [5].

Let us now discuss the works in context-related security mechanisms in the literature. Liao proposed a location data encryption using static locations [25]. In this work, each static location is a mobile node with pre-determined longitude and latitude coordinates. The concept of "Geoencryption" or "location-based encryption" was developed to use in digital film distribution by Logan Scott et al [24]. Omar et. al. presented a geoencryption protocol by restricting the decryption of a message to a particular location and time period [26]. The encryption in this work is similar to [25] where the locations were static which means they are pre-defined in the system. Vijayalakshmi and Palanivelu proposed a secure localization using elliptic curve cryptography (ECC) in wireless sensor networks, where determining the physical positions of sensors is a fundamental and crucial problem in the wireless sensor network operation [27]. Their location based authentication scheme was built on the ID-based cryptography using ECC and ECC key exchange. Karimi presented a geoencryption protocol which allows mobile nodes to communicate with each other by restriction when decoding a message in the specific location and time period [28]. It should be noted that these algorithms provide neither fine-grain access control nor data confidentiality. Moreover, none of the algorithms supports both the static and dynamic attributes together for robust access control.

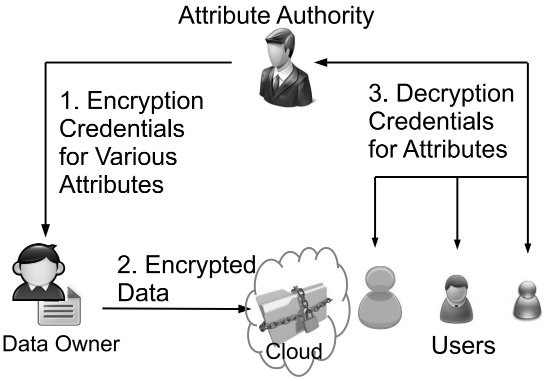
Attribute Authority
1. Encryption Credentials for Various Attributes
3. Decryption Credentials for Attributes
2. Encrypted Data
Data Owner
Cloud
Users

Fig. 1. Single authority ABE scheme.

In contrast to all of the works discussed earlier, the algorithms proposed in this paper combines both the dynamic attributes and static attributes within ABE. To the best of our knowledge, this is the first known results that enhances the secure data access in mobile environment. We present the cryptographic building blocks exploited in the new algorithms in the following sub-sections.

A. Bilinear Pairings

Let $\mathbb{G}_1, \mathbb{G}_2$ be two multiplicative groups of prime order q and let g_1 and g_2 be generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. Let us denote a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. The map has the following three properties.

1) Bilinearity: $\forall x \in \mathbb{G}_1, \forall y \in \mathbb{G}_2$, and $a, b \in \mathbb{Z}_q$, there is $\hat{e}(x^a, y^b) = \hat{e}(x, y)^{ab}$.
2) Non-degeneracy: For $\forall x \in \mathbb{G}_I, \forall y \in \mathbb{G}_2$, there is $\hat{e}(x, y) \neq 1$.
3) Computability: $\hat{e}$ is an efficient computation.

B. Attribute Based Encryption

ABE allows the data to be encrypted in such a way that the encrypted data can only be accessed by individuals who have the credentials for necessary attributes. In ABE scheme, trusted attribute authorities maintain encryption and decryption

Setup $\mathcal{S}$

- Collision-Resistant Hash Function (CRHF) $H : \{0,1\}^* \rightarrow \mathbb{Z}_q$. CRHF can be used to generate user identity u from the user global identity (GID).
- For a given security parameters λ and $\sigma \in \{0,1\}^{poly(\lambda)}$, group bilinear parameters are generated by the attribute authorities as follwos: $q, g_1, g_2, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T \leftarrow \mathcal{S}(1^\lambda; \sigma)$. Now, attribute authorities interact with each other and execute the following:

. k^{th} attribute authority randomly chooses $v_k \in_R \mathbb{Z}_q$ and computes $Y_k = \hat{e}(g_1, g_2)^{v_k}$, and sends Y_k to the other attribute authorities, where each attribute authority computes $Y = \prod Y_k = \hat{e}(g_1, g_2)^{\sum_k v_k}$.

. Each pair of attribute authorities shares a secret, k^{th} attribute authority and j^{th} attribute authority randomly choose $s_{kj} \in \mathbb{Z}_q$ such that $s_{kj} = s_{jk}$.

. k^{th} attribute authority randomly chooses $x_k \in \mathbb{Z}_q$ and computes $y_k = g_1^{x_k}$. Using the shared secret $s_{k,j}$ and u, attribute authorities k and j computes $y_k^{x_j/(s_{kj}+u)}$ and $y_j^{x_k/(s_{kj}+u)}$, respectively.

- k^{th} attribute authority randomly chooses a secret $t_{k,i} \in \mathbb{Z}_q$ for i^{th} attribute, and computes the corresponding public key as $T_{k,i} = g_2^{t_{k,i}}$ ($\forall i \in \{1, ..., N_k\}$ and $k \in \{1, ..., K\}$), where N_k is the number of attributes monitored by authority k.

Key Issuing $\mathcal{KG}$

User u executes the following steps with each authority k:

- For $j \in \{1, ..., K\} \,/\, \{k\}$, user gets the $D_{kj} = g_1^{R_{kj}} y_k^{x_j/(s_{kj}+u)}$ for $k > j$ or $D_{kj} = g_1^{R_{kj}} y_k^{(s_{kj}+u)/x_j}$ if $k < j$, where $R_{k,j} \in \mathbb{Z}_q$ is a random value.
- After obtains all D_{kj}, user computes D_u = $\prod_{(k,j)\in\{1,...,N\}\times\{1,...,N\}/\{k\}} D_{kj} = g_1^{R_u}$, where $R_u = \sum_{(k,j)\in\{1,...,K\}\times\{1,...,K\}/\{k\}} R_{kj}$.
- If user u satisfies d_k number of attributes, then k^{th} attribute authority randomly picks a d_k-degree polynomial $p_{k,u}$ with $p_{k,u}(0) = v_k - \sum_{j\in\{1,...,K\}/\{k\}} R_{kj}$.
- Authority k computes $S_{k,i} = g_1^{p_{k,u}(i)/t_{k,i}}$, $i \in [1, \ldots, N_k]$, $\forall k$.

Encryption $\mathcal{E}$

Data owner encrypts data m for attribute set $A_m = A_A^1 \bigcup A_A^2 \bigcup \ldots A_A^K \bigcup A_C$ as follows (i.e. A_A^k, $\forall$ k denotes the attribute set maintained by kth attribute authority):

* Data owner randomly picks $s_A, s_B \in_R \mathbb{Z}_q$ and encrypts the data as follows: $Enc_m = mY^{s_B}$.
* Data owner computes $E_0 = h(M(a_{a,1})||M(a_{a,2})|| \ldots ||M(a_{a,n}))Y^{s_A+s_B}$, $E_1 = g_2^{s_A}$, $\left\{C_{k,i} = T_{k,i}^{s_A}\right\}, i \in \mathbb{A}_A^k, \forall k \in [1, ..., N]$.
* Now data owner uploads CT_m = $\{Enc_m,\ E_0,\ E_1, C_{k,i} \forall i \in A_A$ and $A_C\}$ into the cloud.

Decryption $\mathcal{D}$

* User downloads CT_m from the cloud and checks the required attributes to decrypt m.
* For each authority k:
  * Using $S_{k,i}$ and the corresponding $C_{k,i}$, user computes $\hat{e}\left(S_{k,i}, C_{k,i}\right) = \hat{e}\left(g_1, g_2\right)^{s_A p_{k,u}(i)}$.
  * User interpolates all $\hat{e}\left(g_1, g_2\right)^{s_A p_{k,u}(i)}$ and gets $P_{k,u}$ = $\hat{e}\left(g_1, g_2\right)^{s_A p_{k,u}(0)} = \hat{e}\left(g_1, g_2\right)^{s_A\left(v_k - \sum_{j\neq k} R_{kj}\right)}$.
* User multiplies all $P_{k,u}$'s together and gets $Q = \hat{e}(g_1, g_2)^{s_A \sum v_k - s_A R_u}$ = $\frac{Y^{s_A}}{\hat{e}\left(g_1^{R_u}, g_2^{s_A}\right)}$.
* Now corporate app installed in users' mobile device computes $h(M(a'_{a,1})||M(a'_{a,2})|| \ldots ||M(a'_{a,n}))$.
* User can decrypt the data as follows (only if $h(M(a'_{a,1})||M(a'_{a,2})|| \ldots ||M(a'_{a,n})) = h(M(a_{a,1})||M(a_{a,2})|| \ldots ||M(a_{a,n})))$

$$Enc_m . \frac{h(M(a'_{a,1})||M(a'_{a,2})||\ldots||M(a'_{a,n}))Q\hat{e}(D_u, E_1)}{E_0} = mY^{s_B} . \frac{h(M(a'_{a,1})||M(a'_{a,2})||\ldots||M(a'_{a,n}))Y^{s_A}}{h(M(a_{a,1})||M(a_{a,2})||\ldots||M(a_{a,n}))Y^{s_A+s_B}} = m.$$

Fig. 2. Algorithm 1: SD-ABAC: Static and Dynamic Attributes Based Access Control Scheme for Secure Data Access in Mobile Environment.

credentials for various attributes. These attribute authorities verify the user attributes before releasing the corresponding credentials for the attributes. Data owner obtains the encryption credentials for a set of attributes from the attribute authority, and encrypts the data using those credentials. Once encryption is successful, the encrypted data can be uploaded into the cloud storage where any users with the decryption credentials will be able to decrypt the data. Fig. 1 shows how data owner, attribute authority and users interact with each other. We propose our new algorithms in the next sections.

III. SD-ABAC: Static and Dynamic Attribute Based Access Scheme

In a traditional ABE system, there is only one attribute authority that monitors all the attributes and issues encryption and decryption credentials for the users. This single authority becomes a fully trusted party to which the users have to prove their attributes in order to obtain the decryption credentials. In such a case, the attribute authority has too much power and it can decrypt all the data and knows all the users' attributes. In the event of corruption, the message confidentiality cannot be achieved and users' privacy can be obtained by the attackers. This is one of the drawbacks in single authority ABE scheme.

It is more convenient and secure to monitor and maintain different sets of attributes by different attribute authorities in reality, e.g., in healthcare one authority can monitor attributes of nurse and doctors while another authority monitors attributes of administrators and human resources [21], [22] or in vehicular adhoc network (VANET), different identities can be monitored by different authorities [9]. Hence, it is more convenient to have multiple attribute authorities where each attribute authority can maintain attributes belonging to one department.

MA-ABE scheme without incorporating the dynamic attributes was proposed in [19]. We develop our new algorithm (i.e., Algorithm 1) by securely incorporating the dynamic attributes within MA-ABE scheme in [19]. The proposed algorithm is composed of four sub-algorithms named as setup, key issuing, encryption and decryption. This new algorithm is presented in Fig. 2. Please note that the steps denoted as * in Fig. 2 are different from the conventional MA-ABE algorithm [19]. Let us briefly explain the functionalities of each sub algorithms below.

Setup: The setup algorithm takes security parameters λ and σ as input, and outputs a bilinear group and a set of parameters. Parameters $q, g_1, g_2, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are public parameters. Parameters v_k and x_k are the private-keys known only to k^{th} attribute authority and the corresponding public-keys are $Y_k = \hat{e}(g_1, g_2)^{v_k}$ and $y_k = g_1^{x_k}$ which are known to all. Two attribute authorities share a private-key s_{jk} which is known only to the two attribute authorities. Parameter $t_{k,i}$ denote the i^{th} attribute maintained by k^{th} attribute authority and the corresponding public-key is $T_{k,i} = g_2^{t_{k,i}}$.

Key Issuing: Data owner computes decryption credential D_{kj} for j^{th} attribute by collaborating with k^{th} attribute authority. Once user obtained all the D_{kj} then she can compute D_u followed by $S_{k,i}$.

Encryption: The encryption algorithm takes attributes maintained by the attribute authorities and attributes defined by the data owner as inputs. Then it output the ciphertext of the data.

Decryption: The decryption algorithm takes the decryption credentials received from attribute authorities and context-related parameters obtained from smart mobile device and the ciphertext as input and output the original data. The behaviour profiling app securely computes the hash value of the required dynamic attributes followed by multiplication with Y^{s_A}. The decryption is successful if and only if $h(M(a'_{a,1})||M(a'_{a,2})||\ldots||M(a'_{a,n})) = h(M(a_{a,1})||M(a_{a,2})||\ldots||M(a_{a,n}))$.

The novelty in our scheme compared to the conventional ABE scheme lies in the encryption and the decryption sub algorithms. Let us denote a set of dynamic attributes defined by the data owner as $A_C = \{a_{c,1}, \ldots, a_{c,n}\}$ where $a_{c,i}$ denotes the i^{th} dynamic attribute. For the sake of simplicity let us consider the following three dynamic attributes: $a_{c,1} =$"location", $a_{c,2} =$"risk level associated with his recent app usage" and $a_{c,3} =$"unlock failures in last two days". Now the data owner defines $A_C = \{a_{c,1} = \text{“}LONDON\text{”}$, $a_{c,2} <$ “3” and $a_{c,3} <$ “2” $\}$ and computes $E_0 = h(LONDON||yes||yes)Y^{s_A+s_B}$. Let us assume that the risk level varies between 1 to 10 where higher risk denoted by larger value. However, different organizations may define the risk level based on their own standards. For example, if a particular document is highly classified then, the organization sets high risk value for that document rather than the risk value of ordinary documents.

IV. LSD-ABAC: Low Complexity Static and Dynamic Attribute Based Access Scheme

In Fig. 2, we assumed that the users will use their mobile devices to access and decrypt the encrypted data. If we recall the key-issuing and decryption processes presented in Fig. 2, it is obvious that the user's mobile device requires to do computationally intensive operations such as bilinear pairing, exponentiation and multiplication. Hence, performing these computationally intensive operations reduce the user experience. Moreover in the key-issuing stage, user's mobile device need to interact with attribute authorities several times in order to retrieve the decryption credentials. However, the mobile data network may not be reliable for all communication and it can cause huge overheads and communication delays.

In our previous work [11], we introduced a semi-trusted authority to outsource the computational and communication cost associated with the users in MA-ABE scheme. In [11], substantial amount of communications and computational costs are outsourced to the semi-trusted authority without compromising the security and privacy of the MA-ABE scheme [19]. The semi-trusted authority interacts with the attribute authorities on behalf of the user and obtains the masked shared-decryption-keys. Later the semi-trusted authority combines all the keys and gets one masked-key which can only be unmasked by a user to decrypt the message. In particular, semi-trusted authority cannot decrypt the message nor determine the attributes of the mobile user, hence, the security and privacy of the proposed MA-ABE scheme is preserved. We are going to use similar semi-trusted authority in this paper to outsource the computational and communications costs associated with the user in Algorithm 1 (Fig. 2). The details of the low complexity static and dynamic MA-ABE based access control scheme is presented in Fig. 3 (Algorithm 2). We briefly explain the functionality of the scheme in the following.

Setup: The setup algorithm executes the same process as the proposed scheme in Fig. 2.

Key Issuing: The sub-algorithm incorporates semi-trusted authority. The semi-trusted authority executes several steps with each attribute authority on behalf of user u. Hence, communication and computational overheads during the key issuing process have been offloaded to cloud based semi-trusted authority.

- For authority $j \in \{1, ..., K\} \,/\, \{k\}$, semi-trusted authority gets $D_{kj} = g_1^{R_{kj}} y_k^{x_j/(s_{kj}+u)}$ for $k > j$ or $D_{kj} = g_1^{R_{kj}} y_k^{(s_{kj}+u)/x_j}$ if $k < j$.
- Semi-trusted authority combines all D_{kj} and computes $D_u = \prod_{(k,j)\in\{1,\dots,N\}\times\{1,\dots,N\}/\{k\}} D_{kj} = g_1^{R_u}$.
- If user u satisfies the required d_k number of attributes, then authority k randomly picks a d_k-degree polynomial $p_{k,u}$.
- Authority k defines $p_{k,u}(0) = v_k + r_k - \sum_{j\in\{1,\dots,K\}/\{k\}} R_{kj}$ and computes $S_{k,i} = g_1^{p_{k,u}(i)/t_{k,i}}$, $i \in [1, \dots, N_k]$.

Encryption: The encryption algorithm takes a set of attributes maintain by attribute authorities as well as a set of context-related attributes defined by data owner as inputs. Then it output the ciphertext of the data. It executes the same process as the proposed scheme in Fig. 2.

Decryption: The decryption algorithm takes the decryption credentials received from attribute authorities and context-related parameters obtained from smart mobile device and the ciphertext as input and output the original data. The decryption algorithm main executes two phases. Firstly, semi-trusted authority uses any required parameters and compute the partial decryption key for the user. Secondly, the smart mobile device securely computes the hash value of the required context-related attributes followed by multiplication with Y^{s_A}. Then, user use the pre-shared secret, partial decryption key received from semi-trusted authority, and the hash value of the context-related attributes.

Decryption process by semi-trusted authority: For each authority k, semi-trusted authority uses $S_{k,i}, C_{k,i}$ to compute $P_{k,u}$. Then semi-trusted authority multiplies all $P_{k,u}$ together and gets Q. In order to reduce the computation work at the mobile device, semi-trusted authority also computes T and sends T to the user.

Decryption process by mobile device: Using the pre-shared secret with authority k, user compute $\prod \hat{e}(g_1^{r_k}, g_2^{s_A}) = \hat{e}(g_1, g_2)^{s_A \sum r_k} = Y^{s_A \sum r_k}$. Now the corporate app installed in user's mobile device captures real-time contextual attributes. The decryption is successful if and only if $h(M(a'_{a,1})||M(a'_{a,2})||\dots||M(a'_{a,n}))$ $= h(M(a_{a,1})||M(a_{a,2})||\dots||M(a_{a,n}))$. Then user obtains $Y^{s_A} = T/Y^{s_A \sum r_k}$. The original data is recovered as follows:

$$Enc_m . \frac{h(M(a'_{a,1})||M(a'_{a,2})||\dots||M(a'_{a,n}))Q\hat{e}\left(D_u, E_1\right)}{E_0 Y^{s_A \sum r_k}} =$$
$$mY^{s_B} . \frac{h(M(a'_{a,1})||M(a'_{a,2})||\dots||M(a'_{a,n}))Y^{s_A + s_A \sum r_k}}{h(M(a_{a,1})||M(a_{a,2})||\dots||M(a_{a,n}))Y^{s_A + s_B + s_A \sum r_k}},$$
$$= \quad m.$$

In the low complexity scheme, the user u and authority k have a pre-shared secret r_k, which is blind to the semi-trusted authority. During the key issuing process, authority k embeds r_k into $p_{k,u}(0) = v_k + r_k - \sum_{j\in\{1,\dots,K\}/\{k\}} R_{kj}$. When semi-trusted authority executes the decryption process, r_k prevents semi-trusted authority from combines keys together to obtain the final decryption key. In the next section, we analyse the performance and security of the proposed algorithms.

V. Performance Analysis

In this section, we analyse the computation and communication costs associated with the schemes proposed schemes. The efficiencies of the proposed algorithms are demonstrated by comparing them against the conventional ABE schemes.

A. Reduction in Computational overhead

In the original MA-ABE, the user involve in key issuing and decryption stages. Hence, we can assume that the setup stage in the Fig. 2 and Fig. 3 can be done in off line for all three schemes (i.e., including conventional MA-ABE scheme). Denote the computational costs in $\mathbb{G}$ for multiplication, exponentiation, and pairing as C_m, C_{ex} and C_p, respectively. Let us also denote the total number of attribute authorities as K and n number of attributes from each authority will be used for encryption (for simplicity, we assumed equal number of attributes across all attribute authorities).

Let us evaluate the computational complexity involved in the remaining three stages. Table I shows all the computational complexities involve in key issuing (for user), encryption (for data owner) and decryption (for user) for all three schemes. Since we introduced a semi-trusted authority in LSD-ABAC, there is no computationally intensive task involved during the key issuing stage for the user. For other two schemes, computational complexities for key issuing are equal. Similarly, all three schemes share almost same computational complexity for encryption stage.

In order to graphically visualize the actual difference between the proposed schemes and conventional algorithm in decryption stage, we plotted the computational complexities given in Table I by varying the number of attributes, n, and number of authorities in Fig. 4. For this comparison, we used the benchmark time values given with popular pairing-based cryptography library namely jPBC in [10], [23]. Time complexities for operations $C_p, C_m,$ and C_{ex}, are 491.2ms, 20ms, and 34.1ms, respectively. The computational complexity is measured in terms of total time required for the the user to decrypt the data. It is obvious from Fig. 4 that our SD-ABAC performs equally well as the conventional MA-ABE scheme.

Setup $\mathcal{S}$ -Same as in Fig. 2

Key Issuing $\mathcal{KG}$

The semi-trusted authority executes the following steps with each authority k on behalf of user u, hence the following communication and computational overheads have been offloaded to semi-trusted authority:

- For $j \in \{1, ..., K\} \, / \, \{k\}$, semi-trusted authority gets the $D_{kj} = g_1^{R_{kj}} y_k^{x_j/(s_{kj}+u)}$ for $k > j$ or $D_{kj} = g_1^{R_{kj}} y_k^{(s_{kj}+u)/x_j}$ if $k < j$, where $R_{k,j} \in \mathbb{Z}_q$ is a random value.
- After receiving all D_{kj}, semi-trusted authority computes $D_u = \prod_{(k,j)\in\{1,...,N\}\times\{1,...,N\}/\{k\}} D_{kj} = g_1^{R_u}$, where $R_u = \sum_{(k,j)\in\{1,...,K\}\times\{1,...,K\}/\{k\}} R_{kj}$.
- If user u satisfies d_k number of attributes, then k^{th} attribute authority randomly picks a d_k-degree polynomial $p_{k,u}$.
- Now using the pre-shared secret r_k, authority k defines $p_{k,u}(0) = v_k + r_k - \sum_{j\in\{1,...,K\}/\{k\}} R_{kj}$
- Authority k computes $S_{k,i} = g_1^{p_{k,u}(i)/t_{k,i}}$, $i \in [1, \ldots, N_k]$.

Encryption $\mathcal{E}$ -Same as in Fig.2

Decryption by semi-trusted authority $\mathcal{DS}$

* Semi-trusted authority downloads CT_m from the cloud and checks the required attributes to decrypt m.

* For each authority k:

  * Using $S_{k,i}$ and the corresponding $C_{k,i}$, user computes $\hat{e}\left(S_{k,i}, C_{k,i}\right) = \hat{e}\left(g_1, g_2\right)^{s_A p_{k,u}(i)}$.

  * semi-trusted authority interpolates all $\hat{e}\left(g_1, g_2\right)^{s_A p_{k,u}(i)}$ and gets $P_{k,u} = \hat{e}\left(g_1, g_2\right)^{s_A p_{k,u}(0)} = \hat{e}\left(g_1, g_2\right)^{s_A\left(v_k + r_k - \sum_{j\neq k} R_{kj}\right)}$.

* Semi-trusted authority multiplies all $P_{k,u}$'s together and gets $Q = \hat{e}(g_1, g_2)^{s_A \sum(v_k + r_k) - s_A R_u} = \frac{Y^{s_A + s_A \sum r_k}}{\hat{e}\left(g_1^{R_u}, g_2^{s_A}\right)}$.

* Semi-trusted authority computes $T = \hat{e}(D_u, E_1) \cdot Q = \hat{e}(g_1^{R_u}, g_2^{s}) \cdot Q = Y^{s_A + s_A \sum r_k}$, and send T to the user.

Decryption by User $\mathcal{DU}$

* User computes $\prod \hat{e}(g_1^{r_k}, g_2^{s_A}) = \hat{e}(g_1, g_2)^{s_A \sum r_k} = Y^{s_A \sum r_k}$

* Now corporate app installed in user's mobile device computes $h(M(a'_{a,1})||M(a'_{a,2})|| \ldots ||M(a'_{a,n}))$.

* User computes $Y^{s_A} = T/Y^{s_A \sum r_k}$

* User can decrypt the data as follows (only if $h(M(a'_{a,1})||M(a'_{a,2})|| \ldots ||M(a'_{a,n})) = h(M(a_{a,1})||M(a_{a,2})|| \ldots ||M(a_{a,n})))$

$$Enc_m . \frac{h(M(a'_{a,1})||M(a'_{a,2})||...||M(a'_{a,n}))Q\hat{e}(D_u,E_1)}{E_0 Y^{s_A \sum r_k}} = mY^{s_B} . \frac{h(M(a'_{a,1})||M(a'_{a,2})||...||M(a'_{a,n}))Y^{s_A + s_A \sum r_k}}{h(M(a_{a,1})||M(a_{a,2})||...||M(a_{a,n}))Y^{s_A + s_B + s_A \sum r_k}} = m.$$

Fig. 3. Algorithm 2: LSD-ABAC: Lightweight Static and Dynamic Attributes Based Access Control Scheme for Secure Data Access in Mobile Environment.

TABLE I

COMPARISON OF COMPUTATION COST OF THE PROPOSED ALGORITHMS AGAINST THE CONVENTIONAL MA-ABE.

	MA-ABE	SD-ABAC	LSD-ABAC
Key Issuing	$(K-1) \times C_m$	$(K-1) \times C_m$	-
Encryption	$(nK+2) \times C_{ex} + C_m$	$(nK+3) \times C_{ex} + 2C_m$	$(nK+3) \times C_{ex} + 2C_m$
Decryption	$(nK+1) \times C_m + (nK+1) \times C_p$	$(nK+3) \times C_m + (nK+1) \times C_p$	$K \times C_p + (K+4) \times C_m$

In LSD-ABAC scheme, since the mobile user outsourced the part of the decryption process to the semi-trusted authority, the computational complexity become independent of number of attributes and substantially lower than other two schemes.

B. Reduction in Communication overhead

The communication overheads is always an important factor in mobile environment. In conventional MA-ABE scheme [19], the user needs to execute $(N-1)$ independent invocations for each authority during the key issuing stage. With the increasing number of authorities, it is generating a large network overhead. In our LSD-ABAC scheme, those communications have been leveraged to the cloud server based semi-trusted authority, hence the mobile devices are not necessary take part in the numerous communication requests and responses. It is reasonable to assumes there will be fixed broadband connections between the semi-trusted authority and attribute authorities. Fig. 5 shows the total number of interaction between user and authorities for both the proposed schemes and conventional scheme.

Number of interactions between user and attribute authorities in MA-ABE and SD-ABAC are almost equal. However, since we introduced a semi-trusted authority in LSD-ABAC, the user only needs to interact with the semi-trusted authority which in fact constant and substantially lower than other schemes.

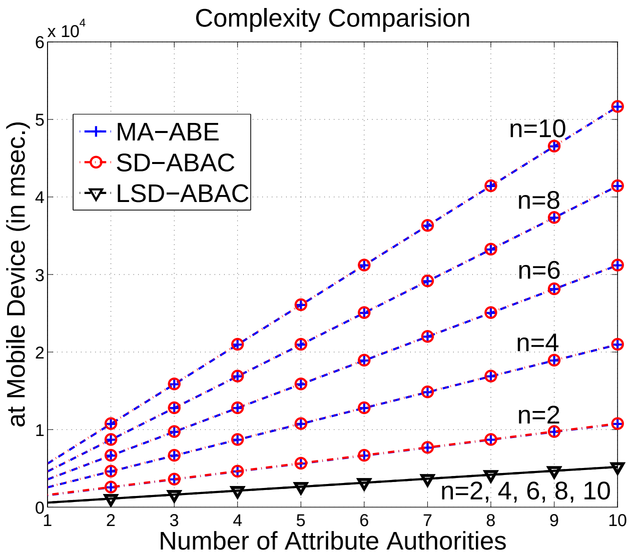

Fig. 4. Comparison of Computational Complexity

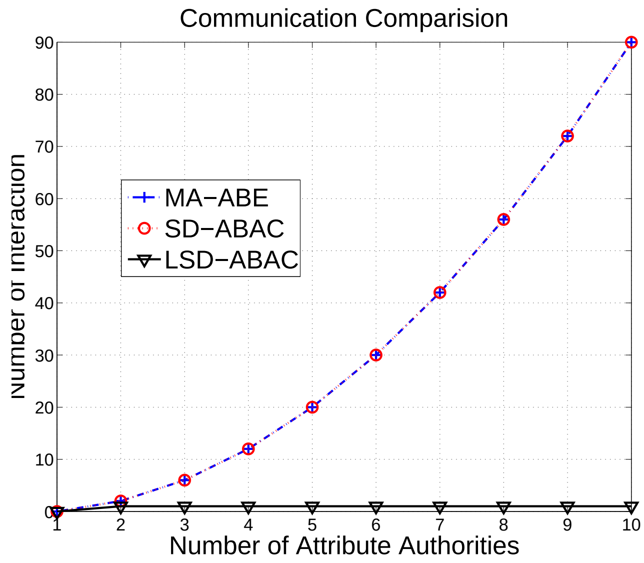

Fig. 5. Comparison of Communication Overheads

C. *Security Analysis*

The proposed static and dynamic ABE schemes were built on top of conventional ABE architectures [7], [19]. Our schemes do not degrade the security and privacy of the encrypted message and the mobile user compared to the original schemes. Further more, it satisfies data confidentiality of encrypted data against unauthorized users and the curious cloud service providers under the selective identity model. It also maintains the collusion resistance against up to $(N-2)$ attribute authorities. Let us explain possible attacks and how our schemes overcome those in the following subsections.

1) Collusion Attacks: ABE system vulnerable for collusion attacks. There are two main types of collusion: (1) attribute authorities collide with each other and aggregate users' attributes (2) users can collide with other and pool their own decryption keys to access data which are not authorized to none of them. Since our schemes were built top of the conventional MA-ABE scheme, the proposed schemes also collusion resistance against up to $(N-2)$ attribute authorities. Hence, let us discuss the user collusion.

- **User collusion in SD-ABAC and LSD-ABAC** During the key-issuing algorithm, user u in SD-ABAC or semi-trusted authority in LSD-ABAC obtains $D_{kj} = g_1^{R_{kj}} y_k^{x_j/(s_{kj}+u)}$. User identity u is embedded within decryption key by inverse exponentiation operation after adding u with random value s_{kj}. Fist of all, it is infeasible to infer $x_j/(s_{kj}+u)$ from $y_k^{x_j/(s_{kj}+u)}$. Secondly, since the user identity is randomized by s_{kj} and incorporated inversely within decryption key prevent malicious user from modifying u. Hence, the user collusion in the proposed schemes is infeasible.

2) Malicious Semi-trusted Authority: During the Key Issuing stage, the semi-trusted authority performs most of the steps instead of user. The k^{th} authority computes $S_{k,i} = g_1^{p_{k,u}(i)/t_{k,i}}$, $i \in [1, \dots, N_k]$ and sends them to semi-trusted authority. If the user satisfies the minimum d_k number of attributes, then the degree of the polynomial chosen by attribute authority is equal to d_k. Hence, d_k number of $S_{k,i}$ can be used to get the secret $p_{k,u}(0) = v_k + r_k - \sum_{j\in\{1,\dots,K\}/\{k\}} R_{kj}$ during the interpolation. If the user does not satisfy the minimum d_k number of attributes then the degree of the polynomial chosen by the k^{th} attribute authority is equal to $N_k + 1$. This is the crucial point, because the k^{th} attribute authority sends only N_k number of $S_{k,i} = g_1^{p_{k,u}(i)/t_{k,i}}$, $i \in [1, \dots, N_k]$ to the semi-trusted authority, where the semi-trusted authority would require $N_k + 1$ number of $S_{k,i}$ to recover the secret $p_{k,u}(0)$. Therefore, the semi-trusted authority cannot be able to distinguish which set of attributes belongs to the mobile user, and furthermore cannot be able to pool all $S_{k,i}$'s from all attribute authorities in order to find the attributes of mobile user. This preserves the privacy of the user.

During the Decryption stage, the semi-trusted authority performs the steps in place of the authority attribute authority. In more detail, the semi-trusted authority only computes $T = \hat{e}(D_u, E_1)\cdot Q = \hat{e}(g_1^{R_u}, g_2^s)\cdot Q = Y^{s_A + s_A \sum r_k}$ in contrast to the Y^s that is computed by the authority in the Chase-Chow scheme. As the required decryption key to decrypt the message m is Y^s, the semi-trusted authority cannot decrypt to obtain the message m, therefore the confidentiality of the message is ensured. More precisely, since the shared secret r_k is only known to the k^{th} attribute authority and the mobile user, and thus the summation $\sum_k r_k$ can only be obtained by a mobile user; therefore the semi-trusted authority cannot obtain Y^s from its known expression of $T = Y^{s+s\sum r_k} = Y^s Y^{s\sum r_k}$.

Moreover, since the data owner enforces dynamic attributes during the encryption process, adversary must satisfy not only the static attributes from authorities but also the dynamic attributes. Without the authorized dynamic attributes, it is impossible to decrypt the message m. Let us discuss whether feeding false values for dynamic attributes in order to decrypt the message is possible in the next subsection.

3) Attribute Cheating: Recently, a novel behavior profiling technique is developed to detect misuse of mobile devices [12], [13]. Mobile user activities such as app usage, network usage, charging times and unlock failures have been used to profile the user behavior. Hence, variations in user activity (i.e., anomalous activity) can be detected. Let us assume that there is an app which combines location information and time stamp together with user behavior profile, and uses machine learning techniques to detect anomaly activities (let us call this app as "behavior-profiling" app). Installing behavior-profiling app in the user mobile device can be used to verify whether the current user is the owner of the mobile device [14]. Samsung and Blackberry introduced robust security softwares called KNOX and Blackberry Enterprise Server (BES), respectively [15]–[17]. These softwares are capable to securely install corporate apps (i.e. behavior-profiling app) on users mobile devices and check for integrity of installed apps. Hence, modifying behavior profiling app to feed false information can be easily detected by data owner using either KNOX or BES.

VI. Conclusions

In this paper, we proposed static and dynamic attribute based access control schemes for the multi-authority scenario. In the proposed schemes, the data owner can incorporate the dynamic attributes together with the conventional attributes which are maintained by attribute authorities. Inclusion of dynamic attributes for encryption provides run-time security to the data stored in the cloud. Hence, even if the user has the credentials from the attribute authorities, the dynamic attributes must be satisfied in order to decrypt the data in the mobile device. We exploited the cloud infrastructure in order to outsource the heavy computational work and communication overheads in mobile user. We showed the proposed schemes enhances the security by adding dynamic attributes to the conventional attribute based encryption while substantially reduce the computational complexity to the mobile user.

Acknowledgements

This work was supported through the EPSRC Grants: "The Uncertainty of Identity: Linking Spatiotemporal Information Between Virtual and Real Worlds (EP/J005266/1)" and "TRUMP: A Trusted Mobile Platform for the Self-Management of Chronic Illness in Rural Areas (EP/J00068X/1)", and City University London's Research Pump-Priming Fund.

References

[1] Cisco Study: IT Saying Yes to BYOD, Cisco, http://newsroom.cisco.com/release/854754/Cisco-Study-IT-Saying-Yes-To-BYOD, May 2012.

[2] G. Thomson. BYOD: Enabling the Chaos, Network Security, vol. 2012, no. 2, pp. 5–8, Feb. 2012.

[3] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based Access Control Models. Computer, vol. 29, no. 2, pp. 38–47, 1996.

[4] K. Church, and B. Smyth. Understanding the Intent Behind Mobile Information Needs. In Proc. 14th Int'l Conf. Intelligent User Interfaces. ACM, 2009: 247-256.

[5] M. Green, S. Hohenberger, and B. Waters. Outsourcing the Decryption of ABE Ciphertexts. USENIX Security Symposium. Aug. 2011.

[6] A. Sahai, and B. Waters. Fuzzy Identity-Based Encryption. Advances in Cryptology EUROCRYPT, vol. 3494, 2005.

[7] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based Encryption for Fine-grained Access Control of Encrypted Data. In Proc. 13th ACM Conf. Comp. Commun. Security, New York, USA, pp. 89–98, 2006.

[8] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy Attribute-Based Encryption. In IEEE Symp. Security and Privacy, SP 07. pp. 321–334, May 2007.

[9] K. Zaidi, Y. Rahulamathavan, and M. Rajarajan. DIVA - Digital identity in VANETs: A Multi-authority Framework for VANETs. In Proc. 19th IEEE Int'l Conf. Netw. (ICON'13), Singapore, Dec. 2013.

[10] R. Lu, X. Liang, X. Li, X. Lin, Xuemin Shen, EPPA: An Efficient and Privacy-Preserving Aggregation Scheme for Secure Smart Grid Communications. In IEEE Trans. Parallel and Distributed Systems, vol.23, no.9, pp.1621-1631, Sep. 2012.

[11] F. Li, Y. Rahulamathavan, M. Rajarajan, R. C.-W Phan. Low Complexity Multi-authority Attribute Based Encryption Scheme for Mobile Cloud Computing. In Proc. IEEE 7th Int'l Symp. Service Oriented System Engineering (SOSE), San Francisco, USA, pp. 573–577, Mar. 2013.

[12] F. Li, N. Clarke, M. Papadaki, and P. Dowland. Misuse Detection for Mobile Devices Using Behaviour Profiling. In Int'l Journal of Cyber Warfare and Terrorism, vol. 1, no. 1, pp. 41–53, Jan.–Mar. 2011.

[13] M. Miettinen, P. Halonen, and K. Hatonen. Host-based Intrusion Detection for Advanced Mobile Devices. In Proc. 20th Int'l Conf. Advanced Information Networking and Applications, pp. 72–76, Washington, DC, USA. 2006.

[14] N. Eagle, and A. S Pentland. Reality Mining: Sensing Complex Social Systems. Journal Personal and Ubiquitous Computing, vol. 10, no. 4, pp. 255 – 268, Mar. 2006.

[15] Y. Rahulamathavan, V. Moonsamy, L. Batten, S. Shunliang and M. Rajarajan. An Analysis of Tracking Service Settings in Blackberry 10 and Windows Phone 8 Smartphones, 19th Australasian Conference on Information Security and Privacy (ACISP), Wollongong, Australia, Jul., 2014.

[16] http://www.samsung.com/global/business/mobile/solution/security/samsung-knox.

[17] http://uk.blackberry.com/business/mobile-device-management.html.

[18] M. Chase. Multi-authority Attribute Based Encryption, In Lecture Notes of Theory of Cryptography in Computer Science, Berlin Heidelberg, pp. 515–534, 2007.

[19] M. Chase, and S. S. M. Chow. Improving Privacy and Security in Multi-authority Attribute-Based Encryption. In Proc. 16th ACM Conf. Comp. Commun. Security, New York, NY, USA, pp. 121–130, 2009.

[20] A. B. Lewko, and B. Waters. Decentralizing Attribute-based Encryption, in EUROCRYPT, ser. LNCS, K. G. Paterson, Ed., vol. 6632. Springer, pp. 568–588, 2011.

[21] C. Burnett, P. Edwards, T. J Norman, L. Chen, Y. Rahulamathavan, M. Jaffray, E. Pignotti. TRUMP: A Trusted Mobile Platform for Self-management of Chronic Illness in Rural Areas. In Trust and Trustworthy Computing, pp. 142–150. Springer Berlin Heidelberg, 2013.

[22] D. Weerasinghe, Y. Rahulamathavan, M. Rajarajan. Secure Trust Delegation for Sharing Patient Medical Records in a Mobile Environment. Health Policy and Technology, vol. 2, pp. 36–44, 2013.

[23] http://gas.dia.unisa.it/projects/jpbc/

[24] L. Scott, and D. E. Denning. A Location Based Encryption Technique and Some of Its Applications. In Proc. National Technical Meeting of The Institute of Navigation, Anaheim, CA, pp. 734–740, Jan. 2003.

[25] L. Hsien-Chou, and C. Yun-Hsiang, A New Data Encryption Algorithm Based on the Location of Mobile Users, Information Technology Journal, vol. 7, no. 1 pp. 63-69, 2008.

[26] Al-Ibrahim, Omar, Ala Al-Fuqaha, D. V. Dyk, and N. Akerman. Mobility Support for Geo-Encryption. In Proc. IEEE Int'l Conf. Commun., pp. 1492–1496, 2007.

[27] V. Vijayalakshmi, and T. G.Palanivelu. Secure Localization Using Elliptic Curve Cryptography in Wireless Sensor Networks. Int'l Journal of Comp. Sciences and Netw. Security, vol. 8, no. 6 pp. 255–261, 2008.

[28] R. Karimi, M. Kalantari. Enhancing Security and Confidentiality in Location-based Data Encryption Algorithms, Applications of Digital Information and Web Technologies (ICADIWT), 2011 Fourth Int'l Conf., pp. 30–35, Aug. 2011.