Computer Communications 68 (2015) 61–72

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Robust access control framework for mobile cloud computing network

Fei Li [a], Yogachandran Rahulamathavan [a,*], Mauro Conti [b], Muttukrishnan Rajarajan [a]

[a] School of Engineering and Mathematical Science, City University London, London, United Kingdom
[b] Department of Mathematics, University of Padova, Padova, Italy

ARTICLE INFO

Article history:
Available online 11 July 2015

Keywords:
Access control
Smart devices
Attributes
Encryption
Cloud computing

ABSTRACT

Unified communications has enabled seamless data sharing between multiple devices running on various platforms. Traditionally, organizations use local servers to store data and employees access the data using desktops with predefined security policies. In the era of unified communications, employees exploit the advantages of smart devices and 4G wireless technology to access the data from anywhere and anytime. Security protocols such as access control designed for traditional setup are not sufficient when integrating mobile devices with organization's internal network. Within this context, we exploit the features of smart devices to enhance the security of the traditional access control technique. Dynamic attributes in smart devices such as unlock failures, application usage, location and proximity of devices can be used to determine the risk level of an end-user. In this paper, we seamlessly incorporate the dynamic attributes to the conventional access control scheme. Inclusion of dynamic attributes provides an additional layer of security to the conventional access control. We demonstrate that the efficiency of the proposed algorithm is comparable to the efficiency of the conventional schemes.

1. Introduction

Nowadays organizations demand a host of tools, including desktop and smart devices, email, instant messaging, voice mail, presence information and audio, video and web conferencing. When these tools are integrated into a system that allows seamless data sharing among devices then it's called unified communications network. Integrating smart devices within traditional network increases productivity among employees as well as new security vulnerabilities. Traditionally, organizations store data in local servers while employees access the data using access control techniques. However, the recent trend towards cloud computing, outsourcing, smart devices or Bring-Your-Own-Devices (BYOD), and high bandwidth mobile broadband has enabled organizations to share information anywhere and anytime. Data could be shared using public data storages such as cloud computing infrastructure which can provide flexible computing capabilities, reduced costs and capital expenditures and charge based on to usage.

In particular, BYOD became a hot topic after the 2012's Cisco survey which has found that 95% of the employees are allowed to use their mobile devices within their organizations [14]. Since then the number users use their personnel device for work has increased exponentially across the globe [15,16]. This trend is against the tradition where employees are allocated with company devices embedded with specific softwares and policies to achieve security. Currently researchers are focusing on developing techniques to securely virtualize the user device hence the corporate data will be protected from data breaches [41,42]. Samsung and BlackBerry use technologies called KNOX, and BES12, respectively to enforce the corporate security policies on user's device [29,30].

This trend requires new ways to control the access of data stored in cloud. Traditionally, we assume that data owners, users, and storage server are in the same domain and also that the server is fully trusted [1–9]. However, in BYOD, 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. To overcome this challenge, the data confidentiality in a distributed environment is achieved via attribute based encryption (ABE) technique [10–13].

ABE is considered as a promising cryptographic technique and supports both the data confidentiality and access control simultaneously [10–13]. Using ABE, the data owners can encrypt the data using fine-grained access policies. 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

* Corresponding author. Tel.: +44 0207040 8377; fax: +44 0207040 8566.
E-mail addresses: fei.li.1@city.ac.uk (F. Li), yogachandran.rahulamathavan.1@city.ac.uk (Y. Rahulamathavan), conti@math.unipd.it (M. Conti), R.Muttukrishnan@city.ac.uk (M. Rajarajan).

http://dx.doi.org/10.1016/j.comcom.2015.07.005
0140-3664/

employee who is a "Manager" employed at "Company A" can decrypt the file. There are two major types of ABE schemes: single authority based ABE [11] and multiple authorities based ABE (MA-ABE) [32] schemes. In a single authority based ABE scheme, only one authority called attribute authority (AA) is responsible for monitoring all the attributes. In MA-ABE, in contrast to the single authority ABE scheme, there are multiple attribute authorities responsible for a disjoint sets of attributes.

When it comes to BYOD, the ABE cannot directly be used to protect the data due to the user's mobility. It should be noted that the data confidentiality in the ABE schemes relies only on predefined static attributes such "Manager", "Finance Office", and "Company A". Let us consider the previous example, where an employee has the long term credentials for the following attributes: "Manager", and "Company A". Hence, she can access the encrypted file while she is traveling in public transport using her personnel mobile device. However, the risk level associated in this context is high. In fact, people in her proximity might easily see confidential data via shoulder surfing. It is also possible for an adversary to steal the employee's mobile device, and get unauthorized access to the corporate data if there is no real-time verification (assuming that the credentials for static attributes are stored within mobile). Hence, evaluating the data collected by smart device's sensors in real-time provides additional layer of security. In particular smart device attributes such as location, app usage patterns, unlock failures, Wi-Fi networks and proximity of devices could be exploited for real-time verification. We refer the attributes collected via smart devices as dynamic attributes since they change every time with the user's mobility.

In this paper, we propose a new algorithm which supports the organizations to incorporate dynamic attributes within the ABE scheme for robust access control. The novelty of our algorithm are listed below:

1. New algorithm enforces the dynamic attributes to the conventional ABE scheme.
2. New algorithm does not compromise the security of the conventional ABE scheme.
3. New algorithm supports both the single authority and multi authority schemes.
4. Performance of the new algorithm is comparable with the conventional ABE scheme.

The reminder of this paper is organized as follows: we review related works in Section 2. We describe the system architecture and various types of attacks in Section 3. In Section 4, we propose the static and dynamic ABE scheme for single authority scheme followed by a MA-ABE scheme in Section 5. We compare the performance of the proposed schemes against the conventional ABE schemes in Section 6. Section 7 is dedicated for analyzing the security and privacy issues of the proposed schemes. Conclusions, limitations and future works are discussed in Section 8.

2. Related work

Access control is a classical security issue. Various access control models have been proposed in literature. In 1996, Sandhu et. al proposed the feasible access control model called role-based access control (RBAC) [1]. It simplifies authorization and administration because a security administrator needs only to revoke and assign the new appropriate role memberships if a user changes her job function. Various improved RBAC models have been proposed and been widely used in practice. Zhang and Parashar extended the RBAC model to support context information called context-aware dynamic access control scheme [2]. In [2], a user is assigned with access credentials based on her roles (i.e., a set of attributes) and context information. The resource maintains a set of roles and assign a potential role with certain permissions to the user.

Similar works have been proposed based on temporal condition called a temporal RBAC in [4] and based on wider range of event and environmental conditions called event-based RBAC in [3]. In [3,4], the event was defined as measurable, dynamic context variables that can influence access decisions besides the location and time variables. In [17], both the spatial and temporal attributes were exploited to support patient-centric access control scheme in e-healthcare. All these works successfully extend the RBAC model to enforce the context information for access control. However, the central architecture of RBAC is not suitable for today's mobile environment since the data and users are not restricted to be in the same environment i.e., outsourcing the data to cloud and usage of smart devices.

Mobile RBAC system which enforces spatially-aware (location) RBAC policies is proposed in [6]. In [6], an object is equipped with a near field communication (NFC) receiver and the user has an NFC enabled handset. Thus, the users can access certain resources by exchanging credentials using NFC protocols. The NFC receiver verifies the location of the user, but also restrains the range of the implementation since the user has to access the resource by going to an access point.

Hasen and Oleshchuk presented an extended version of RBAC model for mobile systems [5]. They extended the RBAC model by introducing the notion of environmental roles in order to control permission sets by activation and/or deactivation of roles based on spatial information. The main difference in their work with others' is that the availability of permission sets depend on spatial information within the same active role. Permissions are dynamically assigned to the role dependent on location. Thus, it reduces the number of roles that needs to be specified within the system.

In [7], the authors presented a more complex location-aware RBAC model. There are two kinds of associations roles are possible with locations: the role can only be assigned to a user when he is in certain designated locations and some roles can only be activated in some specific locations. Both of the works in [5,7] incorporate the spatial-temporal information in the RBAC model, but they do not consider other important contextual attributes which are important in today's mobile environment.

It should be noted that the works discussed so far have not focused on the data confidentiality. These works assumed that the storage server is secure, hence the data stored in the server is not encrypted. As said in the introduction, these traditional access control techniques are not suitable for the current unified communications network. Let us now discuss the existing access control techniques where the data is stored in the encrypted format.

Hsien-Chou and Yun-Hsiang proposed a location-based data encryption technique using static locations [19]. In this work, each static location contains pre-determined longitude and latitude coordinates. The concept of "geoencryption" or "location-based encryption" was developed to use in digital film distribution by Scott and Denning [18]. Al-Ibrahim et al. presented a geoencryption protocol by restricting the decryption of a message to a particular location and time period [20]. The encryption in this work is similar to [19] where the locations were static which means those 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 for the wireless sensor network operation [21]. In [21], the location based authentication scheme was built based on the identity-based cryptography using ECC and ECC key exchange. Karimi and Kalantari [22] presented a geoencryption protocol which allows the mobile nodes to communicate with each other by restriction when decoding a message in the specific location and time period. Similar technique was applied for mobile devices in [23].

In [8], an access control framework is proposed using IEEE 802.11 protocol, whereby the access to a wireless local area network (WLAN)

Table 1
Comparison of related works. The proposed work incorporate support more dynamic attributes than other works.

		Dynamic attributes					
	Static attributes	Spatial OR temporal attribute	App usage	Unlock failure	Proximity	etc	Data confidentiality
Context aware RBAC [2]	✓	✓	×	×	×	×	×
Event driven RBAC [3]	✓	✓	×	×	×	×	×
Temporal RBAC [4]	✓	✓	×	×	×	×	×
Spatial RBAC [5]	✓	✓	×	×	×	×	×
Spatial temporal RBAC [6]	✓	✓	×	×	×	×	×
Location aware RBAC [7]	✓	✓	×	×	×	×	×
Location aware AAC [8]	✓	✓	×	×	×	×	×
Spatial-Temporal and E-health [17]	✓	✓	×	×	×	×	×
Location based encryption [18]	✓	✓	×	×	×	×	✓
Location and mobile [19]	✓	✓	×	×	×	×	✓
Geoencryption [20]	✓	✓	×	×	×	×	✓
Secure localization [21]	✓	✓	×	×	×	×	✓
Location based encryption [23]	✓	✓	×	×	×	×	✓
Proposed scheme	✓	✓	✓	✓	✓	✓	✓

system is granted if and only if the client is located within the areas covered by multiple access points. When client comes to such areas then she will receive decryption credentials via IEEE 802.11 protocol. However, when a user is out of the wireless signal broadcasting range, then she could not get access to the system. The authors in [9] proposed a methodology which examines and analysis whether an access control model is adequately protected. It helps the developer to consider the security when enforcing contextual information in RBAC model. It should be stressed that these encryption based access control methods are not dynamic and not scalable. Table 1 compares these works with the our work which incorporates dynamic attribute into the access control. In Table 1, we denote ✓, ×, as the compatibility of support and no support, respectively. Table 1 clearly shows that our work is significantly different from other existing works. Let us review some of the pioneering works in ABE. ABE was firstly proposed by Sahai and Waters [10], 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 schemes namely key-policy attribute-based encryption which was proposed by Goyal et al. [11], and ciphertext-policy attribute-based encryption which was proposed in [12]. Chase [31] presented a MA-ABE system which allows any polynomial number of independent attribute authorities to monitor attributes and distribute private-keys. The data owner can decide a number d_k and a set of attributes from an AA, and encrypt a message such that only a user with minimum d_k number of attributes from the revelent AA can decrypt the message.

Chase and Chow proposed another work [32] which improved the previous scheme [31]. In [32], central authority was removed, and anonymous key issuing protocol which address the privacy of the user was proposed. Lewko and Waters proposed a fully decentralized ABE scheme, where user could have zero or more attributes from each AA and do not require a trusted server [33]. In their work, the AA can join and leave the system freely without re-initializing the system.

The work in [43] is about outsourcing the computational and communications complexity of the users to a semi-trusted authority. A new cryptographic scheme has been designed to exploit the semi-trusted authority without violating the privacy concern. However, the core idea of coming up with a new scheme to exploit the features of smart devices is presented here. The proposed work systematically identifies the weaknesses of existing access control schemes and validates the importance of the new scheme. More importantly, the suitability of the new scheme is validated using extended simulations using a mobile test bed.

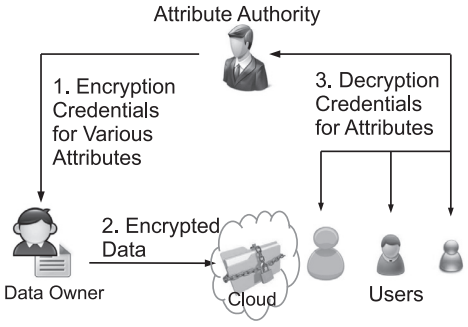

Fig. 1. Single authority ABE scheme.

3. Problem statement

This section presents system architecture, security and privacy threats associated with the propose approach and main assumptions applicable to the rest of this work.

3.1. System architecture

The proposed system consists of the following four components: users (or employees), data owner (or organizations), cloud service provider, and attribute authorities. An illustration of the components of the system as well as relationships between them are provided in Fig. 1.

Users are equipped with one or more smart devices. The smart device has an active Internet connection which enables the user to request and receive any data or services from anywhere and at anytime. Since, the proposed technique uses bilinear paring using elliptic curve cryptography (ECC), any smart device currently performing RSA encryption and decryption (i.e., SSL) can run the proposed technique since the key size required for ECC is 224-bits which substantially lower than 2048-bits RSA key size. Hence, we assume that the smart device has sufficient battery power to carry out required computation in order to decrypt the data.

Data owners upload the encrypted data to the cloud storage and define access policies. In our scheme, the data owner defines access policies based on static attributes obtained from AA together with dynamic attributes. Thus, data owner can decide who is able to decrypt the data from where and in what circumstances.

Cloud service providers provide cloud storage and computational power to both the users and data owners. In our scheme, we assume that data owner will upload the encrypted data to the storage

while the user will download the encrypted data from the storage. It is reasonable to assume that the cloud will provide the processing power to find the encrypted-data file to the user based on keyword searching.

Attribute authorities manage and maintain static attributes of the users. Different authorities manage different sets of attributes. A user needs to prove her attributes to the authorities in order to receive the decryption key for each attribute from corresponding authority. This is achieved using the anonymous key-issuing protocol proposed in [32]. In the proposed scheme, it should be noted that for the dynamic attributes, there is no need to have an AA to distribute encryption and decryption credentials e.g., for location attribute the device uses GPS module to obtain longitude and latitude.

3.2. 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 credentials for various attributes. These attribute authorities verify the user attributes before releasing the corresponding decryption credentials for the attributes. Data owner obtains the encryption credentials for a set of attributes from the AA, 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, AA and users interact with each other.

3.3. Security and privacy threats

There are number of known security and privacy threats hinder the access control schemes. Let us provide a list of potential attacks and relate them to the requirements of the system.

- **Identity-related threats**: The main threat we need to consider is related to the identities of the elements involved in the protocol. Adversary might impersonate as one of the entities and try to establish a connection to a legitimate entity. Our goal is to prevent an adversary from impersonating a legitimate entity.
- **Privacy-related threats**: In order to receive the decryption key, a user needs to provide her attributes to the authority. Hence, a malicious authority might profile the users and their requested attributes. Our aim is to provide privacy guarantee to the users, hence, any AA will not be able to profile the users.
- **Collusion attacks**: Each AA manages a set of attributes in our system, hence, authorities can collide with each other to infer user attributes. This will allow the malicious authorities to profile the user based on different set of attributes user has with the malicious authorities. Similarly, two users can collide and get access to data which are not accessible by the users individually. Our goal is to protect collusion attacks both at the end-user and at the AA.
- **Dynamic attribute cheating**: Smart devices capture dynamic attributes such as unlock failures, app usage, location and near by devices using the local sensors. It is crucial for the data owner to ensure that the user's device is not modified (rooted) or tampered in order to feed false dynamic attributes. Hence, we need to consider a set of technologies in order to guarantee that there is no attribute cheating is possible.
- **Tracking threats**: The app installed within user device needs to collect the sensor data to determine the values for dynamic attributes. If the app is malicious then that can pass the sensor data to third-parties who can then be able to track the user. Hence it is important to have technologies which protect the users from being tracked by third-parties.

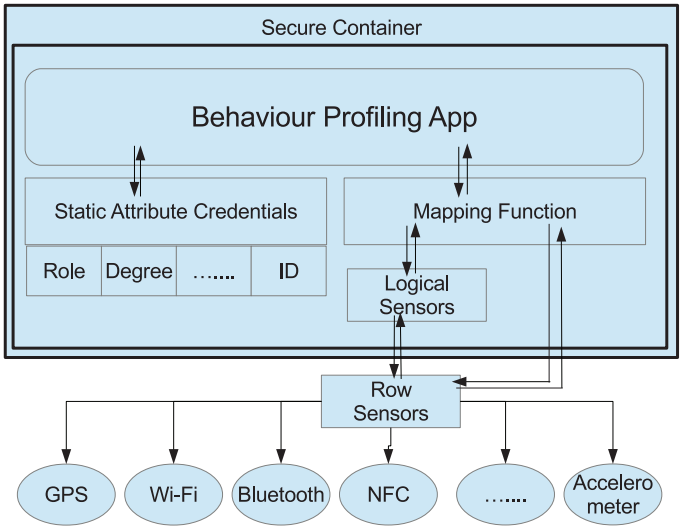

Fig. 2. Proposed algorithm's architecture at high level.

We explain how our new algorithm mitigates all these threats in the security and privacy analysis section later in this paper. In the next section, we define a set of complexity assumptions followed by dynamic and static ABE algorithm for a single AA case.

4. Dynamic and static attribute-based encryption scheme for single AA

4.1. Assumptions and design principles

In the proposed scheme, we assume that users have an app installed on their smart device which captures dynamic attributes. These attributes have been used together with the static attributes of the user to satisfy the access policy defined by the data owner.

4.1.1. Dynamic attributes

Dynamic attributes such as location, time, temperature, noise, light, the presence of other devices, a particular interaction between the user and the smartphone, or a combination of these were used in [37–40] to define fine-grain access policies in smart device environment. In [28], authors proposed an access scheme to dynamically control the device locking timeout and unlocking method based on perceived safety and real-time context in mobile devices. Recently, a novel behavior profiling technique has been developed to detect misuse of mobile devices based on these dynamic attributes [24,25]. 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. The works in [24,25,28,37–40] combines dynamic attributes and time stamp and uses machine learning techniques to detect anomaly activities. This functionality could be incorporated with mobile apps i.e., let us call this app as "behavior-profiling" app.

As shown in Fig. 2 the app will be installed within a secure container. This container is monitored by employer using software platforms such as KNOX or BES12 [29,30]. Static attributes also stored within the secure container. The values for dynamic attributes are obtained from the smartphones *raw sensors* (e.g., GPS sensor) and *logical sensors*. Logical sensors are functions which combine raw data from physical sensors to capture specific user behaviors (such as detecting whether the user is running). We assume that the mapping function as well as the logical sensor reside within secure container to mitigate user's malicious activity. Security and privacy analysis of the behavior-profiling app is discussed in Section 7.4.

4.1.2. Preliminaries

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 $\forall x \in \mathbb{G}_1, \forall y \in \mathbb{G}_2$, there is $\hat{e}(x, y) \neq 1$.
3. Computability: $\hat{e}$ is an efficient computation.

Lagrange interpolation: Shamir's secret share uses Lagrange interpolation technique to obtain the secret from shared-secrets. Suppose that $p(x) \in \mathbb{Z}_p[x]$ is a $(k-1)$ degree polynomial and secret $s = p(0)$. Let us denote $S = \{x_1, x_2, \ldots, x_k\}$ and the Lagrange coefficient for x_i in S as

$$\Delta_{x_i,S}(x) = \prod_{x_j \in S, x_j \neq x_i} \frac{x - x_j}{x_i - x_j}.$$

For a given k different number of values $p(x_1)$, $p(x_2)$, $\ldots$, $p(x_k)$, the polynomial $p(x)$ can be reconstructed as follows:

$$p(x) = \sum_{x_i \in S} p(x_i) \prod_{x_j \in S, x_j \neq x_i} \frac{x - x_j}{x_i - x_j} = \sum_{x_i \in S} p(x_i) \Delta_{x_i,S}(x),$$

hence the secret s can be obtained as:

$$s = p(0) = \sum_{x_i \in S} p(x_i) \prod_{x_j \in S, x_j \neq x_i} \frac{0 - x_j}{x_i - x_j}.$$

Mapping: We consider a linear comparison function namely mapping denoted as M. This function takes two inputs: one from smart device sensors (i.e., location) and the second one from the data owner. Data owner must embed the required sensor data and boolean operations. The output of the mapping function is "*yes*" or "*no*" by comparing the both the inputs. For example, this function can extract data from smart device's GPS module and compare with locations in data owners input and output "*yes*" or "*no*" i.e., $M(\text{"}data\ from\ GPS\text{"} = \text{"}London\text{"}) = no$. It should be noted that this function can be embedded securely within any smartphone apps which calls required sensors at device level. Similar to the XACML policy language, the data owner can even define a range of values for sensor data. However data owner needs embed a boolean function with the data to expedite the comparison process.

4.1.3. Complexity assumption

Decisional Modified Bilinear Diffie-Hellman (MBDH) assumption [10]: Suppose a challenger chooses $a, b, c, z \xleftarrow{R} \mathbb{Z}_p$ at random. The Decisional MBDH assumption is that no polynomial-time adversary is to be able to distinguish the tuple $(A = g^a, B = g^b, C = g^c, Z = e(g,g)^{\frac{ab}{c}})$ from $(A = g^a, B = g^b, C = g^c, Z = e(g,g)^z)$ with more than a negligible advantage.

We will use this property to prove by contradiction that our proposed algorithm is secure against well-known attacks. Later in this paper, we will show that if there is an adversary who can break the proposed algorithm then we can use the adversary indirectly to break the MBDH assumption (i.e., this is a contradiction to the MBDH assumption, hence our proposed algorithm is secure).

4.2. Proposed scheme

In contrast to the conventional ABE scheme described in Section 3.2, we will show in this section that how to efficiently incorporate the dynamic attributes to the conventional ABE scheme, where the data owner can encrypt the data by not only using the credentials obtained from the AA, but also using dynamic attributes. Similar to the conventional ABE [11], the proposed algorithm also composed of four sub-algorithms namely setup, key issuing, encryption, and decryption. The proposed algorithm is given in Fig. 3 (the steps which are different from conventional ABE scheme are denoted as *). Let us briefly explain the functionalities of each sub-algorithms below.

Setup: The setup algorithm takes a security parameter λ as input and output a bilinear group and a set of parameters. Parameters q, g and $\mathbb{G}_1$ are public parameters, $\{T_i = g^{t_{A,i}}\}$ and $Y = \hat{e}(g,g)^y$ are public-keys of the attributes maintained by an authority and y, $t_{A,i} \in Z_q$ for each attribute i $(1 \leq i \leq N)$ are private-keys known only to AA where N is the total number of attributes monitored by the authority.

Key issuing: The AA generates decryption key for a user u who holds a set of attributes by randomly generating an unique polynomial p_u for u. However, $p_u(0) = y$ for all the users. Then, AA will issue a decryption credential $D_i(u)$ to the user u for her ith attribute.

Encryption: The encryption algorithm takes a set of attributes maintained by AA as well as a set of dynamic attributes defined by the data owner as input. Then it output the ciphertext of the data. In this step, data owner generates private-keys s_A and s_B and the corresponding public-keys E_0 and E_i for all attributes. The hash value of dynamic attributes is incorporated in E_0.

Decryption: The decryption algorithm takes the decryption credentials received from AA, dynamic parameters obtained from smart mobile device and the ciphertext as input and then output the original data. The behavior-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 $h(M(a'_{c,1})||M(a'_{c,2})||\ldots||M(a'_{c,n})) = h(M(a_{c,1})||M(a_{c,2})||\ldots||M(a_{c,n}))$.

The novelty in our scheme compared to the conventional ABE scheme lies in the encryption and the decryption sub-algorithms in Fig. 3. Let us denote the dynamic attribute set defined by the data owner as $A_C = \{a_{c,1}, \ldots, a_{c,n}\}$ where $a_{c,i}$ denotes the ith 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 her recent app usage" and $a_{c,3}$ ="unlock failures in last two days". Now the data owner defines $A_C = \{a_{c,1} = \text{"}LONDON\text{"}, a_{c,2} < \text{"3"}$ and $a_{c,3} < \text{"2"}\}$ and computes $E_0 = h(yes||yes||yes)Y^{s_A+s_B}$. Let us assume that the risk level varies between 1 and 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 ordinary documents.

In the decryption phase, "behavior-profiling" app pre-installed in the users' mobile device determines its location. As explained in Section 4.1.2 mapping function, M, which inputs data owners dynamic attribute requirements and smart device readings and output "*yes*" or "*no*", e.g., if the current risk level is less than the threshold defined by the data owner then $M(a_{c,2} < \text{"3"}) = yes$. This ensures that even a user has all the credentials from AA, dynamic attributes enforced by the data owner must be satisfied before the decryption. We analyze the security of the behavior-profiling corporate app in the security analysis section in Section 7.

4.3. Security game

In order to avoid security vulnerabilities, the ABE based schemes must be proved to be secure against selective identity (ID) model [10]. In selective ID model, adversary must provide the challenge identities he wishes to challenge to challenger. Then the challenger (i.e., system) will generate necessary parameters corresponding to the challenge identities and send them to the adversary. Only one requirement is that credentials for at least one attribute in the challenge identity cannot be revealed to the adversary. Then adversary is allowed to make secret queries about challenge identities. If the adversary cannot decrypt the encrypted message at the end with

Setup $\mathcal{S}$

- AA generates group bilinear parameters: $q, g, \mathbb{G}_1, \leftarrow \mathcal{S}(1^\lambda; \sigma)$, for a given security parameter λ and $\sigma \in \{0,1\}^{poly(\lambda)}$.
- AA randomly generates secrets $y, t_{A,i} \in Z_q$ for each attribute i ($1 \le i \le N$) where N is the total number of attributes monitored by the authority.
- AA publishes the corresponding public keys $\{T_i = g^{t_{A,i}}\}\ \forall\ i \in \{1, ..., N\}$ and $Y = \hat{e}(g,g)^y$.

Key Issuing $\mathcal{KG}$

- To issue decryption keys for user u, AA chooses a random polynomial p_u with degree $d-1$ where $p_u(0) = y$. It should be noted that $p_u(0)$ for any users should be equivalent to y.
- AA generates decryption credential for i^{th} attribute for u as: $D_u(i) = g^{p_u(i)/t_{A,i}}$ where $\forall i \in A_U$ (A_U denotes the attributes set of u).

Encryption $\mathcal{E}$

Data owner encrypts data m for attribute set $A_m = A_A \bigcup A_C$, where $A_A = \{a_{a,1}, \ldots, a_{a,l}\}$ denotes the attributes maintained by AA and $A_C = \{a_{c,1}, \ldots, a_{c,n}\}$ denotes the dynamic attributes defined by the data owner, as follows:

* Data owner randomly chooses $s_A,\ s_B \in Z_q$ and encrypts the data as $Enc_m = mY^{s_B}$.

* Data owner computes $E_0 = h(M(a_{c,1})\|M(a_{c,2})\| \ldots \|M(a_{c,n}))Y^{s_A+s_B}$, $E_i = g^{t_i s_A}, \forall i \in A_A$, where $h : \{0,1\} \rightarrow Z_q$ is a secure hash function, M is a mapping function of dynamic attributes and || denotes concatenation.

* Now data owner uploads $CT_m = \{Enc_m,\ E_0,\ E_i, \forall i \in A_A$ and $A_C\}$ into the cloud.

Decryption $\mathcal{D}$

* User downloads CT_m from the cloud and checks the required attributes to decrypt m.

* User computes $\hat{e}(E_i, D_u(i)) = \hat{e}(g,g)^{p_u(i)s_A}$ where $i \in A_A \bigcap A_U$.

* Using interpolation technique, u can compute $Y^{s_A} = \hat{e}(g,g)^{p_u(0)s_A} = \hat{e}(g,g)^{ys_A}$.

* Now corporate app installed in users mobile device computes the hash value of dynamic attributes such as current location, risk-level associated with current location and risk-level associated with user behavior and outputs $h(M(a'_{c,1})\|M(a'_{c,2})\| \ldots \|M(a'_{c,n}))$.

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

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

Fig. 3. Static and dynamic attribute based encryption scheme for single AA.

non-negligible advantage then the proposed scheme is secure against selective ID model. Formally, this is represented by the following game between the adversary and challenger (here we focus on single AA and the game can be extended for multi authority):

Setup: The challenger runs the setup phase of the algorithm and tells the adversary the public parameters.

Secret key queries: The adversary is allowed to make any number of secret key queries. However, the only requirement is that for each user, there must be at least one attribute for which the adversary can get insufficient number secret keys.

Challenge: The adversary sends two messages m_0 and m_1 to the challenger in plain domain. Now the challenger randomly chooses one of the messages and encrypt it and send the ciphertext to the adversary.

More secret key queries: The adversary is allowed to make more secret key queries as long as he satisfy the requirement given earlier.

Guess: Now the adversary guesses which message was encrypted by the challenger. The adversary is said to be successful if he guesses the correct message with probability $\frac{1}{2} + \epsilon$ whereby ϵ is non-negligible function.

5. Dynamic and static attribute-based encryption scheme for multiple attribute authorities

In a single authority scenario, there is only one AA monitors all the attributes and issues encryption and decryption credentials for the data owners and 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 AA has too much power and it can decrypt all the data and knows about all the users' attributes. In the event of corruption, the message confidentiality cannot be achieved and user's privacy can be compromised by the attackers. This is one of the limitations 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 [34] or in vehicular Ad hoc network different identities can be monitored by different authorities [35]. Hence, it is more convenient to have multiple attribute authorities where each AA can maintain attributes belonging to one department. MA-ABE scheme without incorporating the dynamic attributes was proposed in [31,32]. Hence, similar to Fig. 3, static and dynamic attributes based MA-ABE scheme is given in Fig. 4. In our scheme, we assume that there are K number of attribute authorities. Each AA manages N number of different attributes and issues credentials for the users based on their eligible attributes. Let us briefly explain the functionalities of each sub-algorithms below.

Setup: The setup algorithm takes security parameters λ 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 kth AA and the corresponding public-keys $Y_k = \hat{e}(g_1, g_2)^{v_k}$ and $y_k = g_1^{x_k}$ 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 ith attribute maintained by kth AA and the corresponding public-key is $T_{k,i} = g_2^{t_{k,i}}$.

Key issuing: User and AA execute anonymous key-issuing protocol proposed in [32]. User computes decryption credential D_{kj} for jth attribute by collaborating with kth AA. Once user obtained all the D_{kj}, she will compute D_u followed by $S_{k,i}$. Since this is based on anonymous key-issuing protocol, attribute authorities cannot be able to profile the users.

Encryption: The encryption algorithm takes a set of attributes maintained by attribute authorities as well as a set of dynamic attributes defined by data owner as inputs. Then it output the ciphertext of the data. This step is same as the single authority case.

Setup $\mathcal{S}$: For a given security parameters λ and $\sigma \in \{0,1\}^{poly(\lambda)}$, group bilinear parameters are generated by the attribute authorities as follows: $q, g_1, g_2, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T \leftarrow \mathcal{S}(1^\lambda; \sigma)$. Now, attribute authorities interact with each other and execute the following:

- k^{th} AA 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 AA 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} authority and j^{th} authority randomly choose $s_{kj} \in \mathbb{Z}_q$ such that $s_{kj} = s_{jk}$.
- k^{th} 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} AA 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 obtained 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} AA 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 AA):

* 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 $E_0 = h(M(a_{c,1})||M(a_{c,2})|| \ldots ||M(a_{c,n}))Y^{s_A+s_B}$, $E_1 = g_2^{s_A}$, $\left\{C_{k,i} = T_{k,i}^{s_A}\right\}, i \in \mathbb{A}_A^k, \forall k \in [1, ..., N]$.
* 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 $\hat{e}(S_{k,i}, C_{k,i}) = \hat{e}(g_1, g_2)^{s_A p_{k,u}(i)}$.
  * User interpolates all $\hat{e}(g_1, g_2)^{s_A p_{k,u}(i)}$ and gets $P_{k,u} = \hat{e}(g_1, g_2)^{s_A p_{k,u}(0)} = \hat{e}(g_1, g_2)^{s_A\left(v_k - \sum_{j\neq k} R_{kj}\right)}$.
* 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 $h(M(a'_{c,1})||M(a'_{c,2})|| \ldots ||M(a'_{c,n}))$.
* User can decrypt the data as follows (only if $h(M(a'_{c,1})||M(a'_{c,2})|| \ldots ||M(a'_{c,n})) = h(M(a_{c,1})||M(a_{c,2})|| \ldots ||M(a_{c,n}))$)

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

Fig. 4. Static and dynamic attribute based encryption algorithm scheme for multiple attribute authorities.

Decryption: The decryption algorithm takes the decryption credentials received from attribute authorities and dynamic parameters obtained from smart mobile device and the ciphertext as input and output the original data. The behavior-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 $h(M(a'_{c,1})||M(a'_{c,2})|| \ldots ||M(a'_{c,n})) = h(M(a_{c,1})||M(a_{c,2})|| \ldots ||M(a_{c,n}))$.

6. Performance analysis

In this section, we analyze the computation and communication costs associated with both the single and multi-authority algorithms proposed in this paper. As described in the related work section, the works related to the proposed algorithms are the conventional ABE schemes. Hence, the efficiencies of the proposed algorithms are demonstrated by comparing them against the conventional ABE schemes.

6.1. Computational complexity

Let us consider the single authority ABE scheme followed by MA-ABE scheme. In a single authority ABE scheme (i.e., Fig. 1), the user is involved in the computation during the decryption step and the data owner is involved in the encryption step. We can ignore the computational costs involved in the setup and key-issuing steps since those can be done during the idle time. Since, the computational cost for hash function is negligible compared to pairing and exponentiation, let us denote the computational time (in ms) for one multiplication, one exponentiation, and one pairing as C_m, C_{ex}, and C_p, respectively.

Table 2
Time complexity measures for two different testbeds.

	Testbed 1 (ms)	Testbed 2 (ms)
C_p	14.6	491.2
C_{ex}	2.8	34.1
C_m	1.8	20

For comparison, let us use the benchmark time values given with popular pairing-based cryptography library namely jPBC in [36]. Table 2 shows the time values (in ms) for C_m, C_{ex}, and C_p for two different testbeds: testbed 1 uses Intel(R) Core(TM) 2 Quad CPU Q6600 with 2.40 GHz and 3 GB memory running on Ubuntu 10.04 and testbed 2 uses HTC Desire HD A9191 smart phone running on Android 2.2. The time values given in Table 2 are for a symmetric elliptic curve

Table 3
Comparison of computational cost for the single authority ABE scheme and the proposed scheme.

	ABE scheme	Proposed scheme
Enc.	$(n+1)C_{ex}+C_m$	$(n+2)C_{ex}+2C_m$
Dec.	nC_p+nC_m	$nC_p+(n+2)C_m$

Table 4
Comparison of computational cost for the MA-ABE scheme and the proposed scheme.

	MA-ABE scheme	Proposed scheme
Enc.	$(nK+2)C_{ex}+C_m$	$(nK+3)C_{ex}+2C_m$
Dec.	$(nK+1)C_p+(nK+1)C_m$	$(nK+1)C_p+(nK+3)C_m$

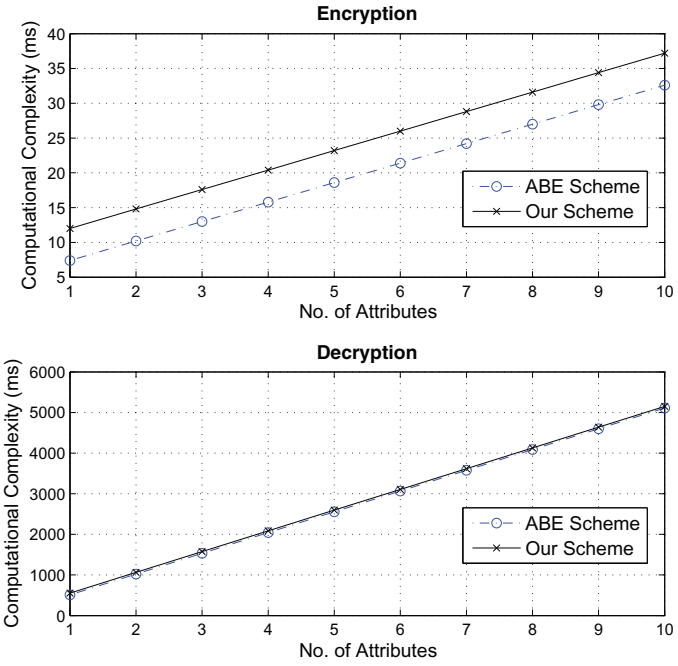

Fig. 5. Computational complexity comparison for single authority case.

called *a*-curve, where the base field size is 512-bit and the embedding degree is 2. The *a*-curve has a 160-bit group order. Let us assume that the data owner uses an environment similar to the testbed 1 for the encryption while user uses a mobile device similar to the testbed 2 for the decryption.

Let us denote the number of attributes used for encryption as n and the total number of dynamic attributes used by data owner as d. Table 3 shows the total time required for encryption (by the data owner) and for decryption (by the user) for the proposed and the conventional ABE schemes for the single AA. In order to graphically visualize the actual difference between proposed and conventional algorithms, we plotted the computational complexities given in Table 3 by varying the number of attributes, n, in Fig. 5.

Fig. 5 shows the computational complexity of the conventional ABE scheme against the proposed scheme. The computational complexity is measured in terms of total time required for the data owner and the user to encrypt and decrypt the data, respectively. For the encryption, our scheme consumes nearly 9 ms more than the conventional ABE. However, the proposed scheme incorporates the dynamic attributes during the encryption which provides run-time security to the data owner. It is worth noting from Fig. 5 that the time difference between our scheme and the conventional ABE for encryption is independent of the number of attributes (i.e., time complexity orders for both the schemes are same for encryption). However, our scheme is capable of including dynamic attributes on top of the regular attributes. For the decryption, it is obvious from Fig. 5 that our scheme performs equally well as the conventional ABE scheme. Since the decryption is performed in smart device (i.e., testbed 2) which is less powerful than desktop computer (i.e., testbed 1), the decryption time is almost 100 times higher than the encryption time.

Now let us compare our static and dynamic MA-ABE based algorithm against conventional MA-ABE scheme. Table 4 compares the computational complexity of the proposed MA-ABE algorithm with Chase and Chow's MA-ABE scheme in [32]. We denote the total number of attribute authorities in the system as K whereby each AA maintains N number of attributes (for simplicity, we assumed that all attribute authorities maintain equal number of attributes).

Fig. 6 compares both the proposed scheme and the conventional MA-ABE scheme in terms of time complexity for different numbers of attribute authorities (i.e., $K=\{2,4,6,8,10\}$). Encryption and decryption time increases with the total number of attribute authorities. For the encryption, similar to single authority case, the time complexity orders of both schemes are same (i.e., our scheme consume nearly 5ms more than conventional MA-ABE scheme irrespective of number of attributes and number of attribute authorities). Moreover, for decryption, our scheme performs equally well as the conventional MA-ABE scheme regardless of number of attribute authorities involved in the encryption.

Remark. One of the drawbacks of the existing ABE schemes is that the complexity increases linearly with the number of static attributes. Since our algorithm was built on top of the existing ABE scheme the same follows. If the data owner or employer wants to use attribute authorities to issue credentials for dynamic attributes then complexity will increase linearly since those dynamic attributes become static attributes. However, in our solution, as seen from Fig. 7, any number of dynamic attributes can be added for negligible cost. Hence, the complexity can be reduced by reducing the number of static attributes and adding more dynamic attributes used for encryption. For example, instead of including ten static attributes from attribute authorities, it is possible in our scheme that the data owner can include five static attributes from attribute authorities and another five dynamic attributes. This approach reduces the complexity by half. However, the proposed scheme adds additional layer of security on top of the conventional ABE schemes. In a nutshell, the proposed schemes do not degrade the performance of conventional ABE while including the dynamic attributes to provide run-time security to the data owner's data.

6.2. Communications complexity

Now we discuss communication costs for the proposed schemes and the conventional ABE schemes. For both the schemes, the communication costs are relying on the key-issuing step and when uploading and downloading the data. Since, key issuing step is purely dependent on the communication between attribute authorities and the data owner, communication costs for our and conventional schemes in this step are equal. During uploading and downloading the data, the additional components added to the proposed schemes are E_0 and A_C. It should be noted that the size of E_0 is 160-bits. A_C denotes the dynamic attributes used during the encryption, hence 2^d number of bits required to represent one dynamic attributes e.g., if the system consider five dynamic attributes then 32-bits required to denote each dynamic attribute (see Figs. 2 and 3). Overall, the increment in the communications cost in the proposed algorithm is negligible.

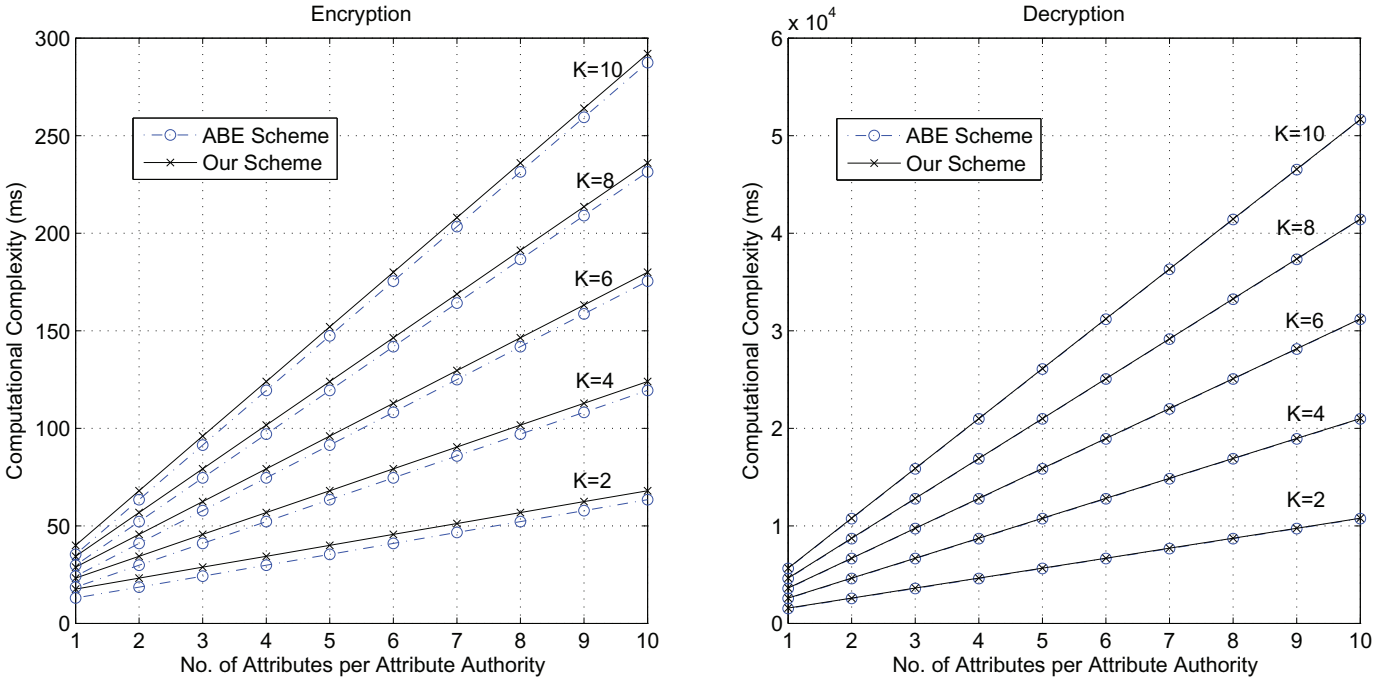

Fig. 6. Computational complexity comparison when there are more than one AA.

7. Security and privacy analysis

In Section 3.3, we categorized the possible security and privacy threats to the proposed algorithms. In this section, we address each issue and validate that our algorithm is robust against those security and privacy threats.

7.1. Mitigate identity threat

Adversary can impersonate as an AA or as users. Let us discuss these in turn. As shown in Figs. 3 and 4, public-keys associated with attribute authorities will be published online and the corresponding private-keys are known only to the authorities. At the same time, according to the modulo arithmetic, it is infeasible to compute private-keys from public-keys. During the encryption and decryption, data owners and users use attribute authorities. Data owner and users can verify the public-keys using well-known techniques such as certificates. Hence, impersonating AA is not possible.

User device might be at the possession of an attacker where user static attribute credentials are stored on the device. However, adversary cannot get access in to the network without satisfying the dynamic attributes introduced in this paper. Adversary behavior may not be similar as the legitimate user, hence, behavior-profiling app running in the user device increases the risk level which will eventually alert the network to deny the service request.

7.2. Mitigate privacy threats

User's privacy is vulnerable when user interacts with attribute authorities in order to get decryption keys i.e., in key issuing sub-algorithms. The proposed schemes were built on top of conventional ABE architectures [32]. In [32], users and attribute authorities execute anonymous key issuing protocol where user can obtain decryption keys for the attributes without revealing identity. In Fig. 4, in key issuing sub-algorithm, decryption key for user u obtained from authority j is $D_{kj} = g_1^{R_{kj}} y_k^{x_j/(s_{kj}+u)}$. This D_{kj} was obtained by executing the anonymous key-issuing protocol where user's identity u was incorporated within decryption key. However, the authority cannot be able to know the identity of the users, which preserve the user privacy.

7.3. Mitigate collusion attacks

Two different types of collusion attacks are possible: (1) attribute authorities can collide with each other and aggregate the user attributes, (2) users can pool their decryption keys to access the data which cannot be accessed by individual users. Since our schemes were built top of the conventional ABE scheme, the proposed schemes also collusion resistance against up to $(N-2)$ attribute authorities. Hence, let us discuss the user collusion. During the key issuing sub-algorithms, due to the inherent anonymous key issuing protocol, user u will obtain only $D_{kj} = g_1^{R_{kj}} y_k^{x_j/(s_{kj}+u)}$ where user identity u incorporated within decryption key by inverse exponentiation operation after adding u with random value s_{kj} (known only to authority). In modulo arithmetic, it is infeasible to infer $x_j/(s_{kj}+u)$ from $y_k^{x_j/(s_{kj}+u)}$. Moreover, the user identity was randomized by s_{kj}, it is impossible to modify u with other user's identity.

7.4. Mitigate dynamic attribute cheating

The behavior-profiling app installed on the user's mobile device can be used to verify whether the current user is the legitimate user of the mobile device [26,27]. However, since the behavior-profiling app is installed within user's device, malicious users might modify the app in order to feed false information for dynamic attributes. Recent technology development in smart device industry already has some working prototype for this kind of security vulnerability i.e., Samsung's KNOX [29] and Blackberry's BES [30]. These softwares are capable of securely installing apps on the users mobile devices and check for integrity of the installed apps without user interruption. Hence, modifying behavior-profiling app in order to feed false information can be easily detected by the data owner using either KNOX or BES. These software platforms are capable of securely installing corporate apps (i.e., behavior-profiling app) on the users mobile devices and check for integrity of the installed apps.

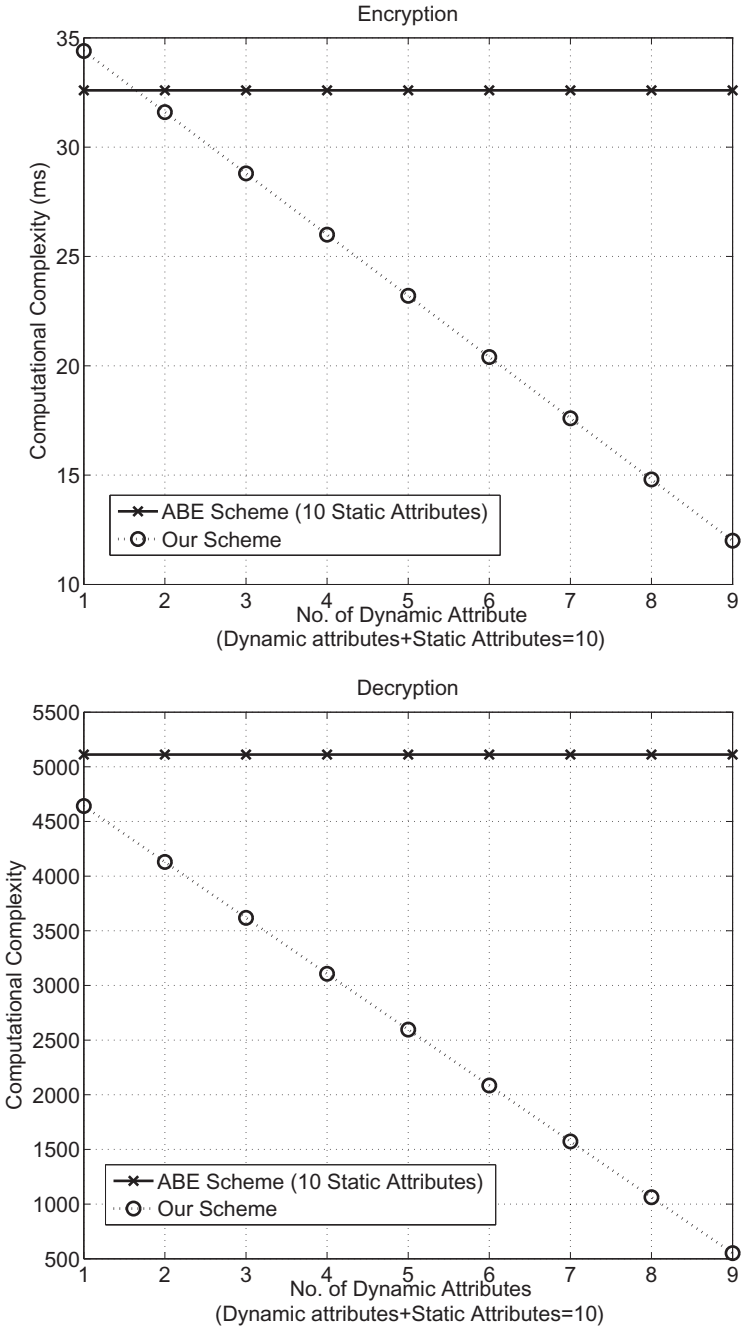

Fig. 7. Complexity can be reduced by our scheme if more dynamic attributes are incorporated within encryption.

7.5. Mitigate tracking threat

In the previous section we considered the user as an adversary. However data owner or employer can also be an adversary since their behavior profiling app collects sensor data from user's device. If the app is malicious then it is obvious that it will send the sensor data to the employer or third-parties who then can monitor or track the user. However, according to the proposed algorithm it is not necessary to send out the sensor data outside the mobile device since mapping is carried out within user device. Employer should certify or validate the app in order to build a trust among users. Since, it is easy to detect whether apps are behaving maliciously [44] we can expect that the employers will not develop an app which send out the sensor data outside the mobile device.

7.6. Security analysis

Theorem 1. *The proposed scheme is semantically secure against chosen plain text attack (CPA) in the selective ID model, if there exist negligible function ϵ such that, in the security game explained earlier any adversary will succeed with probability at most $\frac{1}{2}+\epsilon$.*

Proof. Suppose if there is a probabilistic polynomial time adversary who can break our algorithm then there will be a challenger who can break the decisional MBDH assumption by exploiting the adversary. Let's assume that the challenger is provided with $[g_1^a, g_1^b, g_1^c, Z]$ and if the challenger wants to break the MBDH assumption then he needs to determine whether $Z = e(g,g)^{\frac{ab}{c}}$ or not with at least $\frac{1}{2}+\epsilon$ probability.

Let us assume that there is an adversary who can break the proposed algorithm. In this section, we will show that the challenger can use such an adversary to break the MBDH assumption. In order to exploit such an adversary, the challenger needs to incorporate the given $[g^a, g^b, g^c, Z]$ within the proposed algorithm (i.e., Fig. 3). First of all, let us explain how the challenger incorporates $[g^a, g^b, g^c, Z]$ within global setup, authority setup, and key generation sub-algorithms. We stress here that this incorporation is indistinguishable from the steps provided in Fig. 3.

Initially, as explained in the security game, the adversary must submit a set of attributes he wants to challenge. One of the conditions as given in security game is that at least there will be one attribute for each set whereby the adversary can get insufficient number of decryption credentials [31].

Let's assume the adversary sends a set of attributes, α, to the challenger. The challenger assigns the public key parameters as follows. It sets the parameter $Y = e(g, A) = e(g, g)^a$. For all $i \in \alpha$ it chooses random $\beta_i \underleftarrow{R} \mathbb{Z}_p$ and sets $T_i = C^{\beta_i} = g^{c\beta_i}$. For all other attributes, it chooses random $w_i \underleftarrow{R} \mathbb{Z}_p$ and sets $T_i = g^{w_i}$. It then gives the public parameters to adversary. Notice that from the view of adversary all parameters are chosen at random as in the construction.

Suppose an adversary requests a private key for attribute set γ where $|\gamma \cap \alpha| < d$. We first define three sets Γ, Γ', and S in the following manner: $\Gamma = |\Gamma \cap \alpha|$, Γ' can be any set such that $\Gamma \subseteq \Gamma' \subseteq \gamma$ and $|\Gamma| = d - 1$ and $S = \Gamma' \bigcup \{0\}$. Now let us define decryption keys $D_u(i)$ for $i \in \Gamma'$ as follows: if $i \in \Gamma$ then $D_u(i) = g^{s_i}$ where $s_i \underleftarrow{R} \mathbb{Z}_p$. If $i \in \Gamma' - \Gamma$ then $D_u(i) = g^{\frac{\lambda_i}{w_i}}$ where $\lambda_i \underleftarrow{R} \mathbb{Z}_p$. □

The intuition behind these assignments is that we are implicitly choosing a random $d-1$ degree polynomial $q(x)$ by choosing its value for the $d-1$ points randomly in addition to having $q(0) = a$. For $i \in \Gamma$ we have $q(i) = c\beta_i s_i$ and for $i \in \Gamma' - \Gamma$ we have $q(i) = \lambda_i$. The challenger can calculate the other $D_u(i)$ values where $i \notin \Gamma'$ since the challenger knows the discrete log of T_i. The challenger makes the assignments as follows: If $i \notin \Gamma'$ then

$$D_u(i) = \left(\prod_{j \in \Gamma} C^{\frac{\beta_j s_j \Delta_{j,S(i)}}{w_i}}\right)\left(\prod_{j \in \Gamma' - \Gamma} g^{\frac{\lambda_j \Delta_{j,S(i)}}{w_i}}\right)\left(Y^{\frac{\Delta_{0,S(i)}}{w_i}}\right).$$

Using polynomial interpolation the challenger is able to calculate $D_u(i) = g^{\frac{q(i)}{t_i}}$ for $i \notin \Gamma'$ where $q(x)$ was implicitly defined by the random assignment of the other $d-1$ variables $D_u(i) \in \Gamma$ and the variable Y. Hence, the challenger is able to construct a private key for the attribute set γ. Moreover, the distribution of the private key for γ is identical to that of the original scheme.

Now the adversary submits two challenge messages m_1 and m_0 to the challenger. The simulator flips a fair binary coin, υ, and returns an encryption of m_υ. The ciphertext is output $CT_m = \{E_0 = h(M(a_{c,1})||M(a_{c,1})||\ldots M(a_{c,n}))m_\upsilon Z, E_i = B^{\beta_i}_{i \in \alpha}\}$. If $\upsilon = 0$ then $Z = e(g, g)^{\frac{ab}{c}}$. If we let $s_A + s_B = \frac{b}{c}$, then we have $E_0 = h(M(a_{c,1})||M(a_{c,1})||\ldots M(a_{c,n}))m_\upsilon Y^{s_A+s_B}$ and $E_i = \frac{B^{\beta_i}}{C^{s_B}} = g^{b\beta_i - s_B} = g^{\frac{b}{c}c\beta_i - cs_B} = (T_i)^{s_A}$. Therefore, the ciphertext is a random encryption of the message m_υ under the public key α. Otherwise, if $\upsilon = 1$, then $Z = g^z$. We then have $E_0 = h(M(a_{c,1})||M(a_{c,1})||\ldots M(a_{c,n}))m_\upsilon e(g, g)^z$. Since z is random, E_0 will be a random element and the adversaries view and the message contains no information about m_υ [10].

We stress here that CT_m is a valid encryption of the message m_υ if $Z = e(g, g)^{\frac{ab}{c}}$. Hence, the adversary should have his usual non-negligible advantage ϵ of correctly identifying the message m_υ. However, when $Z \neq e(g, g)^{\frac{ab}{c}}$, then CT_m is just random value, hence, the adversary can have no more than $\frac{1}{2}$ probability of guessing correctly. Hence, if the adversary guesses correctly then challenger guesses that $Z = e(g, g)^{\frac{ab}{c}}$ and if adversary is wrong then challenger guesses that $Z \neq Z = e(g, g)^{\frac{ab}{c}}$, hence, the challenger has an advantage of $\frac{\epsilon}{2}$ in distinguishing whether $Z = e(g, g)^{\frac{ab}{c}}$. Hence, an adversary who breaks our scheme with advantage ϵ implies an algorithm for breaking MBDH assumption with non-negligible advantage $\frac{\epsilon}{2}$. We can conclude that the proposed scheme is selective ID secure.

Similarly this proof can be extended to multi-authority system. As shown in [32], for the multi-authority case, we divide the authorities into two: honest authorities and corrupted authorities. First, we have to set up parameters so that we can set any of our authorities as the one that corresponds to the uncomputable portion of the master key. Then an AA $k*$ chosen at random form its parameters based on this uncomputable value. If it turns out that this is the honest authority from which the adversary requests insufficient attributes for user u, then we are all set, and we can simply reuse the above technique [32].

8. Conclusions, limitations and future works

In this paper, we proposed robust access control technique which incorporates attributes generated by smart devices to secure the conventional access control framework. In the proposed schemes, data owner incorporates smart device's dynamic attributes together with predefined static attributes. This approach adds additional layer of security on top of the security available in conventional access control framework. We showed that the efficiencies of the proposed schemes are comparable to that of the conventional schemes while offering better security and flexibility for mobile computing network.

8.1. Limitations and future works

Collecting and processing the sensor data to determine the values for dynamic attributes increase the time or communication complexity. At present it is assumed that this will be done in off-line or in parallel to downloading the encrypted data from the cloud. Evaluating this latency for different smart devices in various environments could be a potential extension.

Another limitation is the accuracy or number of algorithms available for detecting a user behavior. Potential extension could be on developing an app which aggregates data from all the smart device sensors to profile the user's behavior. Multiple physical activities such as the way individuals walk or the way we take out the phone from pocket can be used to profile a user. Developing a novel algorithm using machine learning techniques to classify users based on behavior is important to bridge the gap between theory and practice.

There are several variants of KP-ABE in literature [45–47]. These variants either enhance the security by adopting fully secure model or improve the complexity by fast decryption technique and outsourcing the pairing computations to the cloud. Hence repeating the proposed technique, i.e., adding dynamic attributes, on top of these schemes will further improve the complexity as well as the security.

Acknowledgments

Mauro Conti is supported by a EU Marie Curie Fellowship for the project PRISM-CODE (grant no. PCIG11-GA-2012-321980). This work has been partially supported by the TENACE PRIN Project (grant no. 20103P34XC) funded by the Italian MIUR.

References

[1] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, C.E. Youman, Role-based access control models, Comput. 2 (1996) 38–47.
[2] G. Zhang, M. Parashar, Context-aware dynamic access control for pervasive applications, in: Proceedings of the Communications Networks and Distributed Systems Modeling and Simulation Conf., 2004, pp. 21–30.
[3] P. Bonatti, C. Galdi, D. Torres, ERBAC: event-driven RBAC, in: Proceedings of the 18th ACM Symposium Access Control Models and Technologies. ACM, 2013, pp. 125–136.

[4] E. Bertino, P.A. Bonatti, E. Ferrari, TRBAC: a temporal role-based access control model, ACM Trans. Inf. Syst. Secur. (TISSEC) 4 (3) (2001) 191–233.
[5] F. Hansen, V. Oleshchuk, SRBAC: a spatial role-based access control model for mobile systems, in: Proceedings of the 7th Nordic Workshop on Secure IT Systems (NORDSEC03), 2003, pp. 129–141.
[6] M.S. Kirkpatrick, E. Bertino, Enforcing spatial constraints for mobile RBAC systems, in: Proceedings of the 15th ACM Symposium Access Control Models and Technologies, 2010, pp. 99–108.
[7] I. Ray, M. Kumar, L. Yu, LRBAC: a location-aware role-based access control model., Information Systems Security, Springer, Berlin Heidelberg, 2006, pp. 147–161.
[8] Y.S. Cho, L. Bao, M.T. Goodrich, LAAC: a location-aware access control protocol, IEEE 3rd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (2006) 1–7.
[9] M. Toahchoodee, et al., Ensuring spatio-temporal access control for real-world applications, in: Proceedings of the 14th ACM Symposium Access Control Models and Technologies, 2009, pp. 13–22.
[10] A. Sahai, B. Waters, Fuzzy Identity-based encryption, Adv. Cryptol. EUROCRYPT 3494 (2005).
[11] V. Goyal, O. Pandey, A. Sahai, B. Waters, Attribute-based encryption for fine-grained access control of encrypted data, in: Proceedings of the 13th ACM Conference on Computer and Communication Security, New York, USA, 2006, pp. 89–98.
[12] J. Bethencourt, A. Sahai, B. Waters, Ciphertext-policy attribute-based encryption, in: IEEE Symposium on Security and Privacy, SP 07, 2007, pp. 321–334.
[13] F. Li, Y. Rahulamathavan, M. Rajarajan, R.C.-W. Phan, Low complexity multi-authority attribute based encryption scheme for mobile cloud computing, in: Proceedings of the IEEE 7th International Symposium on Service Oriented System Engineering (SOSE), San Francisco, USA, 2013, pp. 573–577.
[14] Cisco Study: IT Saying Yes to BYOD, Cisco, http://tinyurl.com/d8fv2uj, May 2012 (accessed on 15 July 2015).
[15] K.W. Miller, J. Voas, G.F. Hurlburt, BYOD: security and privacy considerations, IT Prof. 14 (5) (2012) 53–55.
[16] G. Thomson, BYOD: Enabling the chaos, Net. Secur. 2012 (2) (2012) 5–8.
[17] A. Boonyarattaphan, Y. Bai, S. Chung, R. Poovendran, Spatial-temporal access control for e-health services, in: Proceedings of the 5th IEEE International Conference on Networking, Architecture and Storage (NAS), 2010, pp. 269–276.
[18] L. Scott, D.E. Denning, A location based encryption technique and some of its applications, in: Proceedings of the National Technical Meeting of The Institute of Navigation, Anaheim, CA, 2003, pp. 734–740.
[19] L. Hsien-Chou, C. Yun-Hsiang, A new data encryption algorithm based on the location of mobile users, Inf. Technol. J. 7 (1) (2008) 63–69.
[20] O. Al-Ibrahim, A. Al-Fuqaha, D.V. Dyk, N. Akerman, Mobility support for geo-encryption, in: Proceedings of the IEEE International Conference on Communication, 2007, pp. 1492–1496.
[21] V. Vijayalakshmi, T.G. Palanivelu, Secure localization using elliptic curve cryptography in wireless sensor networks, Int. J. Comp. Sci. Netw. Secur. 8 (6) (2008) 255–261.
[22] R. Karimi, M. Kalantari, Enhancing security and confidentiality in location-based data encryption algorithms, in: Proceedings of the 4th International Conference on Applications of Digital Information and Web Technologies (ICADIWT), 2011, pp. 30–35.
[23] R. Karimi, M. Kalantari, Enhancing security and confidentiality on mobile devices by location-based data encryption, in: Proceedings of the 17th IEEE International Conference on Network, 2011, pp. 241–245.
[24] F. Li, N. Clarke, M. Papadaki, P. Dowland, Misuse detection for mobile devices using behaviour profiling, in: Proceedings of the International Journal of Cyber Warfare and Terrorism, vol. 1, 2011, pp. 41–53.
[25] M. Miettinen, P. Halonen, K. Hatonen, Host-based intrusion detection for advanced mobile devices, in: Proceedings of the 20th International Conference Advanced Information Networking and Applications, Washington, DC, USA, 2006, pp. 72–76.
[26] F. Li, Behaviour profiling for mobile devices, Plymouth University, UK, 2012 Ph.d. thesis.
[27] N. Eagle, A.S. Pentland, Reality mining: sensing complex social systems, J. Pers. Ubiquitous Comput. 10 (4) (2006) 255–268.
[28] A. Gupta, M. Miettinen, N. Asokan, Using context-profiling to aid access control decisions in mobile devices, in: Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), Seattle, WA, 2011, pp. 310–312.
[29] Samsung KNOX, http://tinyurl.com/me93jcv (accessed on 15 July 2015).
[30] BlackBerry BES12, http://tinyurl.com/ls3yxh8 (accessed on 15 July 2015).
[31] M. Chase, Multi-authority attribute based encryption, in: Lecture Notes of Theory of Cryptography in Computer Science, Berlin Heidelberg, 2007, pp. 515–534.
[32] M. Chase, S.S.M. Chow, Improving privacy and security in multi-authority attribute-based encryption, in: Proceedings of the 16th ACM Conference on Computer and Communication Security, New York, NY, USA, 2009, pp. 121–130.
[33] A.B. Lewko, B. Waters, Decentralizing attribute-based encryption, in: EUROCRYPT, Ser. LNCS, vol. 6632, Springer, 2011, pp. 568–588.
[34] 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, Springer, Berlin Heidelberg, 2013, pp. 142–150.
[35] K. Zaidi, Y. Rahulamathavan, M. Rajarajan, DIVA - digital identity in VANETs: a multi-authority framework for VANETs, in: Proceedings of the 19th IEEE International Conference on Network (ICON'13), Singapore, 2013.
[36] The Java Pairing-Based Cryptography Library (JPBC), http://tinyurl.com/ll2p39t (accessed on 15 July 2015).
[37] M. Conti, B. Crispo, E. Fernandes, Y. Zhauniarovich, CRêPE: a system for enforcing fine-grained context-related policies on android, IEEE Trans. Inf. Forensics Secur. vol. 7 (2012) 1426–1438.
[38] K. Ariyapala, M. Conti, C. Keppitiyagama, ContextOS: a context aware operating system for mobile devices, in: Proceedings of the IEEE International Conference on Cyber, Physical and Social Computing, Beijing, China, 2013.
[39] M. Conti, V.T.N. Nguyen, B. Crispo, CRePE: context-related policy enforcement for android, in: Proceedings of the 13th Information Security Conference, Boca Raton, FL, USA, 2010, pp. 331–354.
[40] G. Russello, M. Conti, B. Crispo, E. Fernandes, Y. Zhauniarovich, DEMO: demonstrating the effectiveness of MOSESdroid for separation of execution modes, in: Proceedings of the 19th ACM Conference Computer Communication Security, Raleigh, NC, USA, 2012, pp. 998–1000.
[41] G. Russello, M. Conti, B. Crispo, E. Fernandes, MOSES: supporting operation modes on smartphones, in: Proceedings of the 17th ACM Symposium Access Control Models and Technologies, Newark, NJ, US, 2012, pp. 3–12.
[42] Y. Zhauniarovich, G. Russello, M. Conti, B. Crispo, E. Fernandes, MOSES: supporting and enforcing security profiles on smartphones, IEEE Trans. Dependable Secure Comput 11 (3) (2014) 211–222.
[43] F. Li, Y. Rahulamathavan, M. Rajarajan, LSD-ABAC: Lightweight Static and Dynamic Attributes based Access Control Scheme for Secure Data Access in Mobile Environment, in: Proceedings of the IEEE 39th Conference Local Computer Networks (LCN), 2014, pp. 354–361.
[44] Y. Rahulamathavan, V. Moonsamy, L. Batten, S. Shunliang, M. Rajarajan, An analysis of tracking service settings in blackberry 10 and windows phone 8 smartphones, in: Proceedings of the 19th Australasian Conference on Information Security and Privacy (ACISP), 2014.
[45] S. Hohenberger, B. Waters, Attribute-based encryption with fast decryption, in: Proceedings of the Public-Key Cryptography, 2013, pp. 162–179.
[46] A. Lewko, et al., Advances in cryptology-EUROCRYPT 2010, Springer, Berlin Heidelberg, 2010, pp. 62–91.
[47] Y. Rouselakis, B. Waters, Practical constructions and new proof methods for large universe attribute-based encryption, in: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2013, pp. 463–474.