Decentralized Ciphertext-Policy Attribute-Based Encryption Scheme with Fast Decryption

Y. Rao, Ratna Dutta

Decentralized Ciphertext-Policy Attribute-Based Encryption Scheme with Fast Decryption

Y. Sreenivasa Rao and Ratna Dutta

Department of Mathematics
Indian Institute of Technology Kharagpur
Kharagpur-721302, India
{ysrao,ratna}@maths.iitkgp.ernet.in

Abstract. In this paper, we propose an efficient multi-authority decentralized ciphertext-policy attribute-based encryption scheme dCP-ABE-MAS for monotone access structures (MAS). Our setup is without any central authority (CA) where all authorities function entirely independently and need not even be aware of each other. The scheme makes use of the minimal authorized sets representation of MAS to encrypt messages, and hence the size of ciphertext is linear in the number of minimal authorized sets in MAS and the number of bilinear pairings is *constant* during decryption. We describe several networks that can use dCP-ABE-MAS to control data access from unauthorized nodes. The proposed scheme resists collusion attacks and is secure against chosen plaintext attacks in the generic bilinear group model over prime order bilinear groups.

Keywords: attribute-based encryption, decentralized, multi-authority, monotone access structure.

1 Introduction

In Attribute-Based Encryption (ABE), each user is ascribed a set of descriptive attributes (or credentials), and secret key and ciphertext are associated with an access policy or a set of attributes. Decryption is then successful only when the attributes of ciphertext or secret key satisfy the access policy. ABE is classified as Key-Policy ABE (KP-ABE) [3] or Ciphertext-Policy ABE (CP-ABE) [4] according to whether the secret key or ciphertext is associated with an access policy, respectively. Since the invention of ABE [2], several improved ABE schemes [3–6] have been proposed. All the foregoing ABE schemes make use of a single trusted central authority (CA) to control the universe of attributes and issue secret keys to users that should not be compromised at all. Consequently, the CA can decrypt every ciphertext in the system encrypted under any access policy by calculating the required secret keys at any time, this is the *key escrow* problem of ABE. A solution to help mitigate the key escrow problem is distributing the functionality of the CA over many potentially untrusted authorities in such a way that as long as some of them are honest, the system would still be

secure. An ABE with this mechanism is the so-called *multi-authority* ABE. In this scenario, each authority controls a different domain of attributes and issues attribute-related secret keys to users.

Chase [10] devised the first multi-authority ABE as an affirmative solution to the open problem posed by Sahai and Waters [2] that consists of one fully trusted centralized authority (CA) and multiple (attribute) authorities. Every user is assigned a unique global identifier and the keys from different authorities are bound together by this identifier to counteract the *collusion attack*—multiple users can pool their secret keys obtained from different authorities to decrypt a ciphertext that they are not individually entitled to. As CA holds the system's master secret, it can decrypt all the ciphertexts in the system, thereby cannot the key escrow resists. The first CA-free multi-authority ABE is proposed by Lin et al. [9] wherein Distributed Key Generation (DKG) protocol and Joint Zero Secret Sharing (JZSS) protocol are deployed to remove CA. All authorities must interact to execute DKG and JZSS protocols during system setup phase. However, the scheme is collusion-resistant up to collusion of m users, where m is a system wide parameter that should be fixed during setup, and the number of JZSS protocol executions, the computation and communication costs are all linear in m. Chase and Chow [11] proposed CA-free multi-authority ABE with user privacy that resolves the key escrow problem using distributed Pseudo Random Functions (PRF). In this setting, each pair of authorities will communicate with each other via a 2-party key exchange protocol to generate users' secret keys during setup phase that incurs $\mathcal{O}(N^2)$ communication overhead on the system, where N is the fixed number of authorities. The foregoing constructions [10, 9, 11] can only handle a set of fixed number of authorities at system initialization which exploit AND-gate access policies in key-policy setting to prevent unauthorized data access.

Müller et al. [15] gave two multi-authority CP-ABE schemes which employ one CA and several authorities where the authorities work independently from each other. However, the CA can still decrypt all ciphertexts in the system. The first construction uses Disjunctive Normal Form (DNF) access policies to annotate ciphertexts, thereby achieves constant computation cost during decryption. The second scheme realizes any Linear Secret Sharing Scheme (LSSS) access policy and hence the computation cost for successful decryption is linear in minimum number of attributes required to compute the target vector, i.e., a vector that contains the secret as one of its components. Lewko and Waters [8] proposed a novel multi-authority CP-ABE scheme without CA that is decentralized, where all authorities function entirely independently and need not even be aware of each other. The concept of global identifier introduced by Chase [10] is used to "link" attribute-related secret keys together that are issued to the same user by different authorities, this in turn achieves collusion-resistant among any number of users. The same scheme works on both composite order and prime order bilinear groups. The security of the former is given in random oracle model and the security of latter one is analyzed in the generic group model. In both cases, the monotone access structures are realized by LSSS, the ciphertext size

Table 1. Comparison of [8] with Our (dCP-ABE-MAS) Scheme

Scheme	Key Generation		Encryption			Decryption		Access Policy
	$E_{\mathbb{G}}$	User Secret Key Size	$E_{\mathbb{G}}$	$E_{\mathbb{G}_T}$	Ciphertext Size	$E_{\mathbb{G}_T}$	P_e	
[8]	2γ	$\gamma B_{\mathbb{G}}$	3α	$2\alpha+1$	$2\alpha B_{\mathbb{G}} + (\alpha+1)B_{\mathbb{G}_T} + \tau$	$\mathcal{O}(\beta)$	$\mathcal{O}(\beta)$	LSSS
Our	2γ	$\gamma B_{\mathbb{G}}$	$2k$	k	$2kB_{\mathbb{G}} + kB_{\mathbb{G}_T} + \tau$	-	2	any MAS

$E_{\mathbb{G}}$ (or $E_{\mathbb{G}_T}$) = number of exponentiations in a group $\mathbb{G}$ (or $\mathbb{G}_T$, resp.), P_e = number of pairing computations, $B_{\mathbb{G}}$ (or $B_{\mathbb{G}_T}$) = bit size of an element of $\mathbb{G}$ (or $\mathbb{G}_T$, resp.), α = size of LSSS access structure, β = minimum number of attributes required for decryption, γ = number of attributes annotated to a user secret key, k = number of minimal sets in MAS, τ = size of an access structure.

is linear in the size of the LSSS, and the number of pairings is linear in the minimum number of attributes that satisfy the LSSS. Liu et al. [17] devised a LSSS-realizable multi-authority CP-ABE system which has multiple CAs and authorities. The scheme is adaptively secure without random oracles unlike [8].

In all the multi-authority KP/CP-ABE schemes except the one (CA based) in [15] discussed so far, the size of ciphertext is linear in the size of monotone span program or the number of attributes that are associated with ciphertexts and the number of bilinear pairing computations is linear in the minimum number of attributes required for successful decryption. Constant computation and low communication cost access control schemes are more practical where the computing resources have limited computing power and bandwidth is the primary concern. For these reasons, we provide a solution to help mitigate the problem of large ciphertext size and linear-size number of bilinear pairings in designing multi-authority ABE schemes.

Our Contribution. We propose dCP-ABE-MAS, which is a multi-authority CP-ABE in a decentralized setting for any monotone access structure (MAS). Every MAS, $\mathbb{A}$, can uniquely be represented by a set $\mathbb{A}_0$ of minimal authorized sets in $\mathbb{A}$ (see Section 2.1). This scheme has the same functionality as the most robust and scalable multi-authority CP-ABE [8] to date. Even though the schemes [11, 9] exclude the requirement of the CA, they are not fully decentralized as the number of authorities is fixed ahead of time and all authorities are communicating each other during system setup unlike [8]. That is why we compare (in Table[1] 1) our dCP-ABE-MAS only with the decentralized scheme[2] of [8] in view of prime order bilinear group setting.

The ciphertext size in [8] is linear in the size, α, of LSSS, while the size of ciphertext in our construction grows linearly with k, the number of minimal authorized sets in the MAS. For (t, n)-threshold policy, where $1 < t < n$, the value of $k = n!/(n-t)!\ t!$ which will be larger than n, whereas there exist a

[1] The description of all the symbols in Table 1,3,4 is given at the bottom of Table 1.

[2] The scheme that works on prime order bilinear group and the security is analyzed in the generic group model.

LSSS with size $\alpha = n$ to realize the (t, n)-threshold policy. However, there are several classes of MAS for which the value of k is constant but the size of the monotone span program (or LSSS) computing the MAS is at least polynomial in the number of attributes in the access structure. As a trivial case, if one uses a single AND-gate with n attributes, the value of k will be 1, while the size of LSSS is equal to n, i.e., $\alpha = n$. We now consider some non-trivial cases from [18]. Let $\mathbb{A}_0 = \{B_1 = \{a_1, \ldots, a_{\lceil n/2 \rceil}\}, B_2 = \{a_{\lceil n/2 \rceil + 1}, \ldots, a_n\}\}$ be the set of minimal sets for a MAS, $\mathbb{A}$, over n attributes $a_1, \ldots, a_n$. Then, $k = 2$ and the size, α, of LSSS computing $\mathbb{A}$ is at least $\mathcal{O}(n)$. Similarly, if $\mathbb{A}_0 = \{B_1 = \{a_1, \ldots, a_{\lceil n/3 \rceil}\}, B_2 = \{a_{\lceil n/3 \rceil + 1}, \ldots, a_{\lceil 2n/3 \rceil}\}, B_3 = \{a_{\lceil 2n/3 \rceil + 1}, \ldots, a_n\}\}$ is the set of minimal sets for a MAS, $\mathbb{A}$, then $k = 3$ but the size, α, of LSSS computing $\mathbb{A}$ is at least $\mathcal{O}(n)$ (for more details see Section 2.1 in [18]). Thus, in such cases, our dCP-ABE-MAS scheme exhibits shorter ciphertext. Moreover, our approach requires only 2 pairing computations to decrypt any ciphertext. The user secret key size is linear in the number of attributes associated with the user.

An inherent drawback of [8] is that every authority can independently decrypt every ciphertext in the system, if the set of attributes controlled by the authority satisfies the LSSS access structure associated with the ciphertext. However, this can be avoided if each authorized set contains attributes from at least two different authorities. The same problem can be eliminated in our dCP-ABE-MAS if each minimal authorized set contains attributes from at least two different authorities. This fact follows from satisfiability condition given in Definition 2.

We discuss how our dCP-ABE-MAS can provide attractive solutions to fine-grained access control in various network scenarios and compare our work with the existing works in the area. Additionally, our multi-authority scheme provides a mechanism for packing multiple messages in a single ciphertext. This in turn reduces network traffic significantly. The proposed scheme is proven to be collusion-resistant and is secure against chosen plaintext attacks in the generic bilinear group model. To the best of our knowledge, our proposed multi-authority CP-ABE scheme is the only scheme in a decentralized framework where the decryption time is constant for general MAS.

2 Preliminaries

Definition 1. *Let $\mathbb{G}$ and $\mathbb{G}_T$ be multiplicative cyclic groups of prime order p. Let g be a generator of $\mathbb{G}$. A mapping $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is said to be bilinear if $e(u^a, v^b) = e(u, v)^{ab}$, for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$ and non-degenerate if $e(g, g) \neq 1_T$ (where, 1_T is the unit element in $\mathbb{G}_T$). We say that $\mathbb{G}$ is a bilinear group if the group operation in $\mathbb{G}$ can be computed efficiently and there exists $\mathbb{G}_T$ for which the bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is efficiently computable.*

2.1 Access Structure

In this section, we briefly review the concept of general access structures [7].

Let U be the universe of attributes and $|U| = n$. Let $\mathcal{P}(U)$ be the collection of all subsets of U. Every subset of $\mathcal{P}(U) \setminus \{\emptyset\}$ is called an *access structure.* An access structure $\mathbb{A}$ is said to be *monotone access structure* (MAS) if

$$\{C \in \mathcal{P}(U) | C \supseteq B, \text{ for some B} \in \mathbb{A}\} \subseteq \mathbb{A}.$$

The sets in $\mathbb{A}$ are called the authorized sets and the sets not in $\mathbb{A}$ are called the unauthorized sets with respect to the monotone access structure $\mathbb{A}$. Then every superset of an authorized set is again authorized set in MAS.

A set B in a monotone access structure $\mathbb{A}$ is a *minimal authorized set* in $\mathbb{A}$ if there exists a set $D(\neq B)$ such that $D \subseteq B$, then $D \notin \mathbb{A}$. The set of all minimal authorized sets of $\mathbb{A}$, denoted by $\mathbb{A}_0$, is called the *basis* of $\mathbb{A}$. Then we can generate $\mathbb{A}$ from its basis $\mathbb{A}_0$ as follows:

$$\mathbb{A} = \{C \in \mathcal{P}(U) | C \supseteq B, \text{ for some B} \in \mathbb{A}_0\}. \tag{1}$$

Lemma 1. *The monotone access structure $\mathbb{A}$ given in Eq. (1) is generated uniquely from its basis $\mathbb{A}_0$.*

Proof. Suppose $\mathbb{A}'$ is a monotone access structure generated from $\mathbb{A}_0$. Then $\mathbb{A}' = \{C' \in \mathcal{P}(U) | C' \supseteq B', \text{ for some B}' \in \mathbb{A}_0\}$. We shall prove that $\mathbb{A} = \mathbb{A}'$. Let $C \in \mathbb{A}$. Then by Eq. (1), we have $U \supseteq C \supseteq B$, for some $B \in \mathbb{A}_0$ and hence $C \in \mathbb{A}'$. Therefore, $\mathbb{A} \subseteq \mathbb{A}'$. Similarly, we can have $\mathbb{A}' \subseteq \mathbb{A}$. Thus, $\mathbb{A} = \mathbb{A}'$. □

In sum, every monotone access structure can be represented by its basis.

Definition 2. *Let $\mathbb{A}$ be a monotone access structure and $\mathbb{A}_0$ be its basis. A set, L, of attributes satisfies $\mathbb{A}$, denoted as $L \models \mathbb{A}$ if and only if $L \supseteq B$, for some $B \in \mathbb{A}_0$, and otherwise L does not satisfy $\mathbb{A}$, denoted as $L \not\models \mathbb{A}$.*

3 Decentralized CP-ABE System

A decentralized CP-ABE system is composed mainly of a set $\mathcal{A}$ of authorities, a trusted initializer and users. The *only* responsibility of trusted initializer is generation of system global public parameters, which are system wide public parameters available to every entity in the system, once during system initialization. Each authority $A_j \in \mathcal{A}$ controls a different set U^j of attributes and issues corresponding secret attribute keys to users. We note here that all authorities will work independently. As such, every authority is completely unaware of the existence of the other authorities in the system. Each user in the system is identified with a unique global identity $\mathsf{ID} \in \{0,1\}^*$ and is allowed to request secret attribute keys from the different authorities. At any point of time in the system, each user with identity ID possesses a set of secret attribute keys that reflects a set L_{ID} of attributes, which we call an attribute set of the user with identity ID.

Let $U = \bigcup_{A_j \in \mathcal{A}} U^j$, where $U^{j_1} \cap U^{j_2} = \emptyset$, for all $j_1 \neq j_2$, be the attribute universe of the system. Due to lack of global coordination between authorities,

different authorities may hold the same attribute string. To overcome such scenario, we can treat each attribute as a tuple consisting of the attribute string and the controlling authority identifier, for example ("`supervisor`", j), where the attribute "`supervisor`" is held by the authority A_j. Consequently, the attributes ("`supervisor`", j_1) and ("`supervisor`", j_2) will be considered as distinct as long as $j_1 \neq j_2$.

The decentralized CP-ABE system consists of the following five algorithms.
System Initialization(κ). At the initial system setup phase, a trusted initializer chooses global public parameters GP according to the security parameter κ. Any authority or any user in the system can make use of these parameters GP in order to perform their executions.
Authority Setup(GP, U^j). This algorithm is run by every authority $A_j \in \mathcal{A}$ *once* during initialization. It accepts as input the global public parameters GP and a set of attributes U^j for the authority A_j and outputs public key PubA_j and master secret key MkA_j of the authority A_j.
Authority KeyGen($\mathsf{GP}, \mathsf{ID}, a, \mathsf{MkA}_j$). Every authority executes this algorithm upon receiving a secret attribute key request from the user. It will take as input global public parameters GP, a global identity ID of a user, an attribute a hold by some authority and the master secret key of the corresponding authority. It returns a secret attribute key $\mathsf{SK}_{a,\mathsf{ID}}$ for the identity ID.
Encrypt($\mathsf{GP}, M, \mathbb{A}, \{\mathsf{PubA}_j\}$). This algorithm is run by an encryptor and it takes as input the global public parameters GP, a message M to be encrypted, an access structure $\mathbb{A}$, and public keys of relevant authorities corresponding to all attributes appeared in $\mathbb{A}$. It then encrypts M under $\mathbb{A}$ and returns the ciphertext CT, where $\mathbb{A}$ is embedded into CT.
Decrypt($\mathsf{GP}, \mathsf{CT}, \{\mathsf{SK}_{a,\mathsf{ID}} | a \in L_{\mathsf{ID}}\}$). On receiving a ciphertext CT, a decryptor with identity ID runs this algorithm with the input the global public parameters GP, a ciphertext CT which is an encryption of M under $\mathbb{A}$, and $\{\mathsf{SK}_{a,\mathsf{ID}} | a \in L_{\mathsf{ID}}\}$ is a set of secret attribute keys obtained for the same identity ID. Then it outputs the message M if the user attribute set L_{ID} satisfies the access structure $\mathbb{A}$; otherwise, decryption fails.

3.1 Security Model

Following [8], we define a security model in terms of a game which is carried out between a challenger and an adversary, where the challenger plays the role of all authorities. The adversary can corrupt authorities statically, i.e., the adversary has to announce the list of corrupted authorities before obtaining the public keys of honest authorities, whereas key queries can be made adaptively.
Setup. First, the challenger obtains global public parameters GP. The adversary announces a set $\mathcal{A}' \subset \mathcal{A}$ of corrupt-authorities. Now, the challenger runs Authority Setup algorithm for each honest authority and gives all public keys to the adversary.
Key Query Phase 1. The adversary is allowed to make secret key queries for the attributes coupled with user global identities (a, ID), where the attributes a

are held by honest authorities. The challenger runs Authority KeyGen algorithm and returns the corresponding secret keys $\mathsf{SK}_{a,\mathsf{ID}}$ to the adversary.

Challenge. The adversary submits two equal length messages M_0, M_1 and an access structure $\mathbb{A}$. The access structure $\mathbb{A}$ must obey the following constraint. Let F be a set of attributes belonging to the corrupt-authorities that are in $\mathbb{A}$. For each identity ID, let F_{ID} be the set of attributes in $\mathbb{A}$ for which the adversary has queried (a, ID). For each identity ID, the attribute set $F \cup F_{\mathsf{ID}}$ must not satisfy the access structure $\mathbb{A}$, i.e., $(F \cup F_{\mathsf{ID}}) \not\models \mathbb{A}$. The adversary needs to give the challenger the public keys of corrupt-authorities whose attributes are in $\mathbb{A}$. Now, The challenger flips a random coin $\mu \in \{0, 1\}$ and runs Encrypt algorithm in order to encrypt M_μ under $\mathbb{A}$. The resulting challenge ciphertext CT^* is given to the adversary.

Key Query Phase 2. The adversary can make additional secret key queries for (a, ID) with the same restriction on the challenge access structure stated in Challenge phase.

Guess. The adversary outputs a guess bit $\mu' \in \{0, 1\}$ for the challenger's secret coin μ and wins if $\mu' = \mu$.

The advantage of an adversary in this game is defined to be $|\Pr[\mu' = \mu] - \frac{1}{2}|$, where the probability is taken over all random coin tosses of both adversary and challenger.

Definition 3. *The decentralized CP-ABE system is said to be IND-CPA (ciphertext indistinguishability under chosen plaintext attacks) secure against static corruption of authorities if all polynomial time adversaries have at most a negligible advantage in the above security game.*

4 dCP-ABE-MAS

In this section, we present a decentralized CP-ABE scheme for monotone access structures, dCP-ABE-MAS. Note that every monotone access structure $\mathbb{A}$ is represented by its basis $\mathbb{A}_0$ which is the set of minimal authorized sets in $\mathbb{A}$.

System Initialization(κ). During system initialization phase, a six tuple $\mathsf{GP} = (p, \mathbb{G}, g, \mathbb{G}_T, e, \mathcal{H})$ is chosen as global public parameters, where p is a prime number greater than 2^κ, $\mathbb{G}, \mathbb{G}_T$ are two multiplicative cyclic groups of same prime order p, g is a generator of $\mathbb{G}$, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear map and $\mathcal{H} : \{0, 1\}^* \to \mathbb{G}$ is a collision resistant hash function which will be modeled as a random oracle in our security proof.

Authority Setup(GP, U^j). Each authority $A_j \in \mathcal{A}$ possesses a set of attributes U^j. For each attribute $a \in U^j$, A_j selects two random exponents $t_a, t'_a \in \mathbb{Z}_p$, and computes $P_a = g^{t_a}, P'_a = e(g, g)^{t'_a}$. The public key of A_j is published as $\mathsf{PubA}_j = \{(P_a, P'_a) | a \in U^j\}$. The master secret key of the authority A_j is $\mathsf{MkA}_j = \{(t_a, t'_a) | a \in U^j\}$.

Authority KeyGen($\mathsf{GP}, \mathsf{ID}, a, \mathsf{MkA}_j$). When a user with unique global identity $\mathsf{ID} \in \{0, 1\}^*$requests for a secret key associated with an attribute a which is held by A_j, the authority A_j returns $\mathsf{SK}_{a,\mathsf{ID}} = g^{t'_a} \mathcal{H}(\mathsf{ID})^{t_a}$ to the user.

Encrypt$(\mathsf{GP}, M, \mathbb{A}_0, \{\mathsf{PubA}_j\})$**.** Here $\mathbb{A}_0$ is the basis for a monotone access structure $\mathbb{A}$. Let $\mathbb{A}_0 = \{B_1, B_2, \ldots, B_k\}$, where each $B_i \subset U$ is a minimal authorized set in $\mathbb{A}$. The set $\{\mathsf{PubA}_j\}$ is a set of public keys of all authorities which are managing the attributes in $\mathbb{A}_0$. In order to encrypt a message $M \in \mathbb{G}_T$, the encryptor chooses a random exponent $s_i \in \mathbb{Z}_p$, for each $i, 1 \leq i \leq k$, and computes

$$C_{i,1} = M \cdot \left(\prod_{a \in B_i} P'_a \right)^{s_i}, C_{i,2} = g^{s_i} \text{ and } C_{i,3} = \left(\prod_{a \in B_i} P_a \right)^{s_i}. \quad (2)$$

The encryptor outputs the ciphertext $\mathsf{CT} = \langle \mathbb{A}_0, \{C_{i,1}, C_{i,2}, C_{i,3} | 1 \leq i \leq k\} \rangle$.

Decrypt$(\mathsf{GP}, \mathsf{CT}, \{\mathsf{SK}_{a,\mathsf{ID}} | a \in L_{\mathsf{ID}}\})$**.** When a user with global identity $\mathsf{ID} \in \{0,1\}^*$ receives a ciphertext CT, it first computes $\mathcal{H}(\mathsf{ID})$. Suppose the attribute set L_{ID} of this user satisfies the monotone access structure $\mathbb{A}$ generated by $\mathbb{A}_0 = \{B_1, B_2, \ldots, B_k\}$. Then $L_{\mathsf{ID}} \supseteq B_i$, for some $B_i \in \mathbb{A}_0$. The receiver now aggregates the secret attribute keys associated with the attributes appeared in the minimal authorized set B_i and computes $K_i = \prod_{a \in B_i} (\mathsf{SK}_{a,\mathsf{ID}})$. The message can then be obtained by computing

$$C_{i,1} \cdot \frac{e\left(\mathcal{H}(\mathsf{ID}), C_{i,3}\right)}{e\left(K_i, C_{i,2}\right)} = M \cdot e(g,g)^{s_i b'_i} \cdot \frac{e(\mathcal{H}(\mathsf{ID}), g^{s_i b_i})}{e(g^{b'_i} \mathcal{H}(\mathsf{ID})^{b_i}, g^{s_i})} = M,$$

where $b'_i = \sum_{b \in B_i} t'_a$ and $b_i = \sum_{b \in B_i} t_a$. We will use the notations b'_i and b_i in our security proof.

Remark 1. An encryptor can pack different messages, say $M_1, M_2, \ldots, M_{k'}$, where k' is equal or smaller than the size of a basis of a monotone access structure, in a single ciphertext by using the following encryption algorithm.

multi.Encrypt$(\mathsf{GP}, \{M_1, M_2, \ldots, M_{k'}\}, \mathbb{A}_0, \{\mathsf{PubA}_j\})$. Let $\mathbb{A}$ be a monotone access structure generated by its basis $\mathbb{A}_0 = \{B_1, B_2, \ldots, B_k\}$. For each $i, 1 \leq i \leq k$, the encryptor chooses a random exponent $s_i \in \mathbb{Z}_p$ and computes the ciphertext $\mathsf{CT} = \langle \mathbb{A}_0, \{C_{i,1}, C_{i,2}, C_{i,3} | 1 \leq i \leq k\} \rangle$, where $C_{i,1} = M_i \cdot (\prod_{a \in B_i} P'_a)^{s_i}$, $C_{i,2} = g^{s_i}$ and $C_{i,3} = (\prod_{a \in B_i} P_a)^{s_i}$.

On receiving the ciphertext $\mathsf{CT} = \langle \mathbb{A}_0, \{C_{i,1}, C_{i,2}, C_{i,3} | 1 \leq i \leq k\} \rangle$, the recipient can be recovered respective message M_i by executing the decryption algorithm **Decrypt**$(\mathsf{CT}, \{\mathsf{SK}_{a,\mathsf{ID}} | a \in L_{\mathsf{ID}}\}, \mathsf{GP})$ of dCP-ABE-MAS. The deployment of this mechanism will be discussed in Section 5.

4.1 Security Analysis

In this section, we first argue our dCP-ABE-MAS is secure against collusion attacks. We then prove dCP-ABE-MAS is IND-CPA secure in the generic bilinear group model (we refer the reader to [4] for definition).

Security against collusion attacks. A scheme is said to be collusion-resistant if no two or more recipients can combine their secret keys in order to decrypt a message that they are not entitled to decrypt alone. We will show that if two

users with identities $\mathsf{ID}, \mathsf{ID}'$ try to collude and combine their secret keys, they will fail in decryption process even though their attributes associated with secret keys satisfy the monotone access structure $\mathbb{A}$. Note that $\mathbb{A}_0 = \{B_1, B_2, \ldots, B_k\}$ is a basis for $\mathbb{A}$.

The encryption algorithm blinds the message M with $e(g,g)^{s_i b'_i}$. Consequently, the decryptor needs to recover the blinding term $e(g,g)^{s_i b'_i}$ by coupling their secret keys for attribute and identity pairs (a, ID) with the respective ciphertext components. If the decryptor has a satisfying set of keys with the same identity ID, i.e., $\{\mathsf{SK}_{a,\mathsf{ID}} | a \in B_i\}$, for some i, then the decryptor can recover the blinding term from the following computation.

$$\frac{e(K_i, C_{i,2})}{e(\mathcal{H}(\mathsf{ID}), C_{i,3})} = \frac{e(g,g)^{s_i b'_i} \cdot \prod_{a \in B_i} e(\mathcal{H}(\mathsf{ID}), g)^{s_i t_a}}{\prod_{a \in B_i} e(\mathcal{H}(\mathsf{ID}), g)^{s_i t_a}} = e(g,g)^{s_i b'_i}.$$

Suppose two users with different identities ID and ID' try to collude and combine their secret attribute keys such that $L_{\mathsf{ID}} \not\supset B_i$ and $L_{\mathsf{ID}'} \not\supset B_i$, for any $1 \leq i \leq k$ but $L_{\mathsf{ID}} \cup L_{\mathsf{ID}'} \supseteq B_i$, for some B_i. Then $K_i = \prod_{a \in B_{i,\mathsf{ID}}} \mathsf{SK}_{a,\mathsf{ID}} \cdot \prod_{a \in B_{i,\mathsf{ID}'}} \mathsf{SK}_{a,\mathsf{ID}'}$, where $B_{i,\mathsf{ID}} = L_{\mathsf{ID}} \cap B_i$ and $B_{i,\mathsf{ID}'} = L_{\mathsf{ID}'} \cap B_i$. Consequently, there will be some terms of the form $e(\mathcal{H}(\mathsf{ID}), g)^{s_i t_a}$ in denominator and some terms of the form $e(\mathcal{H}(\mathsf{ID}'), g)^{s_i t_a}$ in numerator which will not cancel with each other as $\mathcal{H}$ is collision resistant, i.e., $\mathcal{H}(\mathsf{ID}) \neq \mathcal{H}(\mathsf{ID}')$, thereby preventing the recovery of the blinding term $e(g,g)^{s_i b'_i}$, so is the message M. This demonstrates that dCP-ABE-MAS scheme is collusion-resistant.

Theorem 1. *The dCP-ABE-MAS scheme is IND-CPA secure against static corruption of authorities in the generic group model.*

Proof. Let $\mathtt{ADV}_1$ be an adversary who plays the original security game, say $\mathtt{GAME}_1$, described in Section 3.1. According to $\mathtt{GAME}_1$, the challenge ciphertext has a component $C_{i,1}$ which is either $M_0 \cdot e(g,g)^{s_i b'_i}$ or $M_1 \cdot e(g,g)^{s_i b'_i}$, and the adversary $\mathtt{ADV}_1$ has to distinguish them. Consequently, we define a modified game, say $\mathtt{GAME}_2$, as follows. Setup, Key Query Phase 1 and Key Query Phase 2 are similar to $\mathtt{GAME}_1$, but the challenge ciphertext component $C_{i,1}$ in Challenge phase is computed as $C_{i,1} = e(g,g)^{s_i b'_i}$ if $\mu = 1$ and $C_{i,1} = e(g,g)^{\delta_i}$ if $\mu = 0$, where δ_i is selected uniformly at random from $\mathbb{Z}_p$, and other ciphertext components are computed in the same way analogous to Encrypt algorithm. Then we have the following claim.

Claim 1: *If* $\mathtt{ADV}_1$ *has advantage* ϵ *to win* $\mathtt{GAME}_1$*, then there is an adversary who wins* $\mathtt{GAME}_2$ *with advantage at least* $\epsilon/2$.

Proof of Claim 1: According to $\mathtt{ADV}_1$, we can construct an adversary $\mathtt{ADV}_2$ as follows. In Setup, Key Query Phase 1 and Key Query Phase 2, $\mathtt{ADV}_2$ forwards all messages it receives from $\mathtt{ADV}_1$ to the challenger and all messages from the challenger to $\mathtt{ADV}_1$. In the Challenge phase, $\mathtt{ADV}_2$ receives two messages M_0 and M_1 from $\mathtt{ADV}_1$ and the challenge ciphertext CT^* from the challenger. Note that CT^* contains $C_{i,1}$ that is either $e(g,g)^{s_i b'_i}$ or $e(g,g)^{\delta_i}$. Now, $\mathtt{ADV}_2$ flips a random coin $\nu \in \{0,1\}$ and replaces $C_{i,1}$ by $C'_{i,1} = M_\nu \cdot C_{i,1}$ in CT^* to compute a modified

ciphertext CT' and finally sends the resulting CT' to the adversary $\mathtt{ADV}_1$.
Guess: $\mathtt{ADV}_1$ outputs his guess $\nu' \in \{0,1\}$ on ν. If $\nu' = \nu$, $\mathtt{ADV}_2$ outputs as its guess $\mu' = 1$; otherwise he outputs $\mu' = 0$.

- In the case where $\mu = 1$, CT' is a correct ciphertext of M_ν. Consequently, $\mathtt{ADV}_1$ can output $\nu' = \nu$ with the advantage ϵ, i.e., $\Pr[\nu' = \nu | \mu = 1] = \frac{1}{2} + \epsilon$. Since $\mathtt{ADV}_2$ guesses $\mu' = 1$ when $\nu' = \nu$, we get $\Pr[\mu' = \mu | \mu = 1] = \frac{1}{2} + \epsilon$.
- In the next case where $\mu = 0$, the challenge ciphertext CT^* is independent of the messages M_0 and M_1, so $\mathtt{ADV}_1$ cannot obtain any information about ν. Therefore, $\mathtt{ADV}_1$ can output $\nu' \neq \nu$ with no advantage, i.e., $\Pr[\nu' \neq \nu | \mu = 0] = \frac{1}{2}$. Since $\mathtt{ADV}_2$ guesses $\mu' = 0$ when $\nu' \neq \nu$, we get $\Pr[\mu' = \mu | \mu = 0] = \frac{1}{2}$.

Thus, advantage of $\mathtt{ADV}_2 = \left|\Pr[\mu' = \mu] - \frac{1}{2}\right| \geq \frac{1}{2} \cdot (\frac{1}{2} + \epsilon) + \frac{1}{2} \cdot \frac{1}{2} - \frac{1}{2} = \frac{\epsilon}{2}$. This proves the claim 1.

This claim demonstrates that any adversary that has a non-negligible advantage in $\mathtt{GAME}_1$ can have a non-negligible advantage in $\mathtt{GAME}_2$. We shall prove that no adversary can have non-negligible advantage in $\mathtt{GAME}_2$. From now on, we will discuss the advantage of the adversary in $\mathtt{GAME}_2$, wherein the adversary must distinguish between $e(g,g)^{s_i b'_i}$ and $e(g,g)^{\delta_i}$.
Simulation in $\mathtt{GAME}_2$: To simulate the modified security game $\mathtt{GAME}_2$, we use the generic bilinear group model given in [4]. Consider two injective random maps $\psi, \psi_T : \mathbb{Z}_p \to \{0,1\}^{\lceil 3\log(p) \rceil}$. In this model every element of $\mathbb{G}$ and $\mathbb{G}_T$ is encoded as an arbitrary random string from the adversary's point of view, i.e., $\mathbb{G} = \{\psi(x) | x \in \mathbb{Z}_p\}$ and $\mathbb{G}_T = \{\psi_T(x) | x \in \mathbb{Z}_p\}$. The adversary is given three oracles to compute group operations of $\mathbb{G}$, $\mathbb{G}_T$ and to compute the bilinear pairing e. The input of all oracles are string representations of group elements. The adversary is allowed to perform group operations and pairing computations by interacting with the corresponding oracles only. It is assumed that the adversary can make queries to the group oracles on input strings that were previously been obtained from the simulator or were given from the oracles in response to the previous queries. This event occurs with high probability. Since $|\psi(\mathbb{Z}_p)| > p^3$ and $|\psi_T(\mathbb{Z}_p)| > p^3$, the probability of the adversary being able to guess an element (which it has not previously obtained) in the ranges of ψ, ψ_T is negligible.

The notations $g^x := \psi(x)$ and $e(g,g)^x := \psi_T(x)$ are used in the rest of the proof. With this notation, g and $e(g,g)$ can be represented as $\psi(1)$ and $\psi_T(1)$, respectively.
Setup: Note that $\mathcal{A}$ is the set of all authorities in the system and U is the attribute universe. The simulator obtains the global public parameters GP from the trusted system initializer and gives $\psi(1)$ to the adversary. The adversary sends a corrupted authority list $\mathcal{A}' \subset \mathcal{A}$ to the simulator. For each attribute $a \in U$ controlled by honest authorities, the simulator chooses two new random values $t_a, t'_a \in \mathbb{Z}_p$, computes g^{t_a}, $e(g,g)^{t'_a}$ using respective group oracles and gives $P_a = \psi(t_a)$, $P'_a = \psi_T(t'_a)$ to the adversary.
Query Phase 1: The adversary issues hash and secret key queries, and consequently the simulator responds as follows.

Hash queries: When the adversary requests $\mathcal{H}(\mathsf{ID})$ for some user identity ID for the first time, the simulator chooses a new, unique random value $u_{\mathsf{ID}} \in \mathbb{Z}_p$,

computes $g^{u_{\mathsf{ID}}} = \psi(u_{\mathsf{ID}})$ using group oracle and gives $\psi(u_{\mathsf{ID}})$ to the adversary as $\mathcal{H}(\mathsf{ID})$. The association between values u_{ID} and the user identities ID is stored in Hlist so that it can reply consistently for subsequent queries in the future.

Secret key queries: If the adversary requests for a secret key of an attribute a with identity ID, the simulator computes $g^{t'_a}\mathcal{H}(\mathsf{ID})^{t_a}$ using the group oracle and returns $\mathsf{SK}_{a,\mathsf{ID}} = \psi(t'_a + u_{\mathsf{ID}}t_a)$ to the adversary. If $\mathcal{H}(\mathsf{ID})$ has not been stored in Hlist, it is determined as above.

Challenge: In order to obtain a challenge ciphertext CT^*, the adversary specifies the basis $\mathbb{A}_0 = \{B_1, B_2, \ldots, B_k\}$ of a monotone access structure $\mathbb{A}$ along with the public keys $g^{t_a}, e(g,g)^{t'_a}$ of attributes $a \in U$ which are controlled by corrupted authorities and appeared in $\mathbb{A}_0$ as members in several B_i. The simulator then checks the validity of these public keys by querying the group oracles. Now, the simulator chooses a random s_i for the i-th minimal set of $\mathbb{A}_0$, for each i, $1 \leq i \leq k$ and computes $b_i = \sum_{a \in B_i} t_a$. The simulator then flips a random coin $\mu \in \{0,1\}$ and if $\mu = 1$, he sets $\delta_i = s_i b'_i$, where $b'_i = \sum_{a \in B_i} t'_a$, otherwise δ_i is set to be a random value from $\mathbb{Z}_p$. The simulator finally computes the components of challenge ciphertext CT^* by using group oracles as follows.

$$C_{i,1} = \psi_T(\delta_i), C_{i,2} = \psi(s_i), C_{i,3} = \psi(s_i b_i) \text{ for all } i, 1 \leq i \leq k.$$

The ciphertext $\mathsf{CT}^* = \langle \mathbb{A}_0, \{C_{i,1}, C_{i,2}, C_{i,3} | 1 \leq i \leq k\}\rangle$ is sent to the adversary.

Query Phase 2: The adversary issues more hash and secret key queries. The simulator responds as in Query Phase 1. We note that if the adversary requests for secret keys of a set of attributes that allow decryption in combination with secret keys obtained from corrupted authorities, then the simulator is aborted.

The adversary now can have in his hand, all values that consists of encodings of random values $\delta_i, 1, u_{\mathsf{ID}}, t_a, t'_a, s_i$ and combination of these values given by the simulator (e.g., $\psi(t'_a + u_{\mathsf{ID}}t_a)$) or results of queries on combination of these values to the oracles. In turn, we can think of each query of the adversary is a multivariate polynomial in the variables $\delta_i, 1, u_{\mathsf{ID}}, t_a, t'_a, s_i$, where a ranges over the attributes controlled by honest authorities, i ranges over the minimal sets in the basis of monotonic access structure and ID ranges over the allowed user identities. We assume that any pair of the adversary's queries on two different polynomials result in two different answers. This assumption is false only when our choice of the random encodings of the variables ensures that the difference of two polynomial queries evaluates to zero. Following the security proof in [4], it can be claimed that the probability of any such collision is at most $\mathcal{O}(q^2/p)$, q being an upper bound on the number of oracle queries made by the adversary during the entire simulation. Therefore, the advantage of the adversary is at most $\mathcal{O}(q^2/p)$. We assume that no such random collisions occur while retain $1 - \mathcal{O}(q^2/p)$ probability mass.

Under this condition, we show that the view of the adversary in $\mathtt{GAME}_2$ is identically distributed when $\delta_i = s_i b'_i$ if $\mu = 1$ and δ_i is random if $\mu = 0$, and hence the adversary cannot distinguish them in the generic bilinear group model. To prove this by contradiction, let us assume that the views are not identically distributed. The adversary's views can only differ when there exists two queries

Table 2. Possible adversary's query terms in $\mathbb{G}_T$ (here, the variables a, a' are possible attributes, $\mathsf{ID}, \mathsf{ID}'$ are authorized user identities and i, i' are indices of the minimal sets in the monotone access structure).

t_a	$t_a t_{a'}$	$u_{\mathsf{ID}} u_{\mathsf{ID}'}$	$b_i(t'_a + u_{\mathsf{ID}} t_a)$	$s_i s_{i'}$
u_{ID}	$t_a u_{\mathsf{ID}}$	$u_{\mathsf{ID}'}(t'_a + u_{\mathsf{ID}} t_a)$	$s_i(t'_a + u_{\mathsf{ID}} t_a)$	$s_i s_{i'} b_{i'}$
$t'_a + u_{\mathsf{ID}} t_a$	$t_{a'}(t'_a + u_{\mathsf{ID}} t_a)$	$u_{\mathsf{ID}} b_i$	$s_i b_i(t'_a + u_{\mathsf{ID}} t_a)$	$s_i s_{i'} b_i b_{i'}$
b_i	$t_a b_i$	$u_{\mathsf{ID}} s_i$	$b_i b_{i'}$	t'_a
s_i	$t_a s_i$	$u_{\mathsf{ID}} s_i b_i$	$s_i b_{i'}$	b'_i
$s_i b_i$	$t_a s_i b_i$	$(t'_a + u_{\mathsf{ID}} t_a)(t'_{a'} + u_{\mathsf{ID}'} t_{a'})$	$s_i b_i b_{i'}$	

q_1 and q_2 in $\mathbb{G}_T$ such that $q_1 \neq q_2$ with $q_1|_{(\delta_i = s_i b'_i)} = q_2|_{(\delta_i = s_i b'_i)}$, for at least one i. Fix one such i. Since δ_i only appears as $\psi_T(\delta_i)$ and elements of ψ_T cannot be used as input of this oracle takes elements of ψ as input, the adversary can only make queries of the following form involving δ_i: $q_1 = c_1\delta_i + q'_1$ and $q_2 = c_2\delta_i + q'_2$, for some q'_1 and q'_2 that do not contain δ_i, and for some constants c_1 and c_2. Since $q_1|_{(\delta_i = s_i b'_i)} = q_2|_{(\delta_i = s_i b'_i)}$, we have $c_1 s_i b'_i + q'_1 = c_2 s_i b'_i + q'_2$ and it gives $q'_2 - q'_1 = (c_1 - c_2) s_i b'_i = c s_i b'_i$, for some constant $c \neq 0$. Therefore, the adversary can construct the query $\psi_T(c s_i b'_i)$, for some constant $c \neq 0$, yielding a contradiction to our claim 2 proved below. Hence the adversary's views in $\mathtt{GAME}_2$ are identically distributed, i.e., the adversary has no non-negligible advantage in $\mathtt{GAME}_2$, so in the original game $\mathtt{GAME}_1$ by claim 1.

Claim 2 : The adversary cannot make a query of the form $\psi_T(c s_i b'_i)$ for any non-zero constant c and any i.

Proof of Claim 2: To establish this claim, we examine the information given to the adversary during the entire simulation and perform case analysis based on that information.

In Table 2, we list all the possible adversary's query terms in $\mathbb{G}_T$ by means of the bilinear map and group elements given to the adversary during the simulation. It can be seen that the adversary can query for an arbitrary linear combination of 1 (which is $\psi_T(1)$), δ_i and the terms given in Table 2. We will now show that no such linear combination can produce a term of the form $c s_i b'_i$ for any non-zero constant c and any i. Note that the adversary knows the values of t_a, t'_a for attributes a that are controlled by the corrupted authorities, so these can appear in a foregoing linear combinations as the coefficients of the terms given in Table 2.

We note that $s_i b'_i = \sum_{a \in B_i} s_i t'_a$. From Table 2 we see that the only way for an adversary to create a term containing $s_i t'_a$ is by pairing s_i with $t'_a + u_{\mathsf{ID}} t_a$. Consequently, the adversary can create a query polynomial of the form

$$\sum_{a \in B} (c_{(i,a)} s_i t'_a + c_{(i,a,\mathsf{ID})} u_{\mathsf{ID}} s_i t_a), \tag{3}$$

for some set of attributes B and non-zero constants $c_{(i,a)}, c_{(i,a,\mathsf{ID})}$. In order to get a query polynomial of the form $c s_i b'_i$ the adversary must add other terms to

cancel the extra terms $\sum_{a \in B} c_{(i,a,\mathsf{ID})} u_{\mathsf{ID}} s_i t_a$. For any terms $c_{(i,a,\mathsf{ID})} u_{\mathsf{ID}} s_i t_a$ where a is an attribute held by a corrupted authority, the value of t_a is revealed to the adversary, thereby the adversary can form the term $-c_{(i,a,\mathsf{ID})} u_{\mathsf{ID}} s_i t_a$ in order to cancel this from the polynomial given in Eq. (3). For terms $c_{(i,a,\mathsf{ID})} u_{\mathsf{ID}} s_i t_a$ where a is an attribute controlled by an uncorrupted authority, the adversary cannot construct terms to cancel these from the polynomial given in Eq. (3) since there is no term in Table 2 that enables the adversary to construct a term of the form $-c_{(i,a,\mathsf{ID})} u_{\mathsf{ID}} s_i t_a$. Consequently, the adversary's query polynomial cannot be of the form $cs_i b'_i$.

Suppose for some identity ID, a set B' of attributes in B belong to the corrupted authorities or the adversary has obtained secret keys $\{\mathsf{SK}_{a,\mathsf{ID}} | a \in B'\}$ such that $B' \supseteq B_i$, for some $i, 1 \leq i \leq k$. Then the adversary can construct a query polynomial of the form

$$\sum_{a \in B_i} (cs_i t'_a + c_{\mathsf{ID}} u_{\mathsf{ID}} s_i t_a), \tag{4}$$

for some non-zero constant c and c_{ID}. The query polynomial given in Eq. (4) is same as $cs_i \sum_{a \in B_i} t'_a + c_{\mathsf{ID}} u_{\mathsf{ID}} s_i \sum_{a \in B_i} t_a = cs_i b'_i + c_{\mathsf{ID}} u_{\mathsf{ID}} s_i b_i$. The extra term $c_{\mathsf{ID}} u_{\mathsf{ID}} s_i b_i$ here will be canceled by using the term $u_{\mathsf{ID}} s_i b_i$ appeared in Table 2. In this case, even though the adversary becomes successful, the constraint mentioned in the Challenge phase of the security game is violated and simulator is aborted.

We have shown that the adversary cannot make a query polynomial of the form $cs_i b'_i$, for any constant $c \neq 0$ and any i, without violating the assumptions stated in the security game. This proves the claim 2 and hence the theorem. □

5 Applications

In this section, we propose an access control scheme in various network scenarios that make use of our dCP-ABE-MAS and then compare our scheme with the existing schemes in the respective areas.

Vehicular Ad Hoc Network: Typically, a vehicular ad hoc network (VANET) mainly consists of three kinds of entities−trusted initializer (TI), road side units (RSUs) and vehicles which are equipped with wireless communication devices, called on-board units (OBUs). During registration phase, each vehicle is assigned by the TI a set of *persistent attributes* (e.g., year, model), which remains constant throughout the lifetime of a vehicle, and a set of different pseudonyms, which preserves location privacy of the vehicle. We assume that each vehicle is capable of changing pseudonyms from time to time. In addition, TI gives each vehicle a set of secret keys associated with the persistent attributes for each pseudonym of that vehicle. These attributes and keys are preloaded into vehicle's OBU.

There are several RSUs which are distributed across the network in a uniform fashion and each RSU provides infrastructure support for a specified region which we call communication range of that RSU. Each RSU controls a set of *dynamic attributes* (e.g., road name, vehicle speed). When a vehicle enters within

communication range of an RSU, the RSU gives it certain dynamic attributes along with corresponding secret attribute keys after receiving a certificate relating the current pseudonym of the vehicle. We assume that there are secure communication channels between vehicles and TI as well as vehicles and RSUs.

Note that the authorities in our dCP-ABE-MAS play the role of RSUs and the attribute universe is combination of all persistent and dynamic attributes involved in the network. Every persistent attribute is different from every dynamic attribute and the attributes controlled by two different RSUs are all different from each other. The pseudonym can be treated as vehicle's identity. The setup and key generation algorithms of TI are same as authorities' setup and key generation algorithms, respectively.

Vehicles can encrypt and decrypt messages. RSUs can also encrypt messages for a set of selected vehicles. When a vehicle wants to send a message M to other vehicles in the network regarding the road situation (e.g., a car accident is ahead), it decides firstly the intended vehicles (e.g., ambulance, police car, breakdown truck) and then formulates an associated MAS in terms of minimal authorized sets over some attributes (both persistent and dynamic), for example, $\mathbb{A}_0 = \{B_1, B_2, B_3\}$, where $B_1 = \{\texttt{ambulance}, \texttt{road1}\}$, $B_2 = \{\texttt{policecar}, \texttt{lane2}\}$ and $B_3 = \{\texttt{breakdowntruck}, \texttt{road2}\}$. The encryptor vehicle then uses the public keys of the attributes occurring in the access structure to encrypt a message and transmits the ciphertext. A recipient vehicle whose attribute set satisfies the access structure will only be able to decrypt the message.

Refer to the above example, consider a scenario where the encryptor vehicle needs to send a different message to each category of vehicles−ambulance, police car, breakdown truck. Consequently, it has to encrypt each message separately under respective access structure for each category. In turn, the number of encryptions will grow linearly with the number of categories. In such cases, the proposed multi.Encrypt algorithm (described in *Remark 1*) can pack multiple messages in a single ciphertext, thereby reduces network traffic significantly, in such a way that the respective message will only be decrypted by the intended category of vehicles. This helps in the widespread dissemination of messages and early decision making in such a highly dynamic network environments.

The comparison of proposed scheme, say Scheme 1 in the VANET scenario, with the existing scheme [12] is presented in Table 3, 4.

Distributed Cloud Network: The cloud storage system is composed of five entities: trusted initializer (TI), key generation authorities (KGAs), cloud, data owner (data provider) and users (data consumers). The only responsibility of TI is generation of global public parameters GP of the system and assignment of a unique global identity ID to each user in the system. Each key generation authority controls a different set of attributes and generates public and secret keys for all attributes that it holds. The KGAs are also responsible to distribute secret keys for users' attribute sets on request according to their role or identity. The KGAs could be scattered geographically far apart and execute assigned tasks independently. The authorities in our dCP-ABE-MAS act as KGAs.

Table 3. Comparison of Computation Costs

Scheme	Key Generation	Encryption			Decryption		
	$E_{\mathbb{G}}$	$E_{\mathbb{G}}$	$E_{\mathbb{G}_T}$	P_e	$E_{\mathbb{G}}$	$E_{\mathbb{G}_T}$	P_e
[14]	$2\gamma + 2$	$4\alpha + 1$	1	-	-	$\mathcal{O}(\beta)$	$\mathcal{O}(\beta)$
[12, 13, 16]	2γ	3α	$2\alpha + 1$	1	-	$\mathcal{O}(\beta)$	$\mathcal{O}(\beta)$
Scheme 1,2	2γ	$2k$	k	-	-	-	2

Table 4. Comparison of Communication Overheads

Scheme	User Secret Key Size	Ciphertext Size	Access Policy	Requirement of CA
[14]	$(\gamma + 2)B_{\mathbb{G}}$	$(3\alpha + 1)B_{\mathbb{G}} + B_{\mathbb{G}_T} + \tau$	LSSS	Yes
[12, 13, 16]	$\gamma B_{\mathbb{G}}$	$(2\alpha)B_{\mathbb{G}} + (\alpha + 1)B_{\mathbb{G}_T} + \tau$	LSSS	No
Scheme 1,2	$\gamma B_{\mathbb{G}}$	$2kB_{\mathbb{G}} + kB_{\mathbb{G}_T} + \tau$	any MAS	No

The cloud is an external storage server that allows the data owners to store their data in the cloud in order to share their data securely to intended users. The data owners enforce an access control policy in the form of a MAS into ciphertext in such a way that only intended users can recover the data and sign the message by employing an efficient attribute-based signature scheme. Finally, the ciphertext along with signature is sent to the cloud. The cloud first verifies the signature and stores the ciphertext if the signature is valid. Each user can obtain ciphertexts from the cloud on demand. However, the users can decrypt the ciphertext only if the set of attributes associated with their secret keys satisfy the access control policy embedded in the ciphertext.

Consider a health-care scenario where the patients can be data providers, and doctors, medical researchers and health insurance companies can be data consumers. For example, a patient wishes to store his medical history in the cloud for specific users as follows: brain scan records, M_1, for any neurologist from hospital X, ECG (Electrocardiography) reports, M_2, for any cardiologist and Ultrasound reports, M_3, for any radiology researcher from any medical research center. In such setting, the multi.Encrypt algorithm (described in *Remark 1*) is well suited to pack all the three messages in a single ciphertext. To this end, the patient first formulates a MAS whose basis is $\mathbb{A}_0 = \{B_1, B_2, B_3\}$, where $B_1 = \{\texttt{neurologist}, \texttt{hospitalX}\}, B_2 = \{\texttt{cardiologist}\}$ and $B_3 = \{\texttt{radiologist}, \texttt{researcher}\}$. Once the policy is specified, multi.Encrypt algorithm is executed with the input the set of messages $\{M_1, M_2, M_3\}$, $\mathbb{A}_0$ and the respective public keys. Finally, the resulting ciphertext will be stored in the cloud. Refer to the decryption algorithm of dCP-ABE-MAS, only the intended users can decrypt the respective messages.

We compare our proposed construction, say Scheme 2 in the context of cloud storage, with the existing schemes [13, 14, 16] in Table 3, 4, where the ciphertext size is considered without signature to make consistent with other schemes.

Acknowledgement. The authors would like to thank the anonymous reviewers of this paper for their valuable comments and suggestions.

References

1. Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. In: CRYPTO 1984. LNCS, vol. 196, pp. 47−53. Springer, Heidelberg (1985)
2. Sahai, A., Waters, B.: Fuzzy Identity-Based Encryption. In: EUROCRYPT 2005. LNCS, vol. 3494, pp. 457−473. Springer, Heidelberg (2005)
3. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute Based Encryption for Fine-Grained Access Control of Encrypted Data. In: ACM Conference on Computer and Communications Security, pp. 89−98 (2006)
4. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-Policy Attribute-Based Encryption. In: IEEE Symposium on Security and Privacy, pp. 321−334 (2007)
5. Waters, B.: Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. In: PKC 2011. LNCS, vol. 6571, pp. 53−70. Springer, Heidelberg (2011)
6. Ibraimi, L., Tang, Q., Hartel, P., Jonker, W.: Efficient and Provable Secure Ciphertext-Policy Attribute-Based Encryption Schemes. In: ISPEC 2009. LNCS, vol. 5451, pp. 1−12. Springer, Heidelberg (2009)
7. Stinson D.R.:Cryptography: Theory and Practice, Third Edition. CRC press (2006)
8. Lewko, A., Waters, B.: Decentralizing Attribute-Based Encryption. Cryptology ePrint Archive, Report 2010/351 (2010)
9. Lin, H., Cao, Z., Liang, X., Shao, J.: Secure Threshold Multi Authority Attribute Based Encryption without a Central Authority. In: INDOCRYPT 2008. LNCS, vol. 5365, pp. 426−436. Springer, Heidelberg (2008)
10. Chase, M.: Multi-authority Attribute Based Encryption. In: TCC 2007. LNCS, vol. 4392, pp. 515−534. Springer, Heidelberg (2007)
11. Chase, M., Chow, S.S.M.: Improving Privacy and Security in Multi-Authority Attribute-Based Encryption. In: ACM Conference on Computer and Communications Security, pp. 121−130. ACM, New York (2009)
12. Ruj, S., Nayak, A., Stojmenovic, I.: Improved Access Control Mechanism in Vehicular Ad Hoc Networks. In: ADHOC-NOW 2011. LNCS, vol. 6811, pp. 191−205. Springer, Heidelberg (2011)
13. Ruj, S., Stojmenovic, M., Nayak, A.: Privacy Preserving Access Control with Authentication for Securing Data in Clouds. 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. pp.556−563 (2012)
14. Yang, K., Jia, X., Ren, K.: DAC-MACS: Effective Data Access Control for Multi-Authority Cloud Storage Systems. Cryptology ePrint Archive, Report 2012/419
15. Müller, S., Katzenbeisser, S., Eckert, C.: On Multi-Authority Ciphertext-Policy Attribute-Based Encryption. Bulletin of the Korean Mathematical Society 46(4), 803−817 (2009)
16. Ruj, S., Nayak, A., Stojmenovic, I.: DACC: Distributed access control in clouds. In: IEEE TrustCom 2011. pp. 91-98, IEEE (2011)
17. Liu, Z., Cao, Z., Huang, Q., Wong, D., Yuen, T.: Fully Secure Multi-authority Ciphertext-Policy Attribute-Based Encryption without Random Oracles. In: ESORICS 2011. LNCS, vol. 6879, pp. 278−297. Springer, Heidelberg (2011)
18. Pandit, T., Barua, R.: Efficient Fully Secure Attribute-Based Encryption Schemes for General Access Structures. In: ProvSec 2012, LNCS 7496, pp. 193−214. Springer, Heidelberg (2012)