Attribute-Based Encryption Optimized for Cloud Computing

Máté Horváth*

Laboratory of Cryptography and System Security (CrySyS Lab)**

Abstract. In this work, we aim to make attribute-based encryption (ABE) more suitable for access control to data stored in the cloud. For this purpose, we concentrate on giving to the encryptor full control over the access rights, providing feasible key management even in case of multiple independent authorities, and enabling viable user revocation, which is essential in practice. Our main result is an extension of the decentralized CP-ABE scheme of Lewko and Waters [LW11] with identity-based user revocation. Our revocation system is made feasible by removing the computational burden of a revocation event from the cloud service provider, at the expense of some permanent, yet acceptable overhead of the encryption and decryption algorithms run by the users. Thus, the computation overhead is distributed over a potentially large number of users, instead of putting it on a single party (e.g., a proxy server), which would easily lead to a performance bottleneck. Besides describing our scheme, we also give a formal proof of its security in the generic bilinear group and random oracle models.

1 Introduction

Recent trends show a shift from using companies' own data centres to outsourcing data storage to cloud service providers. Besides cost savings, flexibility is the main driving force for outsourcing data storage, although in the other hand it raises the issue of security, which leads us to the necessity of encryption. Traditional cryptosystems were designed to confidentially encode data to a target recipient (e.g. from Alice to Bob) and this seems to restrict the range of opportunities and flexibility offered by the cloud environment. Imagine the following scenario: some companies are cooperating on a cryptography project and from each, employees are working together on some tasks. Suppose that Alice wants to share some data of a subtask with those who are working on it, and with the managers of the project from the different companies. We see that encrypting this data with traditional techniques, causes that recipients must be determined formerly, moreover either they has to share the same private key or several encrypted versions (with different keys) must be stored. These undermine the possible security, efficiency and the flexibility which the cloud should provide.

Attribute-based encryption (ABE) proposed by Sahai and Waters [SW05] is intended for one-to-many encryption in which ciphertexts are encrypted for those who are able to fulfil certain requirements. The most suitable variant for fine-grained access control in the cloud is called ciphertext-policy (CP-)ABE, in which ciphertexts are associated with access policies, determined by the encryptor and attributes describe the user, accordingly attributes are embedded in the users' secret keys. A ciphertext can be decrypted by someone if and only if, his attributes satisfy the access structure given in the ciphertext, thus data sharing is possible without prior knowledge of who will be the receiver preserving the flexibility of the cloud even after encryption.

Returning to the previous example, using CP-ABE Alice can encrypt with an access policy expressed by the following Boolean formula: "CryptoProject" AND ("Subtask

* hmate@math.bme.hu

** Budapest University of Technology and Economics Department of Networked Systems and Services, Magyar tudosok krt 2, 1117 Budapest, Hungary. www.crysys.hu

Y" OR "MANAGER"). Uploading the ciphertext to the cloud, it can be easily accessed by the employees of each company, but the data can be recovered only by those who own a set of attributes in their secret keys which satisfies the access policy (e.g. "CRYPTOPROJECT", "SUBTASK Y").

In spite of the promising properties, the adoption of CP-ABE requires further refinement. A crucial property of ABE systems is that they resist collusion attacks. In most cases (e.g. [BSW07, Wat11]) it is achieved by binding together the attribute secret keys of a specific user with a random number so that only those attributes can be used for decryption which contains the same random value as the others. As a result private keys must be issued by one central authority that would need to be in a position to verify all the attributes or credentials it issued for each user in the system. However even our example shows that attributes or credentials issued across different trust domains are essential and these have to be verified inside the different organisations (e.g. "MANAGER" attribute). Later on we are going to make use of the results of Lewko and Waters [LW11] about decentralising CP-ABE.

The other relevant issue is user revocation. In everyday use, a tool for changing a user's rights is essential as unexpected events may occur and affect these. An occasion when someone has to be revoked can be dismissal or the revealing of malicious activity. Revocation is especially hard problem in ABE, since different users may hold the same functional secret keys related with the same attribute set (aside from randomization). We emphasise that user revocation is applied in *exceptional cases* like the above-mentioned, as all other cases can be handled simpler, with the proper use of attributes (e.g. an attribute can include its planned validity like "CRYPTOPROJECT2014").

Contribution. Based on [LW11] and [LSW10] we propose a scheme that adds identity-based user revocation feature to distributed CP-ABE. With this extension, we achieve a scheme with multiple, independent attribute authorities, in which revocation of specific users (e.g. with ID_i) from the system with all of their attributes is possible without updates of attribute public and secret keys (neither periodically, nor after revocation event). We avoid re-encryption of all ciphertexts the access structures of which contain a subset of attributes of the revoked user. The revocation right is given to the encryptor, just like the right to define the access structure which fits to the cloud computing scenario.

Related Work. The concept of attribute-based encryption was first proposed by Sahai and Waters [SW05] as a generalization of identity-based encryption. Bethencourt et al. [BSW07] worked out the first ciphertext-policy ABE scheme in which the encryptor must decide who should or should not have access to the data that she encrypts (ciphertexts are associated with policies, and users' keys are associated with sets of descriptive attributes). This concept was further improved by Waters in [Wat11].

The problem of building ABE systems with multiple authorities was first considered by Chase [Cha07] with a solution that introduced the concept of using a global identifier (GID) for tying users' keys together. Her system relied on a central authority and was limited to expressing a strict AND policy over a pre-determined set of authorities. Decentralized ABE of Lewko and Waters [LW11] does not require any central authority and any party can become an authority while there is no requirement for any global coordination (different authorities need not even be aware of each other) other than the creation of an initial set of common reference parameters. With this it avoids placing absolute trust in a single designated entity, which must remain active and uncorrupted throughout the lifetime of the system. [LCH+11, RNS11] shaped similar multi-authority schemes to the needs of cloud computing, but both lack for efficient user revocation.

Attribute revocation with the help of expiring attributes was proposed by Bethencourt et al. [BSW07]. Ruj et al. [RNS11] and Wang et al. [WLWG11] also show traditional attribute revocation (in multi-authority setting) causing serious computational overhead, because of the need for key re-generation and ciphertext re-encryption. A different approach is identity-based revocation, two types of which are applied to the scheme of Waters [Wat11]. Liang et

al. [LLLS10] gives the right of controlling the revoked set to a "system manager" while Li et al. [LZW+13], follow [LSW10], from the field of broadcast encryption systems and give the revocation right directly to the encryptor. To the best of our knowledge no multi-authority system is integrated with this approach and our work is the first in this direction.

Organization. In section 2 we introduce the theoretical background that we use later and define the security of multi-authority CP-ABE schemes with ID-based revocation. In section 3 the details of our scheme can be found together with efficiency and security analysis. Directions for further research are proposed in the last section.

2 Background

We first briefly introduce bilinear maps, give formal definitions for access structures and relevant background on Linear Secret Sharing Schemes (LSSS). Then we give the algorithms and security definitions of ciphertext policy attribute based encryption (CP-ABE) with identity-based user revocation.

2.1 Bilinear maps

We present the most important fact related to groups with efficiently computable bilinear maps.

Let $\mathbb{G}_0$ and $\mathbb{G}_1$ be two multiplicative cyclic groups of prime order p. Let g be a generator of $\mathbb{G}_0$ and e be a bilinear map (pairing), $e : \mathbb{G}_0 \times \mathbb{G}_0 \to \mathbb{G}_1$, with the following properties:

1. Bilinearity: $\forall u, v \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$
2. Non-degeneracy: $e(g, g) \neq 1$.

We say that $\mathbb{G}_0$ is a bilinear group if the group operation in $\mathbb{G}_0$ and the bilinear map $e : \mathbb{G}_0 \times \mathbb{G}_0 \to \mathbb{G}_1$ are both efficiently computable. Notice that the map e is symmetric since $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

2.2 Access Structures

Definition 1 (Access Structure [Bei96]).
Let $\{P_1, \ldots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1,\ldots,P_n\}}$ is monotone if $\forall B, C$: if $B \in \mathbb{A}$ and $B \subseteq C$ then $C \in \mathbb{A}$. An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection) $\mathbb{A}$ of non-empty subsets of $\{P_1, \ldots, P_n\}$, i.e., $\mathbb{A} \subseteq 2^{\{P_1,\ldots,P_n\}} \setminus \{\emptyset\}$. The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ are called the unauthorized sets.

In our case the access structure $\mathbb{A}$ will contain the authorized sets of attributes, furthermore we restrict our attention to monotone access structures. However, it is possible to (inefficiently) realize general access structures using our techniques by having the not of attributes as separate attributes as well.

2.3 Linear Secret Sharing Schemes (LSSS)

To express the access control policy we will make use of LSSS. Here we adopt the definitions from those given in [Bei96].

Definition 2 (Linear Secret Sharing Scheme). *A secret-sharing scheme Π over a set of parties $\mathcal{P}$ is called linear (over $\mathbb{Z}_p$) if*
i, the shares for each party form a vector over $\mathbb{Z}_p$,

ii, there exists a matrix A with ℓ rows and n columns called the share-generating matrix for Π. For all $i = 1, \ldots, \ell$, the i^{th} row of A let the function ρ defined the party, labelling row i as $\rho(i)$. When we consider the column vector $v = (s; r_2, \ldots, r_n)$, where $s \in \mathbb{Z}_p$ is the secret to be shared, and $r_2, \ldots, r_n \in \mathbb{Z}_p$ are randomly chosen, then $Av = \lambda$ is the vector of ℓ shares of the secret s according to Π. The share $(Av)_i = \lambda_i$ belongs to party $\rho(i)$.

In [Bei96] it is shown that every linear secret sharing-scheme according to the above definition also enjoys the *linear reconstruction property*, defined as follows. Suppose that Π is an LSSS for the access structure $\mathbb{A}$. Let $S \in \mathbb{A}$ be any authorized set, and let $I \subset \{1, 2, \ldots, \ell\}$ be defined as $I = \{i | \rho(i) \in S\}$. Then, there exist constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ such that, if $\{\lambda_i\}$ are valid shares of any secret s according to Π, then $\sum_{i \in I} \omega_i \lambda_i = s$. Furthermore, it is also shown in [Bei96] that these constants $\{\omega_i\}$ can be found in time polynomial in the size of the share-generating matrix A and for unauthorized sets, no such $\{\omega_i\}$ constants exist.

We use the convention that $(1, 0, 0, \ldots, 0)$ is the "target" vector for any linear secret sharing scheme. For any satisfying set of rows I in A, we will have that the target vector is in the span of I, but for any unauthorized set, it is not.

Using standard techniques (see [LW11] - Appendix G) one can convert any monotonic boolean formula into an LSSS representation. An access tree of ℓ nodes will result in an LSSS matrix of ℓ rows.

2.4 Revocation Scheme for Multi-Authority CP-ABE

A multi-authority Ciphertext-Policy Attribute-Based Encryption system with identity-based user revocation is comprised of the following algorithms:

Global Setup$(\lambda) \rightarrow GP$ The global setup algorithm takes in the security parameter λ and outputs global parameters GP for the system.

Central Authority Setup$(GP) \rightarrow (SK^*, PK^*)$ The identity key generator authority runs this algorithm with GP as input to produce its own secret key and public key pair, SK^*, PK^*.

Identity KeyGen$(GP, GID) \rightarrow K^*_{GID}$ The central authority runs this algorithm upon a user request for identity secret key. It checks whether the request is valid and if yes, generates K^*_{GID}

Authority Setup$(GP) \rightarrow (PK, SK)$ Each authority runs the authority setup algorithm with GP as input to produce its own secret key and public key pair, SK, PK.

KeyGen$(PK^*_{GID}, i, SK, GP) \rightarrow K_{i,GID}$ The key generation algorithm takes in an identity GID, the global parameters, an attribute i belonging to some authority, and the secret key SK for this authority. It produces a key $K_{i,GID}$ for this attribute, identity pair.

Encrypt$(\mathcal{M}, (A, \rho), GP, \{PK\}, PK^*, RL) \rightarrow CT$
The encryption algorithm takes in a message $\mathcal{M}$, an access matrix (A, ρ), the set of public keys for relevant authorities, the public key of the central authority, the revoked user list and the global parameters. It outputs a ciphertext CT.

Decrypt$(CT, (A, \rho), \{K_{i,GID}\}, K^*_{GID}, RL, GP) \rightarrow \mathcal{M}$ The decryption algorithm takes in the global parameters, the revoked user list, the ciphertext, identity key and a collection of keys corresponding to attribute, identity pairs all with the same fixed identity GID. It outputs either the message $\mathcal{M}$ when the collection of attributes i satisfies the access matrix corresponding to the ciphertext. Otherwise, decryption fails.

2.5 Security Model

We now define (chosen plaintext) security of multi-authority CP-ABE system with identity-based revocation. Security is defined using the following *Security Game* between an attacker algorithm $\mathcal{A}$ and a challenger. We assume that adversaries can corrupt authorities only

statically, but key queries are made adaptively. The definition reflects the scenario where all users in the revoked set RL get together and collude (this is because the adversary can get all of the private keys for the revoked set). The game is the following:

Setup. The challenger runs the Global Setup algorithm to obtain the global public parameters GP. The attacker specifies a set $AA' \subseteq AA$ of corrupt attribute authorities. For good (non-corrupt) authorities in $AA \setminus AA'$ and the Central ID Generator Authority, the challenger obtains public key, private key pairs by running the Authority Setup and Identity KeyGen algorithms, and gives the public keys to the attacker.

Key Query Phase. $\mathcal{A}$ adaptively issues private key queries for identities GID_k. The challenger gives $\mathcal{A}$ the corresponding identity keys $K^*_{GID_k}$ by running the Identity KeyGen algorithm. Let UL denote the set of all queried GID_k. $\mathcal{A}$ also makes attribute key queries by submitting pairs of (i, GID_k) to the challenger, where i is an attribute belonging to a good authority. The challenger responds by giving the attacker the corresponding key, K_{i,GID_k}.

Challenge. The attacker gives the challenger two messages M_0, M_1, a set $RL \subseteq UL$ of revoked identities and an access matrix (A, ρ).
RL and A must satisfy the following constraints. Let V denote the subset of rows of A labelled by attributes controlled by corrupt authorities. For each identity $GID_k \in UL$, let V_{GID_k} denote the subset of rows of A labelled by attributes i for which the attacker has queried (i, GID_k). For each $GID_k \in UL \setminus RL$, we require that the subspace spanned by $V \cup V_{GID_k}$ must not include $(1, 0, \ldots, 0)$ while for $GID_k \in RL$, it is allowed and we only require that the subspace spanned by V must not include $(1, 0, \ldots, 0)$.
The attacker must also give the challenger the public keys for any corrupt authorities whose attributes appear in the labelling ρ.
The challenger flips a random coin $\beta \in (0, 1)$ and sends the attacker an encryption of M_β under access matrix (A, ρ) with the revoked set RL.

Key Query Phase 2. The attacker may submit additional attribute key queries (i, GID_k), as long as they do not violate the constraint on the challenge revocation list RL and matrix (A, ρ).

Guess. $\mathcal{A}$ must submit a guess β' for β. The attacker wins if $\beta' = \beta$. The attacker's advantage in this game is defined to be $\mathbb{P}(\beta' = \beta) - \frac{1}{2}$.

Definition 3. *We say that a multi-authority CP-ABE system with identity-based revocation is (chosen-plaintext) secure (against static corruption of attribute authorities) if, for all revocations sets RL of size polynomial in the security parameter, all polynomial time adversary have at most a negligible advantage in the above defined security game.*

3 Our Construction

To build our model we will use the prime order group construction of Lewko and Waters [LW11], because of its favourable property of having independent attribute authorities. This scheme is proven to be secure in the generic bilinear group and random oracle models under the restriction that attributes are used only once in the access matrix. A secure scheme allowing reuse of attributes can be obtained by applying a simple transformation (for details, see Appendix B in [LW11]) which also can be applied in case of our construction. We note that this transformation does not lead to an efficient system when the attribute universe is large.

In order to achieve identity-based revocation we supplement the distributed system with a Central Identity Generator Authority. However it seems to contradict with the original aim of distributing the key generation right, this additional central authority would generate only secret keys for global identifiers (GIDs) of users and the attribute key generation remains distributed. Our central authority does not possess any information that alone would give advantage during decryption, in contrast to single authority schemes, where the authority

is able to decrypt all ciphertexts. Regarding this, we can say that our system remains distributed, in spite of launching a central authority.

Our Technique. We face with the challenges of identity-based revocation. To realize the targeted features, we use some ideas from public key broadcast encryption systems [LSW10]. We use secret sharing in the exponent. Suppose an encryption algorithm needs to create an encryption with a revocation set $RL = GID_1^*, \ldots, GID_r^*$ of r identities. The algorithm will create an exponent $s^* \in \mathbb{Z}_p$ and split it into r random shares $s_1, \ldots, s_r$ such that $\sum_{i=1}^{r} s_i = s^*$. It will then create a ciphertext such that any user key with GID_i^* will not be able to incorporate the i^{th} share and thus not decrypt the message.

This approach presents the following challenges. First, we need to make crucial that the decryptor need to do the GID comparisons even if his attributes satisfy the access structure of the ciphertext. Second we need to make sure that a user with revoked identity GID_i^* cannot do anything useful with share i. Third, we need to worry about collusion attacks between multiple revoked users.

To address the first one we are going to take advantage of the technique of [LW11] that is used to prevent collusion attacks. Here the secret s, used for the encryption, is divided into shares, which are further blinded with shares of zero. This structure allows for the decryption algorithm to both reconstruct the main secret and to "unblind" it in parallel. When we would like to make this algorithm necessary, but not enough for decryption it is straightforward to spoil the "unblinding" of the secret by changing the shares of zero in the exponent to shares of an other random number, $s^* \in \mathbb{Z}_p$. Thus we can require an other computation, namely the comparison of the decryptor's and the revoked users' GIDs. If correspondence is found, the algorithm stops, otherwise reveals the blinding, enabling decryption.

The second challenge is addressed by the following method. A user with $GID \neq GID_i^*$ can obtain two linearly independent equations (in the exponent) involving the share s_i, which he will use to solve for the share s_i. However, if $GID = GID_i^*$ he will get two linearly dependent equations and not be able to solve the system.

The third problem is eliminated by using $H(GID)$ as the base of the identity secret key, such that in decryption each user recovers shares $s_k \cdot \log_g H(GID)$ in the exponent, disallowing the combination of shares from different users.

3.1 Our Construction

To make the following algorithms more understandable, in Table 1 we summarize the new (compared to [LW11]) keys and variables which we introduce in our construction.

Table 1. The summary of our new notations

Notation	Meaning	Role
PK^*	$\{g^a, g^{1/b}\}$	public key of the Central ID Generator Authority
SK^*	$\{a, b\}$	secret key of the Central ID Generator Authority
K_{GID}^*	$H(GID)^{(GID+a)b}$	secret Global Identity key of a user
$C_{1,k}^*$	$\left(g^a g^{GID_k^*}\right)^{-s_k}$	revoked user identification in the ciphertext
$C_{2,k}^*$	$g^{s_k/b}$	secret share in the ciphertext
RL	$\{GID_1^*, \ldots, GID_r^*\}$	list of r users, who should be revoked

Global Setup$(\lambda) \rightarrow GP$

In the global setup, a bilinear group $\mathbb{G}_0$ of prime order p is chosen. The global public parameters, GP, are p and a generator g of $\mathbb{G}_0$, and a function H mapping global

identities GID to elements of $\mathbb{G}_0$ (this is modelled as a random oracle in the security proof).

Central Authority Setup$(GP) \rightarrow (SK^*, PK^*)$ The algorithm chooses random exponents $a, b \in \mathbb{Z}_{\text{।}}$, keeps them as a secret and publishes

$$PK^* = (g^a, g^{1/b}).$$

Identity KeyGen$(GP, GID) \rightarrow K^*_{GID}$

Upon the request of a user it first checks whether the user is on the list of revoked users (RL) or it has been queried before, if yes refuses the request, otherwise computes $H(GID)$ and generates the global identity secret key:

$$K^*_{GID} = H(GID)^{(GID+a)b}.$$

Authority Setup$(GP) \rightarrow (PK, SK)$

For each attribute i belonging to the authority (these indices i are not reused between authorities), the authority chooses two random exponents $\alpha_i, y_i \in \mathbb{Z}_p$ and publishes $PK = \{e(g,g)^{\alpha_i}, g^y \ \forall i\}$ as its public key. It keeps $SK = \{\alpha_i, y_i \ \forall i\}$ as its secret key.

KeyGen$(PK^*_{GID}, i, SK, GP) \rightarrow K_{i,GID}$

To create a key for a GID, for attribute i belonging to an authority, the authority computes:

$$K_{i,GID} = g^{\alpha_i} H(GID)^{y_i}$$

Encrypt$(\mathcal{M}, (A, \rho), GP, \{PK\}, PK^*, RL) \rightarrow CT$

The encryption algorithm takes in a message $\mathcal{M}$, an $n \times \ell$ access matrix A with ρ mapping its rows to attributes, the global parameters, the public keys of the relevant authorities, the user identity public key and the most recent list of revoked users.

It chooses random $s, s^* \in \mathbb{Z}_p$ and a random vector $v \in \mathbb{Z}_p^\ell$ with s as its first entry. Let λ_x denote $A_x \cdot v$, where A_x is row x of A. It also chooses a random vector $w \in \mathbb{Z}_p^\ell$ with s^* as its first entry. Let ω_x denote $A_x \cdot w$.

For each row A_x of A, it chooses a random $r_x \in \mathbb{Z}_p$ and supposed that the number of revoked users is $|RL| = r$ it chooses s_i such that $s^* = \sum_{i=1}^{r} s_i$. The CT ciphertext is computed as

$$\begin{gathered} C_0 = \mathcal{M} \cdot e(g,g)^s, \\ C_{1,x} = e(g,g)^{\lambda_x} e(g,g)^{\alpha_{\rho(x)} r_x}, C_{2,x} = g^{r_x}, C_{3,x} = g^{y_{\rho(x)} r_x} g^{\omega_x} \quad \forall x = 1, \ldots, n \\ C^*_{1,k} = \left(g^a g^{GID^*_k}\right)^{-s_k}, C^*_{2,k} = g^{s_k/b} \quad \forall k = 1, \ldots, r. \end{gathered}$$

Decrypt$(CT, (A, \rho), \{K_{i,GID}\}, K^*_{GID}, RL, GP) \rightarrow \mathcal{M}$

We assume the ciphertext is encrypted under an access matrix (A, ρ). If the decryptor is not on the list of revoked users (RL) and has the secret keys K^*_{GID} for his GID and $\{K_{i,GID}\}$ for a subset of rows A_x of A, such that $(1, 0, \ldots, 0)$ is in the span of these rows, then the decryptor proceeds as follows. First chooses constants $c_x \in \mathbb{Z}_p$ such that $\sum_x c_x A_x = (1, 0, \ldots, 0)$ and denoting $r = |RL|$ computes:

$$\frac{\prod_x \left(\frac{C_{1,x} \cdot e(H(GID), C_{3,x})}{e(K_{\rho(x),GID}, C_{2,x})}\right)^{c_x}}{\prod_{k=1}^{r} \left(e(K^*_{GID}, C^*_{2,k}) e(C^*_{1,k}, H(GID))\right)^{1/(GID - GID^*_k)}} = e(g,g)^s$$

The message then can be obtained as : $\mathcal{M} = C_0 / e(g,g)^s$.

To see the soundness of the Decryption algorithm observe the following:

$$\begin{aligned}\mathscr{A} &= \prod_x \left(\frac{C_{1,x}\cdot e(H(GID), C_{3,x})}{e(K_{\rho(x),GID}, C_{2,x})}\right)^{c_x} = \prod_x \left(e(g,g)^{\lambda_x+\omega_x \log_g H(GID)}\right)^{c_x} \\ &= e(g,g)^{\sum_x \lambda_x c_x}\cdot e(H(GID),g)^{\sum_x \omega_x c_x} = e(g,g)^{s+s^* \log_g H(GID)} \\ \mathscr{B} &= \prod_{k=1}^{r} \left(e(K^*_{GID}, C^*_{2,k}) e(C^*_{1,k}, H(GID))\right)^{-1/(GID-GID^*_k)} \\ &= \prod_{k=1}^{r} \left(e(g,g)^{(GID-GID^*_k)s_k \log_g H(GID)}\right)^{-1/(GID-GID^*_k)} \\ &= e(g,g)^{-\sum_{k=1}^{r} s_k \log_g H(GID)} = e(g,g)^{-s^* \log_g H(GID)}\end{aligned}$$

Remark. We note that an almost equivalent result can be achieved, with some different modifications on the decentralized scheme (splitting $C_{1,x}$ into two parts, using $e(g,g)^{\beta s}$ for encryption, where β is the secret of the CA, and publishing g^s) and fitting it to the method of [LZW+13]. However in this way additional modifications are still needed to prevent the CA from being able to decrypt any ciphertext by computing $e(g^\beta, g^s)$.

3.2 Efficiency

Traditional, attribute-based user revocation (e.g. [WLWG11, RNS11]) affects the attributes, thus the revocation of a user may cause the update of all the users' attribute secret keys who had common attribute with the revoked user (a general attribute can affect big proportion of the users) and the re-encryption of all ciphertext the access structure of which contain any of the revoked user's attributes (most of these could not be decrypted by the revoked user).

In our scheme, a revocation event does not have any effect on the attributes as it is based on identity. Although it is a trade-off and in the other hand there is some computational overhead on the encryption and decryption algorithm. In this way the necessary extra computation of authorities is reduced and distributed between the largest set of parties, the users, preventing a possible performance bottleneck of the system. At the same time the extra communication is also reduced to the publication of the revoked user list. Our revocation scheme has the following costs.

The ciphertext has $2r$ additional elements, if the number of revoked users is r. For the computation of these values $3r$ exponentiations and r multiplications are needed in $\mathbb{G}_0$. Alternatively, the revoked user list may contain $g^a g^{GID^*_i}$ instead of the global identifiers. In this case the encryptor need to do only $2r$ additional exponentiations in $\mathbb{G}_0$, compared with the scheme of [LW11], to compute the ciphertext. The overhead of the decryption algorithm is $2r$ pairing operations, r multiplications and exponentiations in group $\mathbb{G}_1$.

Note that, as in all model that uses LSSS to express the access structure, the access matrix and the mapping ρ must be part of the ciphertext, increasing its length. However, it is possible to reduce this length by attaching only a formatted Boolean formula instead and compute the necessary components of LSSS efficiently using the algorithm of Liu and Cao in [LC10].

3.3 Security

Before giving the formal proof we point put that from the point of view of a user, whose attributes have never satisfied the access structure defined in the ciphertext, our construction is at least as secure as the one by [LW11], because the computation of $\mathscr{A}$ is equivalent to the decryption computation given there. However in our case, it is not enough to obtain the message. Changing the first entry of w from zero to a random number (as we did), causes

that the blinding will not cancel out from $\mathscr{A}$, but we need to compute $\mathscr{B}$ which can divide it out. $\mathscr{B}$ can be computed with any GID different from any GID_k^* of the revocation list and we ensure that the decryptor must use the same GID both in $\mathscr{A}$ and $\mathscr{B}$ with the help of $H(GID)$.

We are going to prove the security of our construction in the generic bilinear group model previously used in [BBG05, BSW07, LW11], modelling H as a random oracle. Security in this model assures us that an adversary cannot break our scheme with only black-box access to the group operations and H.

We describe the generic bilinear model as in [BBG05]. We let ψ_0 and ψ_1 be two random encodings of the additive group $\mathbb{Z}_p$. More specifically, each of ψ_0, ψ_1 is an injective map from $\mathbb{Z}_p$ to $\{0,1\}^m$, for $m > 3\log(p)$. We define the groups $\mathbb{G}_0 = \{\psi_0(x) : x \in \mathbb{Z}_p\}$ and $\mathbb{G}_1 = \{\psi_1(x) : x \in \mathbb{Z}_p\}$. We assume we have access to oracles which compute the induced group operations in $\mathbb{G}_0$ and $\mathbb{G}_1$ and an oracle which computes a non-degenerate bilinear map $e : \mathbb{G}_0 \times \mathbb{G}_0 \to \mathbb{G}_1$. We refer to $\mathbb{G}_0$ as a generic bilinear group. To simplify our notations let g denote $\psi_0(1)$, g^x denote $\psi_0(x)$, $e(g,g)$ denote $\psi_1(1)$, and $e(g,g)^y$ denote $\psi_1(y)$.

Theorem 1. *For any adversary $\mathcal{A}$, let q be a bound on the total number of group elements it receives from queries it makes to the group oracles and from its interaction with the security game, described in 2.5. The above described construction is secure according to Definition 3 in the generic bilinear group model and the advantage of $\mathcal{A}$ is $\mathcal{O}(q^2/p)$.*

Proof. In our security game, $\mathcal{A}$ must distinguish $C_0 = M_0 e(g,g)^s$ from $C_0 = M_1 e(g,g)^s$. We can alternatively consider a modified game, where the attacker must distinguish between $C_0 = e(g,g)^s$ or $C_0 = e(g,g)^T$, for T chosen uniformly randomly from $\mathbb{Z}_p$. This is the same modification employed in [BSW07, LW11], and it is justified by a simple hybrid argument.

We now simulate the modified security game in the generic bilinear group model where C_0 is set to be $e(g,g)^T$. We let S denote the set of all authorities, U the universe of attributes and RL the Revocation List. The simulator runs the global setup algorithm, and gives g to the attacker. $\mathcal{A}$ chooses a set $S' \subset S$ of corrupted authorities, and reveals these to the simulator. The simulator randomly chooses values $a, b \in \mathbb{Z}_p$ for the identity key generation and $\alpha_i, y_i \in \mathbb{Z}_p$ for the attributes $i \in U$ controlled by uncorrupted authorities, and it queries the group oracles for $g^a, g^{1/b}$ and for each $g^{y_i}, e(g,g)^{\alpha_i}$ and gives these to the attacker. When the attacker requests $H(GID_k)$ for some GID_k for the first time, the simulator chooses a random value $h_{GID_k} \in \mathbb{Z}_p$, queries the group oracle for $g^{h_{GID_k}}$, and gives this value to the attacker as $H(GID_k)$. It stores this value so that it can reply consistently to any subsequent requests for $H(GID_k)$.

Upon a request for $K^*_{GID_k}$ for some GID_k the simulator uses the group oracle to compute $g^{(GID_k^*+a)bh_{GID_k}}$ and supplies this value to the attacker. A request for a key K_{i,GID_k} for some attribute i and identity GID_k handled similarly, $g^{\alpha_i} H(GID_k)^{y_i}$ is computed using the group oracle and sent to the attacker. In both cases, if $H(GID_k)$ has not been requested before, it is determined as above.

At some point, the attacker specifies an access matrix (A, ρ) for the challenge ciphertext and additionally supplies the simulator with the $g^{y_i}, e(g,g)^{\alpha_i}$ values for any attributes i controlled by corrupt authorities that appear in the image of ρ on the rows of A. The simulator then checks that these are valid group elements by querying the group oracles.

The simulator must now produce the challenge ciphertext. To do so, it chooses random values $s, v_2, \ldots, v_\ell, s^*, w_2, \ldots, w_\ell \in \mathbb{Z}_p$, sets the sharing vector $v = (s, v_2, \ldots, v_\ell)$ and computes the shares $\lambda_x = A_x \cdot v$. Similarly it set the blinding vector $w = (s^*, w_2, \ldots, w_\ell)$ and computes $\omega_x = A_x \cdot w$. Random values $r_x \in \mathbb{Z}_p$ are chosen for each row A_x of A, and a random value $T \in \mathbb{Z}_p$. The values of $s_1, \ldots, s_{r-1} \in \mathbb{Z}_p$ are also chosen randomly, while $s_r = s^* - \sum_{i=1}^{r-1} s_i$ (where $r = |RL|$). Using the group oracles, the simulator can now compute:

$$C_0 = e(g,g)^T,$$

$$C_{1,x} = e(g,g)^{\lambda_x} e(g,g)^{\alpha_{\rho(x)} r_x}, C_{2,x} = g^{r_x}, C_{3,x} = g^{y_{\rho(x)} r_x} g^{\omega_x} \quad \forall x$$
$$C^*_{1,k} = \left(g^a g^{GID^*_k}\right)^{-s_k} C^*_{2,k} = g^{s_k/b} \quad \forall k$$

and the challenge ciphertext is given to the attacker.

We will argue that will all but negligible probability, the attacker's view in the simulation is identically distributed to what it's view would have been if C_0 had been set to $e(g,g)^s$ instead of $e(g,g)^T$. This shows that the attacker cannot attain a non-negligible advantage in the modified security game, and hence in the real one.

We condition on the event that each of the attacker's queries to the group oracles have input values that were given to the attacker during the simulation or were received from the oracles in response to previous queries. Since each ψ_0, ψ_1 is a random injective map from $\mathbb{Z}_p$ into a set of $> p^3$ elements, the probability of the attacker being able to guess an element in the image of ψ_0, ψ_1 which it has not previously obtained is negligible.

Under this condition, we can think of each of the attacker's queries as a multi-variate expressions in the variables $T, y_i, \alpha_i, \lambda_x, r_x, \omega_x, h_{GID_k}, a, b, s_k$, where i ranges over the attributes controlled by uncorrupted authorities, x ranges over the rows of the challenge access matrix, k ranges over the revoked identities. (We can also think of λ, ω_x as linear combinations of the variables $s, v_2, \ldots, v_\ell$ and $s^*, w_2, \ldots, w_\ell$.)

We now further condition on the event that for each pair of queries $\mathcal{A}$ makes corresponding to different polynomials, it receives different answers. In other words, we are conditioning on the event that our random assignment of values to the previous variables does not happen to be a zero of the difference of two query polynomials. Since our polynomials have degree at most 8 (see the possible polynomials later), using the Schwartz-Zippel lemma we have that the probability of a collusion is $\mathcal{O}(1/p)$ and a union bound shows that the probability of that any such collusion happens during the simulation is $\mathcal{O}(q^2/p)$. Now suppose that it does not happen.

Since T only appears as $e(g,g)^T$, the only queries the attacker can make involving T are of the form $cT+$ other terms, where c is a constant. The attacker's view can only differ when $T = s$ if the attacker can make two queries f and f' into $\mathbb{G}_1$ where these are unequal as polynomials but become the same when we substitute s for T. This implies $f - f' = cs - cT$ for some constant c. We may conclude that the attacker can then make the query cs.

We will now show the attacker cannot make a query of the form cs, and therefore arrive at a contradiction. By examining the values given to the attacker during the simulation, [LW11] showed that without a satisfying set of attributes an attacker cannot make a query of the form $c(s+0 \cdot h_{GID_k})$ thus has only a negligible advantage in distinguishing an encoded message from a random group element (in their original scheme). This result implies that in our modified construction, the attacker cannot make a query of the form $c(s + s^* h_{GID_k})$ without a satisfying set of attributes (as the first element of the blinding vector w is changed to s^* from zero) which also shows - following the their reasoning - that cs cannot be formed either. In the other hand, in our case the possession of the necessary attributes are not enough to make a cs query, but $-c(s^* h_{GID_k})$ is also indispensable for this. From now on we assume that $GID_k \in RL$ thus the challenge access structure is satisfied (and simulate that all revoked users are colluding), as the case when $GID_k \in UL \setminus RL$ is equivalent to the original scheme of [LW11]. We will prove that $\mathcal{A}$ cannot be successful by showing it cannot make a query of the form $-c(s^* h_{GID_k})$ and so not cs.

We see that the attacker can form queries which are linear combinations of

$$1, h_{GID_k}, y_i, \alpha_i + h_{GID_k} y_i, \lambda_x + \alpha_{\rho(x)} r_x, r_x, y_{\rho(x)} r_x + \omega_x,$$
$$a, 1/b, b h_{GID_k} (GID^*_k + a), s_k(a + GID^*_k), s_k/b,$$

the product of any two of these and α_i, T. (Note that GID^*_k for all $k = 1, \ldots, r$ and α_i, y_i for attributes i controlled by corrupted authorities are constants, known by the attacker.) In these queries s^* can appear in two different forms: as ω_x and s_k.

In order to gain $s^* h_{GID_k}$ by utilizing ω_x, $\mathcal{A}$ must use the product $h_{GID_k} y_{\rho(x)} r_x + h_{GID_k} \omega_x$ for all rows of A, as these are the only terms which contain $h_{GID_k} \omega_x$ the proper linear

Table 2. Possible relevant query terms

$s_k a + GID_k^* s_k$	s_k/b
$s_k s_l a^2 + GID_k^* GID_l^* s_k s_l + (GID_k^* + GID_l^*) s_k s_l a$	$s_k s_l / b^2$
$s_k a^2 + GID_k^* s_k a$	$s_k a/b$
$s_k a/b + GID_k^* s_k/b$	s_k/b^2
$s_k b h_{GID_l}(a^2 + (GID_k^* + GID_l^*)a + GID_k^* GID_l^*)$	$s_k a h_{GID_l} + GID_l^* s_k h_{GID_l}$
$s_k a h_{GID_l} + GID_k^* s_k h_{GID_l}$	$s_k h_{GID_l}/b$
$s_k s_l a/b + GID_k^* s_k s_l/b$	

combination of which leads to $s^* h_{GID_k}$. To cancel out $h_{GID_k} y_{\rho(x)} r_x$ the attacker should form this product, which is possible only if $y_{\rho(x)}$ or r_x are known constants as otherwise the needed elements appear alone in the above list and besides those, $\mathcal{A}$ can only form the product of any two but not three. However if $y_{\rho(x)}$ or r_x are constants for all x, that contradicts with the rules of the security game as only corrupted attributes would satisfy the access structure.

On s_k, we can make the following observations. (1) In each term, s_k appears as multiplier either in all monads or in none of them. (2) To form $c \cdot s_k h_{GID_l}$ (for a chosen l and all k) as linear combination of different terms, these must contain s_k as multiplier, so terms without s_k are useless (see Table 2 for the possible query terms). (3) In the linear combination there must be a term which contains $s_k h_{GID_l}$ maybe multiplied with some constant.

As it can be seen in Table 2, there are two terms which contain the necessary monad:

$$s_k a h_{GID_l} + GID_k^* s_k h_{GID_l} \text{ and } s_k a h_{GID_l} + GID_l^* s_k h_{GID_l},$$

multiplied each by $c/(GID_k^* - GID_l^*)$ it is possible to gain $c \cdot s_k h_{GID_l}$, if $k \neq l$. However in case of $k = l$ the two terms are equal, and $s_k a h_{GID_l}$ cannot be cancelled out, as no other terms contain this product. We conclude that it is possible to gain $s_k h_{GID_l}$ for all k (thus $-c \sum_{k=1}^{r} s_k h_{GID_l}$) if and only if there exists l which is not from the same set, as k. Here we arrive at a contradiction as both $k, l \in 1, \ldots, r$, otherwise the attacker would have used some $GID_l \notin RL$.

Hence, we have shown that the attacker cannot construct a query of the form cs for a constant c. Therefore, under conditions that hold with all but $\mathcal{O}(q^2/p)$ probability, the attacker cannot distinguish between the cases when T is random or $T = s$ thus the advantage of $\mathcal{A}$ in the security game is at most $\mathcal{O}(q^2/p)$. □

4 Future Work

We proposed a scheme for efficient identity-based user revocation in multi-authority CP-ABE. In the future, our work can be continued in several directions.

As the original work of [LW11] allows to use each attribute only once in the row labelling ρ of the access matrix, the same applies for our extended scheme. However applying a simple transformation (detailed in [LW11], Appendix B) in our case also leads to a system that allows the use of attributes k times, but this transformation does not lead to an efficient system when the attribute universe is large. The elimination of this weakness would lead to a more usable system in practice together with our extension.

The method of identity-based user revocation can be the foundation of a future method that allows non monotonic access structures in multi-authority setting. However our scheme cannot be applied directly for this purpose it may be used to develop ideas in this field.

The security of our construction is proved in the generic bilinear group model, although we believe it would be possible to achieve some stronger assumptions with the use of the Boneh-Boyen-Goh framework [BBG05], analogously to the method of [LSW10]. This type of work would be interesting even if it resulted in a moderate loss of efficiency from our existing system.

Bibliography

[BBG05] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In *Advances in Cryptology–EUROCRYPT 2005*, pages 440–456. Springer, 2005.

[Bei96] Amos Beimel. *Secure schemes for secret sharing and key distribution*. PhD thesis, PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.

[BSW07] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy*, pages 321–334. IEEE Computer Society, 2007.

[Cha07] Melissa Chase. Multi-authority Attribute Based Encryption. In SalilP. Vadhan, editor, *Theory of Cryptography*, volume 4392 of *Lecture Notes in Computer Science*, pages 515–534. Springer Berlin Heidelberg, 2007.

[LC10] Zhen Liu and Zhenfu Cao. On Efficiently Transferring the Linear Secret-Sharing Scheme Matrix in Ciphertext-Policy Attribute-Based Encryption. *IACR Cryptology ePrint Archive*, 2010:374, 2010.

[LCH+11] Zhen Liu, Zhenfu Cao, Qiong Huang, Duncan S Wong, and Tsz Hon Yuen. Fully secure multi-authority ciphertext-policy attribute-based encryption without random oracles. In *Computer Security–ESORICS 2011*, pages 278–297. Springer, 2011.

[LLLS10] Xiaohui Liang, Rongxing Lu, Xiaodong Lin, and Xuemin Sherman Shen. Ciphertext policy attribute based encryption with efficient revocation. *Technical-Report, University of Waterloo*, 2010.

[LSW10] Allison Lewko, Amit Sahai, and Brent Waters. Revocation systems with very small private keys. In *Security and Privacy (SP), 2010 IEEE Symposium on*, pages 273–285. IEEE, 2010.

[LW11] Allison Lewko and Brent Waters. Decentralizing attribute-based encryption. In *Advances in Cryptology–EUROCRYPT 2011*, pages 568–588. Springer, 2011.

[LZW+13] Yang Li, Jianming Zhu, Xiuli Wang, Yanmei Chai, and Shuai Shao. Optimized Ciphertext-Policy Attribute-Based Encryption with Efficient Revocation. *International Journal of Security & Its Applications*, 7(6), 2013.

[RNS11] Sushmita Ruj, Amiya Nayak, and Ivan Stojmenovic. Dacc: Distributed access control in clouds. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on*, pages 91–98. IEEE, 2011.

[SW05] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *Advances in Cryptology–EUROCRYPT 2005*, pages 457–473. Springer, 2005.

[Wat11] Brent Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *Public Key Cryptography–PKC 2011*, pages 53–70. Springer, 2011.

[WLWG11] Guojun Wang, Qin Liu, Jie Wu, and Minyi Guo. Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers. *Computers & Security*, 30(5):320–331, 2011.