Attribute-Based Encryption Optimized for Cloud Computing

Máté Horváth

***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 [8] 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. The formal security proof of our scheme is given in the generic bilinear group and random oracle models.**

***Index Terms*—storage in clouds, access control, attribute-based encryption, user revocation, multi-authority.**

I. 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 have 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 [16] 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 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. [2], [19]) 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 (CA) 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). To overcome this problem, we are going to make use of the results of Lewko and Waters [8] 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 "CRYPTOPROJECT2015").

Simultaneous solutions for these two problems could enhance flexible access control in cloud-based secure data storage. Such "optimized" CP-ABE could hide symmetric keys, which are used to efficiently encode large amounts of data, and reveal them only for authorized users, who can be identified through expressive access policies (for details see Figure 1).

M. Horváth works in the Laboratory of Cryptography and System Security (CrySyS Lab) at Budapest University of Technology and Economics, Department of Networked Systems and Services, Magyar tudósok krt. 2, 1117 Budapest, Hungary. E-mail: mhorvath@crysys.hu

Manuscript received February 12, 2015; revised May 25, 2015

Related Work.: The concept of ABE was first proposed by Sahai and Waters [16] as a generalization of identity-based encryption. Bethencourt et al. [2] 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 [19].

The problem of building ABE systems with multiple authorities was first considered by Chase [5] 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 [8] 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. Several other multi-authority schemes (e.g. [14], [18]) were shaped to the needs of cloud computing, although these lack for efficient user revocation.

Attribute revocation with the help of expiring attributes was proposed by Bethencourt et al. [2]. For single authority schemes Sahai et al. [15] introduced methods for secure delegation of tasks to third parties and user revocation through piecewise key generation. Ruj et al. [14], Wang et al. [18] and Yang et al. [20] 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 were applied to the scheme of Waters [19]. Liang et al. [11] gives the right of controlling the revoked set to a "system manager" while Li et al. [10], follow [7], from the field of broadcast encryption systems and give the revocation right directly to the encryptor. This later was further developed by Li et al. [9] achieving full security with the help of dual system encryption. For this approach, but in key-policy ABE, Qian and Dong [13] showed fully secure solution.

To the best of our knowledge no multi-authority system is integrated with identity-based user revocation and our work is the first in this direction.

Contribution.: Based on [8] and [7] 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 can be given directly to the encryptor, just like the right to define the access structure which fits to the cloud computing scenario.

A preliminary version of this work appeared in [6]. In this paper, we make substantial extensions to the contributions presented in [6], including a new, detailed security analysis of our proposed scheme, with a rigorous proof in the generic bilinear group and random oracle models, as well as proposal for an application approach in the cloud storage scenario and detailed explanations and reflections on related works.

Organization.: In Section II we introduce the later used theoretical background. In Section III the details of our scheme can be found together with efficiency and security analysis. Directions for further research are proposed in the last section.

II. BACKGROUND

We first briefly introduce bilinear maps, and provide the relevant background on access structures and secret sharing schemes. Then we give the algorithms of Ciphertext Policy Attribute-Based Encryption with identity-based user revocation.

A. Bilinear maps

We present the most important facts 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 $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

B. Access Structures and Secret Sharing

The requirements of decryption in an ABE scheme can be expressed using access structures (for formal definition see [1]), which determines all the authorised sets of attributes that allow decryption. Most ABE schemes (like ours) are restricted to *monotone access structures*, meaning that any superset of an authorized set is authorized as well. We note that (inefficiently) general access structures also can be realized using our techniques by having the not of each attribute as separate attribute.

To enforce the access structure, determined by the encryptor, we are going to make essential use of Linear Secret Sharing Schemes (LSSS). Here we adopt the definitions from those given in [1].

Definition 1 (Linear Secret Sharing Scheme [1]): A secret-sharing scheme Π over a set of attributes U is called linear (over $\mathbb{Z}_p$) if

1) the shares for each attribute form a vector over $\mathbb{Z}_p$,
2) there exists a matrix A with ℓ rows and n columns called the share-generating matrix for Π. For all $x = 1, \ldots, \ell$, the x^{th} row of A is labelled by an attribute $\rho(x)$, where ρ is a function from $\{1, \ldots, \ell\}$ to U. 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 $(Av)_x = \lambda_x$ belongs to attribute $\rho(x)$.

In [1] 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 [1] 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 [8] - 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.

C. 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 central authority (CA) runs this algorithm with GP as input to produce its own secret key and public key pair, SK^*, PK^*.

Identity KeyGen$(GP, RL, GID, SK^*) \rightarrow K^*_{GID}$
The central authority runs this algorithm upon a user request for identity secret key. It checks whether the request is valid and if yes (i.e. the user's global identifier, denoted by GID, is not part of the RL revocation list: $GID \notin RL$), generates K^*_{GID} using the global parameters and the secret key of the CA.

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

KeyGen$(GP, SK, GID, i) \rightarrow K_{i,GID}$
The attribute 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$(GP, \mathcal{M}, (A, \rho), \{PK\}, PK^*, RL) \rightarrow CT$
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$(GP, CT, (A, \rho), \{K_{i,GID}\}, K^*_{GID}, RL) \rightarrow \mathcal{M}$
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.

III. OUR RESULTS

To build our model we will use the prime order group construction of Lewko and Waters [8], because of its favourable property of having independent attribute authorities. In order to achieve identity-based revocation we supplement the distributed system with a Central Authority. However it seems to contradict with the original aim of distributing the key generation right, this additional authority would generate only secret keys for global identifiers ($GID \in \mathbb{Z}_p$) 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.

Approach to the Cloud Storage Scenario: We give a high-level description about a possible application of the algorithms that we proposed in Subsection II-C (for graphical depiction see Figure 1). Because of efficiency reasons it is practical to encrypt data using a symmetric cipher, always with fresh random number as key. Access control is achieved by encrypting the symmetric key using CP-ABE and attaching the encrypted key to the ciphertext that is stored by the cloud service provider (CSP). Decryption is possible for users, who can obtain the symmetric key, or with other words those, who possess the necessary attributes and were not revoked. Attribute Authorities are run locally on trusted servers of organisations, that are using the system, while the Central Authority is run by the CSP, which also maintains (archives, publishes) the RL revocation list, based on the revocation requests from authorised parties of the organisations. The ABE encryption always uses the fresh RL and ABE decryption is run with the RL at the encryption time of the ciphertext, which are obtained from the CSP. This approach automatically leads to lazy re-encryption of ciphertext, as fresh symmetric key and RL are used whenever data is edited.

a) 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 [7]. A recent[1] work of Cao and Liu [4] points out an inherent drawback of the [7] scheme, namely that for malicious users it is worth to exchange their decryption keys in order to maximize their interests. However we utilize similar techniques as [7], our system is not vulnerable to this kind of misuse, because unlike in broadcast encryption, where having a non-revoked secret key is the only requirement for decryption, in ABE, users are also required to fulfil requirements related to their attributes. Thus such collusion could have

[1] [4] appeared on ePrint some months later than our work.

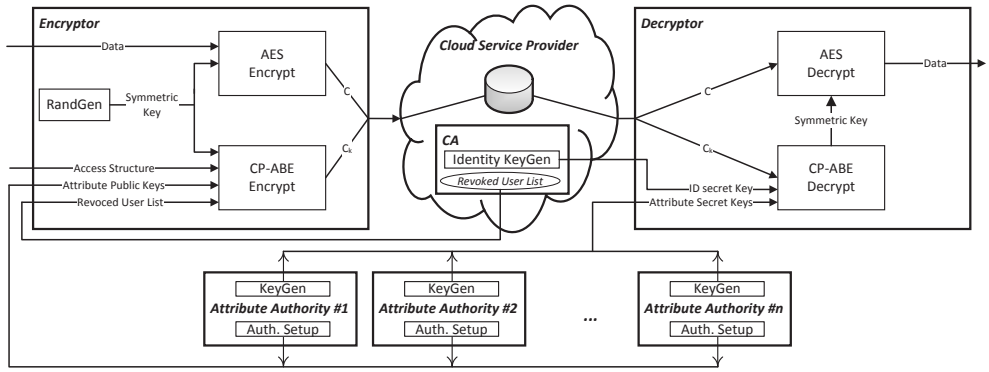

Figure 1. A possible usage of the proposed multi-authority CP-ABE scheme for access control in a cloud storage scenario.

only a restricted benefit[2] as the set of ciphertexts that can be decrypted is also restricted by the used attribute secret keys (which cannot be mixed between different users).[3]

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 $\sum_{k=1}^{r} s_k = s^*$. It will then create a ciphertext such that any revoked user with GID_k^* will not be able to incorporate the k^{th} share and thus not decrypt the message.

This approach presents the following challenges. First, we need to make crucial that the decryptor needs 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_k^* cannot do anything useful with share k. 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 [8] 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. If a user with a particular identifier GID satisfies the access structure, he can reconstruct s in the exponent by raising the group elements to the proper exponents. This operation will simultaneously reconstruct the share of 0 and thus the $e(H(GID), g)$ blinding terms will cancel out. 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 $GID \neq GID_k^*$ can obtain two linearly independent equations (in the exponent) involving the share s_k, which he will use to solve for the share s_k. However, if $GID = GID_k^*$, the obtained equations are going to be linearly dependent and the user will not be able to solve the system.

In the third case, the attack we need to worry about is where a user with GID_k^* processes ciphertext share l, while another user with GID_l^* processes share k, and then they combine their results. To prevent collusion, we use $H(GID)$ 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.

A. Our Construction

To make the following notions more understandable, in Table I we summarize the new keys and variables (compared to [8]) which we introduce in our construction. Based on the above principles, the proposed algorithms are the following:

Table I
THE SUMMARY OF OUR NEW NOTATIONS

Notation	Meaning	Role
PK^*	$\{g^a, g^{1/b}\}$	public key of the Central Authority
SK^*	$\{a, b\}$	secret key of the Central Authority
K^*_{GID}	$H(GID)^{(GID+a)b}$	global identity secret key of a user
$C^*_{1,k}$	$\left(g^a g^{GID_k^*}\right)^{-s_k}$	revoked user identification in CT
$C^*_{2,k}$	$g^{s_k/b}$	k^{th} secret share in the CT
RL	$\{GID_1^*, \ldots, GID_r^*\}$	list of r revoked users

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

[2]Of course, when users reveal their secret keys, we cannot hope for security in any encryption method, but assuming honest users, it is their interest to keep the secrets. As long as the attributes of (still non-revoked) colluding users do not cover all the access policies, our scheme will not reveal all ciphertexts for the malicious group.

[3]We also note that the flaw of [7]'s security proof, mentioned by [4] does not affect our results, as we use different proof technique.

Central Authority Setup$(GP) \rightarrow (SK^*, PK^*)$
The algorithm chooses random exponents $a, b \in \mathbb{Z}_p$, keeps them as secret key $SK^* = \{a, b\}$ and publishes $PK^* = \{g^a, g^{1/b}\}$.

Identity KeyGen$(GP, RL, GID, SK^*) \rightarrow K^*_{GID}$
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 $PK = \{e(g,g)^{\alpha_i}, g^{y_i}\ \forall i\}$ as its public key. It keeps $SK = \{\alpha_i, y_i\ \forall i\}$ as its secret key.

KeyGen$(GP, SK, GID, i) \rightarrow K_{i,GID}$
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$(GP, \mathcal{M}, (A, \rho), \{PK\}, PK^*, RL) \rightarrow CT$
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_k such that $s^* = \sum_{k=1}^{r} s_k$. The CT ciphertext is computed as

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

for all $x = 1, \ldots, n$ and $k = 1, \ldots, r$.

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

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

which equals to $e(g,g)^s$, so the message can be obtained as $\mathcal{M} = C_0 / e(g,g)^s$.

To see the soundness of the Decryption algorithm observe that after substituting the corresponding values we get the following:

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

Remark 1. 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 these to the method of [10]. However in this way additional modifications are still needed to prevent the CA from being able to decrypt any ciphertext by computing $e(g^\beta, g^s)$.

Remark 2. Supposing that we have a honest but curious CSP, which does not collude with the users, it is also possible to achieve indirect revocation (similarly to [11], [15]), with simple modifications on our scheme. With other words, the CSP could fully supervise user revocation based on the revocation requests from parties, authorised for this. We only need to modify the Encrypt algorithm to compute $C, C_0, C_{1,x}, C_{2,x}$ as originally and $C'_{3,x} = g^{y_{\rho(x)} r_x} \quad \forall x = 1, \ldots, n$. These values would form CT' that is sent to the CSP, where the collusion resistant CT with the revocation information is computed and published. CT has the same form as earlier, the only difference is that the blinding vector w is chosen by the CSP, so $\omega_x, C^*_{1,k}, C^*_{2,k}$ (as previously) and $C_{3,x} = C'_{3,x} \cdot g^{\omega_x}$ are computed also by the CSP. The main advantage of this approach is that immediate and efficient (partial) re-encryption can be achieved as only $w, s_k, \omega_x, C^*_{1,k}, C^*_{2,k}$ and $C_{3,x}$ need to be recomputed after a revocation event.

Remark 3. Alternatively, it is also possible to give revocation right directly to the encryptor by simply publishing a user list instead of RL. In this case RL would be defined by the user, separately for each ciphertext, and attached to CT.

B. Efficiency

Traditional, attribute-based user revocation (e.g. [14], [18], [20]) affects 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 algorithms. 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 [8], 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 more efficiently, using the algorithm of Liu and Cao in [12].

C. Security

Before giving the formal proof, we point out 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 [8], 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 the blinding vector 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 $\mathscr{B}$ by using $H(GID)$ in both the identity and attribute secret keys.

1) Security Model: We now define (chosen plaintext) security of multi-authority CP-ABE system with identity-based revocation. Security is defined through a *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). Informally, $\mathcal{A}$ can determine a set of corrupted attribute authorities, ask for any identity and attribute keys and specify messages, on which it will be challenged using the revocation list and access matrix of its choice. The only (natural) restriction in the above choices is that $\mathcal{A}$ cannot ask for a set of keys that allow decryption, in combination with any keys that can be obtained from corrupt authorities in case of a non revoked GID_k. In case of revoked identities we can be less restrictive: corrupted attributes alone cannot satisfy the access policy, but it it might be satisfied together with attributes from honest authorities. $\mathcal{A}$ wins the game if it respects the rules and can decide which of its challenge messages were encrypted by the challenger. The formal security game consists of the following rounds:

Setup. The challenger runs the Global Setup algorithm to obtain the global public parameters GP. $\mathcal{A}$ specifies a set $AA' \subseteq AA$ of corrupt attribute authorities and uses the Authority Setup to obtain public and private keys. For honest authorities in $AA \setminus AA'$ and for the Central Authority, the challenger obtains the corresponding keys by running the Authority Setup and Central Authority Setup algorithms, and gives the public keys to the attacker.

Key Query Phase. $\mathcal{A}$ adaptively issues private key queries for identities GID_k (which denotes the k^{th} GID query). 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 2: 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 has at most a negligible advantage in the above defined security game.

2) Security Analysis: We are going to prove the security of our construction in the generic bilinear group model previously used in [2], [3], [8], modelling H as a random oracle. Security in this model assures us that an adversary cannot break the scheme with only black-box access to the group operations and H. Intuitively, this means that if there are any vulnerabilities in

our construction, then these must exploit specific mathematical properties of elliptic curve groups or cryptographic hash functions used when instantiating the scheme.

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 III-C1. The above described construction is secure according to Definition 2 in the generic bilinear group and random oracle models. The advantage of $\mathcal{A}$ is $\mathcal{O}(q^2/p)$.

In our proof we are going to use the following strategy. First we identify events that occur only with negligible probability, namely that the attacker is able to guess certain values successfully and that the oracle returns the same value for different queries. Assuming that these do not happen, we examine the (exponent) values which the attacker can obtain during the game. We show that $\mathcal{A}$ can recognise the challenge ciphertext only if is has used $GID_K \notin RL$ with a satisfying attribute set or has broken the rules of the game.

Proof:

We describe the generic bilinear model as in [3]. 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 to 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 $\psi_1(y)$.

The challenger and the attacker play the security game (described in III-C1) and compute each value with respect to the generic bilinear group and random oracle models (i.e. send queries to the group oracle that responds with randomly assigned values). When $\mathcal{A}$ requests e.g. $H(GID_k)$ for some GID_k for the first time, the challenger 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 $H(GID_k)$.

We are going to show that in order to determine $\beta \in \{0,1\}$, $\mathcal{A}$ has to be able to compute $e(g,g)^{s^* h_{GID_k}}$ for any $k = 1,\ldots,r$, which is possible only with negligible probability without breaking the rules of the game.

We can assume that each of the attacker's queries to the group oracles either have input values that were given to $\mathcal{A}$ during the security game or were received from the oracles in response to previous queries. This is because of the fact that both ψ_0 and ψ_1 are random injective maps from $\mathbb{Z}_p$ into a set of at least p^3 elements, so 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[4] in the variables $y_i, \alpha_i, \lambda_x, r_x, \omega_x, h_{GID_k}, a, b, s_k$, 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 $s^*, w_2, \ldots, w_\ell$.)

Furthermore we also assume that for each pair of different queries (corresponding to different polynomials), $\mathcal{A}$ receives different answers from the oracle. Since the maximum degree of polynomials is 8 (see the possible polynomials later), using the Schwartz-Zippel lemma [17] we get 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 game is $\mathcal{O}(q^2/p)$, which is negligible. Now suppose that it does not happen.

In order to determine β, the attacker clearly needs to recover s. [8] showed that without a satisfying set of attributes an attacker cannot make a query of the form $c(s + 0 \cdot h_{GID_k})$ (where c is a constant) thus has only negligible advantage in distinguishing an encoded message from a random group element (when using 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 0) which also shows - following their reasoning - that an expression in the form cs cannot be formed either. In our case, however, 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.

It can be seen that the case when $GID_k \in UL \setminus RL$ is equivalent to the original scheme of [8]. Consequently, from now on we can assume that all $GID_k \in RL$ and the challenge access policy is satisfied, thus simulating that all revoked users are colluding and prior to their revocation they were all able to decrypt. We will show that $\mathcal{A}$ cannot make a query of the form $-c(s^* h_{GID_k})$ and so not cs.

Based on the above assumptions 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. (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 shares of s^* can appear in two different forms: as ω_x and s_k, so we investigate whether $\mathcal{A}$ can achieve the desired value from these or not.

1) 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 $h_{GID_k}\omega_x$ and thus which can lead 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, because these 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, because in that case corrupted attributes alone would satisfy the access structure.

[4]These expressions can appear in the exponent of $e(g,g)$.

Table II
POSSIBLE RELEVANT QUERY TERMS

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

2) When trying to obtain $s^* h_{GID_k}$ using s_k, we can observe that in each possible query term, s_k appears as multiplier either in all monads or in none of them. Evidently, terms without s_k are useless (see Table II for the relevant terms) for the attackers purposes and terms containing the $s_k h_{GID_l}$ monad can be useful. As it can be seen in Table II, there are two types of terms which contain the necessary monad:

$$s_k a h_{GID_l} + GID_k^* s_k h_{GID_l}$$

and

$$s_k a h_{GID_l} + GID_l^* s_k h_{GID_l}.$$

Multiplying their subtraction by $c/(GID_k^* - GID_l^*)$ it is possible to gain $c \cdot s_k h_{GID_l}$, if $k \neq l$. 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. Nevertheless, according to our assumption that $GID_l^* \in RL$ for all $l = 1, \ldots, r$ there must be a $k = l$ as k runs over $1, \ldots, r$. We conclude that it is possible to gain $s_k h_{GID_l}$ for all k for any fixed l, if the attacker has used some $GID_l \notin RL$, which is again contradiction.

Hence, we have shown that under conditions that hold with all but $\mathcal{O}(q^2/p)$ probability, $\mathcal{A}$ cannot query $c(s^* h_{GID_k})$ (neither using ω_x nor s_k) therefore cannot get s without breaking the rules of the security game. It follows than, that the advantage of $\mathcal{A}$ is at most $\mathcal{O}(q^2/p)$. ■

IV. CONCLUSION

We proposed a scheme for efficient identity-based user revocation in multi-authority CP-ABE with several advantageous feature compared with attribute-based revocation. Our results fulfil specific needs of the cloud environment, thus optimizes ABE for real world usage. In the future, our work can be continued in several directions.

First and foremost, extensive comparisons are needed between the different revocation schemes proposed for CP-ABE to understand better their performance between different circumstances.

Securely forwarding the revocation related computations to the CSP (or even to the user), as we mentioned in Remark 2, could allow immediate banning of a user, disallowing the decryption of all previously (and later) encrypted ciphertexts. Steps in this direction, without assuming trusted CSP, would be useful.

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 full security by adapting the dual system encryption methodology, which was also used by Lewko and Waters [8] in their composite order group construction. This type of work would be interesting even if it resulted in a moderate loss of efficiency from our existing system.

ACKNOWLEDGMENTS

This work was started as a master thesis at Eötvös Loránd University, in the Security&Privacy program of EIT ICT Labs Masterschool. The author would like to thank all the help and valuable advice of Levente Buttyán from CrySyS Lab. He is also grateful to Viktória Villányi and Péter Ligeti for the useful discussions and to the anonymous reviewers of SOFSEM'15 and Infocommunications Journal for the valuable remarks.

REFERENCES

[1] Amos Beimel. *Secure schemes for secret sharing and key distribution*. PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.
[2] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy*, pages 321–334, 2007.
[3] 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.
[4] Zhengjun Cao and Lihua Liu. Analysis of Lewko-Sahai-Waters Revocation System. Cryptology ePrint Archive, Report 2014/937, 2014. http://eprint.iacr.org/.
[5] Melissa Chase. Multi-authority Attribute Based Encryption. In *Theory of Cryptography*, volume 4392 of *LNCS*, pages 515–534. Springer Berlin Heidelberg, 2007.
[6] Máté Horváth. Attribute-Based Encryption Optimized for Cloud Computing. In G.F. Italiano et al., editor, *SOFSEM 2015: Theory and Practice of Computer Science*, number 8939 in LNCS, pages 566–577. Springer, 2015.
[7] Allison Lewko, Amit Sahai, and Brent Waters. Revocation systems with very small private keys. In *IEEE Symposium on Security and Privacy*, pages 273–285, 2010.
[8] Allison Lewko and Brent Waters. Decentralizing attribute-based encryption. In *Advances in Cryptology–EUROCRYPT 2011*, pages 568–588. Springer, 2011.
[9] Qinyi Li, Hu Xiong, and Fengli Zhang. Broadcast revocation scheme in composite-order bilinear group and its application to attribute-based encryption. *International Journal of Security and Networks*, 8(1):1–12, 2013.
[10] 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.
[11] Xiaohui Liang, Rongxing Lu, Xiaodong Lin, and Xuemin Sherman Shen. Ciphertext policy attribute based encryption with efficient revocation. *TechnicalReport, University of Waterloo*, 2010.
[12] 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.
[13] Jun-lei Qian and Xiao-lei Dong. Fully secure revocable attribute-based encryption. *Journal of Shanghai Jiaotong University (Science)*, 16:490–496, 2011.

[14] Sushmita Ruj, Amiya Nayak, and Ivan Stojmenovic. Dacc: Distributed access control in clouds. In *IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 91–98, 2011.
[15] Amit Sahai, Hakan Seyalioglu, and Brent Waters. Dynamic credentials and ciphertext delegation for attribute-based encryption. In *Advances in Cryptology–CRYPTO 2012*, pages 199–217. Springer, 2012.
[16] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *Advances in Cryptology–EUROCRYPT 2005*, pages 457–473. Springer, 2005.
[17] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, October 1980.
[18] 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.
[19] 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.
[20] Kan Yang, Xiaohua Jia, Kui Ren, and Bo Zhang. DAC-MACS: Effective data access control for multi-authority cloud storage systems. In *INFOCOM, 2013 Proceedings IEEE*, pages 2895–2903, 2013.

Máté Horváth obtained his MSc diploma in computer science in the Security and Privacy program of EIT ICT Labs at the University of Trento (Italy) and Eötvös Loránd University (Hungary). His bachelor degree is in mathematics from the Technical University of Budapest. He has been doing research in the CrySyS Lab under the guidance of prof. Levente Buttyán since 2014.