Efficient Revocable Attribute-Based Encryption with Hidden Policies

1st Dominik Ziegler
Know-Center GmbH
Graz, Austria
dominik.ziegler@tugraz.at

2nd Alexander Marsalek
IAIK, Graz University of Technology
Graz, Austria
alexander.marsalek@iaik.tugraz.at

***Abstract*—We present a novel *Ciphertext-Policy Attribute-Based Encryption* (CP-ABE) scheme, which bridges the gap between highly dynamic (industrial) environments and resource-constrained devices. Our construction combines outsourced-decryption, hidden policies and revocation to cope with the requirements posed by such environments. In contrast to existing schemes, which typically rely on composite order bilinear groups, we present a scheme in prime order groups. The resulting scheme is more efficient as it relies on smaller group orders. We prove our scheme is secure under the *Symmetric External Diffie-Hellman* (SXDH) assumption. Lastly, we compare our scheme against existing schemes and provide timing results of our software implementation. Our evaluation shows that the proposed scheme is flexible enough for the targeted environment while improving performance by an order of magnitude.**

***Index Terms*—access control, revocation, attribute-based encryption, hidden policies,**

I. Introduction

Attribute-Based Encryption (ABE) [1] is a cryptographic approach to achieve fine-grained access control on encrypted data. Recipients are described as a set of attributes. Only parties that hold the necessary attributes and a corresponding secret key can decrypt a ciphertext. There exist two types of ABE: *Key-Policy Attribute-Based Encryption* (KP-ABE) [2] and *Ciphertext-Policy Attribute-Based Encryption* (CP-ABE) [3]. In the former schemes, access control policies are associated with a party's secret key and ciphertexts are tagged with a set of attributes. In the latter schemes, the roles are reversed. Parties obtain secret keys associated with the attributes they hold. In turn, ciphertexts incorporate access policies.

Problem Statement. While ABE promises fine-grained access control, constructions typically rely on expensive bilinear pairing operations to decrypt a ciphertext. As a result, the general tenor is that ABE should only be used for devices with relatively high computing power. However, this cannot possibly be provided in resource-constrained environments such as the *Internet of Things* (IoT). Attempts such as outsourcing decryption to more powerful proxies, as shown in [4], deal with this limitation. They, in turn, entail another problem: The proxy or decryption server can learn the access policy. Consequently, the ciphertext and its associated policy may leak sensitive information.

978-1-XXXX-XXXX-X/20/$31.00 ©2020 IEEE

Additional obstacles become apparent in highly dynamic settings such as industrial applications: Keys need to be efficiently revoked to model changing permissions of involved parties. One promising approach is to manage users in so-called *attribute groups* [5]. To revoke an attribute, the user is removed from said attribute group. All ciphertexts associated with an attribute are then re-encrypted by some proxy.

In reality, combining various approaches into one practical solution is not an easy endeavour. For instance, protocols offering hidden policies, i.e. which hide the attribute information of access structures, such as [6, 7], typically leverage convenient features of bilinear groups of composite order. Most notable, they rely on orthogonal subgroups, a feature not present in prime order groups. However, the security of composite order groups relies on the hardness of factoring the group order. As a result, protocols using composite order groups require large group orders, which, in turn, results in inefficient pairing operations. To make matters worse, composite order groups can only be realised in so-called *Type-1* or *symmetric* pairings, which are typically implemented from supersingular (hyper) elliptic curves. For the most promising families of supersingular curves, it was shown [8] that the discrete logarithm problem is not as difficult as initially anticipated. In general, research therefore suggests to use so-called *Type-3* or *asymmetric pairings* and prime order groups [9]. In literature, it is often argued that protocols using Type-1 pairings can easily be implemented in Type-3. It seems, though, that translating the security proofs fails in practice for many schemes or requires additional effort to translate security proofs correctly [10, 11].

Our Approach. Our approach tackles the problem from three angles. First, we propose a novel CP-ABE solution which combines hidden policies, outsourced decryption and revocation. Our approach thus perfectly copes with requirements posed by highly dynamic and resource-constrained (industrial) environments. Second, we design our scheme around Type-3 pairings. To do so, we rely on *Dual Pairing Vector Spaces* (DPVS) [12], a technique to achieve orthogonality in prime-order groups. As a result, we achieve better performance in practice without sacrificing security. Third, we evaluate our proposed protocol and provide a security proof in a more generalised setting.

Our Contributions. The contributions of this paper are:

- We present a novel CP-ABE scheme suitable for dynamic

environments, with hidden policies, instant revocation through attribute groups and outsourced decryption.
- We design the proposed scheme in prime order groups as opposed to composite order groups.
- We rely on DPVS using a 4-dimensional vector space. However, only two basis vectors are used for the main scheme. The remaining two are used for the security proof. In practice, this means that we can reduce the size of the keys and ciphertext by two group elements.
- We provide a security proof under the *Symmetric External Diffie-Hellman* (SXDH) assumption.
- Lastly, we present a thorough performance evaluation. We compare our scheme against existing schemes on a conceptional level. Furthermore, we provide a purely Java and Kotlin based software implementation. Finally, we compare our construction in practice against several existing implementations. The results show a significant performance gain over similar approaches.

The rest of the paper is organised as follows. First, in Section II, we describe the notation and necessary background. In Section III, we present our approach. Next, in Section IV, we prove the security of our construction. Then, we evaluate the performance of our scheme and compare it against existing solutions in Section V. Finally, in Section VI, we present and discuss related work. We conclude our work in Section VII.

II. Background

In this section, we define the necessary notation, concepts and assumptions used throughout this document.

A. (Asymmetric) Bilinear Groups of Prime Order

Let $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ be multiplicative cyclic groups with prime order p. Let g_1 and g_2 be generators of $\mathbb{G}_1$ and $\mathbb{G}_2$. We define the bilinear map as: $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. The bilinear map e has the following two properties:

- Bilinearity: For all $u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$ and $a, b \in \mathbb{Z}_p^*$, linearity is given: $e(u^a, v^b) = e(u, v)^{ab}$.
- Non-degeneracy: Generators g_1, g_2 must not map to the identity element: $e(g_1, g_2) \neq 1$.

We rely on asymmetric pairings, meaning $\mathbb{G}_1 \neq \mathbb{G}_2$. Furthermore, we define vectors of group elements. Given a vector $\boldsymbol{v} = (v_1, \ldots, v_n)$ and a generator $g_i \in \mathbb{G}_i$, we write $g_i^{\boldsymbol{v}}$ to describe the tuple of elements: $g_i^{\boldsymbol{v}} = (g_i^{v_1}, g_i^{v_2}, \ldots, g_i^{v_n})$. For any element $a \in \mathbb{Z}_p^*$ we let $g_i^{a\boldsymbol{v}} = (g_i^{av_1}, \ldots, g_i^{av_n})$. For the two vectors $\boldsymbol{v}, \boldsymbol{w}$ we define $g_i^{\boldsymbol{v}+\boldsymbol{w}} = (g_i^{v_1+w_1}, \ldots, g_i^{v_n+w_n})$. Lastly, we define the componentwise pairing $e_n(g_1^{\boldsymbol{v}}, g_2^{\boldsymbol{w}}) = \prod_{i=1}^{n} e(g_1^{v_i}, g_2^{w_i}) = e(g_1, g_2)^{\boldsymbol{v}\cdot\boldsymbol{w}}$.

B. Dual Pairing Vector Spaces

Our construction relies on the concept of *Dual Pairing Vector Spaces* (DPVS) [12]. It defines two random bases, $\mathbb{B} = (\boldsymbol{b_1}, \ldots, \boldsymbol{b_n})$ and $\mathbb{B}^* = (\boldsymbol{b_1^*}, \ldots, \boldsymbol{b_n^*})$ in $\mathbb{Z}_n^*$. The bases are "dual orthonormal". That means that $\boldsymbol{b_i} \cdot \boldsymbol{b_j^*} = 0 \pmod p$ if $i \neq j$, and $\boldsymbol{b_i} \cdot \boldsymbol{b_i^*} = \psi$ for all i. We choose $\psi \xleftarrow{R} \mathbb{Z}_p^*$. For generators $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ we note that $e_n(g_1^{\boldsymbol{b_i}}, g_2^{\boldsymbol{b_j^*}}) = 1$, whenever $i \neq j$.

C. Linear Secret Sharing Scheme

A *Linear Secret Sharing Scheme* (LSSS) $\Pi = (\mathbb{A}, \rho)$ is a pair, which, given the probability distribution ρ, allows distributing a secret $s \in \mathbb{Z}_p^*$ to n participants. It is called linear over a set of parties P *iff*:

1) The secret s can be represented as a vector over $\mathbb{Z}_p^*$, over n participants such that each party possesses an element.
2) Every set S of participants in an access structure $\mathbb{A}$ can reconstruct the secret using a linear combination of elements.

D. Symmetric External Diffie-Hellman (SXDH) Assumption

Informally, the SXDH assumption states that, given a bilinear group of prime order, the *Decisional Diffie–Hellman* (DDH) assumption problems are intractable in $\mathbb{G}_1$ and $\mathbb{G}_2$. We formally define the DDH problem for a group $\mathbb{G}_i$ as follows:

Definition 1. *Let G be a bilinear group of prime order p. Given generators $g_i \in \mathbb{G}_i, g_j \in \mathbb{G}_j$ and elements $a, b, c \xleftarrow{R} \mathbb{Z}_p^*$. Let:*

$$\boldsymbol{D} = \langle G, g_i, g_j, g_i^a, g_j^b \rangle$$

Given a distribution $\boldsymbol{D}$, we say that any polynomial time algorithm $\mathcal{B}$ that outputs $\{0, 1\}$ has an advantage ϵ if:

$$\left| Pr[\mathcal{B}(\boldsymbol{D}, T = g_i^{ab}) = 1] - Pr[\mathcal{B}(\boldsymbol{D}, T = g_i^{ab+c}) = 1] \right| \geq \epsilon$$

We henceforth refer to $DDH1$ and $DDH2$ as the Decisional Diffie–Hellman assumptions in $\mathbb{G}_1$ and $\mathbb{G}_2$.

E. (Asymmetric) Subspace Assumption

The *decisional subspace assumption* [11] states that given a bilinear group $\mathbb{G}$ of prime order and an element $g^{\boldsymbol{v}}$, there does not exist any efficient polynomial time algorithm, which can distinguish, whether $\boldsymbol{v}$ is chosen randomly from a span $\boldsymbol{b_1^* b_2^*}$ or a larger span $\boldsymbol{b_1^* b_2^* b_3^*}$. We now describe the subspace assumption from the SXDH assumption in $\mathbb{G}_1$ for our construction formally. It is analogous to the subspace assumption given in [13]. The subspace assumption in $\mathbb{G}_2$ is identical, with the roles of $\mathbb{G}_1$ and $\mathbb{G}_2$ being reversed.

Definition 2. *Given $G = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$, $(\mathbb{B}, \mathbb{B}^*) \xleftarrow{R} Dual(\mathbb{Z}_n^*)$ and elements $\tau_1, \tau_2, \mu_1, \mu_2 \xleftarrow{R} \mathbb{Z}_p^*$. Let:*

$$U_1 = g_2^{\mu_1 \boldsymbol{b_1^*} + \mu_2 \boldsymbol{b_{k+1}^*}}, U_2 = g_2^{\mu_1 \boldsymbol{b_2^*} + \mu_2 \boldsymbol{b_{k+2}^*}}, \ldots, U_k = g_2^{\mu_1 \boldsymbol{b_k^*} + \mu_2 \boldsymbol{b_{2k}^*}},$$

$$V_1 = g_1^{\tau_1 \boldsymbol{b_1}}, V_2 = g_1^{\tau_1 \boldsymbol{b_2}}, \ldots, V_k = g_1^{\tau_1 \boldsymbol{b_k}}$$

$$W_1 = g_1^{\tau_1 \boldsymbol{b_1} + \tau_2 \boldsymbol{b_{k+1}}}, W_2 = g_1^{\tau_1 \boldsymbol{b_2} + \tau_2 \boldsymbol{b_{k+2}}}, \ldots, W_k = g_1^{\tau_1 \boldsymbol{b_k} \tau_2 \boldsymbol{b_{2k}}}$$

$$\boldsymbol{D} = \langle G, g_2^{\boldsymbol{b_1 *}}, \ldots, g_2^{\boldsymbol{b_k *}}, g_2^{\boldsymbol{b_{2k+1}}}, \ldots, g_2^{\boldsymbol{b_n^*}}, g_1^{\boldsymbol{b_1}}, \ldots, g_1^{\boldsymbol{b_n}}, U_1, \ldots, U_k, \mu_2 \rangle$$

A polynomial-time algorithm has an advantage ϵ in solving the subspace problem, if:

$$\left| Pr[\mathcal{B}(\boldsymbol{D}, T = V_1, \ldots, V_k) = 1] - Pr[\mathcal{B}(\boldsymbol{D}, T = W_1, \ldots, W_k) = 1] \right| \geq \epsilon$$

We henceforth refer to $DS1$ and $DS2$ as the decisional subspace *assumptions in $\mathbb{G}_1$ and $\mathbb{G}_2$.*

III. System Definition

We first describe the algorithms for a revocable CP-ABE scheme with outsourced decryption and hidden-policies. Next, we define the security model and present our construction.

A. Algorithms

A revocable CP-ABE scheme with outsourced decryption and hidden policies consists of seven *Probabilistic Polynomial-Time* (PPT) algorithms:

Setup$(1^\lambda, \Sigma) \rightarrow MPK, MSK, \mathcal{K}$. The *Setup* algorithm takes as input a security parameter λ and the attribute space Σ of attributes. It generates the master public key MPK, the master secret key MSK and the set of attribute group keys $\mathcal{K}$.

KeyGen$(MSK, \mathcal{S}) \rightarrow SK_\mathcal{S}$. The *KeyGen* algorithm consumes the master secret key MSK and a set of attributes $\mathcal{S}$. It outputs the secret key $SK_\mathcal{S}$ corresponding to $\mathcal{S}$.

Encrypt$(M, \Pi, MPK) \rightarrow \overline{CT}$. The *Encrypt* algorithm takes as input a message M, a LSSS $\Pi = (\mathbb{A}, \rho)$ and the master public key MPK. It outputs the initial ciphertext $\overline{CT}$.

ReEncrypt$(\overline{CT}, MPK, \mathcal{K}) \rightarrow CT$. *ReEncrypt* consumes the initial ciphertext $\overline{CT}$, the master public key MPK and the attribute group keys $\mathcal{K}$. It incorporates the attribute group keys into the ciphertext and adds random group elements to hide the ciphertext policy. It outputs a ciphertext CT so that only users who are members of a set of attribute groups satisfying the access structure will be able to decrypt the message.

GenTK$(MPK, SK_\mathcal{S}, \overline{CT}, \mathcal{K}_I) \rightarrow TK, RK_\mathcal{S}$. The algorithm takes the master public key MPK, the secret key $SK_\mathcal{S}$, the initial ciphertext $\overline{CT}$ and a (sub-) set of attribute group keys $\mathcal{K}_I$ as argument. It returns a ciphertext specific transformation key TK, where the embedded attribute information is hidden, and a retrieving number $RK_\mathcal{S}$.

OutsourcedDecrypt$(MPK, TK, CT) \rightarrow CT'$. The *Outsourced-Decrypt* algorithm consumes the master public key MPK, a transformation key TK and the final ciphertext CT. The algorithm performs all expensive bilinear operations and returns the partially decrypted ciphertext CT'.

Decrypt$(MPK, CT, CT', RK_\mathcal{S}) \rightarrow M$. The *Decrypt* algorithm consumes the master public key MPK, the initial ciphertext CT, the transformed ciphertext CT' and the retrieving number $RK_\mathcal{S}$. It returns the plaintext M.

B. Security Model

We formally define the security requirements of a revocable CP-ABE scheme with outsourced decryption and hidden policies. Informally, *security* means that an adversary does not learn anything about the plaintext. *Hidden policies* means that the proxy, performing outsourced decryption, does not learn the access policy.

The *security* of the system is described by the following game:

Setup. The challenger runs the ***Setup*** algorithm and generates MPK and MSK. The challenger sends the MPK to the adversary.

Phase 1. The adversary can query the challenger for keys for attribute sets $\mathcal{S}_1, \ldots \mathcal{S}_{q1}$.

Challenge. The adversary generates two equal length messages M_0 and M_1 and two equal length access structures $\mathbb{A}_0^*$, $\mathbb{A}_1^*$. The access structures may not be satisfied by any previously queried attribute set $\mathcal{S}_i$. The challenger flips a coin $\beta \in \{0, 1\}$ and encrypts M_β under the challenge access structure $\mathbb{A}_\beta^*$. The challenger returns the ciphertext CT to the adversary.

Phase 2. The adversary runs key and decryption queries. The queries keys, attribute sets $\mathcal{S}_{q1+1}, \ldots, \mathcal{S}_q$, may not satisfy $\mathbb{A}^*$, respectively.

Guess. The adversary outputs a guess β' for β.

The winning advantage of an adversary in this game is defined to be $Pr[\beta = \beta'] - \frac{1}{2}$.

Definition 3. *The ABE scheme is* Chosen-Plaintext Attack *(CPA) secure and provides hidden policies if the advantage over all PPT algorithms in the aforementioned game is negligible.*

C. Scheme Construction

We provide the construction of our scheme, which is partially based on the construction presented in [14]. Our proposed scheme, however, relies on Type-3 pairings and adapts DPVS to deal with performance issues.

Setup$(1^\lambda, \Sigma) \rightarrow MPK, MSK$. The algorithm performs the following steps:

1. Let $G = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$
2. Sample dual orthonormal bases $(\mathbb{D}, \mathbb{D}^*) \xleftarrow{R} Dual(\mathbb{Z}_p^4)$
3. Let $\boldsymbol{d_1}, \ldots, \boldsymbol{d_4}$ denote elements of $\mathbb{D}$
4. Let $\boldsymbol{d_1^*}, \ldots, \boldsymbol{d_4^*}$ denote elements of $\mathbb{D}^*$
5. Choose $g_1, f_1, \ldots, f_U \xleftarrow{R} \mathbb{G}_1$, $g_2 \xleftarrow{R} \mathbb{G}_2$
6. Choose $a, \alpha \xleftarrow{R} \mathbb{Z}_p^*$.
7. Choose group keys $\kappa_{\lambda_1} \ldots \kappa_{\lambda_U} \xleftarrow{R} \mathbb{Z}_p^*$ for each $U_i \in \mathcal{U}$
8. Output the master public key MPK, the master secret key MSK and the group keys $\mathcal{K}$:

$$MPK : \langle G, g_1^{\boldsymbol{d_1}}, g_2^{\boldsymbol{d_1^*}}, g_2^{\boldsymbol{d_2^*}}, f_1^{\boldsymbol{d_1}}, \ldots, f_U^{\boldsymbol{d_1}}, g_1^{a\boldsymbol{d_1}}, y = e(g_1, g_2)^{\alpha \boldsymbol{d_1} \cdot \boldsymbol{d_1^*}}, y_1 = e(g_1, g_2)^{\boldsymbol{d_2} \cdot \boldsymbol{d_2^*}}), \mathbb{H} \rangle$$
$$MSK = g_1^{\alpha \boldsymbol{d_1}}, \mathcal{K} = \{\kappa_{\lambda_i}\}$$

KeyGen$(MSK, \mathcal{S}) \rightarrow SK_\mathcal{S}$. The algorithm chooses $t \xleftarrow{R} \mathbb{Z}_p^*$. The algorithm outputs:

$$SK_\mathcal{S} = \langle K_1, K_2, \{K_x\}_{x \in \mathcal{S}} \rangle = \langle g_1^{\alpha \boldsymbol{d_1}} \cdot g_1^{at \boldsymbol{d_1}}, g_2^{t \boldsymbol{d_1^*}}, \{f_x^{t \boldsymbol{d_1}}\}_{x_\epsilon \mathcal{S}} \rangle$$

Encrypt$(M, \Pi, MPK) \rightarrow \overline{CT}$. First, the algorithm generates a secret $s \xleftarrow{R} \mathbb{Z}_p^*$. Next, it calculates a share vector $\vec{v} = (s, v_2, \ldots v_n) \in \mathbb{Z}_p^*$. Finally, it computes $\lambda_i = \vec{v} \cdot \mathbb{A}_i$ and generates $r_1, \ldots, r_l \xleftarrow{R} \mathbb{Z}_p^*$. It outputs:

$$\overline{CT} = \langle C = M \cdot y^s, \overline{C'} = g_2^{s \boldsymbol{d_1^*}}, \{\overline{C_i} = g_1^{a \lambda_i \boldsymbol{d_1}} \cdot f_{\rho(i)}^{-r_i \boldsymbol{d_1}}, \overline{D_i} = g_2^{r_i \boldsymbol{d_1^*}}\} \rangle$$

ReEncrypt$(\overline{CT}, MPK, \mathcal{K}) \rightarrow CT$. The algorithm first chooses $\eta_1, \ldots \eta_l, \delta \xleftarrow{R} \mathbb{Z}_p^*$. It outputs the final ciphertext:

$$CT = \langle C = M \cdot y^s, \tilde{C} = g_1^{\mathbb{H}(M)}, C' = g_2^{s \boldsymbol{d_1^*}} g_2^{\delta \boldsymbol{d_2^*}}, \{C_i = g_1^{a \lambda_i \boldsymbol{d_1}} \cdot f_{\rho(i)}^{-r_i \boldsymbol{d_1}} \cdot g_1^{(\eta_i + \kappa_{t_i}) \boldsymbol{d_2}}, \overline{D_i} = g_2^{r_i \boldsymbol{d_1^*}}\} \rangle$$

GenTK$(MPK, SK_\mathcal{S}, \overline{CT}, \mathcal{K}_I) \rightarrow TK, RK_\mathcal{S}$. First, the algorithm computes a set $\{\omega_i \in \mathbb{Z}_p^*\}$, such that $\sum_{i \in I} \omega_i \lambda_i = s$, where $I = \{i : \rho(i) \in \mathcal{S}\}$. Next, it chooses $z, d \xleftarrow{R} \mathbb{Z}_p^*$ and calculates:

$$TK = \langle K_1' = K_1 \cdot g_1^{z\boldsymbol{d_2}}, K_2' = K_2 \cdot g_2^{d\boldsymbol{d_2^*}}, \{(K_\rho(i), \omega_i)\}_{i\in I}, I\rangle$$
$$RK_{\mathcal{S}} = z\delta - d\sum_{i\in I}\eta_i \sum_{i\in I}\kappa_{t_i}$$

OutsourcedDecrypt*(MPK, TK, CT)* $\rightarrow CT'$. The algorithm computes:

$$T = \frac{e_n(K_1', C')}{\prod_{i\in I}(e_n(C_i, K_2') \cdot e_n(K_\rho(i), D_i))^{\omega_i}}$$
$$= e(g_1, g_2)^{\alpha s \boldsymbol{d_1}\cdot\boldsymbol{d_1^*}} \cdot e(g_1, g_2)^{\boldsymbol{d_2}\cdot\boldsymbol{d_2^*}(z\delta - d\cdot\sum_{i\in I}\tau_i \sum_{i\in I}\kappa_{t_i})}$$

Finally the algorithm returns CT':

$$CT' = \langle C', \tilde{C}', T\rangle$$

Decrypt*(MPK, CT, CT', RK*$_{\mathcal{S}}$*)* $\rightarrow M$. First, the algorithm checks if $C = C'$ and $\tilde{C} = \tilde{C}'$. Next, it computes:

$$M = \frac{C \cdot y_1^{RK_S}}{T}$$

If $g_1^{\mathbb{H}(M)} = \tilde{C}'$ the algorithm returns M.

Decryption Correctness:

$$\frac{C \cdot y_1^{RK_S}}{T}$$
$$= \frac{M \cdot y^s \cdot e(g_1, g_2)^{\boldsymbol{d_2}\cdot\boldsymbol{d_2^*}(z\delta - d\cdot\sum_{i\in I}\eta_i\sum_{i\in I}\kappa_{t_i})}}{e(g_1, g_2)^{\alpha s\boldsymbol{d_1}\cdot\boldsymbol{d_1^*}} \cdot e(g_1, g_2)^{\boldsymbol{d_2}\cdot\boldsymbol{d_2^*}(z\delta - d\cdot\sum_{i\in I}\eta_i\sum_{i\in I}\kappa_{t_i})}}$$
$$= \frac{M \cdot e(g_1, g_2)^{\alpha s\boldsymbol{d_1}\cdot\boldsymbol{d_1^*}}}{e(g_1, g_2)^{\alpha s\boldsymbol{d_1}\cdot\boldsymbol{d_1^*}}}$$
$$= M \qquad \square$$

D. Key Update

The key update procedure needs to be initiated whenever a user obtains or loses an attribute. First, the user is removed or added to the affected attribute group. A new attribute group key κ_λ, which is different from the previous attribute group key, is generated. Then, all affected ciphertexts CT are reencrypted using the *ReEncrypt* algorithm. The new attribute group key needs to be delivered to all valid users through a secure communication channel. When a user requests a ciphertext CT afterwards, the newly encrypted ciphertext is sent. This key update procedure ensures that (1) user revocation can be achieved on attribute level and (2) users can still access data with other attributes they hold if they meet the access policy.

IV. Security-Proof

Theorem 1. *If the SXDH assumption holds, the presented CP-ABE scheme is CPA secure. For any adversary $\mathcal{A}$, there exist probabilistic algorithms $\mathcal{B}_0, \mathcal{B}_1, \ldots, \mathcal{B}_v$, which running times are the same as those of $\mathcal{A}$, such that, given the maximum number of key queries v, the following holds:*

$$Adv_{\mathcal{A}}^{ABE}(\lambda) \leq, Adv_{\mathcal{B}_0}^{DDH2}(\lambda) + \sum_{i=1}^{v} Adv_{\mathcal{B}_i}^{DDH1}(\lambda) + \frac{v}{q}$$

We follow the approach by [11]. We rely on *semi-functional ciphertexts* and *semi-functional keys* and provide algorithms to generate them. The provided algorithms are not part of the final system. They do not need to be efficiently computable from MPK or MSK.

KeyGenSF. The algorithm selects $t, z_3, z_4, z, d \xleftarrow{R} \mathbb{Z}_p^*$ and generates a semi-functional secret key as:

$$SK_{\mathcal{S}}^{(SF)} = \langle K_1, K_2, \{K_x\}_{x\in\mathcal{S}}\rangle$$
$$= \langle g_1^{\alpha\boldsymbol{d_1}} \cdot g_1^{at\boldsymbol{d_1}} \cdot g_1^{z_3\boldsymbol{d_3}+z_4\boldsymbol{d_4}}, g_2^{t\boldsymbol{d_1^*}}, \{f_x^{t\boldsymbol{d_1}}\}_{x_\epsilon\mathcal{S}}\rangle$$
$$TK^{(SF)} = \langle K_1' = K_1 \cdot g_1^{z\boldsymbol{d_2}}, K_2' = K_2 \cdot g_2^{d\boldsymbol{d_2^*}}, \{(K_\rho(i), \omega_i)\}_{i\in I}, I\rangle$$
$$RK_{\mathcal{S}} = z\delta - d\sum_{i\in I}\eta_i \sum_{i\in I}\kappa_{t_i}$$

EncryptSF. The algorithm selects $s, t_3, t_4, \eta_1, \ldots, \eta_l, \delta \xleftarrow{R} \mathbb{Z}_p^*$ and encrypts a message M as:

$$CT^{(SF)} = \langle C = M \cdot y^s, C' = g_2^{s\boldsymbol{d_1^*}} \cdot g_2^{\delta\boldsymbol{d_2^*}} \cdot g_2^{t_3\boldsymbol{d_3^*}+t_4\boldsymbol{d_4^*}},$$
$$C_i = g_1^{a\lambda_i\boldsymbol{d_1}} \cdot f_{\rho(i)}^{-r_i\boldsymbol{d_1}} \cdot g_1^{(\eta_i+\kappa_i)\boldsymbol{d_2}}, D_i = g_2^{r_i\boldsymbol{d_1^*}}\}\rangle$$

Decrypting a normal ciphertext with a semi-functional key will succeed because $\boldsymbol{d_3}, \boldsymbol{d_4}$ are orthogonal to all remaining vectors in C'. Likewise, decrypting a semi-functional ciphertext will succeed for a normal key because $\boldsymbol{d_3^*}, \boldsymbol{d_4^*}$ are orthogonal to all the remaining exponents in K_1. If the ciphertext and the key are semi-functional, decryption will fail. The pairing $e_n(K_1', C')$ will then contain an additional term:

$$e(g_1, g_2)^{t_3z_3\boldsymbol{d_3}\cdot\boldsymbol{d_3^*}+t_4z_4\boldsymbol{d_4}\cdot\boldsymbol{d_4^*}} = e(g_1, g_2)^{(t_3z_3+t_4z_4)\psi}$$

We define the following games between any polynomial-time adversary $\mathcal{A}$ and a challenger C:

- $Game_{Real}$: is the real security game
- $Game_i$ for $i = 0, 1 \ldots v$: $Game_i$ is like $Game_{Real}$, except that the challenge ciphertext is semi-functional and the first i keys given to the attacker are semi-functional. For $Game_0$ all keys are normal and for $Game_v$ all keys are semi-functional.
- $Game_{Final}$: Is like $Game_v$, except that the ciphertext is a semi-functional encryption of a random message M in $\mathbb{G}_T$.

We prove the following lemmas to show that for each transition from $Game_{Real}$ to $Game_v$ the attacker's advantage cannot change by a non-negligible amount.

Lemma 1. *Suppose that there exists an adversary $\mathcal{A}$. If the adversary has an advantage $|Adv_{\mathcal{A}}^{Game_{Real}}(\lambda) - Adv_{\mathcal{A}}^{Game_0}(\lambda)| = \epsilon$, there exists an algorithm $\mathcal{B}_0$ such that $Adv_{\mathcal{B}_0}^{DS2}(\lambda) = \epsilon$ with $k = 2$ and $n = 4$.*

Proof. $\mathcal{B}_0$ is given a distribution $\boldsymbol{D}$ and T_1, T_2, as defined in Section II-E:

$$\boldsymbol{D} = \langle G, g_1^{\boldsymbol{b_1}}, g_1^{\boldsymbol{b_2}}, g_2^{\boldsymbol{b_1^*}}, \ldots, g_2^{\boldsymbol{b_4^*}}, U_1, U_2, \mu_2\rangle$$

$\mathcal{B}_0$ needs to decide whether T_1, T_2 are distributed as $g_2^{\tau_1\boldsymbol{b_1^*}}, g_2^{\tau_1\boldsymbol{b_2^*}}$ or $g_2^{\tau_1\boldsymbol{b_1^*}+\tau_2\boldsymbol{b_3^*}}, g_2^{\tau_1\boldsymbol{b_2^*}+\tau_2\boldsymbol{b_4^*}}$. $\mathcal{B}_0$ starts by simulating $Game_{Real}$ or $Game_0$ with $\mathcal{A}$, depending on the distribution of T_1, T_2. Given a security parameter λ, $\mathcal{B}_0$ first generates a random invertible matrix $A \in \mathbb{Z}_p^{2x2}$. It sets the dual orthonormal bases $\mathbb{D}, \mathbb{D}^*$ to:

$$\boldsymbol{d_1} \coloneqq \boldsymbol{b_1}, \quad \boldsymbol{d_2} \coloneqq \boldsymbol{b_2}, \quad (\boldsymbol{d_3}, \boldsymbol{d_4}) \coloneqq (\boldsymbol{b_3}, \boldsymbol{b_4})A,$$
$$\boldsymbol{d_1^*} \coloneqq \boldsymbol{b_1^*}, \quad \boldsymbol{d_2^*} \coloneqq \boldsymbol{b_2^*}, \quad (\boldsymbol{d_3^*}, \boldsymbol{d_4^*}) \coloneqq (\boldsymbol{b_3^*}, \boldsymbol{b_4^*})(A^{-1})^t$$

Now, $\mathcal{B}_0$ chooses random values $a, \alpha \xleftarrow{R} \mathbb{Z}_p^*$ and computes the MPK and MSK according to the ***Setup*** algorithm. Since MSK is known to $\mathcal{B}_0$, it can respond to key queries for a set $\mathcal{S}$ by calling the normal key generation algorithm. We can now build the challenge ciphertext. First, the adversary $\mathcal{A}$ defines the challenge access structures $\mathbb{A}_0$, $\mathbb{A}_1$. It then generates two messages M_0 and M_1 and sends them to $\mathcal{B}_0$. $\mathcal{B}_0$ now flips a coin β. $\mathcal{B}_0$ encrypts M_β under the challenge access structure $\mathbb{A}_\beta$ and implicitly sets $s \coloneqq \tau_1$:

$$CT^{(SF)} = \langle C = M \cdot y^s, C' = g_2^{s\boldsymbol{f_1^*}} \cdot g_2^{\delta \boldsymbol{f_2^*}} \cdot g_2^{t_3 \boldsymbol{d_3^*} + t_4 \boldsymbol{d_4^*}},$$
$$C_i = g_1^{a\lambda_i \boldsymbol{f_1}} \cdot f_{\rho(i)}^{-r_i \boldsymbol{f_1}} \cdot g_1^{(\eta_i + \kappa_i)\boldsymbol{f_2}}, D_i = g_2^{r_i \boldsymbol{f_1^*}} \}\rangle$$

Finally, it returns the ciphertext $\overline{CT}$. If T_1, T_2 are equal to $g_2^{\tau_1 \boldsymbol{b_1^*}}$, $g_2^{\tau_1 \boldsymbol{b_2^*}}$ then $\mathcal{B}_0$ has successfully simulated $Game_{Real}$. However, if T_1,T_2 are equal to $g_2^{\tau_1 \boldsymbol{b_1^*} + \tau_2 \boldsymbol{b_3^*}}, g_2^{\tau_1 \boldsymbol{b_2^*} + \tau_2 \boldsymbol{b_4^*}}$, the ciphertext element $\overline{C'}$ has an additional term $\tau_2 \boldsymbol{b_3^*} + \tau_2 \boldsymbol{b_4^*}$. $\mathcal{B}_0$ can obtain the coefficients in the basis $\boldsymbol{d_3^*}, \boldsymbol{d_4^*}$ by multiplying the matrix A^{-1} with the transpose of the vector. Since A is uniformly random, the coefficients are uniformly random and $\mathcal{B}_0$ has properly simulated $Game_0$. As a result, $\mathcal{B}_0$ can leverage the adversaries' advantage ϵ between $Game_{Real}$ and $Game_0$ against the subspace assumption in $\mathbb{G}_2$: $Adv_{\mathcal{B}_0}^{DS2}(\lambda) = \epsilon$. □

Lemma 2. *Suppose that there exists an adversary $\mathcal{A}$. If the adversary has an advantage $|Adv_{\mathcal{A}}^{Game_{i-1}}(\lambda) - Adv_{\mathcal{A}}^{Game_i}(\lambda)| = \epsilon$. Then there exists an algorithm $\mathcal{B}_i$ such that $Adv_{\mathcal{B}_i}^{DS1}(\lambda) = \epsilon - 1/q$ with $k = 2$ and $n = 4$.*

Proof. $\mathcal{B}_i$ is given a distribution $\boldsymbol{D}$, as defined in Section II-E and and T_1, T_2:

$$\boldsymbol{D} = \langle G, g_2^{\boldsymbol{b_1}*}, g_2^{\boldsymbol{b_2}*}, g_1^{\boldsymbol{b_1}}, \ldots, g_1^{\boldsymbol{b_4}}, U_1, U_2, \mu_2 \rangle$$

$\mathcal{B}_i$ needs to decide whether T_1, T_2 are distributed as $g_1^{\tau_1 \boldsymbol{b_1}}, g_1^{\tau_1 \boldsymbol{b_2}}$ or $g_1^{\tau_1 \boldsymbol{b_1} + \tau_2 \boldsymbol{b_3}}, g_1^{\tau_1 \boldsymbol{b_2} + \tau_2 \boldsymbol{b_4}}$. It begins by simulating $Game_i$ or $Game_{i-1}$ with $\mathcal{A}$, depending on the distribution of T_1, T_2. Given a security parameter λ, $\mathcal{B}_i$ first generates a random invertible matrix $A \in \mathbb{Z}_q^{2x2}$. Then, $\mathcal{B}_i$ sets the dual orthonormal bases $\mathbb{D}, \mathbb{D}^*$ to:

$$\boldsymbol{d_1} \coloneqq \boldsymbol{b_1}, \quad \boldsymbol{d_2} \coloneqq \boldsymbol{b_2}, \quad (\boldsymbol{d_3}, \boldsymbol{d_4}) \coloneqq (\boldsymbol{b_3}, \boldsymbol{b_4})A,$$
$$\boldsymbol{d_1^*} \coloneqq \boldsymbol{b_1^*}, \quad \boldsymbol{d_2^*} \coloneqq \boldsymbol{b_2^*}, \quad (\boldsymbol{d_3^*}, \boldsymbol{d_4^*}) \coloneqq (\boldsymbol{b_3^*}, \boldsymbol{b_4^*})(A^{-1})^t$$

Now, $\mathcal{B}_0$ chooses random values $a, \alpha \xleftarrow{R} \mathbb{Z}_p^*$ and computes the parameters according to the ***Setup*** algorithm. Since MSK is known to $\mathcal{B}_i$, it can respond to key queries for a set $\mathcal{S}$ by calling the normal key generation algorithm. Furthermore, $\mathcal{B}_i$ knows $g_1^{\boldsymbol{d_3}}$ and $g_1^{\boldsymbol{d_4}}$. It can, therefore, easily generate semi-functional keys. For the first $i-1$ key queries, $\mathcal{B}_i$ runs the semi-functional key generation algorithm and returns them to $\mathcal{A}$. For the ith key query for a set of attributes $\mathcal{S}$, $\mathcal{B}_i$ responds with:

$$SK_{\mathcal{S}} = \langle K_1, K_2, \{K_x\}_{x \in \mathcal{S}} \rangle \quad = \langle (g_1^{\boldsymbol{b_1}})^\alpha \cdot (T_1)^t, g_2^{t \boldsymbol{b_1^*}}, \{f_x^{t \boldsymbol{b_1}}\}_{x \in \mathcal{S}} \rangle$$
$$TK_{CT} = \langle K_1' = K_1 \cdot T_2, K_2' = K_2 \cdot (g_2^{\boldsymbol{b_2^*}})^d, \{(K_\rho(i), \omega_i)\}_{i \in I}, I \rangle$$

$\mathcal{B}_0$ implicitly sets $a \coloneqq \tau_1$. If T_1, T_2 are equal to $g_1^{\tau_1 \boldsymbol{b_1}}, g_1^{\tau_1 \boldsymbol{b_2}}$, then the key is properly distributed. However, if T_1,T_2 are equal to $g_1^{\tau_1 \boldsymbol{b_1} + \tau_2 \boldsymbol{b_3}}, g_1^{\tau_1 \boldsymbol{b_2} + \tau_2 \boldsymbol{b_4}}$, the key is semi-functional. The exponent vector then includes an additional term $t\tau_2 \boldsymbol{b_3} + \tau_2 \boldsymbol{b_4}$. For the remaining key queries, $\mathcal{B}_k$ simply runs the normal key generation algorithms.

Next, the adversary $\mathcal{A}$ defines the challenge access structures $\mathbb{A}_0$, $\mathbb{A}_1$. It then generates two messages M_0 and M_1 and sends them to $\mathcal{B}_0$. $\mathcal{B}_1$ now flips a coin β. $\mathcal{B}_1$ implicitly sets $s \coloneqq \tau_1$. It encrypts M_β under the challenge access structure $\mathbb{A}_\beta^*$ as follows:

$$CT = \langle C = M_\beta \cdot \left(e_n(g_1^{\boldsymbol{b_1}}, U_1)\right)^\alpha = M_\beta \cdot \left(e_n(g_1, g_2)^{\alpha \boldsymbol{d_1} \boldsymbol{d_1^*}}\right)^s,$$
$$C' = U_1 \cdot U_2, \{C_i = g_1^{a\lambda_i \boldsymbol{b_1}} \cdot f_{\rho(i)}^{-r_i \boldsymbol{b_1}} \cdot g_1^{(\eta_i + \kappa_{t_i})\boldsymbol{b_2}}, D_i = g_2^{r_i \boldsymbol{b_1^*}}\}\rangle$$

where $\mathcal{B}_1$ implicitly sets $s \coloneqq \mu_1$. The semi-functional part of the ciphertext now contains $\mu_2 \boldsymbol{b_3^*} + \mu_2 \boldsymbol{b_4^*}$. The distribution for the $i-1$ first keys is independent of the random matrix A. Likewise, the challenge ciphertext is independent of the random matrix A. The coefficients are uniformly random (except for $1/q$ probability from [15]). Summarising, $\mathcal{B}_i$ can properly simulate either $Game_{i-1}$ or $Game_i$, depending on the distribution of T_1, T_2. As a result, $\mathcal{B}_0$ can leverage the adversaries' advantage ϵ between $Game_{i-1}$ and $Game_i$ against the subspace assumption in $\mathbb{G}_1$: $Adv_{\mathcal{B}_i}^{DS1}(\lambda) = \epsilon - 1/q$. □

Lemma 3. *For any adversary $\mathcal{A}$, $Adv_{\mathcal{A}}^{Game_i}(\lambda) = Adv_{\mathcal{A}}^{Game_{Final}}(\lambda)$.*

Proof. To prove this lemma, we show that an adversary $\mathcal{A}$ cannot distinguish between the joint distributions of $Game_\nu$: $(MPK, CT^{(SF)}, \{SK_{\mathcal{S}_l}^{(SF)}\}_{l=1,\ldots,i}, \{TK_l^{(SF)}\}_{l=1,\ldots,i})$ and $Game_{Final}$: $(MPK, CT^{(R)}, \quad \{SK_{\mathcal{S}_l}^{(SF)}\}_{l=1,\ldots i}, \{TK_l^{(SF)}\}_{l=1,\ldots i})$, where $CT^{(R)}$ is a semi-functional encryption of a random message in $\mathbb{G}_T$ under a random access structure. We again define a matrix A. This time, we set $A \coloneqq (\xi_{i,j}) \xleftarrow{R} \mathbb{Z}_p^{2x2}$ and define new orthonormal bases $\mathbb{F} \coloneqq (\boldsymbol{f_1}, \ldots, \boldsymbol{f_4})$, and $\mathbb{F}^* \coloneqq (\boldsymbol{f_1^*}, \ldots, \boldsymbol{f_4^*})$:

$$\begin{pmatrix} \boldsymbol{f_1} \\ \boldsymbol{f_2} \\ \boldsymbol{f_3} \\ \boldsymbol{f_4} \end{pmatrix} \coloneqq \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \xi_{1,1} & \xi_{1,2} & 1 & 0 \\ \xi_{2,1} & \xi_{2,2} & 0 & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{d_1} \\ \boldsymbol{d_2} \\ \boldsymbol{d_3} \\ \boldsymbol{d_4} \end{pmatrix}, \begin{pmatrix} \boldsymbol{f_1^*} \\ \boldsymbol{f_2^*} \\ \boldsymbol{f_3^*} \\ \boldsymbol{f_4^*} \end{pmatrix} \coloneqq \begin{pmatrix} 1 & 0 & -\xi_{1,1} & -\xi_{2,1} \\ 0 & 1 & -\xi_{1,2} & -\xi_{2,2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{d_1^*} \\ \boldsymbol{d_2^*} \\ \boldsymbol{d_3^*} \\ \boldsymbol{d_4^*} \end{pmatrix},$$

Next, we express our public parameters, the challenge ciphertext and the secret keys in $Game_\nu$ with the bases of $\mathbb{F}, \mathbb{F}^*$. We change the coefficients (s, δ) in the ciphertext term to random coefficients $(s', s'') \in \mathbb{Z}_p \times \mathbb{Z}_p$ of $\boldsymbol{f_1^*}, \boldsymbol{f_2^*}$. We let:

$$SK_{\mathcal{S}_l}^{(SF)} = \langle K_1, K_2, \{K_x\}_{x \in \mathcal{S}} \rangle$$
$$= \langle g_1^{\alpha \boldsymbol{f_1}} \cdot g_1^{a t_l \boldsymbol{f_1}} \cdot g_1^{z'_{l,3} \boldsymbol{f_3} + z'_{l,4} \boldsymbol{f_4}}, g_2^{t_l \boldsymbol{f_1^*}}, \{f_x^{t_l \boldsymbol{f_1}}\}_{x_\epsilon \mathcal{S}} \rangle$$
$$TK_{CT}^{(SF)} = \langle K_1' = K_1 \cdot g_1^{z_l \boldsymbol{f_2}}, K_2' = K_2 \cdot g_2^{d_l \boldsymbol{f_2^*}}, \{(K_\rho(i), \omega_i)\}_{i \in I}, I \rangle$$
$$RK_{\mathcal{S}} = z_l \delta - d \sum_{i \in I} \eta_i \sum_{i \in I} \kappa_{t_i}$$
$$CT^{(SF)} = \langle C = M \cdot y^s, C' = g_2^{s' \boldsymbol{f_1^*}} \cdot g_2^{s'' \boldsymbol{f_2^*}} \cdot g_2^{t'_3 \boldsymbol{f_3^*} + t'_4 \boldsymbol{f_4^*}},$$
$$C_i = g_1^{a\lambda_i \boldsymbol{f_1}} \cdot f_{\rho(i)}^{-r_i \boldsymbol{f_1}} \cdot g_1^{(\eta_i + \kappa_i)\boldsymbol{f_2}}, D_i = g_2^{r_i \boldsymbol{f_1^*}} \}\rangle$$

We set $s' = s - t_3\xi_{1,1} - t_4\xi_{2,1}$, $s'' = \delta - t_3\xi_{1,2} - t_4\xi_{2,2}$, $\{z'_{l,3} = z_{l,3} + \xi_{1,1}(\alpha + at_l) + z_l\xi_{1,2}$, $z'_{l,4} = z_{l,4} + \xi_{2,1}(\alpha + at_l) + z_l\xi_{2,2}$, $r_1, \ldots, r_l, \lambda_1, \ldots, \lambda_l \xleftarrow{R} \mathbb{Z}_p^* \}_{l=1,\ldots,v}$, . The values $\xi_{1,1}, \ldots, \xi_{2,2}$, $t_{1,3}, \ldots, t_{v,3}$, $t_{1,4}, \ldots, t_{v,4}, r_1, \ldots, r_l, \lambda_1, \ldots, \lambda_l$ are all uniformly distributed in $\mathbb{Z}_p^*$. Thus, the challenge ciphertext can be seen as a semi-functional encryption of a random element in $\mathbb{G}_T$ under a random access structure. From the adversary's perspective, both $(\mathbb{D}, \mathbb{D}^*)$ and $(\mathbb{F}, \mathbb{F}^*)$ are consistent with the same public key. We can therefore express the challenge ciphertext and the queried keys in two ways: In $Game_v$ over the bases $(\mathbb{D}, \mathbb{D}^*)$ and in $Game_{Final}$ over the bases $(\mathbb{F}, \mathbb{F}^*)$. Thus, the adversary cannot distinguish between $Game_v$ and $Game_{Final}$. □

Lemma 4. *For any adversary $\mathcal{A}$, $Adv_{\mathcal{A}}^{Game_{Final}}(\lambda) = 0$.*

Proof. The value of β is independent from the adversary's view in $Game_{Final}$. Hence, $Adv_{\mathcal{A}}^{Game_{Final}}(\lambda) = 0$. □

In $Game_{Final}$ the challenge ciphertext is a semi-functional encryption of a random element in $\mathbb{G}_T$ under a random access structure. It is independent of the provided messages and access structures. Hence, our CP-ABE scheme provides hidden policies.

V. Implementation and Evaluation

Comparison. We first compare the performance of our scheme with similar ABE schemes. Table I gives an overview of the overhead in terms of the number of pairing (P) and exponentiation (E) operations as well as a theoretical overview of the ciphertext size. $|CT|$ denotes the length of the (re-encrypted) ciphertext and $|CT'|$ the length of the ciphertext after outsourced decryption.

From the table we can see that our construction introduces an additional performance overhead. We attribute this to using DPVS. Our implementation requires a 2-dimensional vector space. Every vector pairing operation therefore effectively requires two pairing operations. However, we argue that in practice, the overhead is negligible. We will later see that our construction outperforms constructions based on composite order groups with a similar feature-set in practice by an order of magnitude.

Turning to communication and storage overhead, we see that our approach is comparable with existing solutions. Adding support for attribute groups does not influence the size of the elements. However, relying on DPVS, doubles the required elements. Still, as our construction relies on prime order groups, we can use lower security parameters. As a result, the actual size of the elements is smaller compared to constructions using composite order groups.

Implementation Details. We implemented the proposed architecture in Java and Kotlin. We relied on two providers for bilinear pairings: *Java Pairing-Based Cryptography Library* (JPBC) [16] and the *IAIK ECCelerate*^{TM}[1] library. This approach allows better estimates of the performance we can expect in practice. Furthermore, we implemented the schemes presented in [14, 17, 18]. The schemes offer similar features and give a good estimate of the performance of our construction.

Setting. We used the *Java Microbenchmark Harness* (JMH)[2] on an AMD EPYC 7502 32-Core Processor. For consistent results, we used 5 warmup forks with 5 warmup iterations. For the actual measurement, we used 10 forks with 10 iterations.

Policies. We generated random policies with up to 100 attributes.

Security. We chose a security parameter equivalent to a security level of AES-128. This level is equal to near-term security (at least ten years) as defined by NIST [19]. We used parameters as recommended by Guillevic [20]. For prime order groups the authors recommend to use the following group sizes: $\mathbb{G}_1 = 256$, $\mathbb{G}_2 = 512$, $\mathbb{G}_T = 3072$. We use an embedding degree $k = 12$. For symmetric pairings we set $r = 160$ and $q = 3000$ as recommended in [21]. For composite order groups with two primes, the authors recommend larger group orders: $\mathbb{G} = 2644$ and $\mathbb{G}_T = 5312$. We rely on the embedding degree $k = 2$.

Time Benchmarks. Figure 1 shows the execution time for all algorithms. The x-axis shows the number of attributes used and the logarithmically scaled y-axis shows the execution time in nanoseconds per operation. Looking at the graphs, it is noticeable that the scheme by Zhao et al. [17] exhibits a similar performance as our scheme but outperforms our construction in several cases. However, their scheme does not offer hidden policies. As a result, embedded attribute information might reveal sensitive information. When comparing the schemes, we can see that re-encryption attributes to a large amount of overhead of our construction. We attribute this to the fact that random information needs to be embedded in the ciphertext. In contrast, our construction clearly outperforms the original composite-order group based scheme by Wang and Liu [14] in all cases, as expected. Finally, our scheme outperforms the scheme presented by Zhang et al. [18] in all cases. The scheme offers hidden policies as ours but does not offer revocation. Hence, it does not require ReEncrypt our OutsourcedDecrypt algorithms. Summarising, these results showcase that our construction can compete with similar constructions while being more flexible.

VI. Related-Work

We summarise the major contributions of anonymous ABE and discuss their relevance to our work.

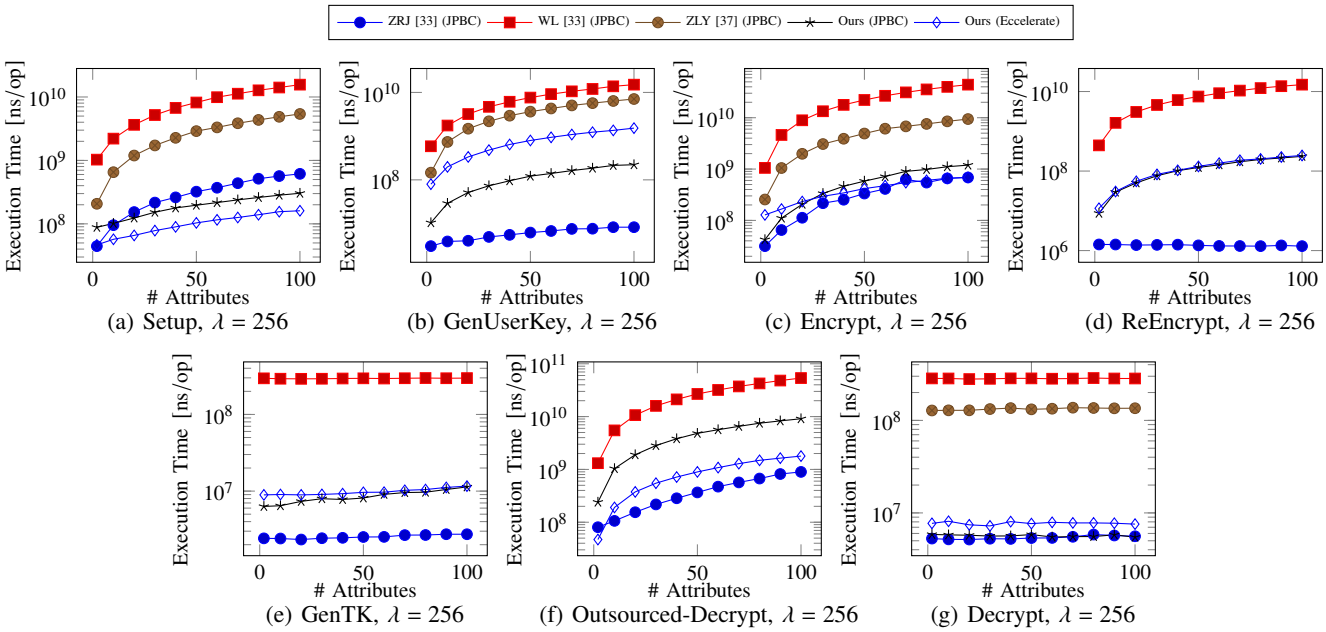
Anonymous Attribute-Based Encryption *(AABE).* In AABE schemes, attribute information in access policies is hidden or embedded in ciphertexts. The concept was first introduced by Kapadia et al. [23]. They presented a scheme in which recipients of ciphertexts cannot learn any information about a message's policy beyond the satisfied number of clauses. The scheme, however, is not resistant to collusion-attacks. Boneh and Waters [24] proposed an encryption scheme based on *Hidden Vector Encryption* (HVE). The scheme can be used for ABE schemes to provide hidden policies. In contrast to our work, the scheme relies on composite order groups and does not consider resource-constrained devices. Nishide et al. [25] presented an improved CP-ABE scheme with partially hidden ciphertext policies and collusion-resistance. The scope of the work does

[1] https://jce.iaik.tugraz.at

[2] https://openjdk.java.net/projects/code-tools/jmh/

TABLE I: Comparison of the proposed scheme with existing schemes. We denote P as the time for a pairing operation and E as the time for an exponentiation.

	GHW [4]	LDGW [22]	ZRJ [17]	WL [14]	ZLY [18]	Ours
$\|CT\|$	$(1+2l)\|\mathbb{G}\| + \|\mathbb{G}_T\|$	$(2+4l)\|\mathbb{G}\| + 2\|\mathbb{G}_T\|$	$\|\mathbb{G}_1\| + 2\|\mathbb{G}_2\|$	$(1+2l)\|\mathbb{G}\| + \|\mathbb{G}_T\|$	$2\|\mathbb{G}\| + 3\|\mathbb{G}_T\|$	$(1+l)\|\mathbb{G}_1\| + 4\|\mathbb{G}_2\|$
$\|CT'\|$	$2\|\mathbb{G}_T\|$	$2\|\mathbb{G}_T\|$	$\|\mathbb{G}_1\| + 2\|\mathbb{G}_2\|$	$4\|\mathbb{G}\| + \|\mathbb{G}_T\|$	-	$4\|\mathbb{G}_2\| + \|\mathbb{G}_T\|$
Encrypt	$1P + (2+3l)E$	$(4+2\cdot 3l)E$	$(4+2l)E$	$(2+3l)E$	$(3+2l)E$	$(4+6l)E$
Outsourced Decrypt	(2 + l)P + 2lE	$(2+4l)P + 2lE$	$3P + lE$	$(1+2l)P + lE$	-	$(1+2\cdot 2l)P + lE$
Decrypt	$1E$	$4E$	$4E$	$1E$	$5P + 1E$	$1E$
Hidden Policies	✗	✗	✗	✓	✓	✓
Revocation	✗	✗	✓	✗	✗	✓
Prime-Order Groups	✓	✓	✓	✗	✓	✓



(a) Setup, $\lambda = 256$ (b) GenUserKey, $\lambda = 256$ (c) Encrypt, $\lambda = 256$ (d) ReEncrypt, $\lambda = 256$

(e) GenTK, $\lambda = 256$ (f) Outsourced-Decrypt, $\lambda = 256$ (g) Decrypt, $\lambda = 256$

Fig. 1: Performance comparison of the proposed scheme with the schemes presented in [14, 17, 18]. The graphs show the timing results for different operations and different implementations. Tests were executed on an AMD EPYC 7502 32-Core Processor.

not include outsourced decryption or revocation like ours does. Chase and Chow [26] dealt with user-privacy in multi-authority ABE schemes. In such schemes, multiple attribute authorities issue different sets of attributes. In the presented scheme it is infeasible for two authorities to tell if they are interacting with the same user. Our scheme differs from their work as we focus on heterogeneous environments. Our main goal is to hide attribute information in access policies. Lai et al. [6] were the first to formally describe the construction of CP-ABE from attribute-hiding inner-product. Their approach is based on composite order groups. Tackling inefficiency of anonymous ABE schemes, the closest scheme to ours was presented by Wang and Liu [14]. They propose a CP-ABE scheme with hidden access policies. They rely on a dedicated proxy to perform most of expensive bilinear pairing operations. The proxy does not learn the policy or the plaintext, like in our work. We improve this work by introducing attribute revocation. Furthermore, our scheme relies on prime-order elliptic curves. Most recently, Zhang et al. [18] presented a CP-ABE scheme which constant sized ciphertexts and hidden-policies. The length of the ciphertext and the decryption overhead are constant. The resulting scheme is efficient and suitable for resource-constrained environments but does not discuss revocation strategies.

Anonymous Attribute-Based Proxy Re-encryption *(AABPRE).* Yu et al. [27] were one of the first to combine CP-ABE with proxy re-encryption [28] to deal with attribute revocation. In their proposed scheme, attribute authorities can revoke user attributes with minimal effort. A similar approach was presented by Luo et al. [29]. Both approaches do not consider resource-constrained environments or privacy of users. Kawai and Takashima [30] introduced a fully-anonymous inner-product proxy re-encryption scheme. Ciphertexts in the scheme do not reveal any attribute information. Like our scheme, the authors rely on DPVS. They do not, however, provide outsourced-decryption and do not discuss revocation strategies. Finally, Zhang et al. [7] introduced and first formalised the notion of anonymous *Ciphertext-Policy Attribute-Based Proxy Re-encryption* (CP-ABPRE). The authors promise fine-grained access control in cloud computing while

preserving user privacy. However, users need to reveal their attribute information to perform proxy re-encryption. Hence, the approach cannot guarantee anonymity of the data owner. The scheme by Chaudhari et al. [31] built upon this approach. It allows re-encrypting ciphertexts without learning the plaintext or access policy. The scheme does not focus on less powerful devices. Clients need to perform all necessary operations. In our scheme, clients are only left with a single constant operation. Summarising, we improve the state-of-the-art by tackling two open problems: First, providing an implementation of CP-ABE with hidden policies and attribute revocation and second, by providing an efficient (prime-order group) construction.

VII. Conclusion

We have presented a novel CP-ABE approach suitable for resource-constrained, heterogeneous environments. Our scheme improves the state-of-the-art by offering hidden policies, attribute revocation and outsourced decryption while relying on Type-3 pairings and bilinear groups of prime order. The timings of our software implementation clearly show the benefit of our approach. In fact, we achieve over 30 times faster operations in comparison to composite order group implementations. This demonstrates that our construction is also applicable in practice. For the future, we will concentrate on further improving the performance. Additionally, we seek for a holistic architecture, which integrates existing solutions well.

References

[1] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," in *Advances in Cryptology – EUROCRYPT 2005*, R. Cramer, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 457–473.

[2] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-grained Access Control of Encrypted Data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, ser. CCS '06. New York, NY, USA: ACM, 2006, pp. 89–98.

[3] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," in *2007 IEEE Symposium on Security and Privacy (SP '07)*, May 2007, pp. 321–334.

[4] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the Decryption of ABE Ciphertexts," in *Proceedings of the 20th USENIX Conference on Security*, ser. SEC'11. Berkeley, CA, USA: USENIX Association, 2011, p. 34.

[5] J. Hur and D. K. Noh, "Attribute-Based Access Control with Efficient Revocation in Data Outsourcing Systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 7, pp. 1214–1221, Jul. 2011.

[6] J. Lai, R. H. Deng, and Y. Li, "Fully Secure Cipertext-Policy Hiding CP-ABE," in *Information Security Practice and Experience*, F. Bao and J. Weng, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 24–39.

[7] Y. Zhang, J. Li, X. Chen, and H. Li, "Anonymous Attribute-based Proxy Re-encryption for Access Control in Cloud Computing," *Sec. and Commun. Netw.*, vol. 9, no. 14, pp. 2397–2411, Sep. 2016.

[8] R. Granger, T. Kleinjung, and J. Zumbrägel, "Breaking '128-bit Secure' Supersingular Binary Curves," in *Advances in Cryptology – CRYPTO 2014*, J. A. Garay and R. Gennaro, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 126–145.

[9] S. D. Galbraith, K. G. Paterson, and N. P. Smart, "Pairings for Cryptographers," *Discrete Applied Mathematics*, vol. 156, no. 16, pp. 3113–3121, 2008.

[10] S. Chatterjee, D. Hankerson, and A. Menezes, "On the Efficiency and Security of Pairing-Based Protocols in the Type 1 and Type 4 Settings," in *Arithmetic of Finite Fields*, M. A. Hasan and T. Helleseth, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 114–134.

[11] A. Lewko, "Tools for Simulating Features of Composite Order Bilinear Groups in the Prime Order Setting," in *Advances in Cryptology – EUROCRYPT 2012*, D. Pointcheval and T. Johansson, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 318–335.

[12] T. Okamoto and K. Takashima, "Homomorphic Encryption and Signatures from Vector Decomposition," in *Pairing-Based Cryptography – Pairing 2008*, S. D. Galbraith and K. G. Paterson, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 57–74.

[13] J. Chen, H. W. Lim, S. Ling, H. Wang, and H. Wee, "Shorter IBE and Signatures via Asymmetric Pairings," in *Pairing-Based Cryptography – Pairing 2012*, M. Abdalla and T. Lange, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 122–140.

[14] Z. Wang and W. Liu, "CP-ABE with Outsourced Decryption and Directionally Hidden Policy," *Sec. and Commun. Netw.*, vol. 9, no. 14, pp. 2387–2396, Sep. 2016.

[15] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption," in *Advances in Cryptology – EUROCRYPT 2010*, H. Gilbert, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 62–91.

[16] A. De Caro and V. Iovino, "jPBC: Java pairing based cryptography," in *Proceedings of the 16th IEEE Symposium on Computers and Communications, ISCC 2011*, Kerkyra, Corfu, Greece, June 28 - July 1, 2011, pp. 850–855.

[17] Y. Zhao, M. Ren, S. Jiang, G. Zhu, and H. Xiong, "An efficient and revocable storage CP-ABE scheme in the cloud computing," *Computing*, vol. 101, no. 8, pp. 1041–1065, 2019.

[18] Y. Zhang, J. Li, and H. Yan, "Constant Size Ciphertext Distributed CP-ABE Scheme With Privacy Protection and Fully Hiding Access Structure," *IEEE Access*, vol. 7, pp. 47 982–47 990, 2019.

[19] E. Barker, "Recommendation for Key Management Part 1: General," National Institute of Standards and Technology, Gaithersburg, MD, Tech. Rep., Jan. 2016.

[20] A. Guillevic, "Comparing the Pairing Efficiency over Composite-Order and Prime-Order Elliptic Curves," in *Applied Cryptography and Network Security*, M. Jacobson, M. Locasto, P. Mohassel, and R. Safavi-Naini, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 357–372.

[21] M. S. Kiraz and O. Uzunkol, "Still Wrong Use of Pairings in Cryptography," *CoRR*, vol. abs/1603.0, 2016.

[22] J. Lai, R. H. Deng, C. Guan, and J. Weng, "Attribute-Based Encryption With Verifiable Outsourced Decryption," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 8, pp. 1343–1354, Aug. 2013.

[23] A. Kapadia, P. P. Tsang, and S. W. Smith, "Attribute-Based Publishing with Hidden Credentials and Hidden Policies," in *Proceedings of the Network and Distributed System Security Symposium*, ser. NDSS, San Diego, California, USA, 2007, pp. 179–192.

[24] D. Boneh and B. Waters, "Conjunctive, Subset, and Range Queries on Encrypted Data," in *Theory of Cryptography*, S. P. Vadhan, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 535–554.

[25] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-Based Encryption with Partially Hidden Encryptor-Specified Access Structures," in *Applied Cryptography and Network Security*, S. M. Bellovin, R. Gennaro, A. Keromytis, and M. Yung, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 111–129.

[26] M. Chase and S. S. M. Chow, "Improving Privacy and Security in Multi-authority Attribute-based Encryption," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, ser. CCS '09. New York, NY, USA: ACM, 2009, pp. 121–130.

[27] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute Based Data Sharing with Attribute Revocation," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, ser. ASIACCS '10. New York, NY, USA: ACM, 2010, pp. 261–270.

[28] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *Advances in Cryptology — EUROCRYPT'98*, K. Nyberg, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 127–144.

[29] S. Luo, J. Hu, and Z. Chen, "Ciphertext Policy Attribute-Based Proxy Re-encryption," in *Information and Communications Security*, M. Soriano, S. Qing, and J. López, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 401–415.

[30] Y. Kawai and K. Takashima, "Fully-Anonymous Functional Proxy-Re-Encryption," *IACR Cryptology ePrint Archive*, vol. 2013, p. 318, 2013.

[31] P. Chaudhari, M. L. Das, and D. Dasgupta, "Privacy-Preserving Proxy Re-encryption with Fine-Grained Access Control," in *Information Systems Security*, R. K. Shyamasundar, V. Singh, and J. Vaidya, Eds. Cham: Springer International Publishing, 2017, pp. 88–103.