Cross-Domain Attribute-Based Access Control Encryption*

Mahdi Sedaghat and Bart Preneel

COSIC, KU Leuven and imec
ssedagha, bart.preneel@esat.kuleuven.be

Abstract. Logic access control enforces who can read and write data; the enforcement is typically performed by a fully trusted entity. At TCC 2016, Damgård et al. proposed Access Control Encryption (ACE) schemes where a predicate function decides whether or not users can read (decrypt) and write (encrypt) data, while the message secrecy and the users' anonymity are preserved against malicious parties. Subsequently, several ACE constructions with an arbitrary identity-based access policy have been proposed, but they have huge ciphertext and key sizes and/or rely on indistinguishability obfuscation. At IEEE S&P 2021, Wang and Chow proposed a Cross-Domain ACE scheme with constant-size ciphertext and arbitrary identity-based policy; the key generators are separated into two distinct parties, called Sender Authority and Receiver Authority. In this paper, we improve over their work with a novel construction that provides a more expressive access control policy based on attributes rather than on identities, the security of which relies on standard assumptions. Our generic construction combines Structure-Preserving Signatures, Non-Interactive Zero-Knowledge proofs, and Re-randomizable Ciphertext-Policy Attribute-Based Encryption schemes. Moreover, we propose an efficient scheme in which the sizes of ciphertexts and encryption and decryption keys are constant and thus independent of the number of receivers and their attributes. Our experiments demonstrate that not only is our system more flexible, but it also is more efficient and results in shorter decryption keys (reduced from about 100 to 47 bytes) and ciphertexts (reduced from about 1400 to 1047 bytes).

Keywords: Access Control Encryption, Ciphertext-Policy Attribute-Based Encryption, Structure-Preserving Signature, Non-Interactive Zero-Knowledge proofs.

* Extended version of Mahdi Sedaghat, Bart Preneel: Cross-Domain Attribute-Based Access Control Encryption. *Cryptology and Network Security (CANS 2021), Mauro Conti, Marc Stevens, Stephan Krenn, Eds, Lecture Notes in Computer Science 13099,* Springer 2021, pp. 3-23.

1 Introduction

Information Flow Control (IFC) systems enforce which parts of the communication amongst the users are allowed to pass over the network [32,30]. As introduced in the seminal work of Bell and LaPadula [7], restrictions have to be imposed on who can receive a message (enforce the No-Read rule) and who can send a message (enforce the No-Write rule). Although encryption guarantees users' privacy by limiting the set of recipients, we need more functionality to control who can write and transfer a ciphertext. Broadcasting of sensitive data by malicious senders is a serious threat for companies that handle highly sensitive data such as cryptocurrency wallets with access to signing keys [13]. Although some advanced cryptographical tools such as *Functional Encryptions* provide fine-grained access to encrypted data, they do not allow to enforce the No-Write rule, hence additional functionalities beyond these cryptographic primitives are required to protect against data leakage.

To achieve this aim, Damgård et al. [15] introduced a novel scheme called *Access Control Encryption (ACE)* to impose information flow control systems using cryptographic tools. They have defined two security notions for an ACE scheme: the No-Read rule and the No-Write rule. Unauthorized receivers cannot decrypt the ciphertext and unauthorized senders are not able to transmit data over the network. The model assumes that all the communications are transmitted through an honest-but-curious third party, called Sanitizer. The Sanitizer follows the protocol honestly but it is curious to find out more about the encrypted message and the identities of the users. The Sanitizer performs some operations on the received messages before transmitting them to the intended recipients without learning any information about the message itself or the identity of the users. More precisely, with a set of senders $\mathcal{S}$ and receivers $\mathcal{R}$, an ACE scheme determines via a hidden Boolean Predicate function $\textsc{Pf} : \mathcal{S} \times \mathcal{R} \to \{0, 1\}$ which group of senders (like $i \in \mathcal{S}$) are allowed to communicate with a certain group of receivers (like $j \in \mathcal{R}$): communication is allowed iff $\textsc{Pf}(i, j) = 1$, else the request will be rejected.

Damgård et al. proposed two ACE constructions that support arbitrary policies. Their first construction takes a brute-force approach that is based on standard number-theoretic assumptions, while the size of the ciphertext grows exponentially in the number of receivers. The second scheme is more efficient: ciphertext length is poly-logarithmic in the number of the receivers, but it relies on the strong assumption of *indistinguishability obfuscation (iO)* [20]. In a subsequent work, Fuchsbauer et

al. [19] proposed an ACE scheme for restricted classes of predicates including equality, comparisons, and interval membership. Although their scheme is secure under standard assumptions in groups with bilinear maps and asymptotically efficient (i.e., the length of the ciphertext is linear in the number of the receivers), the functionalities of their construction are restricted to a limited class of predicates. Tan et al. [37] proposed an ACE scheme based on the *Learning With Error* (LWE) assumption [31]. Since their construction follows the Damgård et al. approach, the ciphertexts in their construction also grow exponentially with the number of receivers. On the positive side, their construction is secure against post-quantum adversaries. Recently, Wang et al. [40], proposed an efficient LWE-based ACE construction from group encryptions. Kim and Wu [27] proposed a generic ACE construction based on standard assumptions such that the ciphertext shrinks to poly-logarithmic size in the number of receivers and with arbitrary policies. Their construction requires Digital Signature, Predicate Encryption, and Functional Encryption schemes to obtain an ACE construction based on standard assumptions. Recently, Wang and Chow [39] proposed a new notion called Cross-Domain ACE in which the keys are generated by two distinct entities, the Receiver-Authority and the Sender-Authority. Structure Preserving Signatures, Non-Interactive Zero-Knowledge proofs, and Sanitizable Identity-Based Encryption schemes constitute the main ingredients in their construction. In [39], the length of the ciphertext is constant, but it fails to preserve the identity of the receivers and also the decryption key size grows linearly.

Our Contributions: In this paper, we propose a generic *Cross-Domain Attribute-Based Access Control Encryption* (CD-ABACE) scheme and then propose an efficient CD-ABACE scheme with a constant ciphertext size and constant key length. Next we explain our results in more detail.

This paper re-defines the way to conceive the predicate function in ACE constructions by considering users' attributes instead of their identities. Based on an *Attribute-Based* predicate function, $\textsc{Pf} : \Sigma_k \times \Sigma_c \rightarrow \{0,1\}$, the senders with a certain ciphertext index value in Σ_c are limited to transmit data only to restricted recipients with a key index Σ_k. In a nutshell, for an attribute space $\mathbb{U}$, s.t. $\Sigma_k, \Sigma_c \subseteq \mathbb{U}$, the sender who owns a secret encryption key for ciphertext index $\mathbb{P} \in \Sigma_c$ can transmit data to those receivers with private decryption key corresponding to key index $\mathbb{B} \in \Sigma_k$, iff $\textsc{Pf}(\mathbb{B}, \mathbb{P}) = 1$, otherwise, the Sanitizer bans the communication between them. One of the main differences between this approach and the identity-based one is that the anonymity of the receivers corresponds

to the level of attribute hiding applied to the underlying *Attribute-Based Encryption (ABE)* scheme.

ABE schemes provide a powerful tool to enforce fine-grained access control over encrypted data; they have been used in several applications [33]. Goyal et al. in [23], proposed two complementary types of ABE schemes: *Key-Policy Attribute-Based Encryption* (KP-ABE) and *Ciphertext-Policy Attribute-Based Encryption* (CP-ABE) schemes. In CP-ABE, the sender embeds a (policy) function $f(\cdot)$ into ciphertext to describe which group of receivers can learn the encrypted message. In this approach, the ciphertext is labeled by an arbitrary function $f(\cdot)$, and secret keys are associated with attributes in the domain of $f(\cdot)$. The decryption algorithm yields the plaintext iff the receivers' attribute set $\mathbb{A}$ satisfies $f(\cdot)$, i.e., $f(\mathbb{A}) = 1$. Conversely, in KP-ABE the secret keys are labeled by the function $f(\cdot)$; this label is set in the setup phase and a ciphertext can only be decrypted with a key whose access structure is satisfied by the set of attributes. In KP-ABE, the access policy cannot be altered after setup phase, while in CP-ABE data owners can control the data access.

Hence, we utilize CP-ABE schemes to limit senders to transmit data to a specific ciphertext index $\mathbb{P}$. While CP-ABE schemes only enable fine-grained access to the encrypted data, they are not equipped to enforce policies for writing a message as well; thus we need additional functionalities to cover the latter by defining secret encryption keys. We utilize a Structure-Preserving Signature to guarantee the given encryption key is valid and one can only get access with a valid signature. A signature of this type allows selective re-randomization of a valid signature, and therefore efficiently proves the validity of this operation. Additionally, the CP-ABE scheme must also be re-randomizable in order to achieve the key-less sanitizability.

Based on realistic application scenarios for ACE constructions, the proposed scheme follows the Cross-Domain key generation method, proposed by Wang and Chow in [39]. In an ACE scheme, the users might belong to two distinct companies with different security levels, so one of them may not be able to grant access rights to the other. In this context, two entities referred to as Sender Authority and Receiver Authority locally generate secret keys for senders and receivers, respectively. Moreover, since users, including senders and receivers, may need to be added to the system later on, the setup phase will be carried out independently of the predicate function. Hence, our approach follows this setup method

and we provide a generic construction of a *Cross-Domain Access Control Encryption* scheme based on *Attribute-Based Encryption* constructions.

We finally propose an efficient CD-ABACE construction with constant key and ciphertext sizes. To obtain a CD-ABACE scheme that is efficient both in the length of the parameters and the computational overhead, we propose a novel CP-ABE scheme with AND-gate circuits. More specifically, we say a Boolean AND-gate circuit is satisfied (i.e, the output is true) iff all the input gates are true. In particular, we say the set of attributes $\mathbb{B} \subset \mathbb{U}$ satisfies the AND-gate circuit with the set of input constraints $\mathbb{P} \subseteq \mathbb{U}$ iff $\mathbb{P}$ is a subset of $\mathbb{B}$, i.e., $\mathbb{P} \subseteq \mathbb{B}$. As a simple example, let $\mathbb{U} = \{U_1, U_2, U_3, U_4\}$, then the set of input wires $\mathbb{B} = \{U_1, U_3, U_4\}$ satisfies the circuit $\mathbb{P} = \{U_1, U_4\}$, because $\mathbb{P} \subseteq \mathbb{B}$. Identity-based encryptions are special cases of AND-gate ABE schemes with an attribute universe consisting of the users' identity and also $|\mathbb{B}| = 1$. Moreover, in this construction the SANITIZER only requires public parameters, but no secret or public keys. Our CD-ABACE scheme has the following properties:

- Predicate function takes as inputs user attributes instead of their identities.
- The length of the ciphertext remains constant regardless of the number of receivers and the number of attributes in the access policy.
- All users' secret keys for encryption and decryption consist of only one group element, regardless of the number of attributes of the users.
- As an additional result, we present an efficient CP-ABE scheme with constant size ciphertexts and keys.

Table 1 compares the efficiency of the proposed construction with related works. As illustrated, in our scheme the lengths of the ciphertext and the key are improved to a constant size. The computational overhead for decryption grows linearly with the number of attributes that a receiver owns, while the encryption cost is constant and completely independent of the number of intended recipients. Our experiments show that the time required to run the encryption and decryption algorithm is only ~ 15 ms and ~ 45 ms, respectively.

Road-map: The rest of the paper is organized as follows. In Sect. 2, we review the preliminaries and definitions and describe the system architecture. The formal definition of the CD-ABACE scheme and its security definitions are described in Sect. 3. In Sect. 4, we propose the generic construction of CD-ABACE schemes and discuss their security features. In Sect. 5 we present an efficient CD-ABACE construction based on a

Table 1. Comparison of Efficiency and Functionality. n is the number of receivers and the total number of attributes in the system. $r \ll n$ indicates the maximum number of receivers that any sender is allowed to communicate with, and $s \ll n$ denotes the maximum number of senders that any receiver can receive a message from. $t \ll n$ indicates the maximum number of attributes that a sender can transmit data to. The maximum number of legitimate attributes that any recipients possesses to decrypt a ciphertext is denoted by $w \ll n$. SS, CD, PF, PE, IB, AB are short for Selectively Secure, Cross-Domain, Predicate Function, Predicate Function, Identity-Based and Attribute-Based, respectively.

Scheme	**Ciph. size**	**Enc. key size**	**Dec. key size**	**San. key size**	**Enc. cost**	**Dec. cost**	**CD**	**PF**	**Assump.**
[15, ‡ 3]	$O(2^n)$	$O(r)$	$O(1)$	$O(1)$	$O(n)$	$O(n)$	✓	IB	DDH/DCR
[15, ‡ 4]	$poly(n)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$	✗	IB	iO
[19]	$O(n)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$	✗	IB	SXDH
[27]	$poly(n)$	$O(1)$	$O(1)$	$O(1)$	$O(n)$	$O(n)$	✗	PE	DDH/LWE
[39] (SS)	$O(1)$	$O(1)$	$O(s)$	0	$O(1)$	$O(s)$	✓	IB	GBDP
Ours (SS)	$O(1)$	$O(1)$	$O(1)$	0	$O(1)$	$O(w)$	✓	AB	MSE-DDH

novel CP-ABE scheme. The performance of the proposed construction is compared in Sect. 6.

2 Preliminaries and Definitions

To detail the CD-ABACE schemes we need to review some preliminaries. Throu-ghout, we suppose the security parameter of the scheme is λ and $\mathsf{negl}(\lambda)$ denotes a negligible function. Let $\mathbb{U} = \{U_1, \ldots, U_n\} \in \mathbb{Z}_p^n$ be a set and for each subset $\mathbb{A} \subset \mathbb{U}$ we denote the i^{th} component scalar of this subset by A_i. We use $Y \leftarrow\!\$\, F(X)$ to denote a probabilistic function F that on input X is uniformly sampled resulting in the output Y. Also, $[n]$ denotes the set of integers between 1 and n. The algorithms are randomized unless expressly stated. "PPT" refers to "Probabilistic Polynomial Time". Two computationally indistinguishable distributions A and B are shown with $A \approx_c B$. We assumed a prime order field $\mathbb{F}$ and denote by $\mathbb{F}_{<d}[X]$ the set of univariate polynomials with degree smaller than d. The i^{th} coefficient of the univariate polynomial $f(x) \in \mathbb{F}_{<d}[X]$ is denoted by f_i and a polynomial with degree d has at most $d+1$ coefficients. The set $\{1, X, X^2, \ldots, X^d\}$ forms the standard basis: it is trivial to show that the representation of the coefficients for a polynomial with degree d as the coefficients of powers X is unique. The vector of A is denoted by $\vec{A}$.

Definition 2.1 (Access Structure [6]). *For a given set of parties $\mathcal{P} = \{p_1, \ldots, p_n\}$, we say a collection $\mathbb{U} \subseteq 2^{\mathcal{P}}$ is monotone if, for all A, B, if $A \in \mathbb{U}$ and $A \subseteq B$ then $B \in \mathbb{U}$. Also, a(n) (monotonic) access structure is a (monotone) collection $\mathbb{U} \subseteq 2^{\mathcal{P}} \setminus \{\emptyset\}$. We call the sets in $\mathbb{U}$ authorized sets and the sets that do not belong to $\mathbb{U}$ are called unauthorized.*

Definition 2.2 (Binary Representation of a subset). *For a given universe set $\mathbb{U}$ of size n, we can represent each subset $\mathbb{A}$ as a binary string of length n. Particularly, the i^{th} the element of the binary string for the subset $\mathbb{A} \subseteq \mathbb{U}$ is equal to 1 (i.e., $a[i] = 1$) if $A_i = U_i$. We show a binary representation set as binary tuple $(a[1], \ldots, a[n]) \in \mathbb{Z}_2^n$.*

Definition 2.3 (Zero-polynomial). *For a finite set $\mathbb{U} = \{k_1, \ldots, k_n\}$, we define the zero-polynomial $Z_{\mathbb{A}}(X)$ for a nonempty subset of $\mathbb{A} \subset \mathbb{U}$ as $Z_{\mathbb{A}}(X) := \prod_{i=1}^{n} (X - k_i)^{\overline{a[i]}}$, where $\overline{a[i]}$ is the binary representation of the complement set $\overline{\mathbb{A}}$. In other words, this univariate polynomial vanishes on all the elements of the set $\mathbb{U}$ for which the binary representation of the subset $\mathbb{A}$ is zero.*

The Zero-polynomial corresponding to subset $\mathbb{A} \subset \mathbb{U}$ is divisible by the Zero-polynomial of subset $\mathbb{B} \subset \mathbb{U}$ iff $\mathbb{A} \subseteq \mathbb{B}$. The result of this division is equal to the Zero-polynomial for the complement set of $(\mathbb{B} \setminus \mathbb{A})$ (i. e., $\overline{(\mathbb{B} \setminus \mathbb{A})}$). As a simple example, let $\mathbb{U} = \{1, 2, 3, 4\}$, $\mathbb{A} = \{2, 3\}$ and $\mathbb{B} = \{1, 2, 3\}$. Then we have $Z_{\mathbb{A}}(x) = (x-1)(x-4)$ and $Z_{\mathbb{B}}(x) = (x-4)$. Obviously, the zero-polynomial $Z_{\mathbb{A}}(x)$ is divisible by $Z_{\mathbb{B}}(x)$ and the result of this division is $Z_{\overline{(\mathbb{B} \setminus \mathbb{A})}}(x) = (x-1)$. Conversely, the inverse of this division is rational and we cannot represent it in the standard basis.

Definition 2.4 (Bilinear Groups [12]). *A Type-III [1] bilinear group generator $\mathcal{BG}(\lambda)$ returns a tuple $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathsf{p}, \hat{e})$, such that $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are cyclic groups of the same prime order p, and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ such that $\hat{e}(G, H) \neq 1$ is an efficiently computable bilinear map with the following properties;*

- $\forall\, a, b \in \mathbb{Z}_{\mathsf{p}}$, $\hat{e}(G^a, H^b) = \hat{e}(G, H)^{ab} = \hat{e}(G^b, H^a)$,
- $\forall\, a, b \in \mathbb{Z}_{\mathsf{p}}$, $\hat{e}(G^{a+b}, H) = \hat{e}(G^a, H)\hat{e}(G^b, H)$.

We use the bracket notation: for randomly selected generators $G \in \mathbb{G}_1$ and $H \in \mathbb{G}_2$ we denote $x \cdot G \in \mathbb{G}_1$ with $[x]_1$, and we write $\hat{e}\left(G^a, H^b\right) = [a]_1 \bullet [b]_2$.

[1] For the two distinct cyclic groups $\mathbb{G}_1 \neq \mathbb{G}_2$, there is neither efficient algorithm to compute a nontrivial homomorphism in both directions.

2.1 System Architecture

The proposed scheme's architecture is based on the Cross-Domain ACE technique described in [39]. In a Cross-Domain ACE setting, two distinct entities generate the keys to determine which group of senders can send data to a certain group of receivers and control which group of receivers can read this data. There are five entities in this system as follows:
Receiver Authority (RA) as a trusted third party generates and distributes system parameters and the secret decryption keys for the Receivers. For this aim, based on a certified predicate function $\textsc{Pf}(.,.)$, it authorizes the claimed attributes by the receivers and returns the corresponding secret decryption keys.
Sender Authority (SA) as a semi-trusted entity generates the pair of SA's public parameters and master secret keys; it publishes the former, while it keeps the latter secret. Moreover, it generates the secret encryption keys for the Senders based on a predicate function $\textsc{Pf}(.,.)$ and SA's master secret keys.
Sanitizer is an honest-but-curious party in the network that checks the validity of the communication links and acts based on the predicate function $\textsc{Pf}(.,.)$. If the sender does not allow to transmit a message to the recipients, then the Sanitizer bans the request, else it broadcasts the received ciphertexts. The Sanitizer is semi-honest which means that it follows the protocol honestly but tries to infer some sensitive information including the identities of the users (Senders and Receivers) or compromise the secrecy of a message.
Senders: to share a secret message among a group of receivers, they encrypt data and send the resulting ciphertext to the Sanitizer along with a proof to ensure that they possess a valid encryption key generated by the SA.
Receivers: by having access to the ciphertexts, they can recover the plaintexts using their own attributes and the corresponding secret key for decryption. Conversely, if the receiver does not satisfy the access policy then the ciphertext never reveals any meaningful information about the encrypted message.

In a nutshell, RA sets up the global public parameters of the network and publishes them, while it securely stores its master secret key. After authorizing the receivers' attribute set, RA computes the decryption secret keys corresponding to their attribute sets. From the public parameters issued by RA, SA generates the rest of parameters required for the authorization of the senders. Also, SA uses its master secret key to create the authorized secret encryption keys for the senders based on the

predicate function $\textsc{Pf}(.,.)$. Since RA is generating the main parameters of the system, it can compromise the security requirements, so we assume this entity is fully-trusted. The sender who wants to share a message securely among a group of receivers re-randomizes the signature (to ensure sender anonymity), then encrypts the plaintext and proves the validity of the claimed hidden witness. The SANITIZER receives the sender's request, and checks the validity of the proof and the signature to decide on rejecting the unauthorized senders without learning their identities. Otherwise, if the sender – based on the predicate function – is authorized to communicate with the selected group of receivers, the SANITIZER re-randomizes the received ciphertext and then passes the sanitized ciphertext on the recipients. Finally, the receivers who are allowed to decrypt a ciphertext can run the decryption algorithm and retrieve the message, else they learn nothing about it. It is assumed the SANITIZER is honest-but-curious: while it follows the protocol honestly, it is unable to compromise the message secrecy and anonymity of the users.

3 Cross-Domain Attribute-Based ACE scheme

Next we introduce the notion of *Cross-Domain Attribute-Based Access Control Encryption* (CD-ABACE) schemes. The high-level idea behind the definition of a CD-ABACE is that we can generalize the concept of Boolean relations in the plain CP-ABE schemes (see App. B.3) to the predicate function in an ACE construction. In this scenario, the encryption key generator allows the sender to talk to a restricted group of receivers based on a given predicate function. By contrast with the original approach of specifying the ciphertext access rights during the encryption phase, in the present approach, the Sender Authority declares the access right during the encryption key generation phase. Moreover, the generated encryption keys are signed by the SA, and no one can convincingly assert ownership unless they have a correct signature.

Definition 3.1 (CD-ABACE schemes). *A* CD-ABACE *scheme,* $\Psi_{\text{CD-ABACE}}$*, over the message space* $\mathcal{M}$*, the ciphertext space* $\mathcal{C}$ *and a predicate function* $\textsc{Pf} : \Sigma_k \times \Sigma_c \to \{0,1\}$ *has the following PPT algorithms:*

- $(\mathsf{pp}_{ra}, \mathsf{msk}_{ra}) \leftarrow \mathsf{RAgen}(\mathbb{U}, \lambda)$*: This randomized algorithm takes as inputs the security parameter* λ *and the universe attribute set* $\mathbb{U}$*, and outputs the public parameters* pp_{ra} *and master secret key* msk_{ra}*.*
- $(\mathsf{pp}_{sa}, \mathsf{msk}_{sa}) \leftarrow \mathsf{SAgen}(\lambda, \mathsf{pp}_{ra})$*: This randomized algorithm takes the security parameter* λ *and RA's public parameters* pp_{ra} *as inputs and*

generates the pair of SA's public parameters pp_{sa} *and SA's master secret key* msk_{sa}.

- $(\mathsf{dk}_{\mathbb{B}}) \leftarrow \mathsf{DecKGen}(\mathsf{msk}_{ra}, \mathbb{B})$*: This randomized algorithm takes RA's master secret key* msk_{ra} *and the authorized set of attributes* $\mathbb{B} \in \Sigma_k$ *as inputs and outputs the corresponding private decryption key* $\mathsf{dk}_{\mathbb{B}}$.
- $(\mathsf{ek}_{\mathbb{P}}, \sigma, W) \leftarrow \mathsf{EncKGen}(\mathsf{pp}_{ra}, \mathsf{pp}_{sa}, \mathsf{msk}_{sa}, \mathbb{P}, \mathrm{P\small F})$*: This algorithm takes the public parameters* pp_{ra} *and* pp_{sa}*, the SA's master secret key* msk_{sa}*, authorized ciphertext index* $\mathbb{P} \in \Sigma_c$*, and predicate function* $\mathrm{P\small F}(.,.)$ *as inputs. It returns the secret encryption key* $\mathsf{ek}_{\mathbb{P}}$ *that enforces that only the sender can send a message to those receivers who satisfy* $\mathbb{P}$ *along with the signature* σ *and its underlying re-randomizing token* W.
- $(\pi, \mathsf{x}) \leftarrow \mathsf{Enc}(\mathsf{pp}_{ra}, \mathsf{pp}_{sa}, m, \mathsf{ek}_{\mathbb{P}}, \sigma, W)$*: This algorithm takes as inputs the public parameters, a message* $m \in \mathcal{M}$*, the encryption key corresponding to the attribute set of* $\mathbb{P}$*, a valid signature* σ *and the token* W*. It returns a request including a proof* π *along with its underlying public instance* x.
- $\left(\tilde{\mathsf{Ct}}, \perp\right) \leftarrow \mathsf{San}(\mathsf{pp}_{ra}, \mathsf{pp}_{sa}, \pi, \mathsf{x}, \mathrm{P\small F})$*: This algorithm takes as inputs the public parameters* pp_{ra} *and* pp_{sa}*, a ciphertext along with a proof* π *and its corresponding instance* x*. Afterwards, the algorithm either re-randomizes the ciphertext to* $\tilde{\mathsf{Ct}}$ *or rejects the request. To this end, it checks the validity of the proof and, if it allows this flow based on the predicate function* $\mathrm{P\small F}(.,.)$*, it transfers the ciphertext* $\tilde{\mathsf{Ct}} \in \mathcal{C}$ *to the receivers, else it returns* $\perp$.
- $(m', \perp) \leftarrow \mathsf{Dec}(\mathsf{pp}_{ra}, \mathsf{pp}_{sa}, \tilde{\mathsf{Ct}}, \mathsf{dk}_{\mathbb{B}})$*: The decryption algorithm takes as inputs the public parameters* pp_{ra} *and* pp_{sa}*, a re-randomized ciphertext* $\tilde{\mathsf{Ct}}$ *and the decryption key* $\mathsf{dk}_{\mathbb{B}}$*. If* $\mathrm{P\small F}(\mathbb{B}, \mathbb{P}) = 1$*, then it returns a message* $m' \in \mathcal{M}$*, otherwise it responds by* $\perp$*. In other words, a recipient with a wrong decryption key learns nothing from the output of this algorithm.*

3.1 Security Definitions

Next we present the required security properties for a CD-ABACE scheme under only CPA-based definitions, where $\mathcal{A}$ has access to encryption, encryption-key generation, and decryption-key generation oracles. Noted that the following security games are motivated by the notion of co-selective CPA security in [5], such that $\mathcal{A}$ has to declare q decryption key queries before the Initialization phase, while it can select the target challenge ciphertext, adaptively. We slightly modify the extended security notions introduced in [39] to adapt them to the CD-ABACE system model.

Definition 3.2 (Correctness). *For a given attribute universe* $\mathbb{U}$ *and predicate function* $\text{PF} : \Sigma_k \times \Sigma_c \to \{0,1\}$*, we say that* $\Psi_{\text{CD-ABACE}}$ *over message space* $\mathcal{M}$ *and ciphertext space* $\mathcal{C}$ *is correct if we have,*

$$\Pr\left[\mathsf{Dec}\left(\mathsf{dk}_{\mathbb{B}}, \mathsf{San}(\mathsf{Enc}(m, \mathsf{ek}_{\mathbb{P}}, \mathbb{P}))\right) = m : \text{PF}(\mathbb{B}, \mathbb{P}) = 1\right] \approx_c 1$$

Correctness captures the feature that a sender with an encryption key $\mathsf{ek}_{\mathbb{P}}$ is able to deliver a message to those receivers for which the attribute set $\mathbb{B}$ satisfies $\text{PF}(\mathbb{B}, \mathbb{P}) = 1$ with a high probability. In this case, the SANITIZER should pass the information on and a receiver with decryption key $\mathsf{dk}_{\mathbb{B}}$ should be able to retrieve the message correctly from a re-randomized ciphertext.

Definition 3.3 (No-Read Rule). *Consider* $\Psi_{\text{CD-ABACE}}$ *over the attribute universe* $\mathbb{U}$*, message space* $\mathcal{M}$*, a ciphertext space* $\mathcal{C}$ *and a predicate function* $\text{PF} : \Sigma_k \times \Sigma_c \to \{0,1\}$*. For a security parameter* λ*, we say that a PPT adversary* $\mathcal{A}$ *wins the defined* NO-READ *rule security game described in Fig. 1 with access to the oracles in the same table, if she guesses the random bit* b *better than by chance. It is assumed that for a challenge access structure* $\mathbb{P}^*$*,* $\mathcal{A}$ *would not request the decryption key for attribute set* $\mathbb{B}_j$*, such that* $\text{PF}(\mathbb{B}_j, \mathbb{P}^*) = 1$*.* $\Psi_{\text{CD-ABACE}}$ *satisfies the* NO-READ *rule if for all PPT adversaries* $\mathcal{A}$ *with advantage* $Adv^{\text{NO-READ}}_{\Psi_{\text{CD-ABACE}},\mathcal{A}}(1^\lambda, b) = (\Pr[\mathcal{A}$ *wins the* NO-READ *game*$] - 1/2)$ *we have,* $\left| Adv^{\text{NO-READ}}_{\Psi_{\text{CD-ABACE}},\mathcal{A}}(1^\lambda, b = 0) - Adv^{\text{NO-READ}}_{\Psi_{\text{CD-ABACE}},\mathcal{A}}(1^\lambda, b = 1) \right| \approx_c 0$*, where we call* $\mathcal{A}$ *wins the defined security game, iff* $b' == b$*.*

Similar to the ID-based ACE constructions, the NO-READ rule in an attribute-based model enforces that only eligible recipients who satisfy a certain access structure, should learn the message while the other participants learn nothing. In particular, not only should an unauthorized receiver be unable to read the messages, combining the decryption secret keys of a group of unauthorized receivers should not reveal any information about the message. Also, this property has to hold even if the recipients collude with the SANITIZER.

Definition 3.4 (Parameterized No-Write Rule). *Consider* $\Psi_{\text{CD-ABACE}}$ *over* $\mathbb{U}$*, a message space* $\mathcal{M}$*, ciphertext space* $\mathcal{C}$ *and a predicate function* $\text{PF} : \Sigma_k \times \Sigma_c \to \{0,1\}$*. We say a* $\Psi_{\text{CD-ABACE}}$ *scheme satisfies the Parameterized* NO-WRITE *rule, if no PPT adversary* $\mathcal{A}$ *with access to the oracles in Fig. 1 has a non-negligible advantage in winning the* NO-WRITE *game, i.e, under the advantage* $Adv^{\text{NO-WRITE}}_{\Psi_{\text{CD-ABACE}},\mathcal{A}}(1^\lambda, b) =$

$(\Pr[\mathcal{A} \text{ wins No-Write}] - 1/2)$ *we have,*

$$\left| Adv^{\text{No-Write}}_{\Psi_{\text{CD-ABACE}},\mathcal{A}}(1^\lambda, b=0) - Adv^{\text{No-Write}}_{\Psi_{\text{CD-ABACE}},\mathcal{A}}(1^\lambda, b=1) \right| \approx_c 0.$$

We say $\mathcal{A}$ wins the defined No-Write *game iff $b' == b$ under the condition that for all queried secret encryption keys $\mathbb{P}_i \in \mathcal{Q}_\mathcal{E} \cup \{\mathbb{P}^*\}$ and all requested private decryption keys $\mathbb{B}_j \in \mathcal{Q}_\mathcal{D}$, along with the challenge access structure $\mathbb{P}^*$, we have $\text{Pf}(\mathbb{B}_j, \mathbb{P}_i) = 0$. The function* fix(.) *accepts a ciphertext* Ct *as input and generates auxiliary information* aux *of* Ct *that is not sanitizable [39]. By seeding an encryption algorithm with this auxiliary information, the resulting ciphertext has also the same auxiliary information.*

$\text{No-Read}^{\mathcal{A}}_{\text{CD-ABACE}}(1^\lambda, \mathbb{U})$

$(\mathsf{pp}_{ra}, \mathsf{msk}_{ra}) \leftarrow \mathsf{RAgen}(1^\lambda, \mathbb{U})$
$(\mathsf{pp}_{sa}, \mathsf{msk}_{sa}) \leftarrow \mathsf{SAgen}(\mathsf{pp}_{ra}, \mathbf{R_L})$
$\mathbb{P}^* \leftarrow \mathcal{A}(\mathsf{pp}_{ra}, \mathsf{pp}_{sa})$
$(m_0, m_1) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{pp}_{ra}, \mathsf{pp}_{sa})$
$(\mathsf{ek}_{\mathbb{P}^*}, \sigma^*, W^*) \leftarrow \mathsf{EncKGen}(\mathbb{P}^*)$
$b \leftarrow\$ \{0,1\}$
$(\pi_b, \mathsf{x}_b) \leftarrow\$ \mathsf{Enc}(\mathsf{pp}_{ra}, \mathsf{pp}_{sa}, \mathsf{ek}_{\mathbb{P}^*}, m_b)$
$b' \leftarrow\$ \mathcal{A}^{\mathcal{O}}(\pi_b, \mathsf{x}_b)$

$\text{No-Write}^{\mathcal{A}}_{\text{CD-ABACE}}(1^\lambda, \mathbb{U})$

$(\mathsf{pp}_{ra}, \mathsf{msk}_{ra}) \leftarrow \mathsf{RAgen}(1^\lambda, \mathbb{U})$
$(\mathsf{pp}_{sa}, \mathsf{msk}_{sa}) \leftarrow \mathsf{SAgen}(\mathsf{pp}_{ra}, \mathbf{R_L})$
$(\pi^*, \mathsf{x}^*, \mathbb{P}^*) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{pp}_{ra}, \mathsf{pp}_{sa})$
$(\pi_0, \mathsf{x}_0) := (\pi^*, \mathsf{x}^*)$
$(\mathsf{ek}_{\mathbb{P}^*}, \sigma^*, W^*) \leftarrow \mathsf{EncKGen}(\mathbb{P}^*)$
$m^* \leftarrow\$ \mathcal{M},\ \mathsf{aux} \leftarrow \mathsf{fix}(\mathtt{Ct}_0)$
$(\pi_1, \mathsf{x}_1) \leftarrow \mathsf{Enc}(\mathsf{ek}_{\mathbb{P}^*}, m^*, \mathsf{aux})$
$b \leftarrow\$ \{0,1\}, \tilde{\mathtt{Ct}}_b \leftarrow \mathsf{San}(\pi_b, \mathsf{x}_b)$
$b' \leftarrow\$ \mathcal{A}^{\mathcal{O}}(\tilde{\mathtt{Ct}}_b)$

Oracle $\mathcal{O}_{\mathsf{DecKGen}}(\mathbb{B}_j)$

Initialize $\mathcal{Q}_\mathcal{D} = \{\emptyset\}$
if $\mathbb{B}_j \notin \mathcal{Q}_\mathcal{D}$:
  $\mathsf{dk}_{\mathbb{B}_j} \leftarrow \mathsf{DecKGen}(\mathbb{B}_j)$
  return $(\mathsf{dk}_{\mathbb{B}_j}) \wedge \mathcal{Q}_\mathcal{D} = \mathcal{Q}_\mathcal{D} \cup \{\mathbb{B}_j\}$
else : **return** $(\mathsf{dk}_{\mathbb{B}_j})$

Oracle $\mathcal{O}_{\mathsf{EncKGen}}(\mathbb{P}_i)$

Initialize $\mathcal{Q}_\mathcal{E} = \{\emptyset\}$
if $\mathbb{P}_i \notin \mathcal{Q}_\mathcal{E}$:
  $(\mathsf{ek}_{\mathbb{P}_i}, \sigma_i, W_i) \leftarrow \mathsf{EncKGen}(\mathbb{P}_i, \text{Pf})$
  return $(\mathsf{ek}_{\mathbb{P}_i}, \sigma_i, W_i) \wedge \mathcal{Q}_\mathcal{E} = \mathcal{Q}_\mathcal{E} \cup \{\mathbb{P}_i\}$
else : **return** $(\mathsf{ek}_{\mathbb{P}_i}, \sigma_i, W_i)$

Oracle $\mathcal{O}_{\mathsf{Enc}}(m, \mathbb{P}_i)$

$(\mathsf{ek}_{\mathbb{P}_i}, \sigma_i, W_i) \leftarrow \mathsf{EncKGen}(\mathbb{P}_i, \text{Pf})$
$(\pi, \mathsf{x}) \leftarrow \mathsf{Enc}(\mathsf{ek}_{\mathbb{P}_i}, m)$
return (π, x)

Fig. 1. No-Read and No-Write Security Games

Remark 3.1. With regard to the security definitions, the anonymity of the sender is guaranteed and the SANITIZER cannot deduce the identity of the sender while the receivers' anonymity relies on the CP-ABE construction. Note that the same type of property is known as weak attribute hiding in the context of ABE constructions [29]. Although an IND-CPA-secure CP-ABE satisfies the payload hiding property, a stronger security concept, called attribute-hiding CP-ABE, ensures that the set of attributes associated with each ciphertext is also obscured [26]. The latter increases the ciphertext size incrementally and the identity-based encryptions reveal the receivers' identity in plain.

4 Generic Construction

Our generic construction for a general predicate function and universal CP-ABE is built from following constructions:

1. An EUF-CMA-secure SPS construction, $\mathcal{SPS}.(\mathsf{Pgen}, \mathsf{KG}, \mathsf{Sign}, \mathsf{Randz}, \mathsf{Vf})$ (see App. B.1).
2. A computational Knowledge-Sound NIZK proof, $\mathcal{ZK}.(\mathsf{K}_{\vec{\mathsf{crs}}}, \mathsf{P}, \mathsf{V}, \mathsf{Sim})$ (see App. B.2).
3. A publicly re-randomizable CP-ABE scheme, $r\mathcal{ABE}.(\mathsf{Pgen}, \mathsf{KGen}, \mathsf{Col}, \mathsf{Enc}, \mathsf{Randz}, \mathsf{Dec})$ (see App. B.3).

For a given predicate function $\text{PF} : \Sigma_k \times \Sigma_c \to \{0,1\}$, and message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$, the generic construction consists of the following PPT algorithms:

- **RA setup** ($\mathsf{RAgen}(\mathbb{U}, \lambda)$): Takes as inputs the security parameter λ and an attribute universe $\mathbb{U}$, and runs the $r\mathcal{ABE}.\mathsf{Pgen}(\lambda, \mathbb{U})$ algorithm to generate the global and CP-ABE parameters. It outputs RA's master secret key set $\mathsf{msk}_{ra} = (\mathsf{msk}_{r\mathcal{ABE}})$ and RA's public parameters $\mathsf{pp}_{ra} = (\mathsf{pp}_{r\mathcal{ABE}})$.
- **SA setup** ($\mathsf{SAgen}(\mathsf{pp}_{ra}, \mathbf{R_L})$): Takes as inputs RA's public parameters pp_{ra} and relation $\mathbf{R_L}$ and runs the $\mathcal{ZK}.\mathsf{K}_{\vec{\mathsf{crs}}}(\mathbf{R_L})$, $\mathcal{SPS}.\mathsf{Pgen}(\lambda)$ and $\mathcal{SPS}.\mathsf{KG}(\mathsf{pp})$ algorithms and returns $\mathsf{pp}_{sa} = (\mathsf{pp}, \mathsf{vk}, \vec{\mathsf{crs}})$ and $\mathsf{msk}_{sa} = (\vec{\mathsf{ts}}, \mathsf{sk})$ as outputs. The underlying relation $\mathbf{R_L}$ is defined corresponding to the NP-language $\mathbf{L}$ for the statement $\mathsf{x} = (\sigma', \mathsf{vk}', \mathsf{ek}', \mathtt{Ct})$ and witness $\mathsf{w} = (\sigma, \mathsf{ek}, m, r, t)$. We say the relation is satisfied, i.e. $\mathbf{R_L}(\mathsf{x}, \mathsf{w}) = 1$, if the following conditions hold:

$$\mathsf{ek}_{\mathbb{P}} \leftarrow r\mathcal{ABE}.\mathsf{Col}(\mathsf{pp}, \mathbb{P}) \wedge \mathcal{SPS}.\mathsf{Vf}(\mathsf{vk}, \mathsf{ek}_{\mathbb{P}}, \sigma) = 1 \wedge$$
$$(\sigma', \mathsf{vk}') \leftarrow \mathcal{SPS}.\mathsf{Randz}(\sigma, W; t) \ \wedge \ \mathtt{Ct} \leftarrow r\mathcal{ABE}.\mathsf{Enc}(\mathsf{ek}_{\mathbb{P}}, m; r) \ .$$

- **Decryption KGen** ($\mathsf{DecKGen}(\mathsf{msk}_{ra}, \mathbb{B})$): Takes as inputs RA's master secret key msk_{ra} and a key index $\mathbb{B} \in \Sigma_k$. It generates the private decryption key $\mathsf{dk}_{\mathbb{B}}$ by executing the algorithm $r\mathcal{ABE}.\mathsf{KGen}(\mathsf{msk}_{ra}, \mathbb{B})$.
- **Encryption KGen** ($\mathsf{EncKGen}(\mathsf{pp}_{ra}, \mathsf{msk}_{sa}, \mathbb{P}, \textsc{Pf})$): Takes as inputs pp_{ra}, msk_{sa} and a ciphertext index $\mathbb{P} \in \Sigma_c$ that indicates to whom the sender is allowed to talk based on predicate function $\textsc{Pf}(.,.)$. It executes the collector algorithm $r\mathcal{ABE}.\mathsf{Col}(\mathsf{pp}_{ra}, \mathbb{P})$ to obtain the aggregated value $\mathsf{ek}_{\mathbb{P}}$ and then signs it by running the algorithm $\mathcal{SPS}.\mathsf{Sign}(\mathsf{sk}, \mathsf{ek}_{\mathbb{P}})$. It returns both the encryption key and the underlying signature to the sender.
- **Encryption** ($\mathsf{Enc}\,(\mathsf{pp}_{sa}, \mathsf{pp}_{ra}, m, \mathsf{ek}_{\mathbb{P}}, \sigma, W)$): Takes as inouts the secret encryption key $\mathsf{ek}_{\mathbb{P}}$ and the underlying signature σ, the public parameters and a message $m \in \mathcal{M}$. It re-randomizes σ under an initial random string μ by running $\mathcal{SPS}.\mathsf{Randz}(\mathsf{pp}_{sa}, \mathsf{ek}_{\mathbb{P}}, \sigma, W; \mu)$. Next it runs the re-randomizable CP-ABE encryption algorithm $r\mathcal{ABE}.\mathsf{Enc}(\mathsf{pp}_{ra}, m, \mathsf{ek}_{\mathbb{P}})$ and proves knowledge of hidden values by executing the $\mathcal{ZK}.\mathsf{P}(\mathbf{R_L}, \vec{\mathsf{crs}}, \mathsf{w}, \mathsf{x})$ algorithm. It returns the instance and underlying proof (π, x) as outputs.
- **Sanitization** ($\mathsf{San}(\mathsf{pp}_{sa}, \mathsf{pp}_{ra}, \pi, \mathsf{x})$): Takes as inputs the proof π and the instance x: if $\mathcal{SPS}.\mathsf{Vf}(\mathsf{pp}, \mathsf{vk}', \sigma', \mathsf{ek}') = 1$ and $\mathcal{ZK}.\mathsf{V}(\mathbf{R_L}, \vec{\mathsf{crs}}, \pi, \mathsf{x}) = 1$, it runs the algorithm $r\mathcal{ABE}.\mathsf{Randz}(\mathsf{pp}_{ra}, \mathtt{Ct})$ and returns the sanitized ciphertext $\tilde{\mathtt{Ct}}$ as output; otherwise it rejects the link and returns $\perp$.
- **Decryption** ($\mathsf{Dec}(\mathsf{pp}_{sa}, \mathsf{pp}_{ra}, \tilde{\mathtt{Ct}}, \mathsf{dk}_{\mathbb{B}})$): Takes as inputs the public parameters, a sanitized ciphertext $\tilde{\mathtt{Ct}}$ and the decryption key $\mathsf{dk}_{\mathbb{B}}$. It returns the plaintext $m \in \mathcal{M}$ by executing $r\mathcal{ABE}.\mathsf{Dec}(\mathsf{pp}_{ra}, \tilde{\mathtt{Ct}}, \mathsf{dk}_{\mathbb{B}})$ algorithm if and only if $\textsc{Pf}(\mathbb{B}, \mathbb{P}) = 1$; otherwise this algorithm returns $\perp$.

Theorem 4.1. *The proposed generic* CD-ABACE *construction is correct.*

Proof. The proof can be found in App. C.1. □

Theorem 4.2. *The proposed generic* CD-ABACE *scheme satisfies the* No-Read *rule of Definition 3.3.*

Proof. The proof can be found in App. C.2. □

Theorem 4.3. *No PPT adversary $\mathcal{A}$ can win the* No-Write *security game of Definition 3.4 for the proposed* CD-ABACE *scheme under a fixed predicate function* $\textsc{Pf}(.,.)$.

Proof. The proof can be found in App. C.3. □

5 An Efficient CD-ABACE Scheme

In this section, we propose a CD-ABACE scheme such that the key and ciphertext sizes are constant. It primarily comes from a novel CP-ABE scheme; we believe that this is a result that is valuable by itself. Following on from Sect. 4, there are three main cryptographic primitives that are listed below;

Structure-Preserving Signature (SPS): In this paper, we use a variant of the selectively re-randomizable SPS scheme of Abe et al. [2] (see App. B.1) as an efficient, unified and selectively re-randomizable SPS. Since in the proposed CD-ABACE construction the generator of the first cyclic group is hidden and the message is a second group element over the Type-III bilinear groups, we need to slightly modify this scheme with the following PPT algorithms:

- $(\mathsf{pp}) \leftarrow \mathcal{SPS}.\mathsf{Pgen}(\lambda)$: This algorithm takes as input the security parameter λ and picks a random integer $\alpha \leftarrow\!\$\, \mathbb{Z}_p^*$ and a group generator $Y \leftarrow\!\$\, \mathbb{G}_2$. It returns the public parameters pp by running a Type-III bilinear group generator $\mathcal{BG}(\lambda) = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathsf{p}, \hat{e})$ and publishes $\mathsf{pp} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathsf{p}, \hat{e}, \left[\alpha^2\right]_1, Y)$, while it keeps α secret.
- $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathcal{SPS}.\mathsf{KG}(\mathsf{pp})$: Samples $v \leftarrow\!\$\, \mathbb{Z}_p$ and publishes the public verification key $\mathsf{vk} = \left[v\alpha^2\right]_1$ while it securely stores the secret signing key $\mathsf{sk} = v$.
- $(\sigma, W) \leftarrow \mathcal{SPS}.\mathsf{Sign}(\mathsf{pp}, \mathsf{sk}, m)$: This algorithm takes as inputs the public parameters pp, the secret key sk and a message $m \in \mathbb{G}_2$. It samples $r \leftarrow\!\$\, \mathbb{Z}_p^*$, computes $\sigma = (R, S, T) = \left(\left[r\alpha^2\right]_1, m^{v/r} Y^{1/r}, S^{v/r}\, [1/r]_2\right)$, and outputs $(\sigma, W = [1/r]_2)$.
- $(\sigma', W') \leftarrow \mathcal{SPS}.\mathsf{Randz}(\mathsf{pp}, \sigma, W)$: The re-randomizing algorithm takes as inputs the public parameters pp, a signature $\sigma \in \mathcal{S}$ along with token W, picks a random integer $t \leftarrow\!\$\, \mathbb{Z}_p^*$ and computes the re-randomized signature as $\sigma' = (R', S', T') = (R^{1/t}, S^t, T^{t^2} W^{t(1-t)})$ and returns it along with a new token $W' = W^t$.
- $(0, 1) \leftarrow \mathcal{SPS}.\mathsf{Vf}(\mathsf{pp}, \mathsf{vk}, \sigma', m)$: The verification algorithm takes as inputs pp, either a plain signature σ or a re-randomized signature σ', a message m and the verification key vk. It first checks $m, S', T' \in \mathbb{G}_2$, $R' \in \mathbb{G}_1$ and then checks the pairing equations $R' \bullet S' = (\mathsf{vk} \bullet m)(\left[\alpha^2\right]_1 \bullet Y)$ and $R' \bullet T' = (\mathsf{vk} \bullet S')(\left[\alpha^2\right]_1 \bullet [1]_2)$. If both conditions hold, then it returns 1, otherwise it responds with 0 (rejecting the signature).

The proof of correctness is identical to that of Abe et al.'s SPS construction, where a message is part of the second rather than the first

group. As the first group generator is hidden in the proposed CD-ABACE scheme, we need to take $\left[\alpha^2\right]_1$ instead of $[1]_1$ to generate and verify signatures.

Non-Interactive Zero-Knowledge (NIZK) proofs: As discussed in App. B.2, Zero-Knowledge proofs [22] allow a prover to convince the verifier about the validity of a statement without revealing any other information. We use a standard Schnorr proof [34] to prove the knowledge of exponents in the random oracle model. To convert an interactive protocol to a non-interactive framework, we utilize the Fiat-Shamir heuristic [18]. More precisely, the prover has access to a hash function, modeled as a random function ($\mathcal{O}$), to generate the challenges instead of receiving them from the verifier. For a given cyclic group $\mathbb{G}_i$ of order p with generator g_i, we denote by $\text{PoK}\{(\mathsf{w}) : \mathbf{R_L}(\mathsf{x}, \mathsf{w}) = 1\}$, the proof of knowledge of a hidden witness w that satisfies a given relation $\mathbf{R_L}$. Fig. 2 formalizes a NIZK in ROM for proof of exponentiation.

$\mathsf{K}_{\mathsf{crs}}(\mathbf{R_L}, \lambda)$	$\text{PROVE}(\mathbf{R_L}, \mathsf{x}, \mathsf{w})$	$\text{VERIFIER}(\mathbf{R_L}, \pi, \mathsf{x})$
1 : **Instance**$(\mathsf{x}) : y \in \mathbb{G}_i$	1 : Parse $(\mathbf{R_L}, \mathsf{x}, \mathsf{w})$	1 : Parse $(\mathbf{R_L}, \pi, \mathsf{x})$
2 : **Witness**$(\mathsf{w}) : x \in \mathbb{Z}_\mathsf{p}$	2 : $r \leftarrow\$\, \mathbb{Z}_\mathsf{p}$	2 : Computes $c = \mathcal{O}(y, t)$
3 : **Statement:**	3 : $t := g_i^r$	3 : **if** $\{y, t \in \mathbb{G}_i \ \wedge \ z \in \mathbb{Z}_\mathsf{p}$
4 : Knwl of $x := \log_{g_i} y$	4 : $c := \mathcal{O}(y, t)$	4 : $\quad \wedge\, y^c t == g_i^z\}$:
5 : **return** (x, w)	5 : $z := cx + r \mod \mathsf{p}$	5 : **return** (ACCEPT)
	6 : **return** $\pi = (t, z)$	6 : **else** : (REJECT)

Fig. 2. Proof of Knowledge of Exponents

An efficient Re-randomizable CP-ABE: In what follows, we define a new IND-CPA-secure CP-ABE scheme with a constant key and ciphertext size. The Boolean function of this scheme is applied in AND-gate circuits. Although Guo et al. in [24] took a similar approach and presented a constant-key size CP-ABE scheme, the ciphertext size in their scheme increases linearly with the total number of attributes. Our construction consists of the following algorithms:

- $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathcal{ABE}.\mathsf{Pgen}(\mathbb{U}, \lambda)$: Takes as inputs an attribute space $\mathbb{U}$ with size n along with the security parameter λ, and runs a Type-III bilinear group generator $\mathcal{BG}(\lambda) = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathsf{p}, \hat{e})$. It also selects a standard collision-resistant hash function $\mathsf{H} \leftarrow\$\, \mathcal{H}$ that is modeled as a

random oracle in the security proofs. For a randomly selected integer $\alpha \leftarrow\$ \mathbb{Z}_p^*$, it computes $h_i = \left[\alpha^i\right]_2$ as the set of monomials in $\mathbb{G}_2$ and $g_2 = \left[\alpha^2\right]_1$. It returns the master secret key $\mathsf{msk} = \left([1]_1, \alpha\right)$ and the system's public parameters $\mathsf{pp} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathsf{p}, \hat{e}, g_2, \{h_i\}_{i=0}^n, [\alpha]_T, \mathsf{H})$.

- $(\mathsf{dk}_{\mathbb{B}}) \leftarrow \mathcal{ABE}.\mathsf{KGen}(\mathsf{msk}, \mathbb{B})$: Takes as inputs msk and generates a secret decryption key corresponding to attribute set $\mathbb{B} \in \Sigma_k$, such that $|\mathbb{B}| < n-1$. It first computes the Zero-Polynomial $Z_{\mathbb{B}}(x) = \prod_{i=1}^n (x-k_i)^{\overline{b[i]}}$ such that $k_i = \{\mathsf{H}(U_i)\}_{U_i \in \mathbb{U}}$. It returns the secret decryption key $\mathsf{dk}_{\mathbb{B}} = [1/Z_{\mathbb{B}}(\alpha)]_1$.
- $(\mathtt{Ct}) \leftarrow \mathcal{ABE}.\mathsf{Enc}\,(\mathsf{pp}, m, \mathbb{P})$: Takes as inputs the message $m \in \mathcal{M}$, the public parameters pp and an access structure $\mathbb{P} \in \Sigma_c$. It first samples $r \leftarrow\$ \mathbb{Z}_p^*$, calculates $Z_{\mathbb{P}}(x) = \sum_{j=0}^n z_j x^j$ and returns the ciphertext as a tuple $\mathtt{Ct} = (\mathbb{P}, C, C_1, C_2) = (\mathbb{P}, m\,[r\alpha]_T, (\prod_{j=0}^n h_{j+1}^{z_j})^r = [r\alpha Z_{\mathbb{P}}(\alpha)]_2, g_2^{-r} = \left[-r\alpha^2\right]_1)$. We define the collector algorithm as $\mathsf{Col}(\mathsf{pp}, \mathbb{P}) = [\alpha Z_{\mathbb{P}}(\alpha)]_2$.
- $(m', \perp) \leftarrow \mathcal{ABE}.\mathsf{Dec}\,(\mathsf{pp}, \mathtt{Ct}, \mathsf{dk}_{\mathbb{B}})$: This algorithm takes as input the public parameters pp, a ciphertext $\mathtt{Ct}$ and a secret decryption key $\mathsf{dk}_{\mathbb{B}}$. If $\mathbb{P} \subseteq \mathbb{B}$, it computes, $F_{\mathbb{B},\mathbb{P}}(x) = \prod_{i=1}^n (x-k_i)^{c[i]} = \sum_{j=0}^n f_j x^j$ for $c[i] = b[i] - p[i]$ and returns $m' = C \cdot \left(\left(C_2 \bullet \prod_{i=1}^n (h_{i-1})^{f_i}\right) \cdot (\mathsf{dk}_{\mathbb{B}} \bullet C_1)\right)^{\frac{-1}{f_0}}$; otherwise it responds with $\perp$.

Theorem 5.1. *Based on Definition B.9, the proposed CP-ABE scheme is correct.*

Proof. The proof can be found in App. C.4. □

Theorem 5.2. *Under the $(l, m, t)-$`MSE-DDH` assumption, defined in Definition A.1, a PPT adversary $\mathcal{A}$ cannot win the* IND-CPA*-security game from Definition B.10 for the proposed CP-ABE scheme in the random oracle model.*

Proof. The proof can be found in App. C.5. □

Next we modify the re-randomizing phase of our CP-ABE scheme; the other algorithms are the same, except that the decryption algorithm can take either $\tilde{\mathtt{Ct}}$ or $\mathtt{Ct}$ as input.

- $(\tilde{\mathtt{Ct}}) \leftarrow r\mathcal{ABE}.\mathsf{Randz}(\mathsf{pp}, \mathtt{Ct})$: Takes as inputs pp and a ciphertext $\mathtt{Ct}$ under access structure $\mathbb{P} \in \Sigma_c$. To re-randomize the ciphertext $\mathtt{Ct} \in \mathcal{C}$, it samples an initial random integer $s \leftarrow\$ \mathbb{Z}_p^*$ and computes the Zero-polynomial $Z_{\mathbb{P}}(x)$. Then it outputs $\tilde{\mathtt{Ct}} = (\tilde{C}, \tilde{C}_1, \tilde{C}_2) = (C \cdot [s\alpha]_T, C_1 \cdot [sZ_{\mathbb{P}}(\alpha)]_2, C_2 \cdot g_2^{-s})$.

Remark 5.1. The proposed construction guarantees that no PPT adversary can obtain the receiver's identity, deterministically. This is the same as the notion of "weak attribute-hiding" in the context of Attribute-Based Signatures [36]. Indeed, the access policy corresponding to a ciphertext only reveals the list of receivers who satisfy a specific set of attributes, even though it never leaks any information about the identity of the receivers. Under the assumption that there is more than one user who satisfies a set of certain attributes, the adversary is unable to deduce for which specific receiver the challenge ciphertext is intended.

Related Works: The first CP-ABE scheme, which allows the data owners to implement an arbitrary and fine-grained access policy in terms of any monotonic formula for each message was proposed by Bethencourt et al. at IEEE S&P 2007 in [8]; its security was proven in the *Generic Group Model* (GGM). In a subsequent work, Cheung et al. [14] constructed a CP-ABE scheme in the standard model, which is however restricted to a single AND-gate. Waters [41] introduced an asymptotically efficient CP-ABE scheme in the standard model, which is based on a *Linear Secret Sharing Scheme* (LSSS) to establish an arbitrary access policy. Lewko and Waters [28] introduced a secure construction based on LSSS in which the length of the ciphertext, the size of users' secret keys, and the number of required pairings to decrypt a ciphertext correspond to the size of the *Monotone Span Program* (MSP) that defines the access structure. Some recent works have extended the functionality of these schemes for various applications [35,25]. While these CP-ABE schemes allow to define in an effective way the right to access data, either the key or the ciphertext size grows linearly in the number of attributes. Therefore, CP-ABE schemes based on AND-gate circuits are considered promising candidates to address this downside. In this approach the sender defines a specific Boolean AND-gate circuit such that a recipient can learn the encrypted data iff they satisfy all the attributes, otherwise the decryption algorithm returns nothing. Considering AND-gate circuits provides a constant ciphertext length; several CP-ABE schemes are proposed based on this approach [17,38,24,3].

The proposed CD-ABACE scheme: At this point, we can wrap up the construction described in Fig. 3 by taking a family of collision-resistant hash functions $\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$. Our CD-ABACE scheme is built under a CP-ABE scheme based on AND-gate circuits with constant key and ciphertext sizes. The primary motivation behind this circuit choice is to construct a fully constant ACE within the context of CD-

$(\mathsf{pp}_{ra}, \mathsf{msk}_{ra}) \leftarrow \mathsf{RAgen}(\mathbb{U}, \lambda)$

1 : Run $\mathcal{BG}(\lambda) = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathsf{p}, \hat{e})$
2 : $\mathsf{H} \leftarrow\$\, \mathcal{H}, \alpha \leftarrow\$\, \mathbb{Z}_p^*$
3 : $h_i = \left[\alpha^i\right]_2$
4 : $g_2 = \left[\alpha^2\right]_1$
5 : $\mathsf{msk}_{ra} = \left([1]_1, \alpha\right)$
6 : $\mathsf{pp}_{ra} = (g_2, \{h_i\}_{i=0}^n, [\alpha]_T, \mathsf{H})$
7 : **return** $(\mathsf{msk}_{ra}, \mathsf{pp}_{ra})$

$(\mathsf{pp}_{sa}, \mathsf{msk}_{sa}) \leftarrow \mathsf{SAgen}(\lambda, \mathsf{pp}_{ra}, \mathbf{R_L})$

1 : Parse $(\mathcal{BG}(\lambda), \mathsf{pp}_{ra})$
2 : $Y \leftarrow\$\, \mathbb{G}_2$
3 : $\mathsf{sk} := v \leftarrow\$\, \mathbb{Z}_\mathsf{p}$
4 : $\mathsf{vk} = g_2^v = \left[\alpha^2 v\right]_1$
5 : $(\vec{\mathsf{crs}}, \vec{\mathsf{ts}}) \leftarrow\$\, \mathcal{ZK}.\mathsf{K}_{\vec{\mathsf{crs}}}(\lambda, \mathbf{R_L})$
6 : $\mathsf{msk}_{sa} = (\mathsf{sk}, \vec{\mathsf{ts}})$
7 : $\mathsf{pp}_{sa} = (\mathbf{R_L}, \vec{\mathsf{crs}}, Y, \mathsf{vk})$
8 : **return** $(\mathsf{msk}_{sa}, \mathsf{pp}_{sa})$

$(\mathsf{dk}_\mathbb{B}) \leftarrow \mathsf{DecKGen}(\mathsf{msk}_{ra}, \mathbb{B})$

1 : Parse $(\mathcal{BG}(\lambda), \mathsf{msk}_{ra})$
2 : $Z_\mathbb{B}(x) = \prod_{i=1}^{n} (x - k_i)^{\overline{b[i]}}$
3 : $\mathsf{dk}_\mathbb{B} = [1/Z_\mathbb{B}(\alpha)]_1$
4 : **return** $(\mathsf{dk}_\mathbb{B})$

$(\mathsf{ek}_\mathbb{P}, \sigma, W) \leftarrow \mathsf{EncKGen}(\mathsf{pp}_{ra}, \mathsf{pp}_{sa}, \mathsf{msk}_{sa}, \mathbb{P}, \mathrm{P}_\mathrm{F})$

1 : Parse $(\mathcal{BG}(\lambda), \mathsf{pp}_{ra}, \mathsf{msk}_{sa})$
2 : $Z_\mathbb{P}(x) = \prod_{i=1}^{n} (x - k_i)^{\overline{p[i]}} = \sum_{j=0}^{n} z_i x^i$
3 : $\mathsf{ek}_\mathbb{P} = \mathsf{Col}(\mathsf{pp}, \mathbb{P}) = \prod_{i=0}^{n} h_{i+1}^{z_i} = [\alpha Z_\mathbb{P}(\alpha)]_2$
4 : $t_u \leftarrow\$\, \mathbb{Z}_p^*, W = [1/t_u]_2$
5 : $(R, S, T) = (g_2^{t_u}, \mathsf{ek}_\mathbb{P}^{\mathsf{sk}/t_u} Y^{1/t_u}, S^{\mathsf{sk}/t_u}\, [1/t_u]_2)$
6 : **return** $(\mathsf{ek}_\mathbb{P}, \sigma = (R, S, T), W)$

$(\pi, \mathsf{x}) \leftarrow \mathsf{Enc}\,(\mathsf{pp}_{sa}, \mathsf{pp}_{ra}, m, \mathsf{ek}_\mathbb{P}, \sigma, W)$

1 : Parse $(\mathcal{BG}(\lambda), \mathsf{pp}_{ra}, \mathsf{pp}_{sa})$
2 : $r, t \leftarrow\$\, \mathbb{Z}_\mathsf{p}^*$
3 : $(C, C_1, C_2) = (m\,[r\alpha]_T, \mathsf{ek}_\mathbb{P}^r, g_2^{-r})$
4 : $R' = R^{1/t},\ S' = S^t,\ T' = T^{t^2} W^{t(1-t)}$
5 : $\sigma' = (R', S', T')$
6 : $\mathsf{vk}' = \mathsf{vk}^{1/t},\ \mathsf{ek}'_\mathbb{P} = \mathsf{ek}_\mathbb{P}^t$
7 : $\mathsf{x} = (\sigma', \mathsf{vk}', \mathsf{ek}'_\mathbb{P}, \mathsf{Ct} = (\mathbb{P}, C, C_1, C_2))$
8 : $\mathsf{w} = (\mathsf{ek}_\mathbb{P}, \sigma, m, r, t)$
9 : $\pi \leftarrow \mathrm{PoK}\{(\mathsf{w}) : \mathbf{R_L}(\mathsf{x}, \mathsf{w}) = 1\}$
10 : **return** (π, x)

$(\tilde{\mathsf{Ct}}, \perp) \leftarrow \mathsf{San}(\mathsf{pp}_{sa}, \mathsf{pp}_{ra}, \pi, \mathsf{x})$

1 : Parse $(\mathcal{BG}(\lambda), \mathsf{pp}_{ra}, \mathsf{pp}_{sa})$
2 : **if** $\{R' \in \mathbb{G}_1 \wedge \mathsf{ek}'_\mathbb{P}, S', T' \in \mathbb{G}_2 \wedge$
3 : $R' \bullet S' = (\mathsf{vk}' \bullet \mathsf{ek}'_\mathbb{P})(g_2 \bullet Y) \wedge$
4 : $R' \bullet T' = (\mathsf{vk} \bullet S')(g_2 \bullet [1]_2) \wedge$
5 : $\mathcal{ZK}.\mathsf{V}(\mathbf{R_L}, \vec{\mathsf{crs}}, \pi, \mathsf{x}) = 1\}$:
6 : $s \leftarrow\$\, \mathbb{Z}_\mathsf{p}^*, \tilde{C} = C \cdot [s\alpha]_T$
7 : $\tilde{C}_1 = C_1 \cdot [s\alpha Z_\mathbb{P}(\alpha)]_2$
8 : $\tilde{C}_2 = C_2 \cdot g_2^{-s}$
9 : **return** $\tilde{\mathsf{Ct}} = (\mathbb{P}, \tilde{C}, \tilde{C}_1, \tilde{C}_2)$
10 : **else** : **abort**

$(m', \perp) \leftarrow \mathsf{Dec}\big(\mathsf{pp}_{sa}, \mathsf{pp}_{ra}, \tilde{\mathsf{Ct}}, \mathsf{dk}_\mathbb{B}\big)$

1 : Parse $(\mathcal{BG}(\lambda), \mathsf{pp}_{ra}, \mathsf{pp}_{sa})$
2 : **if** $\mathbb{P} \subseteq \mathbb{B}$:
3 : $c[i] = b[i] - p[i], F_{\mathbb{B},\mathbb{P}}(x) = \prod_{i=1}^{n} (x - k_i)^{c[i]} = \sum_{j=0}^{n} f_j x^j$
4 : **return** $m' = C \left(\left(C_2 \bullet \prod_{i=1}^{n} (h_{i-1})^{f_i} \right) \cdot (\mathsf{dk}_\mathbb{B} \bullet C_1) \right)^{-1/f_0}$
5 : **else** : **abort**

Fig. 3. The proposed CD-ABACE scheme

ABACE schemes. Note that we can build more universal circuit levels using the generic model discussed in Sect. 4.

Remark 5.2. However, while the proposed CD-ABACE scheme achieves a weak notion of receiver anonymity, it improves Wang and Chow's weak point where recipients' identities are public. In order to resolve this issue we can use the existing CP-ABE schemes with a more universal circuit level, but this compromises the efficiency. For instance, according to Garg et al. [21], we can fully anonymize the receiver using our generic construction based on multilinear maps and iO assumptions. In App. D we specify a CD-ABACE scheme using Waters's CP-ABE [41], which is defined under Linear Secret Sharing Schemes; we compare it with our proposed CD-ABACE scheme in Sect. 6.

6 Performance Analysis

In this section, we examine how the performance of our proposed fully-constant CD-ABACE scheme and a CD-ABACE variant of Waters's CP-ABE [41] (see App. D) compares to the selectively-secure scheme of Wang and Chow [39], which is the only implemented ACE construction to date.

We obtained the benchmarks for our proposed CD-ABACE scheme on Ubuntu 20.04.2 LTS with an Intel Core i7-9850H CPU @ 2.60 GHz with 16 GB of memory. We applied the Barreto-Naehrig (BN) curve, type F, $y^2 = x^3 + b$ over the field $\mathbb{F}_q$ of order p with embedding curve degree $k = 12$ and 1920-bit DLog security. For simplicity the bit-lengths of expressions of access policies and computations over $\mathbb{Z}_{\mathsf{p}}$ are not taken into account. We implemented the proposed construction using the Charm-Crypto framework [4], a Python library for Pairing-based Cryptography[2]. Fig. 4 consists of six graphs depicting the following relationships:

- *Total number of Attributes/Users versus RA Setup time*: The top left graph displays the relationship between the total number of attributes/users and time required to generate the parameter of the Receiver Authority. As can be seen, in our scheme and [39] scheme the time required to run this algorithm grows linearly with the total number of attributes/users, and for a generous consideration of 1000 attributes, it only requires $\sim$ 200 milliseconds (ms) and $\sim$ 300 ms, respectively. However, for an ACE variation of Waters' CP-ABE [41] construction (see App. D) this time is constant and less than 30 ms.

[2] `https://github.com/CDABACE`

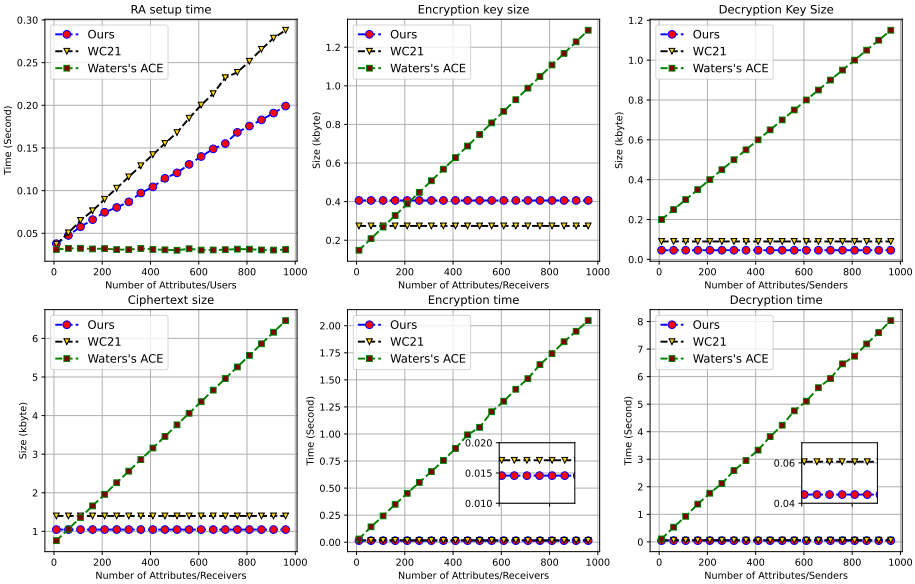

Fig. 4. Running time of attribute size dependence algorithms

- *Maximum number of Attributes/Receivers versus Encryption key size*: The top centre graph of Fig. 4 shows the relationship between the total number of attributes/receivers that a sender can send to them and the size of the stored encryption key. As can be seen, this relationship in Fig. 5 construction is linear, however the our proposed construction and [39] require a constant storage. Assuming 1000 attributes/receivers to be the highest number used by a sender, the required memory for storing this key for [39], Fig. 5 and our scheme is ~ 300, ~ 1200 and ~ 400 bytes, respectively.
- *Maximum number of Attributes/Senders versus Decryption key size*: The top right graph of Fig. 4 shows the relationship between maximum the number of attributes/senders for each receiver and the size of the decryption key. As can be seen, in Fig. 5 scheme this relationship grows linearly with number of attributes while in both our scheme and [39] the requires storage is constant independent of the number of attributes/senders; for instance, this size for a user having 1000 attributes/senders is equal to ~ 50, ~ 100 bytes, while Fig. 5 scheme is equal to ~ 1.2 KB.
- *Number of Attributes/Receivers versus ciphertext size*: The bottom left graph of Fig. 4 depicts the relationship between the total number of attributes/receivers in the policy and the length of ciphertext. As

can be seen, in Waters' ACE scheme this relationship is linear while our scheme and [39] achieve a constant ciphertext size. For instance, a ciphertext with 100 embedded attributes/receivers in the policy has a ciphertext of size ~ 1, ~ 1.4, ~ 7 KB in our scheme, [39] and Waters' ACE scheme.

- *Number of Attributes/Receivers versus Encryption time*: The bottom centre graph of Fig. 4 shows the relationship between the total number of attributes/receivers of in the embedded policy and the encryption time. As can be seen, the time required to encrypt a ciphertext in our scheme and [39] is constant, while in Waters's ACE variation it grows linearly with the total number of attributes. For example, a sender in Waters' ACE, [39] and our scheme requires ~ 2000, ~ 18, ~ 15 ms to encrypt a message with 1000 embedded attributes/receivers.
- *Number of Attributes/Senders versus Decryption time*: The bottom right graph of Fig. 4 shows the relationship between the maximum number of attributes/senders of each receiver and the decryption time. As can be seen, the time required to decrypt a ciphertext in Fig. 5's ACE grows linearly with the maximum number of attributes, while this overhead in our scheme and [39] is constant. For instance, a receiver in [41], [39] and our proposed construction requires ~ 8000, ~ 60, ~ 45 ms to decrypt a ciphertext with 1000 attributes in the policy.

Overall, our scheme has improved the receivers' key length and privacy level from identity-based to attribute-based. The ciphertext size has also been reduced, along with the number of public parameters. Since the second group generator is hidden in [39], the SA has to choose a new generator to create the SPS parameters. In contrast, the proposed variant of Abe et al.'s SPS [2] requires no new generator for the second cyclic group, and the intended NIZK proof cuts out the need for a target group proof of exponentiation.

7 Conclusion

In this work, we proposed a generic and efficient *Cross-Domain Attribute-Based Access Control Encryption* scheme based on attribute-based predicate functions. In comparison with earlier works, the length of the secret decryption keys and the ciphertext size has been substantially reduced to less than ~ 50 and ~ 1000 bytes as compared to Wang and Chow scheme where the size was ~ 100 and ~ 1400 bytes, respectively. Moreover, the computational overhead of encryption and decryption is linear in

the number of the policy attributes and user attributes, respectively. Also, it is formally proved that the proposed scheme satisfies the No-Read and the No-Write rules based on standard assumptions. We leave the construction of a CD-ABACE scheme based on a Boolean circuit instead of AND-gate circuits with the same performance as an interesting open problem. As we discussed, the main downside for AND-gate circuits is that the attribute sets in plain may reveal some meaningful information about the intended constraints and consequently, applying a Boolean circuit can result in stronger anonymity guarantees for the receivers.

Acknowledgements

We would like to thank Sherman S. M. Chow, Georg Fuchsbauer, Karim Baghery, Ward Beullens, Pavel Hubáček and anonymous reviewers for their helpful discussions and valuable comments. This work was supported by Flanders Innovation & Entrepreneurship through the Spearhead Cluster Flux50 ICON project PrivateFlex. In addition, this work was supported in part by the Research Council KU Leuven C1 on Security and Privacy for Cyber-Physical Systems and the Internet of Things with contract number C16/15/058 and by CyberSecurity Research Flanders with reference number VR20192203.

References

1. Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Structure-preserving signatures and commitments to group elements. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 209–236. Springer, Heidelberg, August 2010.
2. Masayuki Abe, Jens Groth, Miyako Ohkubo, and Mehdi Tibouchi. Unified, minimal and selectively randomizable structure-preserving signatures. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 688–712. Springer, Heidelberg, February 2014.
3. Seyed Farhad Aghili, Mahdi Sedaghat, Dave Singelée, and Maanak Gupta. MLS-ABAC: Efficient Multi-Level Security Attribute-Based Access Control scheme. *Future Generation Computer Systems*, 131:75–90, 2022.
4. Joseph A. Akinyele, Christina Garman, Ian Miers, Matthew W. Pagano, Michael Rushanan, Matthew Green, and Aviel D. Rubin. Charm: a framework for rapidly prototyping cryptosystems. *Journal of Cryptographic Engineering*, 3(2):111–128, June 2013.
5. Nuttapong Attrapadung and Benoît Libert. Functional encryption for inner product: Achieving constant-size ciphertexts with adaptive security or support for negation. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010*, volume 6056 of *LNCS*, pages 384–402. Springer, Heidelberg, May 2010.
6. Amos et al. Beimel. *Secure schemes for secret sharing and key distribution.* Technion-Israel Institute of technology, Faculty of computer science, 1996.

7. D Elliott Bell and Leonard J LaPadula. Secure computer systems: Mathematical foundations. Technical report, DTIC document, 1973.
8. John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *2007 IEEE Symposium on Security and Privacy*, pages 321–334. IEEE Computer Society Press, May 2007.
9. Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *20th ACM STOC*, pages 103–112. ACM Press, May 1988.
10. Dan Boneh and Xavier Boyen. Efficient selective identity-based encryption without random oracles. *Journal of Cryptology*, 24(4):659–693, October 2011.
11. Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 440–456. Springer, Heidelberg, May 2005.
12. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, Heidelberg, August 2001.
13. Michael Brengel and Christian Rossow. Identifying key leakage of bitcoin users. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 623–643. Springer, 2018.
14. Ling Cheung and Calvin C. Newport. Provably secure ciphertext policy ABE. In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *ACM CCS 2007*, pages 456–465. ACM Press, October 2007.
15. Ivan Damgård, Helene Haagh, and Claudio Orlandi. Access control encryption: Enforcing information flow with cryptography. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 547–576. Springer, Heidelberg, October / November 2016.
16. Cécile Delerablée and David Pointcheval. Dynamic threshold public-key encryption. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 317–334. Springer, Heidelberg, August 2008.
17. Keita Emura, Atsuko Miyaji, Akito Nomura, Kazumasa Omote, and Masakazu Soshi. A ciphertext-policy attribute-based encryption scheme with constant ciphertext length. In *International Conference on Information Security Practice and Experience*, pages 13–23. Springer, 2009.
18. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Heidelberg, August 1987.
19. Georg Fuchsbauer, Romain Gay, Lucas Kowalczyk, and Claudio Orlandi. Access control encryption for equality, comparison, and more. In Serge Fehr, editor, *PKC 2017, Part II*, volume 10175 of *LNCS*, pages 88–118. Springer, Heidelberg, March 2017.
20. Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, October 2013.
21. Sanjam Garg, Craig Gentry, Shai Halevi, Amit Sahai, and Brent Waters. Attribute-based encryption for circuits from multilinear maps. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 479–499. Springer, Heidelberg, August 2013.
22. Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

23. Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 89–98. ACM Press, October / November 2006. Available as Cryptology ePrint Archive Report 2006/309.
24. Fuchun Guo, Yi Mu, Willy Susilo, Duncan S Wong, and Vijay Varadharajan. CP-ABE with constant-size keys for lightweight devices. *IEEE Transactions on Information Forensics and Security, volume=9, number=5, pages=763–771, year=2014, publisher=IEEE.*
25. Hanshu Hong and Zhixin Sun. An efficient and secure attribute based signcryption scheme with LSSS access structure. *SpringerPlus*, 5(1):644, 2016.
26. Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 146–162. Springer, Heidelberg, April 2008.
27. Sam Kim and David J. Wu. Access control encryption for general policies from standard assumptions. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 471–501. Springer, Heidelberg, December 2017.
28. Allison B. Lewko and Brent Waters. Decentralizing attribute-based encryption. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 568–588. Springer, Heidelberg, May 2011.
29. Tatsuaki Okamoto and Katsuyuki Takashima. Adaptively attribute-hiding (hierarchical) inner product encryption. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 591–608. Springer, Heidelberg, April 2012.
30. Sylvia Osborn, Ravi Sandhu, and Qamar Munawer. Configuring role-based access control to enforce mandatory and discretionary access control policies. *ACM Transactions on Information and System Security (TISSEC)*, 3(2):85–106, 2000.
31. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.
32. Andrei Sabelfeld and Andrew C Myers. Language-based information-flow security. *IEEE Journal on selected areas in communications*, 21(1):5–19, 2003.
33. Amit Sahai and Brent R. Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, Heidelberg, May 2005.
34. Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 239–252. Springer, Heidelberg, August 1990.
35. Seyyed Mahdi Sedaghat, Mohammad Hassan Ameri, Javad Mohajeri, and Mohammad Reza Aref. An efficient and secure data sharing in smart grid: Ciphertext-policy attribute-based signcryption. In *2017 Iranian Conference on Electrical Engineering (ICEE)*, pages 2003–2008. IEEE, 2017.
36. Siamak Fayyaz Shahandashti and Reihaneh Safavi-Naini. Threshold attribute-based signatures and their application to anonymous credential systems. In Bart Preneel, editor, *AFRICACRYPT 09*, volume 5580 of *LNCS*, pages 198–216. Springer, Heidelberg, June 2009.
37. Gaosheng Tan, Rui Zhang, Hui Ma, and Yang Tao. Access control encryption based on LWE. In *Proceedings of the 4th ACM International Workshop on ASIA Public-Key Cryptography*, pages 43–50. ACM, 2017.

38. Phuong Viet Xuan Tran, Thuc Nguyen Dinh, and Atsuko Miyaji. Efficient ciphertext-policy ABE with constant ciphertext length. In *2012 7th International Conference on Computing and Convergence Technology (ICCCT)*, pages 543–549. IEEE, 2012.
39. X. Wang and S. M. Chow. Cross-Domain Access Control Encryption: Arbitrary-Policy, Constant-Size, Efficient. In *2021 2021 IEEE Symposium on Security and Privacy (SP)*, pages 388–401, Los Alamitos, CA, USA, may 2021. IEEE Computer Society.
40. Xiuhua Wang, Harry W. H. Wong, and Sherman S. M. Chow. Access control encryption from group encryption. In *Applied Cryptography and Network Security*, pages 417–441, Cham, 2021. Springer International Publishing.
41. Brent Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *PKC 2011*, volume 6571 of *LNCS*, pages 53–70. Springer, Heidelberg, March 2011.

A Omitted Definitions

A.1 Multi-Sequence of Exponents Diffie-Hellman

The following definition is proposed by [16] as a general Diffie-Hellman exponent theorem [11] for an asymmetric bilinear group. This definition is non-interactive and falsifiable. It is also demonstrated to hold for the generic group model similar to the BDH, q-BDHI and $(l,m,t)-\texttt{MSE-DDH}$ assumptions.

Definition A.1 (Multi-Sequence of Exponents Diffie-Hellman ($(l,m,t)-\texttt{MSE-DDH}$) assumption [16]). *For security parameter λ, consider an asymmetric bilinear group generator $\mathcal{BG}(\lambda)=(\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,\mathsf{p},\hat{e})$. For three given integers l,m,t, consider two univariate composite polynomials f and h of degree l and m that vanish on pairwise distinct points $\vec{x}=(x_1,\ldots,x_l)$ and $\vec{y}=(y_1,\ldots,y_m)$, respectively. For randomly chosen integers $\alpha,\delta,k\leftarrow\$\,\mathbb{Z}_p^*$, the $(l,m,t)-$ MSE-DDH assumption states that no PPT adversary $\mathcal{A}$ can distinguish between $\Gamma=[kf(\alpha)]_T$ and a random element $\Gamma\leftarrow\$\,\mathbb{G}_T$ with a non-negligible advantage, when given:*

$$\begin{aligned}
\vec{v_1} &= \left([1]_1,[\alpha]_1,\left[\alpha^2\right]_1,\ldots,\left[\alpha^{l+t-2}\right]_1,[k\alpha f(\alpha)]_1\right)\\
\vec{v_2} &= \left([\delta]_1,[\delta\alpha]_1,\left[\delta\alpha^2\right]_1,\ldots,\left[\delta\alpha^{l+t}\right]_1\right)\\
\vec{v_3} &= \left([1]_2,[\alpha]_2,\left[\alpha^2\right]_2,\ldots,\left[\alpha^{m-2}\right]_2\right)\\
\vec{v_4} &= \left([\delta]_2,[\delta\alpha]_2,\left[\delta\alpha^2\right]_2,\ldots,\left[\delta\alpha^{2m-1}\right]_2,[kh(\alpha)]_2\right)\quad .
\end{aligned}$$

The adversary $\mathcal{A}$ can solve the $(l,m,t)-$ MSE-DDH assumption with the advantage of:

$$\left| \Pr\left[\mathcal{A}^{\textit{MSE-DDH}}(\vec{x},\vec{y},\vec{v}_{1-4},\Gamma=[kf(\alpha)]_T)=1\right] - \Pr\left[\mathcal{A}^{\textit{MSE-DDH}}(\vec{x},\vec{y},\vec{v}_{1-4},\Gamma\leftarrow\!\!\$\,\mathbb{G}_T)=1\right]\right| \leq \mathsf{negl}(\lambda) \ ,$$

where $\vec{v}_{1-4}$ denotes the four vectors $\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{v}_4$.

B Main Building Blocks

In this section, we formally define the cryptographical primitives needed for the proposed construction and their security requirements.

B.1 Structure-Preserving Signatures

In a Structure-Preserving Signature (SPS) [1], the signature and signed message are both group elements; the verification requires a pairing-product process.

Definition B.1 (Structure-Preserving Signatures [1]). *A Structure-Preserving Signature (SPS) scheme $\Psi_{\mathcal{SPS}}$ in a type-III bilinear group, over message space $\mathcal{M}$ and signature space $\mathcal{S}$ consists of five PPT algorithms* $(\mathsf{Pgen}, \mathsf{KG}, \mathsf{Sign}, \mathsf{Randz}, \mathsf{Vf})$*, defined as follows:*

- $(\mathsf{pp}) \leftarrow \mathcal{SPS}.\mathsf{Pgen}(\lambda)$*: This algorithm takes the security parameter λ as input, and generates the public parameters* pp*.*
- $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathcal{SPS}.\mathsf{KG}(\mathsf{pp})$*: Key generation is a probabilistic algorithm which takes the public parameters* pp *as input. It returns a key-pair* $(\mathsf{sk}, \mathsf{vk})$ *composed of the secret signing key and the public verification key.*
- $(\sigma, W) \leftarrow \mathcal{SPS}.\mathsf{Sign}(\mathsf{pp}, \mathsf{sk}, m)$*: The signing algorithm takes the public parameters* pp*, the secret signing key* sk *and a message $m \in \mathcal{M}$ as inputs and outputs a signature $\sigma \in \mathcal{S}$ along with a re-randomizing token W.*
- $(\sigma', W') \leftarrow \mathcal{SPS}.\mathsf{Randz}(\mathsf{pp}, \sigma)$*: The re-randomizing algorithm takes the public parameters* pp*, a signature σ and a token W as inputs, returns a re-randomized signature $\sigma' \in \mathcal{S}$ and generates a new token W'.*

- $(0,1) \leftarrow \mathcal{SPS}.\mathsf{Vf}(\mathsf{pp}, \mathsf{vk}, \sigma, m)$*: The deterministic verification algorithm takes the public parameters* pp*, a signature* σ*, the message* $m \in \mathcal{M}$ *and a public verification key* vk *as inputs. It responds by either 0 (reject) or 1 (accept).*

The primary security requirements for an SPS scheme are *Correctness* and *Existential Unforgeability under Chosen Message Attack* (EUF-CMA) that are defined as follows:

Definition B.2 (Correctness). *An SPS scheme* $\Psi_{\mathcal{SPS}}$ *is called* correct *if,*

$$\Pr\begin{bmatrix}(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KG}(\mathsf{pp}), \forall\ m \in \mathcal{M}, \\ \mathsf{Vf}\left(\mathsf{pp}, \mathsf{vk}, m, \mathsf{Sign}(\mathsf{pp}, \mathsf{sk}, m)\right) = 1\end{bmatrix} \approx_c 1\ .$$

Definition B.3 (Existential Unforgeability under Chosen Message Attack (EUF-CMA**)).** *An SPS scheme* $\Psi_{\mathcal{SPS}}$ *is called an* EUF-CMA*-secure scheme if for all PPT adversaries* $\mathcal{A}$*,* $Adv^{\text{EUF-CMA}}_{\mathcal{A},\mathcal{SPS}}(1^\lambda)$*, we have the following advantage function,*

$$\Pr\begin{bmatrix}(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KGen}(\mathsf{pp}), (\sigma^*, m^*) \leftarrow\!\$\ \mathcal{A}^{\mathcal{O}_{\mathsf{Sign}}}(\mathsf{pp}) : \\ m^* \notin \mathcal{Q}_{msg}\ \wedge \mathsf{Vf}(\mathsf{vk}, \sigma^*, m^*) = 1\end{bmatrix}\ .$$

The signature oracle $\mathcal{O}_{\mathsf{Sign}}$ *takes a message* $m \in \mathcal{M}$ *and returns the corresponding signature by running the* $\mathsf{Sign}(\mathsf{pp}, \mathsf{sk}, m)$ *algorithm. All the queried messages are kept track of via a query set* $\mathcal{Q}_{msg}$*. An SPS is called an* EUF-CMA*-secure if for all PPT adversaries we have,* $Adv^{\text{EUF-CMA}}_{\mathcal{A},\mathcal{SPS}}(1^\lambda) \approx_c 0$*.*

In the following, the Abe et al. [2] SPS construction is outlined, as a selectively re-randomizable SPS. This SPS behaves as a unified type, which means that it can be instantiated under either type of bilinear group. This scheme has been proven to be EUF-CMA-secure. A valid re-randomization token enables one to re-randomize the signature without needing to know the secret signing key. The scheme consists of the following algorithms:

- $(\mathsf{pp}) \leftarrow \mathcal{SPS}.\mathsf{Pgen}(\lambda)$: This algorithm takes as input the security parameter λ, picks $X \leftarrow\!\$\ \mathbb{G}_1$, and runs a Type-III bilinear group generator $\mathcal{BG}(\lambda) = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathsf{p}, \hat{e})$. It returns the public parameters of the system $\mathsf{pp} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathsf{p}, \hat{e}, X)$.
- $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathcal{SPS}.\mathsf{KG}(\mathsf{pp})$: The key generation algorithm takes as input pp, picks $v \leftarrow\!\$\ \mathbb{Z}_p$ and computes $V = [v]_2$. It returns the public verification key $\mathsf{vk} = V$ and the secret signing key $\mathsf{sk} = v$.

- $(\sigma, W) \leftarrow \mathcal{SPS}.\mathsf{Sign}(\mathsf{pp}, \mathsf{sk}, m)$: The signing algorithm takes as inputs the public parameters pp, the secret signing key sk and a message $m \in \mathbb{G}_1$. It samples $r \leftarrow\!\$\, \mathbb{Z}_p^*$ and computes $\sigma = (R, S, T) = \left([r]_2, m^{v/r} X^{1/r}, S^{v/r}\, [1/r]_1\right)$. It outputs the pair of $(\sigma, W = [1/r]_1)$, where W is a token for re-randomizing the signature.
- $(\sigma', W') \leftarrow \mathcal{SPS}.\mathsf{Randz}(\mathsf{pp}, \sigma, W)$: The re-randomizing algorithm takes as inputs pp, signature $\sigma \in \mathcal{S}$ along with a token W, picks a random integer $t \leftarrow\!\$\, \mathbb{Z}_p^*$ and computes a re-randomized signature as $\sigma' = (R', S', T') = (R^{1/t}, S^t, T^{t^2} W^{t(1-t)})$. It returns σ' along with a new token $W' = W^t$ as the outputs.
- $(0, 1) \leftarrow \mathcal{SPS}.\mathsf{Vf}(\mathsf{pp}, \mathsf{vk}, \sigma', m)$: The verification algorithm takes as inputs the public parameters pp, a re-randomized signature σ', the message $m \in \mathcal{M}$ and the verification key vk. It first checks $m, S', T' \in \mathbb{G}_1$, $R' \in \mathbb{G}_2$ and whether the pairing equations $S' \bullet R' = (m \bullet V)(X \bullet [1]_2)$, $T' \bullet R' = (S' \bullet V)([1]_1 \bullet [1]_2)$ hold or not. If both conditions hold then it returns 1, otherwise it responds with 0.

The correctness of the scheme is trivial and the re-randomized signature is perfectly indistinguishable from the original signature.

B.2 Non-Interactive Zero-Knowledge proofs

A Zero-Knowledge proof as a two-party protocol is a fundamental and powerful cryptographic tool. It allows a prover to convince a verifier about the validity of a statement without revealing any other information. Non-Interactive Zero-Knowledge (NIZK) proofs remove the interaction between the parties in two settings: either the Random Oracle Model (ROM) [18] or the Common Reference String (CRS) model [9]. Since the construction of NIZK proofs in the CRS model requires a trusted setup phase that outputs some public parameters, known as the CRS, that are shared with the prover and verifier to respectively generate and verify the proof in a single communication round we use them in ROM.

For a security parameter λ, let $\mathcal{R}$ be a relation generator, such that $\mathcal{R}(1^\lambda)$ returns an efficiently computable binary relation $\mathbf{R_L} = \{(\mathsf{x}, \mathsf{w})\}$, where x is the instance and w is the corresponding witness. Let $\mathbf{L} = \{\mathsf{x} : \exists\, \mathsf{w} \mid (\mathsf{x}, \mathsf{w}) \in \mathbf{R}\}$ be the **NP**-language consisting of the statements in the relation $\mathbf{R_L}$. Formally, a NIZK proof Ψ_{NIZK} under the relation generator $\mathcal{R}$ consists of the following PPT algorithms:

- $(\vec{\mathsf{crs}}, \vec{\mathsf{ts}}) \leftarrow \mathcal{ZK}.\mathsf{K}_{\vec{\mathsf{crs}}}(\mathbf{R_L})$: The CRS generator is a probabilistic algorithm that, given relation $(\mathbf{R_L})$, first samples the simulation trapdoor

$\vec{\mathsf{ts}}$, and generates $\vec{\mathsf{crs}}$. It securely stores the former while publishing the latter.

- $(\pi, \perp) \leftarrow \mathcal{ZK}.\mathsf{P}(\mathbf{R_L}, \vec{\mathsf{crs}}, \mathsf{x}, \mathsf{w})$: Prove is a probabilistic algorithm that takes as input $(\mathbf{R_L}, \vec{\mathsf{crs}}, \mathsf{x}, \mathsf{w})$ and if $(\mathsf{x}, \mathsf{w}) \in \mathbf{R_L}$, outputs a proof π, otherwise, it returns $\perp$.
- $(0, 1) \leftarrow \mathcal{ZK}.\mathsf{V}(\mathbf{R_L}, \vec{\mathsf{crs}}, \mathsf{x}, \pi)$: The verification algorithm is a deterministic process that returns 1 if the given proof is correct ($(\mathsf{x}, \mathsf{w}) \in \mathbf{R_L}$) and 0 if it is incorrect ($(\mathsf{x}, \mathsf{w}) \notin \mathbf{R_L}$).
- $(\pi') \leftarrow \mathcal{ZK}.\mathsf{Sim}(\mathbf{R_L}, \vec{\mathsf{crs}}, \vec{\mathsf{ts}}, \mathsf{x})$: The simulator is an algorithm, that given the tuple $(\mathbf{R_L}, \vec{\mathsf{crs}}, \vec{\mathsf{ts}}, \mathsf{x})$, outputs a simulated proof π' without knowing the witness. It is computationally infeasible for a PPT adversary to distinguish between π and π'.

Next we recall the security requirements for a NIZK proof.

Definition B.4 (Completeness). *A NIZK proof Ψ_{NIZK} is called* complete*, if for all λ, and $(\mathsf{x}, \mathsf{w}) \in \mathbf{R_L}$ we have,*

$$\Pr\left[\begin{matrix}(\mathbf{R_L}) \leftarrow \mathcal{R}(1^\lambda), (\vec{\mathsf{crs}}, \vec{\mathsf{ts}}) \leftarrow \mathsf{K}_{\vec{\mathsf{crs}}}(\mathbf{R_L}) : \\ \mathsf{V}(\mathbf{R_L}, \vec{\mathsf{crs}}, \mathsf{x}, \mathsf{P}(\mathbf{R_L}, \vec{\mathsf{crs}}, \mathsf{x}, \mathsf{w})) = 1\end{matrix}\right] \approx_c 1 \ .$$

Definition B.5 (Soundness). *A NIZK proof Ψ_{NIZK} is called* Sound*, if for all adversaries $\mathcal{A}$, we have,*

$$\Pr\left[\begin{matrix}(\mathbf{R_L}) \leftarrow \mathcal{R}(1^\lambda), (\vec{\mathsf{crs}}, \vec{\mathsf{ts}}) \leftarrow \mathsf{K}_{\vec{\mathsf{crs}}}(\mathbf{R_L}), \\ (\mathsf{x}, \pi) \leftarrow \mathcal{A}(\mathbf{R_L}, \vec{\mathsf{crs}}) : \mathsf{V}(\mathbf{R_L}, \vec{\mathsf{crs}}, \mathsf{x}, \pi) = 1 \ \wedge \ \mathsf{x} \notin \mathbf{L}\end{matrix}\right] \approx_c 0 \ .$$

Definition B.6 (Statistically Zero-Knowledge). *A NIZK proof Ψ_{NIZK} is called* statistically Zero-Knowledge*, if for all adversary $\mathcal{A}$, $\varepsilon_0^{unb} \approx_c \varepsilon_1^{unb}$, where,*

$$\varepsilon_b^{unb} = \Pr\left[(\vec{\mathsf{crs}} \,\|\, \vec{\mathsf{ts}}) \leftarrow \mathsf{K}_{\vec{\mathsf{crs}}}(\mathbf{R_L}) : \mathcal{A}^{\mathcal{O}_b(\cdot,\cdot)}(\mathbf{R_L}, \vec{\mathsf{crs}}) = 1\right] \ .$$

Here, the oracle $\mathcal{O}_0(\mathsf{x}, \mathsf{w})$ returns $\perp$ (reject) if $(\mathsf{x}, \mathsf{w}) \notin \mathbf{R_L}$, and else it returns $\mathsf{P}(\mathbf{R_L}, \vec{\mathsf{crs}}, \mathsf{x}, \mathsf{w})$. Similarly, $\mathcal{O}_1(\mathsf{x}, \mathsf{w})$ returns $\perp$ (reject) if $(\mathsf{x}, \mathsf{w}) \notin \mathbf{R_L}$, else it returns $\mathsf{Sim}(\mathbf{R_L}, \vec{\mathsf{crs}}, \mathsf{x}, \vec{\mathsf{ts}})$.

Intuitively, a NIZK proof Ψ_{NIZK} is zero-knowledge if it does not leak extra information beyond the validity of the statement. Now we recall the definitions of *Knowledge Soundness* as a stronger notion of *Soundness*.

Definition B.7 (Computational Knowledge-Soundness). *A NIZK proof Ψ_{NIZK} is computationally (adaptively)* knowledge-sound*, if for every PPT adversary $\mathcal{A}$, there exists an extraction trapdoor $\vec{\mathsf{te}}$ and an extractor $\mathsf{Ext}_{\mathcal{A}}$, s.t. for all λ we have,*

$$\Pr\left[\begin{matrix}(\mathbf{R_L}) \leftarrow \mathcal{R}(1^\lambda), (\vec{\mathsf{crs}} \,\|\, \vec{\mathsf{te}}) \leftarrow \mathsf{K}_{\vec{\mathsf{crs}}}(\mathbf{R_L}), (\mathsf{x}, \pi) \leftarrow \mathcal{A}(\mathbf{R_L}, \vec{\mathsf{crs}}), \\ (\mathsf{w}) \leftarrow \mathsf{Ext}_{\mathcal{A}}(\mathbf{R_L}, \vec{\mathsf{crs}}, \vec{\mathsf{te}}, \pi) : (\mathsf{x}, \mathsf{w}) \notin \mathbf{R_L} \wedge \mathsf{V}(\mathbf{R_L}, \vec{\mathsf{crs}}, \mathsf{x}, \pi) = 1\end{matrix}\right] \approx_c 0 \ .$$

B.3 Re-randomizable CP-ABE schemes

Next, we capture a unified definition of Ciphertext-Policy Attribute-Based Encryption (CP-ABE) schemes and their security requirements. Then, we recall a re-Randomizable CP-ABE scheme, in which one party can re-randomize a ciphertext without needing the secret key.

Definition B.8. *(Ciphertext-Policy Attribute-Based Encryption schemes [8]): For a given attribute universe $\mathbb{U}$ with size n, let Σ_c and $\Sigma_k = 2^{\mathbb{U}}$ be any collection of access structures and key indices over the attribute space $\mathbb{U}$, respectively. A CP-ABE scheme for a Boolean function* $\textsc{Bf} : \Sigma_k \times \Sigma_c \to \{0,1\}$ *over message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$ consists of the following algorithms:*

- $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathcal{ABE}.\mathsf{Pgen}(\lambda, \mathbb{U})$*: The parameter generation algorithm takes the security parameter λ and attribute space $\mathbb{U}$ as inputs and outputs the public parameters* pp *and the master secret key* msk*.*
- $(\mathsf{dk}_{\mathbb{B}}) \leftarrow \mathcal{ABE}.\mathsf{KGen}(\mathsf{msk}, \mathbb{B})$*: The key generation algorithm takes the master secret key* msk *and an authorized key index $\mathbb{B} \in \Sigma_k$ as inputs and returns* $\mathsf{dk}_{\mathbb{B}}$*.*
- $(\mathtt{Ct}) \leftarrow \mathcal{ABE}.\mathsf{Enc}(\mathsf{pp}, m, \mathbb{P})$*: The Encryption algorithm takes the public parameters* pp*, a message $m \in \mathcal{M}$ and a ciphertext index $\mathbb{P} \in \Sigma_c$ as inputs. It returns a ciphertext $\mathtt{Ct} \in \mathcal{C}$ along with the access structure $\mathbb{P}$.*
- $(m', \perp) \leftarrow \mathcal{ABE}.\mathsf{Dec}(\mathsf{pp}, \mathtt{Ct}, \mathsf{dk}_{\mathbb{B}}, \mathbb{B}, \mathbb{P})$*: The decryption algorithm takes the public parameters* pp*, a ciphertext $\mathtt{Ct} \in \mathcal{C}$ and its corresponding collection $\mathbb{P} \in \Sigma_c$ along with a private decryption key $\mathsf{dk}_{\mathbb{B}}$ for the key index $\mathbb{B} \in \Sigma_k$ as inputs. It responds with $m' \in \mathcal{M}$ iff $\textsc{Bf}(\mathbb{B}, \mathbb{P}) = 1$, otherwise $\perp$.*

We give a standard definition of the security properties for CP-ABE schemes namely, Correctness and Indistinguishability under Chosen Plaintext Attack (IND-CPA) as follows:

Definition B.9 (Correctness [23]). *A CP-ABE scheme Ψ_{CP-ABE}, over message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$ is called correct iff for all $m \in \mathcal{M}$ and $\mathtt{Ct} \in \mathcal{C}$ we have:*

$$\Pr\left[\begin{matrix}(\mathsf{pp},\mathsf{msk}) \leftarrow \mathsf{Pgen}(\lambda), (\mathsf{dk}_{\mathbb{B}}) \leftarrow \mathsf{KGen}(\mathsf{msk},\mathbb{B}),\\ \mathsf{Dec}\left(\mathsf{dk}_{\mathbb{B}}, \mathsf{Enc}(\mathsf{pp}, m, \mathbb{P}), \mathbb{P}\right) = m : \textsc{Bf}(\mathbb{B},\mathbb{P}) = 1\end{matrix}\right] \approx_c 1 \ .$$

Definition B.10 (Indistinguishability under Chosen Plaintext Attack (IND-CPA) [23]). *Let Ψ_{CP-ABE} be defined for the attribute universe $\mathbb{U}$, message space $\mathcal{M}$ and a Boolean relation $\textsc{Bf} : 2^{\mathbb{U}} \times \Sigma_c \to \{0,1\}$. For a security parameter λ and a PPT adversary $\mathcal{A}$, we define* IND-CPA *as follows:*

Initialization: *The Challenger samples the pair of public parameters and the master secret key by running the algorithm $(\mathsf{pp},\mathsf{msk}) \leftarrow \mathsf{Pgen}(\lambda,\mathbb{U})$ and gives pp to $\mathcal{A}$, while keeping msk secure.*

1^{st} ***Query Phase:*** *On a polynomially bounded number of requests, the adversary $\mathcal{A}$ chooses a key index $\mathbb{B} \in \Sigma_k$ and queries the key generation oracle. The challenger executes $\mathsf{KGen}(\mathsf{msk},\mathbb{B})$ and returns $\mathsf{dk}_{\mathbb{B}}$.*

Challenge: *$\mathcal{A}$ selects two messages of the same length $(m_0, m_1) \leftarrow\!\!\$\, \mathcal{M} \times \mathcal{M}$ and a challenge ciphertext index $\mathbb{P}^*$ such that $\textsc{Bf}(\mathbb{B},\mathbb{P}^*) = 0$ for all queried key indices in the first query phase. Then $\mathcal{B}$ flips a fair coin, produces a random bit $b \leftarrow\!\!\$\, \{0,1\}$, runs $\mathsf{Enc}(\mathsf{pp}, m_b, \mathbb{P}^*)$ and sends $\mathtt{Ct}^*$ back to $\mathcal{A}$.*

2^{nd} ***Query Phase:*** *After receiving the challenge ciphertext, $\mathcal{A}$ still is allowed to request more decryption keys for key indices $\mathbb{B}$ with the limitation $\textsc{Bf}(\mathbb{B},\mathbb{P}^*) = 0$.*

Guess. *$\mathcal{A}$ returns a bit b' to $\mathcal{B}$. The advantage of $\mathcal{A}$ is $Adv^{\text{IND-CPA}}_{\mathcal{A},\Psi_{CP-ABE}}(1^\lambda, b) = 2\left|\Pr\left[b = b'\right] - \frac{1}{2}\right|$, where the probability is taken over all coin flips. We say Ψ_{CP-ABE} is* IND-CPA *if for all PPT adversaries $\mathcal{A}$ we have,*

$$\left| Adv^{\text{IND-CPA}}_{\mathcal{A},\Psi_{CP-ABE}}(1^\lambda, b=0) - Adv^{\text{IND-CPA}}_{\mathcal{A},\Psi_{CP-ABE}}(1^\lambda, b=1) \right| \approx_c 0 \ .$$

Remark B.1. To be more concrete, we say a CP-ABE scheme is adaptively secure if, for each request, the adversary $\mathcal{A}$ can query the key generation algorithm such that its queries may depend on the information it gathered in its previous requests. In a selective secure CP-ABE as a weaker security notion [10], $\mathcal{A}$ should select the challenge access policy $\mathbb{P}^*$ before the initialization phase, while the decryption key queries can be still adaptive. We call a CP-ABE scheme co-selective IND-CPA secure [5], if

$\mathcal{A}$ declares q decryption key queries before the initialization phase, but she can adaptively select the challenge index $\mathbb{P}^*$ afterward. In this paper, we prove the security of the proposed CP-ABE construction based on the co-selective notion.

For a given CP-ABE scheme, we can define an optional sub-algorithm driven by the Encryption algorithm as follows:

- $(\mathsf{ek}_{\mathbb{P}}) \leftarrow \mathcal{ABE}.\mathsf{Col}(\mathsf{pp}, \mathbb{P})$: The Collector algorithm takes the public parameters pp and a ciphertext index $\mathbb{P} \in \Sigma_c$ as inputs. It returns a public aggregated value $\mathsf{ek}_{\mathbb{P}}$ for the ciphertext index $\mathbb{P}$.

Definition B.11 (Re-randomizable CP-ABE (rCP-ABE)). *For a given attribute universe $\mathbb{U}$ with size n, let Σ_c be any collection of access structures over the attribute space $\mathbb{U}$ and Σ_k be the key index set. A re-randomizable CP-ABE scheme, $\Psi_{r\mathcal{ABE}}$, for a Boolean relation* $\textsc{Bf} : \Sigma_k \times \Sigma_c \rightarrow \{0,1\}$*, over message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$, coincides with the algorithms from Definition B.8; the following algorithm supports this expansion:*

- $(\tilde{\mathsf{Ct}}) \leftarrow r\mathcal{ABE}.\mathsf{Randz}(\mathsf{pp}, \mathsf{Ct}, \mathbb{P})$*: The Re-randomization algorithm takes the public parameters* pp *and a valid ciphertext* Ct *under the access structure $\mathbb{P} \in \Sigma_c$ as inputs. It returns a re-randomized ciphertext $\tilde{\mathsf{Ct}} \in \mathcal{C}$ based on internal randomness without requiring any secret information.*

The Correctness and IND-CPA security of a Re-randomizable CP-ABE derives naturally from the initial CP-ABE, specified in Definitions B.9 and B.10. The decryption functions in the former can thus accept either a ciphertext Ct or a re-randomized ciphertext $\tilde{\mathsf{Ct}} \in \mathcal{C}$, but they both yield the same output parameters. A re-randomizable CP-ABE scheme also guarantees that no PPT adversary $\mathcal{A}$ can distinguish between $\tilde{\mathsf{Ct}}$ and Ct.

C Proofs

C.1 Proof of Theorem 4.1

Proof. Our evaluation of the correctness of the scheme occurs in two phases. We claim that Sanitizer confirms a sender with a valid and signed secret encryption key for attribute set $\mathbb{P}$ to transmit data to a group of receivers with attribute set $\mathbb{B}$ so that they satisfy it if $\textsc{Pf}(\mathbb{B}, \mathbb{P}) = 1$. Moreover, a target recipient with a private decryption key $\mathsf{dk}_{\mathbb{B}}$ can decrypt

the message entirely. The former relies on two properties including the correctness of the SPS construction of Definition B.2 as well as the completeness of the NIZK proof of Definition B.4. The latter also comes from the consistency of the CP-ABE scheme and consequently, the correctness of its re-randomizable variant. Thus we can conclude the proposed generic CD-ABACE scheme is correct. □

C.2 Proof of Theorem 4.2

Proof. We wish to prove that for all PPT adversaries $\mathcal{A}$, no player can distinguish between two possible scenarios in the No-Read security game: the case $b = 0$ constitutes one scenario denoted by H_0, and the case that b is fixed to 1, called H_1, i.e., $(\pi_0, \mathsf{x}_0) \approx_c (\pi_1, \mathsf{x}_1)$. We do so by defining several hybrid experiments and by demonstrating that each of them is computationally indistinguishable from the previous one.

- H_0^1**:** In this game, we modify H_0 by creating the challenge NIZK proof π_0 and running $\pi_0' \leftarrow \mathsf{Sim}(\vec{\mathsf{crs}}, \vec{\mathsf{ts}}, \mathsf{x}_0)$.

The Zero-Knowledge property of the NIZK proofs (cf. Definition B.6) guarantees that this experiment is indistinguishable from the one for H_0.

- H_1^1**:** In this game, we modify H_1 by simulating the proof π_1 by running the simulator $\pi_1' \leftarrow \mathsf{Sim}(\vec{\mathsf{crs}}, \vec{\mathsf{ts}}, \mathsf{x}_1)$.

According to the Zero-Knowledge property of the NIZK proofs, this experiment is indistinguishable from H_1.

- H**:** In this game, we modify H_b^1 by assuming the challenger runs the encryption algorithm under m_{1-b} instead of m_b.

According to the IND-CPA security property of the proposed CP-ABE scheme, this experiment is indistinguishable from H_b^1. To be more concrete, $\mathcal{A}$ cannot distinguish between $\mathtt{Ct}_b$ and $\mathtt{Ct}_{1-b}$ even if the proofs are simulated. Thereby we can conclude, $H_0 \approx_c H_0^1 \approx_c H \approx_c H_1^1 \approx_c H_1$. □

C.3 Proof of Theorem 4.3

Proof. The proof technique is inspired by the No-Write proof strategies of [27,39]. The following experiments rely on the security properties of the cryptographic primitives, namely the knowledge soundness of the NIZK, the existential unforgeability of the SPS and the IND-CPA security of the rCP-ABE. By playing a sequence of indistinguishable games between a PPT adversary $\mathcal{A}$ and the challengers $\mathcal{B}_{\text{KS}}$, $\mathcal{B}_{\text{EUF-CMA}}$ and $\mathcal{B}_{\text{IND-CPA}}$, we gradually turn the No-Write rule game into the security features of the underlying primitives.

- G_0**:** The first security game is the No-Write game of Definition 3.4, thus we can write, $Adv^{\text{No-Write}}_{\Psi_{\text{CD-ABACE}},\mathcal{A}}(1^\lambda, b) = \Pr[\mathcal{A} \text{ Wins } G_0]$.
- G_1**:** In this game, we modify G_0 such that the existence of an extraction trapdoor is assumed. In this case, there exists an extractor that takes $\vec{\mathsf{te}}$ and the received tuple (π_0, x_0), and returns the corresponding witness $(\mathsf{w}_0) \leftarrow \mathsf{Ext}(\vec{\mathsf{te}}, \mathsf{x}_0, \pi_0)$ such that $\mathsf{w}_0 = (\mathsf{ek}_{\mathbb{P}^*}, \sigma^*, m_0, r_0, t_0)$. The indistinguishability of G_0 and G_1 can be proven via the *Knowledge Extraction* property of NIZK proofs, specified in Definition B.7. This property guarantees the existence of an efficient extractor under non-falsifiable assumptions and we can write, $\Pr[\mathcal{A} \text{ Wins } G_0] \approx_c \Pr[\mathcal{A} \text{ Wins } G_1]$. This advantage consequently depends on two possible cases:

  $$\Pr[\mathcal{A} \text{ Wins } G_1] =$$
  $$\Pr[\mathcal{A} \text{ Wins } G_1 : (\mathsf{w}_0, \mathsf{x}_0) \in \mathbf{R_L}] + \Pr[\mathcal{A} \text{ Wins } G_1 : (\mathsf{w}_0, \mathsf{x}_0) \notin \mathbf{R_L}] \ .$$

  The probability of an adversary in the latter case can be bounded by the advantage a soundness attacker faces under the NIZK proof, i.e.,

  $$Adv^{\text{No-Write}}_{\Psi_{\text{CD-ABACE}},\mathcal{A}}(1^\lambda, b) \leq$$
  $$\Pr[\mathcal{A} \text{ Wins } G_1 : (\mathsf{w}_0, \mathsf{x}_0) \in \mathbf{R_L}] + Adv^{\text{KS}}_{\text{NIZK}}(\mathcal{B}_{\text{KS}}) \ .$$

  Hence the game is won by the adversary when the former is the case.
- G_2**:** This is the game G_1, except for a valid pair of witness and statement in $\mathbf{R_L}$; one can reduce it to a forgery attack for the underlying SPS scheme, if the extracted signature is created under a fresh attribute set. More specifically, if $\mathcal{A}$ does not query the encryption key for the attribute set $\mathbb{P}^*$, i.e. $\mathbb{P}^* \notin \mathcal{Q}_\mathcal{E}$, then $\mathcal{B}_{\text{EUF-CMA}}$ returns the pair $(\mathsf{ek}_{\mathbb{P}^*}, \mathbb{P}^*)$ as a forgery for the EUF-CMA security game of Definition B.3. We can write,

  $$Adv^{\text{No-Write}}_{\Psi_{\text{CD-ABACE}},\mathcal{A}}(1^\lambda, b) \leq$$
  $$Adv^{\text{KS}}_{\text{NIZK}}(\mathcal{B}_{\text{KS}}) + \Pr[\mathcal{A} \text{ Wins } G_1 : (\mathsf{w}_0, \mathsf{x}_0) \in \mathbf{R_L} \wedge \mathbb{P}^* \notin \mathcal{Q}_\mathcal{E}]$$
  $$+ \Pr[\mathcal{A} \text{ Wins } G_1 : (\mathsf{w}_0, \mathsf{x}_0) \in \mathbf{R_L} \wedge \mathbb{P}^* \in \mathcal{Q}_\mathcal{E}] \leq Adv^{\text{KS}}_{\text{NIZK}}(\mathcal{B}_{\text{KS}}) +$$
  $$Adv^{\text{EUF-CMA}}_{\text{SPS}}(\mathcal{B}_{\text{EUF-CMA}}) + \Pr[\mathcal{A} \text{ Wins } G_1 : (\mathsf{w}_0, \mathsf{x}_0) \in \mathbf{R_L} \wedge \mathbb{P}^* \in \mathcal{Q}_\mathcal{E}] \ .$$

- G_3**:** This game is the same as the previous game G_2, except for a random message $m^* \leftarrow\!\!\$\, \mathcal{M}$ and the random bit $b \leftarrow\!\!\$\, \{0, 1\}$. The challenger executes the sanitization algorithm under $\mathtt{Ct}_{1-b}$. Then, the difference between the views in G_2 and G_3 is bounded by $Adv^{\text{IND-CPA}}_{rCP-ABE}(\mathcal{B}_{\text{IND-CPA}})$ and we can write:

  $$Adv^{\text{No-Write}}_{\Psi_{\text{CD-ABACE}},\mathcal{A}}(1^\lambda, b) \leq Adv^{\text{KS}}_{\text{NIZK}}(\mathcal{B}_{\text{KS}}) +$$
  $$Adv^{\text{EUF-CMA}}_{\text{SPS}}(\mathcal{B}_{\text{EUF-CMA}}) + Adv^{\text{IND-CPA}}_{rCP-ABE}(\mathcal{B}_{\text{IND-CPA}}) \ .$$

Thereby we can conclude, $Adv^{\text{No-Write}}_{\Psi_{\text{CD-ABACE}},\mathcal{A}}(1^\lambda, b) \leq \mathsf{negl}(\lambda)$. □

C.4 Proof of Theorem 5.1

Proof. We demonstrate that a receiver who owns the set of attributes $\mathbb{B} \subset \mathbb{U}$ can correctly decrypyt the ciphertext if and only if the attribute set $\mathbb{B}$ satisfies the access structure $\mathbb{P}$ (i.e., $\mathbb{P} \subseteq \mathbb{B}$). In the decryption phase we have:

$$V_1 = \left(C_2 \bullet \prod_{i=1}^{n} (h_{i-1})^{f_i}\right) = \left(\left[-r\alpha^2\right]_1 \bullet \left[\left(\sum_{i=1}^{n} f_i \alpha^{i-1}\right) + f_0/\alpha - f_0/\alpha\right]_2\right)$$
$$\left(\left[-r\alpha^2\right]_1 \bullet \left[(F_{\mathbb{B},\mathbb{P}}(\alpha) - f_0)/\alpha\right]_2\right) = \left[r\alpha(f_0 - F_{\mathbb{B},\mathbb{P}}(\alpha))\right]_T \ .$$
$$V_2 = (\mathsf{dk}_{\mathbb{B}} \bullet C_1) = \left[1/Z_{\mathbb{B}}(\alpha)\right]_1 \bullet \left[r\alpha Z_{\mathbb{P}}(\alpha)\right]_2 = \left[r\alpha Z_{\mathbb{P}}(\alpha)/Z_{\mathbb{B}}(\alpha)\right]_T =$$
$$\left[r\alpha F_{\mathbb{B},\mathbb{P}}(\alpha)\right]_T \ .$$
$$m' = C \cdot (V_1 \cdot V_2)^{-1/f_0} = C\left(\left[r\alpha f_0\right]_T \cdot \left[-rF_{\mathbb{B},\mathbb{P}}(\alpha)\right]_T \cdot \left[rF_{\mathbb{B},\mathbb{P}}(\alpha)\right]_T\right)^{-1/f_0}$$
$$= m \cdot \left[r\alpha\right]_T \cdot \left[-r\alpha\right]_T = m \ .$$

□

More precisely, the univariate polynomial $Z_{\mathbb{B}}(x)$ vanishes in the point $k_i = \mathsf{H}(U_i)$ for those attributes that are not in the set $\mathbb{B}$, i.e., this polynomial has $n - |\mathbb{B}|$ roots. In a similar way, the polynomial $Z_{\mathbb{P}}(x)$ has degree $n - |\mathbb{P}|$ with factors $(x - k_j)$ for those attributes that are in $\overline{\mathbb{P}}$. The Boolean relation Bf in the proposed CP-ABE enforces that to decrypt a ciphertext the subset $\mathbb{P}$ has to be a subset of $\mathbb{B}$ and we have $|n - \mathbb{B}| \leq |n - \mathbb{P}|$. Since all the attributes which are out of the set $\mathbb{B}$ are equal to all the attributes out of the set $\mathbb{P}$, all the factors of the polynomial $Z_{\mathbb{B}}(x)$ simplify by the polynomial $Z_{\mathbb{P}}(x)$. Since $|\mathbb{P}| \leq |\mathbb{B}|$ and the result of the division $Z_{\mathbb{P}}(x)/Z_{\mathbb{B}}(x)$ is not rational and equal to $F_{\mathbb{B},\mathbb{P}}(x)$, we can evaluate this polynomial from the second group, $\left[F_{\mathbb{B},\mathbb{P}}(\alpha)\right]_2$, by knowing the monomial set $\{\left[\alpha^i\right]_2\}_{i\in[0,n]}$. Moreover, the univariate polynomial $F_{\mathbb{B},\mathbb{P}}(x)$ vanishes on those k_i for which $\mathbb{B}$ and $\mathbb{P}$ are disjoint. Conversely, if $\mathbb{P} \not\subseteq \mathbb{B}$ then there exists at least one root for the polynomial $Z_{\mathbb{B}}(x)$ that does not cancel out by the numerator $Z_{\mathbb{P}}(x)$. Hence the result of the division is rational and the receiver cannot compute the evaluated polynomial based on the defined standard basis in the point of α from the second group.

Moreover, as in a traditional security evaluation of ABE schemes, we evaluate the possibility of multiple users colluding. More precisely, in a collusion-resistance ABE scheme, malicious users cannot acquire an encrypted message for which access is denied by the access right embedded

in the ciphertext, implying that they cannot retrieve the original plaintext by pooling their secret decryption keys. The proposed construction is secure against this attack because the secret value α is hidden and for an attribute universe size n, the size of key indices is assumed to be strictly smaller than $n-1$, i.e. $|\mathbb{B}| < n-1$. It follows naturally that since the degree of underlying zero-polynomial is at least 2 and the key indices are disjoint on at least one element, a term of α will appear in the nominator and a malicious user would need to guess α correctly to cancel out this term.

C.5 Proof of Theorem 5.2

Proof. We prove this theorem by reduction. Assume there exists a Probabilistic Polynomial Time (PPT) adversary, $\mathcal{A}$, who can break the proposed scheme in the introduced security game in Definition B.10 with a non-negligible advantage ϵ. Then we will show how a PPT adversary, $\mathcal{B}$, can solve the $(l, m, t) - \texttt{MSE-DDH}$ problem with a non-negligible advantage of at least $\frac{\epsilon}{2}$. In fact, $\mathcal{B}$ takes on the role of the challenger and utilizes the adversary $\mathcal{A}$ in order to solve the mentioned hard problem.

Let the challenger $\mathcal{C}$ of the Decisional $(l, m, t) - \texttt{MSE-DDH}$ hard assumption run the asymmetric bilinear group generator $\mathcal{BG}(\lambda)$ for the security parameter λ and take $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathsf{p}, \hat{e})$ such that $[1]_1$, $[1]_2$ and $[1]_T$ are the generators of the defined cyclic groups. The challenger $\mathcal{C}$ first chooses three integers l, m, t, along with two univariate coprime polynomials f and h of degree l and m with pairwise distinct roots $\vec{x} = (x_1, \ldots, x_l)$ and $\vec{y} = (y_1, \ldots, y_m)$. It samples integers $\alpha, \delta, k \leftarrow\!\!\$\, \mathbb{Z}_p^*$ uniformly at random and then flips a fair coin, $\beta \leftarrow\!\!\$\, \{0, 1\}$, outside $\mathcal{B}$'s view. If $\beta = 0$, $\mathcal{C}$ sets $\Gamma = [kf(\alpha)]_T$, otherwise, it sets $\Gamma = R$, where R is a random element of the target cyclic group $\mathbb{G}_T$. The challenger $\mathcal{C}$ sends Γ and the pair of vectors $\vec{x}$ and $\vec{y}$ along with the vectors:

$$
\begin{aligned}
\vec{v_1} &= \left([1]_1, [\alpha]_1, \left[\alpha^2\right]_1, \ldots, \left[\alpha^{l+t-2}\right]_1, [k\alpha f(\alpha)]_1\right) \\
\vec{v_2} &= \left([\delta]_1, [\delta\alpha]_1, \left[\delta\alpha^2\right]_1, \ldots, \left[\delta\alpha^{l+t}\right]_1\right) \\
\vec{v_3} &= \left([1]_2, [\alpha]_2, \left[\alpha^2\right]_2, \ldots, \left[\alpha^{m-2}\right]_2\right) \\
\vec{v_4} &= \left([\delta]_2, [\delta\alpha]_2, \left[\delta\alpha^2\right]_2, \ldots, \left[\delta\alpha^{2m-1}\right]_2, [kh(\alpha)]_2\right)
\end{aligned}
\tag{1}
$$

Initialization: In this phase, the simulator $\mathcal{B}$ sets the universe attribute set of $\mathbb{U}$ as all the possible attributes in the defined network and a collision-resistant Hash function $\mathsf{H} \leftarrow\!\!\$\, \mathcal{H}$ that is modelled as a ROM. We just program this random oracle to give us structured attribute indices,

enabling us to form the challenge ciphertext. $\mathcal{B}$ publishes $\mathbb{U}$ and she receives back the challenge access policy $\mathbb{P}^*$ along with the query set Q as a group of attribute sets $\mathbb{B}_i \subseteq \mathbb{U}$ for $i \in [s]$, such that $|\mathbb{B}_i| \leq e$ from $\mathcal{A}$ and also $\mathbb{P}^* \not\subseteq \mathbb{B}_i$ (i.e., $\text{BF}(\mathbb{B}_i, \mathbb{P}^*) = 0$). In order to publish the public parameters, $\mathcal{B}$ computes the public parameters and computes the following polynomial that is the multiplication of zero-polynomials corresponding to the chosen subsets $\mathbb{B}_i$ in the query set Q:

$$Y_Q(x) = \prod_{i=1}^{s} Z_{\mathbb{B}_i}(x) = \prod_{i=1}^{s} \left(\prod_{j=1}^{n-e} (x - k_j)^{\overline{b_i[j]}} \right) \quad .$$

Here $b_i[j]$ represents the j^{th} binary representation of subset $\mathbb{B}_i$. The degree of the univariate polynomial $Y_Q(x)$ is upper bounded by $s(n-e)$. Moreover, since we know $h(x) = \prod_{i=0}^{m} (x - y_i)$, it is assumed $Z_{\mathbb{P}^*}(x) = Y_Q(x)h(x)$ such that $|\mathbb{P}^*| = n - (s(n-e) + m)$. This can be feasible by modelling the hash function H as a Random Oracle. Then she assumes $G = [f(\alpha)Y_Q(\alpha)]_1$ as a new generators for the first cyclic group $\mathbb{G}_1$. In this case, the Challenger $\mathcal{B}$ calculates $f(x)Y_Q(x) = \sum_{j=0}^{l+s(n-e)} p_j x^j$ to compute g_2 based on the newly defined generator as follows:

$$G_2 = \prod_{j=0}^{l+s(n-e)} \left[\alpha^{j+2}\right]_1^{p_j} = [f(\alpha)Y_Q(\alpha)]_1^{\alpha^2} = G^{\alpha^2} \quad .$$

We emphasize that the generator of the first cyclic group is not public. Based on the knowledge of the vector $\vec{v_1}$, $\mathcal{B}$ can compute the above equation if $t \geq s(n-e) + 4$. Consequently, the challenger defines $h_i = \left[\alpha^i\right]_2$. Finally, the public parameters based on the new generator are $\mathsf{pp} = \left\{G_2, \{h_i\}_{i=0}^{n}, [\alpha f(\alpha)Y_Q(\alpha)]_T, \mathsf{H}\right\}$. Next she securely stores the master secret key $\mathsf{msk} = \{G\}$.

1^{st} **Query phase.** After receiving the public parameters, the adversary $\mathcal{A}$ has access to the following oracles for a polynomially bounded number of queries:

- **Simulating the $\mathcal{O}_{\mathsf{DecKGen}}(\mathbb{B}_i)$ oracle.** The adversary $\mathcal{A}$ has access to this oracle which is provided by $\mathcal{B}$, to receive the secret decryption key corresponding to the attribute set $\mathbb{B}_i \in Q$. In this end, in order to simulate the secret decryption key, $\mathcal{B}$ calculates the univariate polynomial $\Lambda_i(x)$, such that $f(x)Y_Q(x) = \Lambda_i(x) \cdot Z_{\mathbb{B}_i}(x)$. Based on the definition of the polynomial $Y_Q(x)$ we know it is divisible by $Z_{\mathbb{B}_i}(x)$, and we can rewrite the above equation as $\Lambda_i(x) = (f(x)Y_Q(x))/Z_{\mathbb{B}_i}(x)$. Since

the polynomial $\Lambda_i(x)$ is not rational, we can take the coefficients in the standard basis as $\Lambda_i(x) = \sum_{j=0}^{q} \lambda_j x^j$. Finally, the challenger returns the following equation as the simulated secret decryption key corresponding to $\mathbb{B}_i$.

$$\mathsf{dk}_{\mathbb{B}_i} = \prod_j \left[\alpha^j\right]_1^{\lambda_j} = [\Lambda_i(\alpha)]_1 = [f(\alpha)Y_Q(\alpha)]_1^{\frac{1}{Z_{\mathbb{B}_i}(\alpha)}} = G^{\frac{1}{Z_{\mathbb{B}_i}(\alpha)}} \ .$$

– **Simulating the $\mathcal{O}_{\mathsf{Enc}}(m, \mathbb{P})$ oracle.** The adversary $\mathcal{A}$ can adaptively request to encrypt arbitrary messages from the message space $\mathcal{M}$ under a certain access structure $\mathbb{P}$. The challenger $\mathcal{B}$ samples a random integer $r \leftarrow\!\$\, \mathbb{Z}_p^*$, uniformly and computes:

$$C = m \left(\prod_{i=0}^{s(n-e)+l} \left(\left[\alpha^i\right]_1 \bullet [\alpha]_2\right)^{p_i} \right)^r = m\,[r\alpha f(\alpha)Y_Q(\alpha)]_T$$
$$C_1 = \left(\prod_{i=0}^{n} \left[\alpha^{i+1} z_i\right]_2 \right)^r = [r\alpha Z_{\mathbb{P}}(\alpha)]_2$$
$$C_2 = G_2^{-r} \ .$$

It sends back the tuple $\mathtt{Ct} = (\mathbb{P}, C, C_1, C_2)$ to $\mathcal{A}$. The only condition is that $(m-2) \geq n - |\mathbb{P}| + 1$, i.e., $|\mathbb{P}| \geq n - m + 3$.

– **Simulating the $\mathcal{O}_{\mathsf{Dec}}(\mathtt{Ct}, \mathbb{B}_j)$ oracle.** The adversary $\mathcal{A}$ has access to this oracle to receive the decryption of ciphertext $\mathtt{Ct}$ by providing an attribute set $\mathbb{B}_j \in Q$. To this end, $\mathcal{B}$ executes $\mathsf{dk}_{\mathbb{B}_j} \leftarrow \mathsf{DecKGen}(\mathsf{msk}, \mathbb{B}_j)$ and takes the set $\mathbb{P}$, defines $c[i] = b_j[i] - p[i]$ and calculates, $F_{\mathbb{B}_j,\mathbb{P}}(x) = \prod_{i=1}^{n} (x - k_i)^{c[i]} = \sum_i f_i x^i$. Whence she returns the decrypted message m' as follows:

$$m' = C \cdot \left(C_2 \bullet (\prod_{i=1}^{n} h_{i-1})^{f_i} \cdot (\mathsf{dk}_{\mathbb{B}} \bullet C_1) \right)^{-1/f_0} \ .$$

Challenge: The adversary $\mathcal{A}$ chooses two plaintexts with the same length $\{m_0, m_1\} \leftarrow\!\$\, \mathcal{M} \times \mathcal{M}$ and sends them to $\mathcal{B}$. Then $\mathcal{B}$ flips a fair coin to generate the bit $b \leftarrow\!\$\, \{0,1\}$, and computes the challenge ciphertext $\mathtt{Ct}^* = (\mathbb{P}^*, C^*, C_1^*, C_2^*)$ as follows:

$$C^* = m_b \Gamma, C_1^* = [kh(\alpha)]_2\,, C_2^* = [-k\alpha f(\alpha)]_1 \ .$$

The randomness of the challenge ciphertext is assumed to be $r^* = k/(\alpha Y_Q(\alpha))$. In a nutshell, based on the $(l, m, t) - \mathtt{MSE\text{-}DDH}$ assumption,

there are two cases for the received challenge Γ with the same probability $1/2$. If $\Gamma = [kf(\alpha)]_T$ then we have:

$$C^* = m_b\,[kf(\alpha)]_T = m_b\,[r^*\alpha f(\alpha)Y_Q(\alpha)]_T$$
$$C_1^* = [kh(\alpha)]_2 = [r^*\alpha Y_Q(\alpha)h(\alpha)]_2 = [r^*\alpha Z_{\mathbb{P}^*}(\alpha)]_2$$
$$C_2^* = [-k\alpha f(\alpha)]_1 = \left[-r^*\alpha^2 Y_Q(\alpha)f(\alpha)\right]_1 = G_2^{-r^*}\ .$$

While in the case of an independent and random element in the group $\mathbb{G}_T$, the computed C^* is a random element out of the construction and the adversary cannot do better than guessing with success probability $1/2$.

2^{nd} **Query phase.** After receiving the challenge ciphertext $\mathtt{Ct}^*$, the adversary $\mathcal{A}$ has access to the queries defined in the first query phase on the condition that she cannot query the key generation oracle for those attribute sets that are sufficient for the challenge access policy $\mathbb{P}^*$.

Guess. Afterwards, $\mathcal{A}$ returns either 1 or 0. Let b' and β' be the values that are guessed respectively by $\mathcal{A}$ for b and by $\mathcal{B}$ for β. If $b' == b$, the adversary $\mathcal{B}$ outputs $\beta' = 0$, otherwise she returns $\beta' = 1$, which indicates that she receives a random element in the target group as the challenge. When $\beta = 1$, the adversary $\mathcal{A}$ obtains no information about b. So she can guess it and we have $\Pr[b' == b \mid \beta = 1] = 1/2$. On the other hand, when $b' \neq b$, $\mathcal{B}$ returns $\beta' = 1$, hence we have $\Pr[\beta' == \beta \mid \beta = 1] = 1/2$. Particularly, if $\beta = 0$, $\mathcal{A}$ can distinguish with a non-negligible advantage ϵ because she has received the true format of the ciphertext for the challenge message m_b. Thus, we have $\Pr[b' == b \mid \beta = 0] \geq \epsilon + 1/2$. As $\mathcal{B}$ correctly guesses β, when $\beta = 0$, we have $\Pr[\beta' == \beta \mid \beta = 0] \geq \epsilon + 1/2$. Therefore, the overall advantage of the adversary $\mathcal{B}$ in solving the $(l, m, t) - \mathtt{MSE\text{-}DDH}$ problem is:

$$Adv_{\mathcal{B}}^{\mathtt{MSE\text{-}DDH}}(1^\lambda) = \Big(\Pr[\beta = 0]\Pr[\beta' == \beta \mid \beta = 0] +$$
$$\Pr[\beta = 1]\Pr\left[\beta' == \beta \mid \beta = 1\right]\Big) - 1/2 \geq$$
$$1/2\,(\epsilon + 1/2) + (1/2 \cdot 1/2) - 1/2 \geq \frac{\epsilon}{2}\ .$$

Therefore, the adversary $\mathcal{B}$ can play the $(l, m, t) - \mathtt{MSE\text{-}DDH}$ game with a non-negligible advantage $\frac{\epsilon}{2}$. By contradiction, since we know there is no PPT adversary $\mathcal{B}$ to break the $(l, m, t) - \mathtt{MSE\text{-}DDH}$ assumption with a non-negligible advantage, the proposed CP-ABE scheme in Sect. 5 is secure in the IND-CPA game of Definition B.10. □

D A CD-ABACE with LSSS

In Fig. 5, we propose an ACE variation of Waters's CP-ABE scheme [41]. This scheme is built under the original Abe et al.'s SPS scheme [2], Waters's CP-ABE [41] and NIZK proofs in the ROM. We provide an overview of the LSSS in the following definition and refer readers to [41] for more details.

Definition D.1 (Linear Secret Sharing Scheme (LSSS) [6]). *A secret-sharing scheme Ψ_A for the access structure A over a set of attributes $\mathbb{P}$ is called linear (over $\mathbb{Z}_{\mathsf{p}}^*$) if:*

- *The shares of a secret $s \in \mathbb{Z}_{\mathsf{p}}^*$ for the set of attributes form a vector over $\mathbb{Z}_{\mathsf{p}}^*$.*
- *There exists a matrix $\mathcal{M}$, with dimension $\ell \times d$, called the share-generating matrix for Ψ_A. The i^{th} row of $\mathcal{M}$, $\mathcal{M}_i$, is labeled by $\rho(i)$ where $\rho(.)$ is a function from $\{1, 2, \ldots, \ell\}$ to $\mathbb{P}$. We consider the column vector $\vec{v} = \{s, r_2, \ldots, r_d\}$, where $s \in \mathbb{Z}_{\mathsf{p}}^*$ is the secret to be shared and $\{r_2, \ldots, r_d\} \in \mathbb{Z}_{\mathsf{p}}^*$ are randomly chosen. So $\mathcal{M}_{l\times d} \times \vec{v}^{\top}$ is the vector of ℓ shares of the secret s according to Ψ_A. The share $\lambda_i = (\mathcal{M}_{\ell\times d} \times \vec{v}^{\top})_i$ corresponds to the attribute $\rho(i)$.*

Let $S \in A$ be any authorized set, and let $\mathcal{I} \subset \{1, \ldots, \ell\}$ be defined as $\mathcal{I} = \{i \mid \rho(i) \in S\}$. Then, there exist constant values $\{\omega_i \in \mathbb{Z}_p\}_{i\in\mathcal{I}}$ such that, if $\lambda_{i\in\mathcal{I}}$ are valid shares of the secret s according to Ψ_A, then $\sum_{i\in\mathcal{I}} \omega_i \cdot \lambda_i = s$. Let $\mathcal{M}_i$ denotes the i^{th} row of $\mathcal{M}_{\ell\times d}$, then $\sum_{i\in\mathcal{I}} \omega_i \mathcal{M}_i = (1, 0, \ldots, 0)_{1\times d}$. The constants $\{\omega_i\}$ can be found with polynomial time complexity in term of the size of the share-generation matrix $\mathcal{M}_{\ell\times d}$, where ℓ is the number of attributes and d is the level of the access structure. We refer to [41] for more details about the access structure and the LSSS technique.

$(\mathsf{pp}_{ra}, \mathsf{msk}_{ra}) \leftarrow \mathsf{RAgen}(\mathbb{U}, \lambda)$

1 : Run $\mathcal{BG}(\lambda) = (\mathbb{G}, \mathbb{G}, \mathbb{G}_T, \mathsf{p}, \hat{e})$
2 : $\mathsf{H} \leftarrow\$\, \mathcal{H}, \alpha \leftarrow\$\, \mathbb{Z}_p^*, \beta \leftarrow\$\, \mathbb{Z}_p^*$
3 : $\{h_i \leftarrow \mathbb{G}\}_{i\in[|\mathbb{U}|]}$
4 : $\mathsf{msk}_{ra} = ([\alpha]_1)$
5 : $\mathsf{pp}_{ra} = (g, \{h_i\}_{i=1}^{|\mathbb{U}|}, [\alpha]_T, g^{\beta}, \mathsf{H})$
6 : **return** $(\mathsf{msk}_{ra}, \mathsf{pp}_{ra})$

$(\mathsf{pp}_{sa}, \mathsf{msk}_{sa}) \leftarrow \mathsf{SAgen}(\lambda, \mathsf{pp}_{ra}, \mathbf{R_L})$

1 : Parse $(\mathcal{BG}(\lambda), \mathsf{pp}_{ra})$
2 : $X \leftarrow\$\, \mathbb{G}$
3 : $\mathsf{sk} := v \leftarrow\$\, \mathbb{Z}_{\mathsf{p}}$
4 : $\mathsf{vk} = [v]_1$
5 : $(\vec{\mathsf{crs}}, \vec{\mathsf{ts}}) \leftarrow\$\, \mathcal{ZK}.\mathsf{KG}_{\vec{\mathsf{crs}}}(\lambda, \mathbf{R_L})$
6 : $\mathsf{msk}_{sa} = (\mathsf{sk}, \vec{\mathsf{ts}})$
7 : $\mathsf{pp}_{sa} = (\mathbf{R_L}, \vec{\mathsf{crs}}, X, \mathsf{vk})$
8 : **return** $(\mathsf{msk}_{sa}, \mathsf{pp}_{sa})$

$(\mathsf{dk}_{\mathbb{S}}) \leftarrow \mathsf{DecKGen}(\mathsf{msk}_{ra}, \mathbb{S})$

1 : Parse $(\mathcal{BG}(\lambda), \mathsf{msk}_{ra})$
2 : $t \leftarrow\$\, \mathbb{Z}_{\mathsf{p}}^*$
3 : $K = [\alpha + \beta t]_1$
4 : $L = [t]_1$
5 : $\forall\ x \in \mathbb{S}\ K_x = h_x^t$
6 : **return** $\mathsf{dk}_{\mathbb{S}} = (K, L, \{K_x\}_{x\in\mathbb{S}})$

$(\mathsf{ek}_{\mathbb{P}}, \sigma, W) \leftarrow \mathsf{EncKGen}(\mathsf{pp}_{ra}, \mathsf{msk}_{sa}, (\mathcal{M}, \rho), \mathrm{PF})$

1 : Parse $(\mathcal{BG}(\lambda), \mathsf{pp}_{ra}, \mathsf{msk}_{sa})$
2 : $\mathsf{Col}(\mathsf{pp}, (\mathcal{M}, \rho)) = \{\mathsf{ek}_i = h_{\rho(i)}\}_{i=1}^{\ell}$
3 : $t_u \leftarrow\$\, \mathbb{Z}_p^*, W = [1/t_u]_2$
4 : $R = g_2^{t_u}$
5 : $\{S_i = \mathsf{ek}_i^{\mathsf{sk}/t_u} X^{1/t_u}\}_{i\in\ell}$
6 : $\{T_i = S^{\mathsf{sk}/t_u}\,[1/t_u]_2\}_{i\in[\ell]}$
7 : **return** $(\mathsf{ek}_{\mathbb{P}}, \sigma = (R, \{S_i, T_i\}_{i\in[\ell]}), W)$

$(\pi, \mathsf{x}) \leftarrow \mathsf{Enc}\,(\mathsf{pp}_{sa}, \mathsf{pp}_{ra}, m, \mathsf{ek}_{\mathbb{S}}, \sigma, W)$

1 : Parse $(\mathcal{BG}(\lambda), \mathsf{pp}_{ra}, \mathsf{pp}_{sa})$
2 : $t_i \leftarrow\$\, \mathbb{Z}_p^*, r_i \leftarrow\$\, \mathbb{Z}_p^*, s \leftarrow\$\, \mathbb{Z}_{\mathsf{p}}^*$
3 : $\vec{v} = (s, y_2, \ldots, y_n) \in \mathbb{Z}_{\mathsf{p}}^n,\ \lambda_i = \vec{v} \cdot \mathcal{M}_i$
4 : $(C, C') = (m\,[s\alpha]_T, [s]_1)$
5 : $\{(C_i, D_i) = ([\beta\lambda_i]_1\, \mathsf{ek}_i^{-r_i}, [r_i]_1)\}_{i\in[\ell]}$
6 : $R_i' = R_i^{1/t_i},\ S_i' = S_i^{t_i},$
7 : $T_i' = T_i^{t_i^2} W^{t_i(1-t_i)}$
8 : $\sigma' = \{(R_i', S_i', T_i')\}_{i\in[\ell]}$
9 : $\mathsf{x} = (\sigma', \mathtt{Ct} = ((\mathcal{M}, \rho), C, C', \{C_i\}_{i\in\ell}))$
10 : $\mathsf{w} = (\sigma, m, \{r_i, t_i\}_{i\in\ell})$
11 : $\pi \leftarrow \mathrm{PoK}\{(\mathsf{w}) : \mathbf{R_L}(\mathsf{x}, \mathsf{w}) = 1\}$
12 : **return** (π, x)

$(\tilde{\mathtt{Ct}}, \perp) \leftarrow \mathsf{San}(\mathsf{pp}_{sa}, \mathsf{pp}_{ra}, \pi, \mathsf{x})$

1 : Parse $(\mathcal{BG}(\lambda), \mathsf{pp}_{ra}, \mathsf{pp}_{sa})$
2 : **if** $\{R_i', \mathsf{ek}_{\mathbb{S}}, S_i', T_i' \in \mathbb{G} \wedge$
3 : $R_i' \bullet S_i' = (\mathsf{vk} \bullet \mathsf{ek}_i)(X \bullet [1]_1) \wedge$
4 : $R_i' \bullet T_i' = (\mathsf{vk} \bullet S_i')\,[1]_T\ \wedge$
5 : $\mathcal{ZK}.\mathsf{V}(\mathbf{R_L}, \vec{\mathsf{crs}}, \pi, \mathsf{x}) = 1\}$:
6 : $\quad \mu_i \leftarrow\$\, \mathbb{Z}_{\mathsf{p}}^*$
7 : $\quad \tilde{C}_i = C_i \cdot \mathsf{ek}_i^{-\mu_i}$
8 : $\quad \tilde{D}_i = D_i \cdot [\mu_i]_1$
9 : $\quad$ **return**
10 : $\quad \tilde{\mathtt{Ct}} = ((\mathcal{M}, \rho), C, C', \{\tilde{C}_i, \tilde{D}_i\}_{i\in\ell})$
11 : **else** : **abort**

$(m', \perp) \leftarrow \mathsf{Dec}\big(\mathsf{pp}_{sa}, \mathsf{pp}_{ra}, \tilde{\mathtt{Ct}}, \mathsf{dk}_{\mathbb{S}}\big)$

1 : Parse $(\mathcal{BG}(\lambda), \mathsf{pp}_{ra}, \mathsf{pp}_{sa})$
2 : **if** $\mathbb{S}$ Satisfies $(\mathcal{M}, \rho)$:
3 : $\quad I = \{i : \rho(i) \in \mathbb{S}\} \subset \{1, \ldots, \ell\},$ Let $\{\omega_i \in \mathbb{Z}_{\mathsf{p}}\}_{i\in I}$ then $\sum_{i\in I} \omega_i \lambda_i = s$
4 : $\quad$ **return** $m' = \dfrac{C}{\big((C' \bullet K)\,/\,\big(\prod_{i\in I}((C_i \bullet L)(D_i \bullet K_{\rho(i)}))^{\omega_i}\big)\big)}$
5 : **else** : **abort**

Fig. 5. A CD-ABACE based on Waters's CP-ABE scheme