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 construction combines Structure-Preserving Signatures, Non-Interactive Zero-Knowledge proofs, and Re-randomizable Ciphertext-Policy Attribute-Based Encryption schemes. The sizes of ciphertexts and encryption and decryption keys are constant and thus independent of the number of receivers and their attributes. Not only is our system more flexible, it also is more efficient and results in shorter keys.

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

1 Introduction

Information Flow Control (IFC) systems enforce which parts of the communication amongst the users are allowed to pass over the network [SM03, OSM00]. As introduced in the seminal work of Bell and LaPadula [BL73] restrictions have to be imposed on who can send a message (enforce the No-Write rule) and who can receive a message (enforce the No-Read rule). Although encryption guarantees users' privacy by limiting the set of recipients, we need more functionality to control the access to information. Broadcasting of sensitive data by malicious senders is a serious threat for companies that handle highly sensitive data such cryptocurrency wallet with access to signing keys. Moreover, data regulations that are country-dependent have brought new concerns for Cloud providers [YKM14], hence it is vital to enforce potentially complex security policies. It is crucial to protect data against unauthorized access and to control which group of users is allowed to use certain services. Although some advanced cryptographical tools such *Functional Encryption* schemes provide fine-grained access to encrypted data, they do not allow to enforce the No-Write property, hence additional functionalities beyond these cryptographic primitives are required to protect against data leakage and abuse.

To achieve this aim, Damgård et al. [DHO16] have 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 if and only if $\textsc{Pf}(i, j) = 1$, else the request will be banned.

Damgård et al. presented 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 and the ciphertext length is poly-logarithmic in the number of the receivers, though, it relies on the strong assumption of *indistinguishability obfuscation (iO)* [GGH$^+$16]. In a subsequent work, Fuchsbauer et al. [FGKO17] 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. [TZMT17] proposed an ACE scheme based on the *Learning With Error* (LWE) assumption [Reg09]. 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. Kim and Wu in [KW17] 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 [WC21] 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 this scheme, the length of the ciphetexts are constant, but their construction fails to preserve the identity of the receivers and also the size of the stored decryption key grows linearly. This paper proposes a modified version of Wang and Chow's construction [WC21] under the form of an *Attribute-Based Access Control Encryption* scheme that supports cross-domain key generation and that is based on users' attributes instead of their identities.

Attribute-Based Encryption (ABE) schemes provide a powerful tool to enforce fine-grained access control over encrypted data; they have been used in several applications [SW05]. Goyal et al. in [GPSW06], 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

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 in any access policy that a sender can transmit data. The maximum number of legitimate attributes that any recipients possesses to decrypt a ciphertext is denoted by $w \ll n$. PF and CD are short for Predicate Function and Cross-Domain, respectively.

Scheme	**PF**	**Ciph. size**	**Enc. Key**	**Dec. Key**	**San. Key**	**Enc. cost**	**Dec. cost**	**CD**	**Assump.**
[DHO16, ‡ 3]	arbitrary ID-based	$O(2^n)$	$O(r)$	$O(1)$	$O(1)$	$O(n)$	$O(n)$	Yes	DDH or DCR
[DHO16, ‡ 4]	arbitrary ID-based	$poly(n)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$	No	iO
[FGKO17]	restricted ID-based	$O(n)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$	No	SXDH
[KW17]	arbitrary ID-based	$poly(n)$	$O(1)$	$O(1)$	$O(1)$	$O(n)$	$O(n)$	No	DDH or LWE
[WC21]	arbitrary ID-based	$O(1)$	$O(1)$	$O(s)$	0	$O(1)$	$O(s)$	Yes	GBDP
This work (Selectively Secure)	arbitrary Attribute-based	$O(1)$	$O(1)$	$O(1)$	0	$O(1)$	$O(w)$	Yes	MSE-DDH

yields the plaintext if and only if the receivers' attribute set $\mathbb{A}$ satisfies $f(\cdot)$, i.e., $f(\mathbb{A}) = 1$. On the other hand, in KP-ABE the secret keys are labeled by the function $f(\cdot)$; this label is set in the setup phase and the sender is not able to change it. In KP-ABE, the access policy cannot be deduced by an encrypted actor, while in CP-ABE data owners can define the right to access and control data, hence it is a more suitable setting for ACE schemes. 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. in [BSW07]; its security was proven in the *Generic Group Model* (GGM). In a subsequent work, Cheung et al. [CN07] constructed a CP-ABE scheme in the standard model, which is however restricted to a single AND-gate. Waters introduced in [Wat11] 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 [LW11] 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 [SAMA17, LZN+20, HS16, AC17, RD14]. 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 for addressing this downside. In this approach the sender defines a specific Boolean AND-gate circuit such that a recipient can learn the encrypted data if and only if 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 based on this approach [EMN+09, TDM12, CZF11, GMS+14]. 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; additional functionalities are required to cover the latter.

1.1 Overview of Our Techniques

In this paper, we propose an efficient and secure *Cross-Domain Attribute-Based Access Control Encryption* (CD-ABACE) scheme. Based on an Attribute-Based predicate function, the senders with a group of attributes are limited to transmit data to only a restricted group of the recipients. To obtain a CD-ABAC scheme that is efficient both in the length of the parameters and the computational overhead, we utilize circuits with AND-gate circuits. More specifically, we say a Boolean AND-gate circuit is satisfied (i.e, the output is true) if and only if all the input gates are true. Particularly, for an attribute space $\mathbb{U}$ we say the set of attributes $\mathbb{B} \subset \mathbb{U}$ satisfies the AND-gate circuit with the set of input constraints $\mathbb{P} \subseteq \mathbb{U}$ if and only if $\mathbb{P}$ 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 $\mathbb{P} \subseteq \mathbb{B}$. The main downside for this kind of circuits is that the attribute sets in plain may reveal some meaningful information about the intended constraints.

Based on realistic application scenarios for ACE constructions, the proposed scheme follows the Cross-Domain key generation method, proposed by Wang and Chow in [WC21]. 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.

The paper re-defines the way to conceive the predicate function in ACE constructions by considering users' attributes instead of their identities. In a nutshell, the sender who owns a secret encryption key for ciphertext index of $\mathbb{P} \subset \mathbb{U}$ can transmit data to those of receivers with private decryption key corresponding to key index $\mathbb{B}$, 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 original one is that Sanitizer would never learn the identity of the receivers while it brings a weaker notion of anonymity. For this aim, we give a generic construction of a *Cross-Domain Access Control Encryption* scheme inspired by *Attribute-Based Encryption* schemes and then propose an efficient construction with a constant key and ciphertext sizes. Moreover, in this construction the Sanitizer only requires public parameters, but no secret or public keys. Our main contributions can be summarized as follows:

- 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.
- The predicate function takes as inputs user attributes instead of their identities.
- Every receiver needs to execute exactly two pairings to learn the message, while in the encryption phase, the sender does not need to compute any pairings.
- 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 and the recent works in the literature. 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. The predicate function takes as input the user attributes. Although the ciphertext access right preserves the identity of receivers in a weak notion, it does not reveal their identity.

1.2 Road-map

The rest of the paper is organized as follows. In Sect. 2, we review the relevant preliminaries and definitions and describe the system architecture. The main building blocks and a formal definition of CD-ABACE schemes are described in Sect. 3. The security definitions are described in Sect. 4. Then in Sect. 5, we present the construction of the proposed CD-ABACE scheme and prove its security features in Sect. 6. The performance of the proposed construction is compared in Sect. 7. Finally, we wrap up with conclusion in Sect. 8.

2 Preliminaries and Definitions

To detail the *Cross-Domain Attribute-Based Access Control Encryption* (CD-ABACE) schemes we need to review some preliminaries. Throughout, 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 the output Y. Also, $[n]$ denotes the set of integers between 1 and n, i.e, the set $\{1, \ldots, n\}$. 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 we have at most $d+1$ coefficients for a polynomial with degree d. 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.

Definition 1 (Access Structure [B+96])**.** 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 $B \subseteq A$ 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 (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 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 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 components of the set $\mathbb{U}$ such that 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 $\mathbb{B} \subset \mathbb{U}$ if and only if $\mathbb{A} \subseteq \mathbb{B}$. 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 4 (Bilinear Groups [BF01])**.** 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 $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is an efficiently computable bilinear map with the following properties;

- $\forall\, a, b \in \mathbb{Z}_p,\ [a]_1 \bullet [b]_2 = [ab]_T = [b]_1 \bullet [a]_2$,
- $[1]_T \neq 1$.

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 $e\left([a]_1, [b]_2\right) = [a]_1 \bullet [b]_2$.

The following definition is proposed by [DP08] in an asymmetric bilinear group as a general Diffie-Hellman exponent theorem [BBG05]. 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)-MSE-DDH assumptions.

Definition 5 (Multi-Sequence of Exponents Diffie-Hellman ((l, m, t)-MSE-DDH) assumption [DP08])**.** Under security parameter λ, let an asymmetric bilinear group generator $\mathcal{BG}(\lambda) = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T \mathsf{p}, \hat{e})$. For given three 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,

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

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

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

Where $\vec{v}_{1-4}$ denotes all vectors $\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{v}_4$.

2.1 System Architecture

The proposed scheme's architecture is based on the Cross-Domain ACE technique described in [WC21]. 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. To this end, there are five entities in this system:

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 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.

[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, that is, from $\mathbb{G}_1 \to \mathbb{G}_2$ and $\mathbb{G}_2 \to \mathbb{G}_1$.

Moreover, it generates the secret encryption keys for the Senders based on a predicate function 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 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 secrete key for decryption. Conversely, if the receiver does not satisfy the access policy then the ciphertext never reveals any 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 sender the authorization. Also, SA uses its master secret key to create the authorized secret encryption keys for the senders corresponding to the predicate function 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. RA, then it is assumed The sender who wants to share a message securely among a group of receivers encrypts the plaintext and proves the validity of the claimed secret encryption key. The SANITIZER receives the sender's request, and checks the validity of the proof and the signature to decide on banning the unauthorized senders without learning their identities. Otherwise, if the sender is – based on the predicate function – authorized to communicate with the selected group of receivers, the SANITIZER re-randomizes the received ciphertext and then passes it on the recipients. Finally, the receivers who are allowed to decrypt ciphertext based on PF, can run the decryption algorithm and retrieve the message, else they learn nothing about it. It is assumed the SANITIZER is not fully trusted: while it follows the protocol honestly, it is unable to compromise the message secrecy and anonymity of the users.

3 Background

In this section, we formally define the primitives required for the proposed construction. Also, we propose a novel CP-ABE scheme with a constant key length and constant ciphertext size. We believe that this is a result of that is valuable by itself.

3.1 Structure-Preserving Signatures

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

Definition 6 (Structure-Preserving Signatures [AFG+10])**.** An SPS scheme $\Pi_{\mathcal{SPS}}$ in a type-III bilinear group, over message space $\mathcal{M}$ and signature space $\mathcal{S}$ consists of four PPT algorithms $(\mathsf{Pgen}, \mathsf{KG}, \mathsf{Sign}, \mathsf{Vf})$, 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.

- $(\sigma) \leftarrow \mathcal{SPS}.\mathsf{Sign}(\mathsf{pp}, \mathsf{sk}, m)$: The signing algorithm takes the public parameters pp, secret signing key sk and a message $m \in \mathcal{M}$ as inputs and outputs a signature $\sigma \in \mathcal{S}$.

- $(0, 1) \leftarrow \mathcal{SPS}.\mathsf{Vf}(\mathsf{pp}, \mathsf{vk}, \sigma, m)$: The verification is a deterministic algorithm that 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 against chosen message attack* that are defined as follows,

Definition 7 (Correctness)**.** An SPS scheme $\Pi_{\mathcal{SPS}}$ is called *correct* if we have,

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

Definition 8 (Existential Unforgeability against Chosen Message Attack (EUF-CMA))**.** An SPS scheme $\Pi_{\mathcal{SPS}}$ is called EUF-CMA if for all PPT adversaries $\mathcal{A}$, $Adv^{\text{EUF-CMA}}_{\mathcal{A},\mathcal{SPS}}(1^\lambda)$, we have the following advantage function,

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

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 to be EUF-CMA-secure if for all PPT adversaries we have, $Adv^{\text{EUF-CMA}}_{\mathcal{A},\mathcal{SPS}}(1^\lambda) \leq \mathsf{negl}(\lambda)$.

In the following, the Abe et al. [AGOT14] SPS construction is outlined, as a selectively re-randomizable SPS in Type-III bilinear groups. This scheme has been proven to be EUF-CMA-secure. If one possess a valid re-randomization token one can re-randomize the signature without needing to know the secret signing key.

Abe et al. SPS scheme [AGOT14] This construction consists of the following PPT algorithms,

- $(\mathsf{pp}) \leftarrow \mathcal{SPS}.\mathsf{Pgen}(\lambda)$: This algorithm takes as input 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, either a plain or a re-randomized signature σ or 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. Since in our main construction the generator of the first group is hidden, then we use a variant of the selectively re-randomizable Abe et al.'s SPS scheme with the same public parameters in the Type-III bilinear group.

- $(\mathsf{pp}) \leftarrow \mathcal{SPS}.\mathsf{Pgen}(\lambda)$: 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 $\mathcal{BG}(1^\lambda) = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathsf{p}, \hat{e})$ 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.
- $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathcal{SPS}.\mathsf{KG}(\mathsf{pp})$: It 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 $\mathsf{sk} = v$.
- $(\sigma, W) \leftarrow \mathcal{SPS}.\mathsf{Sign}(\mathsf{pp}, \mathsf{sk}, m)$: The signing algorithm takes the public parameters pp, the secret key sk and a message $m \in \mathbb{G}_2$ as inputs. 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 a 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 accepts pp, either a plain signature σ or a re-randomized signature σ', a message m and the verification key vk as inputs. 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 scheme, where instead of picking a random X from $\mathbb{G}_1$, we utilize $\left[\alpha^2\right]_1$. Additionally, the proposed variant of Abe et al.'s SPS scheme is EUF-CMA-secure, similar to the original construction of Abe et al. This follows from the observation that α is sampled uniformly random and $\left[\alpha^2\right]_1$ has the same distribution as $[1]_1$. Since α is hidden, no PPT adversary can distinguish between the case $[1]_1$ and $\left[\alpha^2\right]_1$.

3.2 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) arguments remove the interaction between the parties in two possible settings either the Random Oracle Model (ROM) [FS87] or the Common Reference String (CRS) model [BFM88]. The construction of NIZK arguments in the CRS model requires a trusted setup phase that outputs some public parameters, known as the CRS, that it shared with the prover and verifier to respectively generate and verify the proof in a single communication round.

We adopt the definition of Zero-Knowledge Non-Interactive Succinct Argument of Knowledge (zk-SNARK) as an efficient family of the NIZK arguments from [Gro16]. For a security parameter λ, let $\mathcal{R}$ be a relation generator, such that $\mathcal{R}(1^\lambda)$ returns an efficiently computable binary relation $\mathbf{R} = \{(\mathsf{x}, \mathsf{w})\}$, where x is the statement and w is the corresponding witness. Let $\mathbf{L_R} = \{\mathsf{x} : \exists\, \mathsf{w} \mid (\mathsf{x}, \mathsf{w}) \in \mathbf{R}\}$ be the **NP**-language consisting of the statements in the relation $\mathbf{R}$. Formally, a NIZK argument Π_{NIZK} under the relation generator $\mathcal{R}$ consists of the following PPT algorithms:

- $(\vec{\mathsf{crs}}, \vec{\mathsf{ts}}) \leftarrow \mathsf{K}_{\vec{\mathsf{crs}}}(\mathbf{R})$: The CRS generator is a probabilistic algorithm that, given relation $(\mathbf{R})$, first samples the simulation trapdoor $\vec{\mathsf{ts}}$, and generates $\vec{\mathsf{crs}}$. It securely stores the former while publishing the latter.

- $(\pi, \perp) \leftarrow \mathsf{P}(\mathbf{R}, \vec{\mathsf{crs}}, \mathsf{x}, \mathsf{w})$: Prove is a probabilistic algorithm that takes as input ($\mathbf{R}$, $\vec{\mathsf{crs}}, \mathsf{x}, \mathsf{w}$) and if $(\mathsf{x}, \mathsf{w}) \in \mathbf{R}$, outputs a proof π, otherwise, it returns $\perp$.

- $(0, 1) \leftarrow \mathsf{V}(\mathbf{R}, \vec{\mathsf{crs}}, \mathsf{x}, \pi)$: The verification algorithm is a deterministic process that returns 1 if the given proof is correct ($(\mathsf{x}, \mathsf{w}) \in \mathbf{R}$) and 0 if it is incorrect ($(\mathsf{x}, \mathsf{w}) \notin \mathbf{R}$).

- $(\pi') \leftarrow \mathsf{Sim}(\mathbf{R}, \vec{\mathsf{crs}}, \vec{\mathsf{ts}}, \mathsf{x})$: Simulator is an algorithm, that given the tuple $(\mathbf{R}, \vec{\mathsf{crs}}, \vec{\mathsf{ts}}, \mathsf{x})$, outputs a simulated argument 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 argument in the CRS model.

Definition 9 (Completeness)**.** A NIZK argument Π_{NIZK} is called *complete*, if for all λ, and $(\mathsf{x}, \mathsf{w}) \in \mathbf{R}$ we have,

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

Definition 10 (Soundness)**.** A NIZK argument Π_{NIZK} is called *Sound*, if for all adversary $\mathcal{A}$, we have,

$$\Pr\begin{bmatrix}(\mathbf{R}) \leftarrow \mathcal{R}(1^\lambda), (\vec{\mathsf{crs}}, \vec{\mathsf{ts}}) \leftarrow \mathsf{K}_{\vec{\mathsf{crs}}}(\mathbf{R}), \\ (\mathsf{x}, \pi) \leftarrow \mathcal{A}(\mathbf{R}, \vec{\mathsf{crs}}) : \mathsf{V}(\mathbf{R}, \vec{\mathsf{crs}}, \mathsf{x}, \pi) = 0 \ \wedge \ \mathsf{x} \notin \mathbf{L_R}\end{bmatrix} \approx_c 1 \ .$$

Definition 11 (Statistically Zero-Knowledge)**.** A NIZK proof Π_{NIZK} is called *statistically Zero-Knowledge*, if for all adversary $\mathcal{A}$, $\varepsilon_0^{unb} \approx_c \varepsilon_1^{unb}$, where

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

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

Intuitively, a NIZK argument Π_{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 12 (Computational Knowledge-Soundness)**.** A NIZK argument Π_{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,

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

In this paper, we utilize a special and highly efficient class of NIZK arguments in the CRS model, with small proof size and low verification cost, called *Succinct Non-Interactive Arguments of Knowledge* (zk-SNARK). The most efficient zk-SNARK to date has been proposed by Groth [Gro16]: its proof contains only three group elements.

3.3 Re-randomizable CP-ABE schemes

In what follows, we capture a unified definition of Ciphertext-Policy Attribute-Based Encryption (CP-ABE) schemes and their security requirements. Then, we define a novel CP-ABE scheme and its variant as a re-Randomizable CP-ABE scheme, in which one party can re-randomize the generated ciphertext without needing the secret key.

Definition 13. (Ciphertext-Policy Attribute-Based Encryption schemes [BSW07]): 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 $\textsc{Bf} : \Sigma_k \times \Sigma_c \rightarrow \{0, 1\}$ 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 the private decryption key $\mathsf{dk}_{\mathbb{B}}$.
- $(\mathtt{Ct}) \leftarrow \mathcal{ABE}.\mathsf{Enc}(\mathsf{pp}, m, \mathbb{P})$: 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 $m' \in \mathcal{M}$ if and only if $\textsc{Bf}(\mathbb{B}, \mathbb{P}) = 1$, otherwise $\perp$.

We give a standard definition of the security properties for CP-ABE schemes namely, Correctness and Indistinguishability against Chosen Ciphertext Attack (IND-CCA) for CP-ABE schemes.

Definition 14. (Correctness [GPSW06]). Let a CP-ABE scheme for a given security parameter λ and attribute space $\mathbb{U}$, all $\mathbb{B} \in \Sigma_k$ and all $\mathbb{P} \in \Sigma_c$. We say that $\Pi_{\text{CP-ABE}}$ over message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$ is correct if and only if for all $m \in \mathcal{M}$ and $\mathtt{Ct} \in \mathcal{C}$ we have,

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

Definition 15. (IND-CCA [GPSW06]). Let $\Pi_{\text{CP-ABE}}$ be defined for the attribute universe $\mathbb{U}$, message space $\mathcal{M}$ and a Boolean relation $\textsc{Bf} : 2^{\mathbb{U}} \times \Sigma_c \rightarrow \{0, 1\}$. For a security parameter λ and a PPT adversary $\mathcal{A}$, we define the Indistinguishability game under a Chosen Ciphertext Attack (IND-CCA) 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 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 indexes 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 $Adv^{\text{IND-CCA}}_{\mathcal{A},\Pi_{\text{CP-ABE}}}(1^\lambda, b) = 2\left|\Pr[b = b'] - \frac{1}{2}\right|$, where the probability is taken over all coin flips. We say $\Pi_{\text{CP-ABE}}$ is IND-CCA if for all PPT adversaries $\mathcal{A}$ we have,

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

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 [BB04, CHK03], $\mathcal{A}$ 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-CCA secure [AL10], if $\mathcal{A}$ declares q decryption key queries before the initialization phase, but she can adaptively select the challenge index $\mathbb{P}^*$ afterward.

Although an IND-CCA-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 [KSW08]. Since the latter increases the ciphertext size, this will not be considered in this work and our construction only satisfies a weaker notion of attribute-hiding. More precisely, in an ACE construction, the receiver anonymity ensures that the identity of the receivers remains anonymous even against the Sanitizer and the malicious parties. 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 [SSN09]. 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 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.

In the following, we define a new IND-CCA 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 [GMS+14] took a similar approach and presented a constant-key size CP-ABE scheme, the ciphertext size in their scheme increases linearly with total number of possible attributes.

A Constant-size ciphertext CP-ABE scheme This construction consists of the following PPT algorithms,

- $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathcal{ABE}.\mathsf{Pgen}(\mathbb{U}, \lambda)$: Takes an attribute space $\mathbb{U}$ with size n along with 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 collision-resistant hash function $\mathsf{H} \leftarrow\!\!\$\, \mathcal{H}$. 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 $\mathsf{msk} = ([1]_1, \alpha)$ 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 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 $\mathsf{dk}_\mathbb{B} = [1/Z_\mathbb{B}(\alpha)]_1$.

- $(\mathtt{Ct}) \leftarrow \mathcal{ABE}.\mathsf{Enc}(\mathsf{pp}, m, \mathbb{P})$: This algorithm generates a ciphertext for message $m \in \mathcal{M}$, takes 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 $\mathtt{Ct} = (\mathbb{P}, C, C_1, C_2) = (\mathbb{P}, m\,[r\alpha]_T\,, (\prod_{j=0}^{n} h_{j+1}^{z_j})^r = [r\alpha Z_{\mathbb{P}}(\alpha)]_2\,, g_2^{-r} = \left[-r\alpha^2\right]_1)$.
- $(m', \perp) \leftarrow \mathcal{ABE}.\mathsf{Dec}(\mathsf{pp}, \mathtt{Ct}, \mathsf{dk}_{\mathbb{B}})$: 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 $c[i] = b[i] - p[i]$. Then it 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 1. *The proposed CP-ABE scheme is consistent.*

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,

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

□

More precisely, the univariate polynomial $Z_{\mathbb{B}}(x)$ vanishes on the point $k_i = \mathsf{H}(U_i)$ for those attributes that are not in the set of $\mathbb{B}$, 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 polynomial $Z_{\mathbb{B}}(x)$ simplify by the polynomial $Z_{\mathbb{P}}(x)$. Since $|\mathbb{P}| \leq |\mathbb{B}|$ and the result of division $Z_{\mathbb{P}}(x)/Z_{\mathbb{B}}(x)$ is not rational and it is equal to $F_{\mathbb{B},\mathbb{P}}(x)$, hence we can evaluate this polynomial in from the second group by knowing the monomial set $\left[\alpha^i\right]_2$. 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 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, 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. This is because defying the secret value α that is encased as a master secret key thus no two secret keys can create another which benefits more universally. It follows naturally that a malicious user would need to guess correctly the α to cancel out the numerator polynomial caused by the multiplication of least common factor of two distinct decryption attribute set.

Theorem 2. *Under the (l, m, t)-MSE-DDH assumption, a PPT adversary $\mathcal{A}$ cannot win the security game* $\text{IND-CCA}^{\mathcal{A}}_{\textit{CP-ABE}}(1^\lambda, \mathbb{U})$ *from Definition 15 for the proposed CP-ABE scheme.*

Proof. We plan to prove this theorem by reduction. Let there exists a Probabilistic Polynomial Time (PPT) adversary, $\mathcal{A}$, who can break the proposed scheme in the introduced security game in Definition 15 with a non-negligible advantage ϵ. Then we will show that how a PPT adversary, $\mathcal{B}$, can solve the (l,m,t)-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 Decisional (l,m,t)-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 $[1]_T$ be 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 $\vec{v_1} = \left([1]_1, [\alpha]_1, \ldots, \left[\alpha^{l+t-2}\right]_1, [k\alpha f(\alpha)]_1\right)$, $\vec{v_2} = \left([\delta]_1, [\delta\alpha]_1, \ldots, \left[\delta\alpha^{l+t}\right]_1\right)$, $\vec{v_3} = \left([1]_2, [\alpha]_2, \ldots, \left[\alpha^{m-2}\right]_2\right)$ and $\vec{v_4} = \left([\delta]_2, [\delta\alpha]_2, \ldots, \left[\delta\alpha^{2m-1}\right]_2, [kh(\alpha)]_2\right)$ to the adversary $\mathcal{B}$.

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 $\mathsf{H} \leftarrow\$ \mathcal{H}$. $\mathcal{B}$ 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., $\textsc{Bf}(\mathbb{B}_i, \mathbb{P}^*) = 0$). In order to publish the public parameters, $\mathcal{B}$ computes the public parameters computes the following polynomial that is the multiplication of zero-polynomials corresponding for the chosen subsets $\mathbb{B}_i$ in the query set Q.

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

Here $b_i[j]$ represents the j^{th} binary representation of subset $\mathbb{B}_i$. The degree of 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 defining the hash function H in the Random Oracle Model. 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,

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

We have to emphasize that the generator of the first cyclic group is not public. $\mathcal{B}$ by knowing the vector $\vec{v_1}$ 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 $\mathsf{pp} = \left\{G_2, \{h_i\}_{i=0}^{n}, [\alpha f(\alpha)Y_Q(\alpha)]_T, \mathsf{H}\right\}$. While 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 the following equations and sends back the tuple $\mathtt{Ct} = (\mathbb{P}, C, C_1, C_2)$ to $\mathcal{A}$.

$$C = m \left(\prod_{i=0}^{s(n-e)+l} \left(\left[\alpha^i\right]_1 \bullet [\alpha]_2 \right)^{p_i} \right)^r = m \left[r\alpha f(\alpha)Y_Q(\alpha)\right]_T .$$

$$C_1 = \left(\prod_{i=0}^{n} \left[\alpha^{i+1} z_i\right]_2 \right)^r = [r\alpha Z_{\mathbb{P}}(\alpha)]_2 .$$

$$C_2 = G_2^{-r} .$$

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,

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

Challenge: The adversary $\mathcal{A}$ chooses two same length plaintexts $\{m_0, m_1\} \leftarrow\!\$\, \mathcal{M} \times \mathcal{M}$ and sends them to $\mathcal{B}$. Then $\mathcal{B}$ flips a fair coin to have the biased 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 $r^* = k/(\alpha Y_Q(\alpha))$ as the randomness for the challenge ciphertext. In a nutshell, based on the (l, m, t)-MSE-DDH assumption, there are two cases for the received challenge Γ with the same probability 1/2. If $\Gamma = [kf(\alpha)]_T$ then $C^* = m_b [kf(\alpha)]_T = m_b [r^*\alpha f(\alpha)Y_Q(\alpha)]_T$ is in the correct format. Also, $C_1^* = [kh(\alpha)]_2 = [r^*\alpha Y_Q(\alpha)h(\alpha)]_2 = [r^*\alpha Z_{\mathbb{P}^*}(\alpha)]_2$ and $C_2^* = [-k\alpha f(\alpha)]_1 = \left[-r^*\alpha^2 Y_Q(\alpha)f(\alpha)\right]_1 = G_2^{-r^*}$. 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 can distinguish by chance.

2^{nd} **Query phase.** After receiving the challenge ciphertext $\mathtt{Ct}^*$, the adversary $\mathcal{A}$ has access to the queries defined in the first phase on the condition that she cannot query the decryption oracle for the received challenge ciphertext.

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)-MSE-DDH problem is,

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

Therefore, the adversary $\mathcal{B}$ can play the (l, m, t)-MSE-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)-MSE-DDH assumption with a non-negligible advantage, then the proposed CP-ABE scheme in Sect. 3 is secure in the IND-CCA game in Definition 15. $\square$

Definition 16 (Re-randomizable CP-ABE schemes (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, $\Pi_{r\mathcal{ABE}}$, for a Boolean relation $\textsc{Bf} : \Sigma_k \times \Sigma_c \to \{0, 1\}$, over message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$, coincides with the algorithms from Definition 13; the following algorithm supports this expansion:

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

The Correctness and IND-CCA-security of a Re-randomizable CP-ABE derives naturally from the initial CP-ABE, specified in Definitions 14 and 15. The decryption functions in the former can thus accept either a ciphertext $\texttt{Ct}$ or a re-randomized ciphertext $\tilde{\texttt{Ct}} \in \mathcal{C}$, 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 a re-randomized ciphertext and the initial ciphertext.

A Re-randomizable CP-ABE In what follows, we define a variant of the proposed IND-CCA-secure CP-ABE scheme in Section 3.3, that supports the re-randomizing phase as follows:

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

The other algorithms are the same, except the decryption algorithm can take either $\tilde{\texttt{Ct}}$ or $\texttt{Ct}$ as input, and the same security properties hold.

3.4 Cross-Domain Attribute-Based Access Control Encryption scheme

We introduce the notion of *Cross-Domain Attribute-Based Access Control Encryption* (CD-ABACE) schemes as an extended version of a re-randomizable CP-ABE construction. 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 to the predicate function in an ACE construction. In this scenario, the encryption key generator allows the sender the ability to send a message to only those receivers who align with a certain access structure based on a given predicate function. Properly, in place of the original approach of specifying the ciphertext access rights in the encryption phase, in this approach, it is specified by Sender Authority in the encryption key generator 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 17 (Cross-Domain Attribute-Based Access Control Encryption schemes)**.** A CD-ABACE scheme $\Pi_{\text{CD-ABACE}}$ 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 pair of public parameters pp_{ra} and master secret key msk_{ra} of the RA.

- $(\mathsf{pp}_{sa}, \mathsf{msk}_{sa}) \leftarrow \mathsf{SAgen}(\mathsf{pp}_{ra})$: This probabilistic algorithm takes pp_{ra} as input and generates the parameters of the NZIK and the SPS. parameters. It returns pp_{sa} and msk_{sa} as the SA's public parameters and the master secret key, respectively.

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

- $(\mathsf{ek}_{\mathbb{P}}, \sigma, W) \leftarrow \mathsf{EncKGen}(\mathsf{pp}_{ra}, \mathsf{pp}_{sa}, \mathsf{msk}_{sa}, \mathbb{P}, \mathbf{Pf})$: This algorithm takes the public parameters, the Sender Authority's master secret key msk_{sa}, and the authorized set of attributes $\mathbb{P} \in \Sigma_c$ along with the predicate function $\textsc{Pf}$ 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.

- $(\mathtt{Ct}, \pi, \mathsf{x}) \leftarrow \mathsf{Enc}(\mathsf{pp}_{ra}, \mathsf{pp}_{sa}, m, \mathsf{ek}_{\mathbb{P}}, \sigma, W)$: 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 the ciphertext $\mathtt{Ct}$ and a NIZK proof π along with its underlying statement.

- $(\tilde{\mathtt{Ct}}, \perp) \leftarrow \mathsf{Sanitization}(\mathsf{pp}_{ra}, \mathsf{pp}_{sa}, \mathtt{Ct}, \pi, \mathsf{x}, \mathbf{Pf})$: This algorithm takes as inputs the public parameters pp_{ra} and pp_{sa}, a ciphertext along with a NIZK proof π and its corresponding statement x. Afterwards, the algorithm either re-randomizes the ciphertext to $\tilde{\mathtt{Ct}}$ or bans the request. To this end, it checks the validity of the proof and, if it allows this flow based on the predicate function $\textsc{Pf}$, it transfers the ciphertext $\tilde{\mathtt{Ct}} \in \mathcal{C}$ to the receivers, else it returns $\perp$.

- $(m', \perp) \leftarrow \mathsf{Dec}(\mathsf{pp}_{ra}, \mathsf{pp}_{sa}, \tilde{\mathtt{Ct}}, \mathsf{dk}_{\mathbb{B}})$: The decryption algorithm takes as inputs the public parameters pp_{ra} and pp_{sa}, a re-randomized ciphertext $\tilde{\mathtt{Ct}}$ and the decryption key $\mathsf{dk}_{\mathbb{B}}$. If $\textsc{Pf}(\mathbb{B}, \mathbb{P}) = 1$, 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.

4 Security Definitions

In this section, we present the required security properties for a CD-ABACE scheme: Correctness, No-Read rule and No-Write rule. It must be noted that the following security games are motivated by the notion of co-selective security in [AL10], 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 [WC21] to adapt them to the CD-ABACE system model.

Table 2: Security Games

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

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

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 $(\mathsf{dk}_{\mathbb{B}_j})$

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

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

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

Initialize $\mathcal{Q}_{\mathcal{E}} = \{\emptyset\}$
if $\mathbb{P}_i \notin \mathcal{Q}_{\mathcal{E}}$:
  $(\mathsf{ek}_{\mathbb{P}_i}, \sigma_i, W_i) \leftarrow \mathsf{EncKGen}(\mathbb{P}_i, \textsc{Pf})$
  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 $\mathcal{O}_{\mathsf{Sanitization}}(m, \mathbb{P}_i)$

$(\mathsf{ek}_{\mathbb{P}_i}, \sigma_i, W_i) \leftarrow \mathsf{EncKGen}(\mathbb{P}_i, \textsc{Pf})$
$(\tilde{\mathtt{Ct}}) \leftarrow \mathsf{Sanitization}(\mathsf{Enc}(m, \mathsf{ek}_{\mathbb{P}_i}))$
return $(\tilde{\mathtt{Ct}})$

Oracle $\mathcal{O}_{\mathsf{Dec}}(\tilde{\mathtt{Ct}}, \mathbb{B}_j)$

if $\textsc{Pf}(\mathbb{B}_j, \mathbb{P}_i) = 1$:
  $\mathsf{dk}_{\mathbb{B}_j} \leftarrow \mathsf{DecKGen}(\mathbb{B}_j)$
  $m \leftarrow \mathsf{Dec}(\tilde{\mathtt{Ct}}, \mathsf{dk}_{\mathbb{B}_j})$
else :
  return $\perp$

Definition 18 (Correctness)**.** For a given attribute universe $\mathbb{U}$ and predicate function $\textsc{Pf} : \Sigma_k \times \Sigma_c \to \{0,1\}$, we say that $\Pi_{\text{CD-ABACE}}$ over message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$ is correct if we have,

$$\Pr\left[\mathsf{Dec}\left(\mathsf{dk}_{\mathbb{B}}, \mathsf{Sanitization}(\mathsf{Enc}(m, \mathsf{ek}_{\mathbb{P}}, \mathbb{P}))\right) = m : \textsc{Pf}(\mathbb{B}, \mathbb{P}) = 1 \quad\right] \approx_c 1 \ .$$

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 $\textsc{Pf}(\mathbb{B}, \mathbb{P}) = 1$ with a high probability. In this case, the $\textsc{Sanitizer}$ 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 19 ($\textsc{No-Read}$ Rule)**.** Consider $\Pi_{\text{CD-ABACE}}$ over the attribute universe $\mathbb{U}$, message space $\mathcal{M}$, a ciphertext space $\mathcal{C}$ and a predicate function $\textsc{Pf} : \Sigma_k \times \Sigma_c \to \{0,1\}$. For a security parameter λ, we say that a PPT adversary $\mathcal{A}$ wins the defined $\textsc{No-Read}$ rule security game described in Table 2 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

$\textsc{Pf}(\mathbb{B}_j, \mathbb{P}^*) = 1$. We say $\Pi_{\text{CD-ABACE}}$ satisfies the No-Read rule if for all PPT adversaries $\mathcal{A}$ with advantage $Adv^{\textsc{No-Read}}_{\Pi_{\text{CD-ABACE}},\mathcal{A}}(1^\lambda, b) = (\Pr[\mathcal{A} \text{ wins the No-Read game}] - 1/2)$ we have,

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

When we call $\mathcal{A}$, it 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 an unauthorized receiver should be unable to read the message but also combining the decryption secret keys of a group of unauthorized receivers should not reveal any information about the message. Moreover, this property has to hold even if the recipients collude with the Sanitizer.

Definition 20 (Parameterized No-Write Rule)**.** Consider $\Pi_{\text{CD-ABACE}}$ over the attribute universe $\mathbb{U}$, a message space $\mathcal{M}$, ciphertext space $\mathcal{C}$ and a predicate function $\textsc{Pf} : \Sigma_k \times \Sigma_c \to \{0, 1\}$. We say a $\Pi_{\text{CD-ABACE}}$ scheme satisfies the Parameterized No-Write rule, if no PPT adversary $\mathcal{A}$ with access to the oracles in Table 2 has a non-negligible advantage in winning the No-Write game, i.e, under the advantage $Adv^{\textsc{No-Write}}_{\Pi_{\text{CD-ABACE}},\mathcal{A}}(1^\lambda, b) = (\Pr[\mathcal{A} \text{ wins No-Write}] - 1/2)$ we have,

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

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 $\textsc{Pf}(\mathbb{B}_j, \mathbb{P}_i) = 0$. The function $\mathsf{fix}(.)$ accepts a ciphertext $\mathtt{Ct}$ as input and generate auxiliary information aux of $\mathtt{Ct}$ that is not sanitizable [WC21]. By seeding an encryption algorithm with this auxiliary information, the resulting ciphertext has also the same auxiliary information.

5 The proposed construction

In this section we propose the CD-ABACE scheme such that the key and ciphertext size are constant. Select a predicate function $\textsc{Pf} : \Sigma_k \times \Sigma_c \to \{0, 1\}$ and a family of collision-resistant hash functions $\mathcal{H} : \{0, 1\}^* \to \mathbb{Z}_p^*$. The proposed CD-ABACE scheme consists of the following six PPT algorithms:

$(\mathsf{pp}_{ra}, \mathsf{msk}_{ra}) \leftarrow \mathsf{RAgen}(\mathbb{U}, \lambda)$: This algorithm is run by the RA to set up the public parameters as follows,

$(\mathsf{pp}_{ra}, \mathsf{msk}_{ra}) \leftarrow \mathsf{RAgen}(\mathbb{U}, \lambda)$	
1 :	Run $\mathcal{BG}(\lambda) = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathsf{p}, \hat{e})$ // Type-III bilinear group
2 :	$\mathsf{H} \leftarrow\$ \mathcal{H}$ // Collision-resistant hash function
3 :	$\alpha \leftarrow\$ \mathbb{Z}_p^*$ // Samples a random integer
4 :	$h_i = \left[\alpha^i\right]_2$ // The set of monomials in $\mathbb{G}_2$
5 :	$g_2 = \left[\alpha^2\right]_1$ // only one element in $\mathbb{G}_1$
6 :	$\mathsf{msk}_{ra} = ([1]_1, \alpha)$ // Store them securely
7 :	$\mathsf{pp}_{ra} = (g_2, \{h_i\}_{i=0}^n, [\alpha]_T, \mathsf{H})$ // generator of $\mathbb{G}_1$ is secret
8 :	**return** $(\mathsf{msk}_{ra}, \mathsf{pp}_{ra})$

$(\mathsf{pp}_{sa}, \mathsf{msk}_{sa}) \leftarrow \mathsf{SAgen}(\mathsf{pp}_{ra})$: The SA takes the public parameters pp_{ra} and runs the following algorithm to generate pp_{sa} and msk_{sa}.

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

- Parse $(\mathcal{BG}(\lambda), \mathsf{pp}_{ra})$
- $Y \leftarrow\$\ \mathbb{G}_2$ // Samples a random generator
- $\mathsf{sk} = v \leftarrow\$\ \mathbb{Z}_{\mathsf{p}}$ // Picks a random integer uniformly
- $\mathsf{vk} = g_2^v = \left[\alpha^2 v\right]_1$ // SPS's verification key
- $(\vec{\mathsf{crs}}, \vec{\mathsf{ts}}) \leftarrow\$\ \mathsf{KG}_{\vec{\mathsf{crs}}}(\mathbf{R})$ // CRS generator phase
- $\mathsf{msk}_{sa} = (\mathsf{sk}, \vec{\mathsf{ts}})$
- $\mathsf{pp}_{sa} = (\mathbf{R_L}, \vec{\mathsf{crs}}, Y, \mathsf{vk})$
- **return** $(\mathsf{msk}_{sa}, \mathsf{pp}_{sa})$

$(\mathsf{dk}_{\mathbb{B}}) \leftarrow \mathsf{DecKGen}(\mathsf{msk}_{ra}, \mathbb{B})$: The RA executes the following algorithm to generate the private decryption key $\mathsf{dk}_{\mathbb{B}}$ underlying the key index $\mathbb{B} \subset \mathbb{U}$.

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

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

$(\mathsf{ek}_{\mathbb{P}}, \sigma, W) \leftarrow \mathsf{EncKGen}(\mathsf{pp}_{ra}, \mathsf{pp}_{sa}, \mathsf{msk}_{sa}, \mathbb{P}, \mathbf{Pf})$: The SA takes as inputs the public parameters along with msk_{sa} and executes this algorithm to decide the authorized receivers for each sender based on a given predicate function PF.

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

- Parse $(\mathcal{BG}(\lambda), \mathsf{pp}_{ra}, \mathsf{msk}_{sa})$
- $Z_{\mathbb{P}}(x) = \prod_{i=1}^{n} (x - k_i)^{\overline{p[i]}} = \sum_{j=0}^{n} z_i x^i$
- $\mathsf{ek}_{\mathbb{P}} = \prod_{i=0}^{n} h_{i+1}^{z_i} = [\alpha Z_{\mathbb{P}}(\alpha)]_2$ // User u can write to $\mathbb{P}$
- $t_u \leftarrow\$\ \mathbb{Z}_p^*$ // Unique random integer
- $\sigma = (R, S, T) = (g_2^{t_u}, \mathsf{ek}_{\mathbb{P}}^{\mathsf{sk}/t_u} Y^{1/t_u}, S^{\mathsf{sk}/t_u}\ [1/t_u]_2)$
- $W = [1/t_u]_2$ // Re-randomizing token
- **return** $(\mathsf{ek}_{\mathbb{P}}, \sigma = (R, S, T), W)$

$(\mathtt{Ct}, \pi, \mathsf{x}) \leftarrow \mathsf{Enc}\,(\mathsf{pp}_{sa}, \mathsf{pp}_{ra}, m, \mathsf{ek}_{\mathbb{P}}, \sigma, W)$: The sender who owns the secret encryption key $\mathsf{ek}_{\mathbb{P}}$ runs this algorithm and sends the ciphertext and the NIZK proof (π, x) to the SANITIZER.

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

Parse $(\mathcal{BG}(\lambda), \mathsf{pp}_{ra}, \mathsf{pp}_{sa})$

$r, t \leftarrow\$ \mathbb{Z}_p^*$ // Samples two random integers

$\mathsf{Ct} = (\mathbb{P}, C, C_1, C_2) = (\mathbb{P}, m\,[r\alpha]_T, \mathsf{ek}_{\mathbb{P}}^r, g_2^{-r})$

$R' = R^{1/t},\ S' = S^{t^2},\ T' = T^{t^2} \cdot W^{t(1-t)}$

$\sigma' = (R', S', T')$ // Re-randomized signature

$\mathsf{vk}' = \mathsf{vk}^{1/t}$ // Re-randomizes the verification key

$\mathsf{ek}'_{\mathbb{P}} = \mathsf{ek}_{\mathbb{P}}^t$ // Re-randomizes the encryption key

$\mathsf{x} = (R', S', T', \mathsf{vk}', \mathsf{ek}'_{\mathbb{P}}, C, C_2, C_1)$ // Public Statement

$\mathsf{w} = (\mathsf{ek}_{\mathbb{P}}, R, S, T, m, r, t)$ // Secret Witnesses

$\pi \leftarrow \mathsf{P}(\mathbf{R_L}, \vec{\mathsf{crs}}, \mathsf{w}, \mathsf{x})$ // NIZK prove function

return $(\mathsf{Ct}, \pi, \mathsf{x})$

$(\tilde{\mathsf{Ct}}, \perp) \leftarrow \mathsf{Sanitization}(\mathsf{pp}_{sa}, \mathsf{pp}_{ra}, \mathsf{Ct}, \pi, \mathsf{x})$**:** The Sanitizer decides by running this probabilistic algorithm whether to re-randomize and transfer the received ciphertext or to ignore it.

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

Parse $(\mathcal{BG}(\lambda), \mathsf{pp}_{ra}, \mathsf{pp}_{sa})$

if $\{R' \in \mathbb{G}_1 \wedge \mathsf{ek}'_{\mathbb{P}}, S', T' \in \mathbb{G}_2\ \wedge R' \bullet S' = (\mathsf{vk}' \bullet \mathsf{ek}'_{\mathbb{P}})(g_2 \bullet Y) \wedge$

$R' \bullet T' = (\mathsf{vk}' \bullet S')(g_2 \bullet [1]_2)\ \wedge \mathsf{V}(\mathbf{R_L}, \vec{\mathsf{crs}}, \pi, \mathsf{x}) = 1\}$:

$$Z_{\mathbb{P}}(x) = \prod_{i=1}^{n} (x - k_i)^{\overline{p[i]}} = \sum_{j=0}^{n} z_j x^j$$

$s \leftarrow\$ \mathbb{Z}_{\mathsf{p}}^*$ // Samples a secret random integer

$(\tilde{C}, \tilde{C}_1, \tilde{C}_2) = (C \cdot [s\alpha]_T, C_1 \cdot [s\alpha Z_{\mathbb{P}}(\alpha)]_2, C_2 \cdot g_2^{-s})$

return $\tilde{\mathsf{Ct}} = (\mathbb{P}, \tilde{C}, \tilde{C}_1, \tilde{C}_2)$

else : **abort**

$(m', \perp) \leftarrow \mathsf{Dec}(\mathsf{pp}_{sa}, \mathsf{pp}_{ra}, \tilde{\mathsf{Ct}}, \mathsf{dk}_{\mathbb{B}})$: A receiver can run this algorithm to decrypt a ciphertext $\tilde{\mathsf{Ct}}$. The receiver who owns the attribute set of $\mathbb{B}$ can learn the message if and only if $\mathbb{P} \subseteq \mathbb{B}$, otherwise this algorithm returns $\perp$.

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

Parse $(\mathcal{BG}(\lambda), \mathsf{pp}_{ra}, \mathsf{pp}_{sa})$

if $\mathbb{P} \subseteq \mathbb{B}$: $c[i] = b[i] - p[i]$

$$F_{\mathbb{B},\mathbb{P}}(x) = \prod_{i=1}^{n} (x - k_i)^{c[i]} = \sum_{j=0}^{n} f_j x^j$$

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

else : **abort**

Our suggested CD-ABACE scheme is under an AND-gate circuit CP-ABE scheme with constant key and ciphertext sizes, although this can be considered as a generic construction that can be used with any CP-ABE scheme based on more substantial circuit-level such as LSSS, Boolean, etc.

6 Security Analysis

This section examines the security requirements of the proposed CD-ABACE scheme based on three theorems.

Theorem 3. *The proposed construction in Section 5, satisfies the correctness property of Definition 18.*

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 $\mathrm{PF}(\mathbb{B}, \mathbb{P}) = 1$. 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 7 and also the completeness of intended NIZK proof of Definition 9. The latter also comes from the consistency of the proposed CP-ABE scheme that we discussed in Theorem 1 and consequently, its re-randomizable variant. Thus we can conclude the proposed CD-ABAC scheme is correct. □

Theorem 4. *The proposed CD-ABACE scheme satisfies the* NO-READ *rule of Definition 19.*

Proof. We wish to make the argument that for all PPT adversaries $\mathcal{A}$, no player can distinguish between two possible scenarios: the case that in the NO-READ security game $b = 0$ constitutes one scenario which we denote by H_0, and the case that b is fixed to 1, called H_1. I.e., $(\mathtt{Ct}_0, \pi_0, \mathsf{x}_0) \approx_c (\mathtt{Ct}_1, \pi_1, \mathsf{x}_1)$. 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 NIZK arguments defined in Definition 11 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 NIZK arguments, 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 message m_{1-b} instead of m_b.

According to the IND-CCA 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 , $H_0 \approx_c H_0^1 \approx_c H \approx_c H_1^1 \approx_c H_1$. □

Theorem 5. *No PPT adversary $\mathcal{A}$ can win the* NO-WRITE *security game of Definition 20 for the proposed CD-ABACE scheme under a fixed predicate function* $\mathrm{PF} : \Sigma_k \times \Sigma_c \rightarrow \{0, 1\}$.

Proof. The proof technique is inspired by [KW17, WC21]'s NO-WRITE rule proof strategies. The following experiments rely on security properties of the cryptographic primitives, namely the knowledge soundness of the NIZK, the existential unforgeability of the SPS and the IND-CCA security of the rCP-ABE. By playing a sequence of indistinguishable games between a PPT adversary $\mathcal{A}$ and the challengers $\mathcal{B}_{\mathrm{KS}}$, $\mathcal{B}_{\mathrm{EUF\text{-}CMA}}$ and $\mathcal{B}_{\mathrm{IND\text{-}CCA}}$, we gradually turn the NO-WRITE rule game into the security features of the underlying primitives.

- G_0**:** The first security game is the defined No-Write game in Definition 20, thus we can write,

$$Adv^{\text{No-Write}}_{\Pi_{\text{CD-ABAC}},\mathcal{A}}(1^\lambda, b) = \Pr[\mathcal{A} \text{ Wins } G_0] \ .$$

- 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 $(\mathtt{Ct}_0, \pi_0, \mathsf{x}_0)$, and returns the corresponding witness $(\mathsf{w}_0) \leftarrow \mathsf{Ext}(\vec{\mathsf{te}}, \mathtt{Ct}_0, \pi_0)$ 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 arguments, specified in Definition 10. 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.,

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

  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 8. We can write,

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

- G_3**:** This game is the same as 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 $Adv^{\text{IND-CCA}}_{rCP-ABE}(\mathcal{B}_{\text{IND-CCA}})$ and we can write,

$$\begin{aligned}
&Adv^{\text{No-Write}}_{\Pi_{\text{CD-ABAC}},\mathcal{A}}(1^\lambda, b) \le Adv^{\text{KS}}_{\text{NIZK}}(\mathcal{B}_{\text{KS}}) + \\
&Adv^{\text{EUF-CMA}}_{\text{SPS}}(\mathcal{B}_{\text{EUF-CMA}}) + Adv^{\text{IND-CCA}}_{rCP-ABE}(\mathcal{B}_{\text{IND-CCA}}) \ .
\end{aligned}$$

Thereby we can conclude,

$$Adv^{\text{No-Write}}_{\Pi_{\text{CD-ABAC}},\mathcal{A}}(1^\lambda, b) \le \mathsf{negl}(\lambda) \ .$$

□

7 Performance Analysis

In this section, we examine how the performance of our proposed CD-ABACE scheme compares to the selectively-secure Wang and Chow scheme [WC21], which is the only implemented ACE construction to date.

As Table 4 illustrates, 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. In [WC21], since the second group generator is hidden, the SA requires selecting a new generator to create the parameters of the signature scheme. In contrast, the proposed variant of Abe et al.'s SPS requires no new generator for the second cyclic group, and the intended NIZK proof cuts out the need for target group operations.

We analyze the scheme's performance based on the implementation results on Wang and Chow scheme [WC21], which was conducted on Windows 10 Enterprise with an Intel Core i7-3770 CPU at 3.40 GHz with 16 GB of memory. The paper applies the JPBC framework [DCI11], a Java library for the Pairing-based Cryptography [Lyn06] in order to achieve portability. Table 3 lists the size of the groups' elements and the exponentiation running time and pairing cost. Note that for exponentiation it is taken into account pre-processing, but for pairing there is no pre-processing

Table 3: Size of elements and Cost of operations [WC21]

Parameter	$\lvert\mathbb{Z}_\mathsf{p}\rvert$	$\lvert\mathbb{G}_1\rvert$	$\lvert\mathbb{G}_2\rvert$	$\lvert\mathbb{G}_T\rvert$
size (byte)	58	116	232	696
Parameter	$\mathbf{E}_1$	$\mathbf{E}_2$	$\mathbf{E}_T$	$\mathbf{P}$
Time (ms)	3	5	22	468

Based on the experiments in Table 3 and the performance given in Table 4, we can determine the overhead introduced by the ciphertext's length, encryption and secret decryption key and the public parameters sizes (and compare them with [WC21]). As an example, assume $n = 1000$ as the total number of attributes (the total number of users), and $t = 400$ as the maximum number of attributes specified in the access policy ($r = 400$ as the maximum number of receivers that any sender is allowed to communicate with), and $w = 500$ as the maximum number of attributes owned by a receiver ($s = 500$ as the maximum number of senders that any receiver can receive a message from). The size of the public parameters in the network is equal to 140 360 bytes (140 476 bytes). The ciphertext size in our construction is 1972 bytes independent of the intended attributes and the number of receivers (while the ciphertext in [WC21] is 3712 bytes long). Moreover, the memory required to store the secret encryption and private decryption keys is 1044 bytes (696 bytes) and 116 bytes (116 000 bytes), respectively. The encryption algorithm's runtime in a pre-processed setting is 56 ms (71 ms), and the decryption algorithm takes 1458 ms (3458 ms) to process.

Although the authors in [WC21] examined the NIZK proof system in the Random Oracle Model, for the evaluation of the intended NIZK argument, we assess zk-SNARKs based on the pairing-friendly elliptic curve `BLS12-381`. We use the JubJub curve [jub20] explored by Zcash for fast elliptic-curve arithmetic operations in the circuit. The JubJub curve is a twisted Edwards curve defined over $\mathbb{F}_q$ with q being the prime order of `BLS12-381`. Among other features, the Sapling algorithm in Zcash uses the Jubjub curve to prove relations of the form $y = \beta g^\alpha$ to determine that α is in the correct interval for the witness α [jub20]. The first part of the relation can be expressed with 756 constraints, but the latter is made of 252 constraints, hence a total of 1008 constraints [jub20, Section A.4]. The former is all we need in our setting; it requires 756 constraints for each case of exponentiation. In the encryption phase, the sender should prove the knowledge of exponent for eight different relations including, $(C, C_1, C_2, R', S', T', \mathsf{vk}', \mathsf{ek}'_{\mathbb{P}})$. In total, this

Table 4: Performance Analysis. $|\mathbb{G}_i|$: The bit length of elements and $\mathbf{E}_i$: The exponentiation cost in $\mathbb{G}_i$ for $i \in \{1, 2, T\}$. $\mathbf{P}$: The pairing cost.

Scheme	Public parameters	Ciphertext size	Enc. Key size	Dec. Key size	Enc. cost	Dec. cost
[WC21]	$(r+3)\|\mathbb{G}_1\| + \|\mathbb{G}_T\| + (r+1)\|\mathbb{G}_2\|$	$10\|\mathbb{Z}_\mathsf{p}\| + 7\|\mathbb{G}_1\| + \|\mathbb{G}_2\| + 3\|\mathbb{G}_T\|$	$4\|\mathbb{G}_1\| + \|\mathbb{G}_2\|$	$s\|\mathbb{G}_2\|$	$4\mathbf{E}_1 + 3\mathbf{E}_2 + 2\mathbf{E}_T$	$s\mathbf{E}_2 + \mathbf{E}_T + 2\mathbf{P}$
This work	$2\|\mathbb{G}_1\| + \|\mathbb{G}_T\| + (n-t+1)\|\mathbb{G}_2\|$	$3\|\mathbb{G}_1\| + 4\|\mathbb{G}_2\| + \|\mathbb{G}_T\|$	$\|\mathbb{G}_1\| + 4\|\mathbb{G}_2\|$	$\|\mathbb{G}_1\|$	$3\mathbf{E}_1 + 5\mathbf{E}_2 + \mathbf{E}_T$	$(w-t)\mathbf{E}_2 + \mathbf{E}_T + 2\mathbf{P}$

circuit requires 6048 constraints and we have implemented the most efficient zk-SNARK to date proposed by Groth [Gro16] using the libSNARK library [lib14].

The proof systems on the instance R1CS are benchmarked with 6048 constraints and 6048 variables, of which 10 are input variables. A CPU with a clock speed of 2.50 GHz and 16 GB of RAM was used in the benchmarks. At the bandwidth level, the CRS generation phase requires 906.5 ms, and the generated CRS is 1 207 662 bytes long. The proof phase takes 964 ms, while the length of proof is equal to 127 bytes (three group elements) independent of the number of attributes and any other variables. Moreover, the verification algorithm can be performed in 1.1 ms.

8 Conclusion

In this work, we proposed an efficient and secure *Cross-Domain Attribute-Based Access Control Encryption* scheme that is based on the set of attributes that the users possess. In comparison with the previous paper, the length of the secret decryption and encryption keys and the ciphertext size has been substantially reduced to a constant number of cyclic groups elements. 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 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. Consequently, applying a Boolean circuit can be one step ahead to improve the anonymity of the receivers to a stronger notion.

Acknowledgements

We would like to thank Sherman S. M. Chow, Georg Fuchsbauer, Karim Baghery, Ward Beullens and Pavel Hubáček 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

[AC17] Shashank Agrawal and Melissa Chase. Fame: fast attribute-based message encryption. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 665–682, 2017.

[AFG$^+$10] Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Structure-preserving signatures and commitments to group elements. In *Annual Cryptology Conference*, pages 209–236. Springer, 2010.

[AGOT14] Masayuki Abe, Jens Groth, Miyako Ohkubo, and Mehdi Tibouchi. Unified, minimal and selectively randomizable structure-preserving signatures. In *Theory of Cryptography Conference*, pages 688–712. Springer, 2014.

[AL10] Nuttapong Attrapadung and Benoît Libert. Functional encryption for inner product: Achieving constant-size ciphertexts with adaptive security or support for negation. In *International Workshop on Public Key Cryptography*, pages 384–402. Springer, 2010.

[B$^+$96] Amos Beimel et al. Secure schemes for secret sharing and key distribution. 1996.

[BB04] Dan Boneh and Xavier Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *International conference on the theory and applications of cryptographic techniques*, pages 223–238. Springer, 2004.

[BBG05] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 440–456. Springer, 2005.

[BF01] Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. In *Annual international cryptology conference*, pages 213–229. Springer, 2001.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 103–112. ACM, 1988.

[BL73] D Elliott Bell and Leonard J LaPadula. Secure computer systems: Mathematical foundations. Technical report, MITRE CORP BEDFORD MA, 1973.

[BSW07] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *2007 IEEE symposium on security and privacy (SP'07)*, pages 321–334. IEEE, 2007.

[CHK03] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 255–271. Springer, 2003.

[CN07] Ling Cheung and Calvin Newport. Provably secure ciphertext policy ABE. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 456–465, 2007.

[CZF11] Cheng Chen, Zhenfeng Zhang, and Dengguo Feng. Efficient ciphertext policy attribute-based encryption with constant-size ciphertext and constant computation-cost. In *International Conference on Provable Security*, pages 84–101. Springer, 2011.

[DCI11] Angelo De Caro and Vincenzo Iovino. jpbc: Java pairing based cryptography. In *2011 IEEE symposium on computers and communications (ISCC)*, pages 850–855. IEEE, 2011.

[DHO16] Ivan Damgård, Helene Haagh, and Claudio Orlandi. Access control encryption: Enforcing information flow with cryptography. In *Theory of Cryptography Conference*, pages 547–576. Springer, 2016.

[DP08] 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.

[EMN$^+$09] 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.

[FGKO17] Georg Fuchsbauer, Romain Gay, Lucas Kowalczyk, and Claudio Orlandi. Access control encryption for equality, comparison, and more. In *IACR International Workshop on Public Key Cryptography*, pages 88–118. Springer, 2017.

[FS87] 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.

[GGH+16] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *SIAM Journal on Computing*, 45(3):882–929, 2016.

[GMS+14] 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*, 9(5):763–771, 2014.

[GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 89–98, 2006.

[Gro16] Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 305–326. Springer, Heidelberg, May 2016.

[HS16] Hanshu Hong and Zhixin Sun. An efficient and secure attribute based signcryption scheme with LSSS access structure. *SpringerPlus*, 5(1):644, 2016.

[jub20] Zcash protocol specification: [overwinter+sapling+blossom+heartwood]. 2020.

[KSW08] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *annual international conference on the theory and applications of cryptographic techniques*, pages 146–162. Springer, 2008.

[KW17] Sam Kim and David J Wu. Access control encryption for general policies from standard assumptions. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 471–501. Springer, 2017.

[lib14] Scipr lab. libsnark: a c++ library for zksnark proofs. Available: https://github.com/sciprlab/libsnark, First release Jun 2, 2014.

[LW11] Allison Lewko and Brent Waters. Decentralizing attribute-based encryption. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 568–588. Springer, 2011.

[Lyn06] Ben Lynn. Pbc library manual 0.5. 11, 2006.

[LZN+20] Jiguo Li, Yichen Zhang, Jianting Ning, Xinyi Huang, Geong Sen Poh, and Debang Wang. Attribute based encryption with privacy protection and accountability for Cloud-IoT. *IEEE Transactions on Cloud Computing*, 2020.

[OSM00] 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.

[RD14] Y Sreenivasa Rao and Ratna Dutta. Expressive bandwidth-efficient attribute based signature and signcryption in standard model. In *Australasian Conference on Information Security and Privacy*, pages 209–225. Springer, 2014.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):1–40, 2009.

[SAMA17] 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.

[SM03] Andrei Sabelfeld and Andrew C Myers. Language-based information-flow security. *IEEE Journal on selected areas in communications*, 21(1):5–19, 2003.

[SSN09] Siamak F Shahandashti and Reihaneh Safavi-Naini. Threshold attribute-based signatures and their application to anonymous credential systems. In *International Conference on Cryptology in Africa*, pages 198–216. Springer, 2009.

[SW05] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 457–473. Springer, 2005.

[TDM12] 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.

[TZMT17] 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.

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

[WC21] Xiuhua Wang and Sherman S. M. Chow. Cross-domain access control encryption: Arbitrary-policy, constant-size, efficient. *IEEE Symposium on Security and Privacy (S&P)*, 2021.

[YKM14] Younis A Younis, Kashif Kifayat, and Madjid Merabti. An access control model for cloud computing. *Journal of Information Security and Applications*, 19(1):45–60, 2014.