Cross-Domain Attribute-Based Access Control Encryption

Mahdi Sedaghat(✉) and Bart Preneel

imec-COSIC, KU Leuven, Leuven, Belgium
{ssedagha,bart.preneel}@esat.kuleuven.be

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

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 [23,25]. As introduced in the seminal work of Bell and LaPadula [5], restrictions have to be imposed on who can receive a message (enforce the No-Read rule) and who

M. Conti et al. (Eds.): CANS 2021, LNCS 13099, pp. 3–23, 2021.
https://doi.org/10.1007/978-3-030-92548-2_1

can send a message (enforce the No-Write rule). Although encryption guarantees users' privacy by limiting the set of recipients, we need more functionality to control who can write and transfer a ciphertext. Broadcasting of sensitive data by malicious senders is a serious threat for companies that handle highly sensitive data such as cryptocurrency wallets with access to signing keys [8].

Although some advanced cryptographical tools such as *Functional Encryption* provide fine-grained access to encrypted data, they do not allow to enforce the No-Write rule, hence additional functionalities beyond these cryptographic primitives are required to protect against data leakage.

To achieve this aim, Damgård et al. [10] 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 $\text{Pf} : \mathcal{S} \times \mathcal{R} \rightarrow \{0, 1\}$ which group of senders (like $i \in \mathcal{S}$) are allowed to communicate with a certain group of receivers (like $j \in \mathcal{R}$): communication is allowed iff $\text{Pf}(i, j) = 1$, else the request will be rejected.

Damgård et al. proposed two ACE constructions that support arbitrary policies. Their first construction takes a brute-force approach that is based on standard number-theoretic assumptions, while the size of the ciphertext grows exponentially in the number of receivers. The second scheme is more efficient: ciphertext length is poly-logarithmic in the number of the receivers, but it relies on the strong assumption of *indistinguishability obfuscation (iO)* [13]. In a subsequent work, Fuchsbauer et al. [12] 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. [31] proposed an ACE scheme based on the *Learning With Error* (LWE) assumption [24]. Since their construction follows the Damgård et al. approach, the ciphertexts in their construction also grow exponentially with the number of receivers. Recently, Wang et al. [34], proposed an efficient LWE-based ACE construction from group encryptions. Kim and Wu [20] 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 [33] 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 [33], the length of the ciphertext is constant, but it fails to preserve the identity of the receivers and also the decryption key size grows linearly.

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

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

ABE schemes provide a powerful tool to enforce fine-grained access control over encrypted data; they have been used in several applications [26]. Goyal et al. in [16], proposed two complementary types of ABE schemes: *Key-Policy Attribute-Based Encryption* (KP-ABE) and *Ciphertext-Policy Attribute-Based Encryption* (CP-ABE) schemes. In CP-ABE, the sender embeds a (policy) function $f(\cdot)$ into ciphertext to describe which group of receivers can learn the encrypted message. In this approach, the ciphertext is labeled by an arbitrary function $f(\cdot)$, and secret keys are associated with attributes in the domain of $f(\cdot)$. The decryption algorithm yields the plaintext iff the receivers' attribute set $\mathbb{A}$ satisfies $f(\cdot)$, i.e., $f(\mathbb{A}) = 1$. Conversely, in KP-ABE the secret keys are labeled by the function $f(\cdot)$; this label is set in the setup phase and a ciphertext can only be decrypted with a key whose access structure is satisfied by the set of attributes. In KP-ABE, the access policy cannot be altered after setup phase, while in CP-ABE data owners can control the data access.

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

Based on realistic application scenarios for ACE constructions, the proposed scheme follows the Cross-Domain key generation method, proposed by Wang and Chow in [33]. In an ACE scheme, the users might belong to two distinct companies with different security levels, so one of them may not be able to grant access rights to the other. In this context, two entities referred to as Sender Authority and Receiver Authority locally generate secret keys for senders and receivers, respectively. Moreover, since users, including senders and receivers, may need to be added to the system later on, the setup phase will be carried out independently of the predicate function. Hence, our approach follows this setup method and we provide a generic construction of a *Cross-Domain Access Control Encryption* scheme based on *Attribute-Based Encryption* constructions.

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

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

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

Road-map: The rest of the paper is organized as follows: In Sect. 2, we review the preliminaries and definitions and describe the system architecture. The formal definition of the CD-ABACE scheme and its security definitions are described in Sect. 3. In Sect. 4, we propose the generic construction of CD-ABACE schemes and discuss their security features. In Sect. 5 we present an efficient CD-ABACE construction based on a novel CP-ABE scheme. The performance of the proposed construction is compared in Sect. 6.

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

Scheme	Ciph. size	Enc. key size	Dec. key size	San. key size	Enc. cost	Dec. cost	CD	PF	Assump.
[10, ‡ 3]	$O(2^n)$	$O(r)$	$O(1)$	$O(1)$	$O(n)$	$O(n)$	✓	IB	DDH/DCR
[10, ‡ 4]	$poly(n)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$	✗	IB	iO
[12]	$O(n)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$	✗	IB	SXDH
[20]	$poly(n)$	$O(1)$	$O(1)$	$O(1)$	$O(n)$	$O(n)$	✗	PE	DDH/LWE
[33] (SS)	$O(1)$	$O(1)$	$O(s)$	0	$O(1)$	$O(s)$	✓	IB	GBDP
Ours (SS)	$O(1)$	$O(1)$	$O(1)$	0	$O(1)$	$O(w)$	✓	AB	MSE-DDH

2 Preliminaries and Definitions

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

Definition 1 (Access Structure *[4]***).** *For a given set of parties $\mathcal{P} = \{p_1, \ldots, p_n\}$, we say a collection $\mathbb{U} \subseteq 2^{\mathcal{P}}$ is monotone if, for all A, B, if $A \in \mathbb{U}$ and $A \subseteq B$ then $B \in \mathbb{U}$. Also, a(n) (monotonic) access structure is a (monotone) collection $\mathbb{U} \subseteq 2^{\mathcal{P}} \setminus \{\emptyset\}$. We call the sets in $\mathbb{U}$ authorized sets and the sets that do not belong to $\mathbb{U}$ are called unauthorized.*

Definition 2 (Binary Representation of a subset). *For a given universe set $\mathbb{U}$ of size n, we can represent each subset $\mathbb{A}$ as a binary string of length n. Particularly, the i^{th} the element of the binary string for the subset $\mathbb{A} \subseteq \mathbb{U}$ is equal to 1 (i.e., $a[i] = 1$) if $A_i = U_i$. We show a binary representation set as binary tuple $(a[1], \ldots, a[n]) \in \mathbb{Z}_2^n$.*

Definition 3 (Zero-polynomial). *For a finite set* $\mathbb{U} = \{k_1, \ldots, k_n\}$*, we define the zero-polynomial* $Z_{\mathbb{A}}(X)$ *for a nonempty subset of* $\mathbb{A} \subset \mathbb{U}$ *as* $Z_{\mathbb{A}}(X) := \prod_{i=1}^{n} (X - k_i)^{\overline{a[i]}}$*, where* $\overline{a[i]}$ *is the binary representation of the complement set* $\overline{\mathbb{A}}$*. In other words, this univariate polynomial vanishes on all the elements of the set* $\mathbb{U}$ *for which the binary representation of the subset* $\mathbb{A}$ *is zero.*

Definition 4 (Bilinear Groups *[7]***).** *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* $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ *such that* $\hat{e}(G, H) \neq 1$ *is an efficiently computable bilinear map with the following properties;*

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

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

System Architecture. The proposed scheme's architecture is based on the Cross-Domain ACE technique described in [33]. In a Cross-Domain ACE setting, two distinct entities generate the keys to determine which group of senders can send data to a certain group of receivers and control which group of receivers can read this data. There are five entities in this system as follows:

Receiver Authority (RA) as a trusted third party generates and distributes system parameters and the secret decryption keys for the Receivers. For this aim, based on a certified predicate function $\textsc{Pf}(.,.)$, it authorizes the claimed attributes by the receivers and returns the corresponding secret decryption keys.

Sender Authority (SA) as a semi-trusted entity generates the pair of SA's public parameters and master secret keys; it publishes the former, while it keeps the latter secret. Moreover, it generates the secret encryption keys for the Senders based on a predicate function $\textsc{Pf}(.,.)$ and SA's master secret keys.

Sanitizer is an honest-but-curious party in the network that checks the validity of the communication links and acts based on the predicate function $\textsc{Pf}(.,.)$. If the sender does not allow to transmit a message to the recipients, then the Sanitizer bans the request, else it broadcasts the received ciphertexts. The Sanitizer is semi-honest which means that it follows the protocol honestly but tries to infer some sensitive information including the identities of the users (Senders and Receivers) or compromise the secrecy of a message.

Senders: to share a secret message among a group of receivers, they encrypt data and send the resulting ciphertext to the Sanitizer along with a proof to ensure that they possess a valid encryption key generated by the SA.

Receivers: by having access to the ciphertexts, they can recover the plaintexts using their own attributes and the corresponding secret key for decryption. Conversely, if the receiver does not satisfy the access policy then the ciphertext never reveals any meaningful information about the encrypted message.

[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.

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

3 Cross-Domain Attribute-Based ACE Scheme

Next we introduce the notion of *Cross-Domain Attribute-Based Access Control Encryption* (CD-ABACE) schemes. The high-level idea behind the definition of a CD-ABACE is that we can generalize the concept of Boolean relations in the plain CP-ABE schemes (see full version [28]) to the predicate function in an ACE construction. In this scenario, the encryption key generator allows the sender to talk to a restricted group of receivers based on a given predicate function. By contrast with the original approach of specifying the ciphertext access rights during the encryption phase, in the present approach, the Sender Authority declares the access right during the encryption key generation phase. Moreover, the generated encryption keys are signed by the SA, and no one can convincingly assert ownership unless they have a correct signature.

Definition 5 (CD-ABACE schemes). *A* CD-ABACE *scheme* $\Psi_{\text{CD-ABACE}}$ *over the message space* $\mathcal{M}$*, the ciphertext space* $\mathcal{C}$ *and a predicate function* $\textsc{Pf} : \Sigma_k \times \Sigma_c \to \{0,1\}$ *has the following PPT algorithms:*

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

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

3.1 Security Definitions

Next we present the required security properties for a CD-ABACE scheme under only CPA-based definitions, where $\mathcal{A}$ has access to encryption, encryption-key generation, and decryption-key generation oracles. Noted that the following security games are motivated by the notion of co-selective CPA security in [3], 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 [33] to adapt them to the CD-ABACE system model.

Definition 6 (Correctness). *For a given attribute universe* $\mathbb{U}$ *and predicate function* $\mathrm{P\small F} : \Sigma_k \times \Sigma_c \rightarrow \{0, 1\}$*, we say that* $\Psi_{\text{CD-ABACE}}$ *over message space* $\mathcal{M}$ *and ciphertext space* $\mathcal{C}$ *is correct if we have,*

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

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

Definition 7 (No-Read Rule). *Consider* $\Psi_{\text{CD-ABACE}}$ *over the attribute universe* $\mathbb{U}$*, message space* $\mathcal{M}$*, a ciphertext space* $\mathcal{C}$ *and a predicate function* $\text{PF} : \Sigma_k \times \Sigma_c \rightarrow \{0,1\}$*. For a security parameter* λ*, we say that a PPT adversary* $\mathcal{A}$ *wins the defined* No-Read *rule security game described in Fig. 1 with access to the oracles in the same table, if she guesses the random bit* b *better than by chance. It is assumed that for a challenge access structure* $\mathbb{P}^*$*,* $\mathcal{A}$ *would not request the decryption key for attribute set* $\mathbb{B}_j$*, such that* $\text{PF}(\mathbb{B}_j, \mathbb{P}^*) = 1$*.* $\Psi_{\text{CD-ABACE}}$ *satisfies the* No-Read *rule if for all PPT adversaries* $\mathcal{A}$ *with advantage* $Adv^{\text{No-Read}}_{\Psi_{\text{CD-ABACE}},\mathcal{A}}(1^\lambda, b) = (\Pr[\mathcal{A} \text{ wins the No-Read game}] - 1/2)$ *we have,*

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

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 should an unauthorized receiver be unable to read the messages, combining the decryption secret keys of a group of unauthorized receivers should not reveal any information about the message. Also, this property has to hold even if the recipients collude with the Sanitizer.

Definition 8 (Parameterized No-Write Rule). *Consider* $\Psi_{\text{CD-ABACE}}$ *over* $\mathbb{U}$*, a message space* $\mathcal{M}$*, ciphertext space* $\mathcal{C}$ *and a predicate function* $\text{PF} : \Sigma_k \times \Sigma_c \rightarrow \{0,1\}$*. We say a* $\Psi_{\text{CD-ABACE}}$ *scheme satisfies the Parameterized* No-Write *rule, if no PPT adversary* $\mathcal{A}$ *with access to the oracles in Fig. 1 has a non-negligible advantage in winning the* No-Write *game, i.e., under the advantage* $Adv^{\text{No-Write}}_{\Psi_{\text{CD-ABACE}},\mathcal{A}}(1^\lambda, b) = (\Pr[\mathcal{A} \text{ wins No-Write}] - 1/2)$ *we have,*

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

We say $\mathcal{A}$ *wins the defined* No-Write *game iff* $b' == b$ *under the condition that for all queried secret encryption keys* $\mathbb{P}_i \in \mathcal{Q}_\mathcal{E} \cup \{\mathbb{P}^*\}$ *and all requested private decryption keys* $\mathbb{B}_j \in \mathcal{Q}_\mathcal{D}$*, along with the challenge access structure* $\mathbb{P}^*$*, we have* $\text{PF}(\mathbb{B}_j, \mathbb{P}_i) = 0$*. The function* $\mathsf{fix}(.)$ *accepts a ciphertext* $\mathtt{Ct}$ *as input and generates auxiliary information* $\mathtt{aux}$ *of* $\mathtt{Ct}$ *that is not sanitizable [33]. By seeding an encryption algorithm with this auxiliary information, the resulting ciphertext has also the same auxiliary information.*

Remark 1. With regard to the security definitions, the anonymity of the sender is guaranteed and the Sanitizer cannot deduce the identity of the sender while the receivers' anonymity relies on the CP-ABE construction. Note that the same type of property is known as weak attribute hiding in the context of ABE constructions [22]. Although an IND-CPA-secure CP-ABE satisfies the payload hiding property, a stronger security concept, called attribute-hiding CP-ABE, ensures that the set of attributes associated with each ciphertext is also obscured [19]. The latter increases the ciphertext size incrementally and the identity-based encryptions reveal the receivers' identity in plain.

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

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

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

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

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

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

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

1 : $(\mathsf{ek}_{\mathbb{P}_i}, \sigma_i, W_i) \leftarrow \mathsf{EncKGen}(\mathbb{P}_i, \text{PF})$
2 : $(\pi, \mathsf{x}) \leftarrow \mathsf{Enc}(\mathsf{ek}_{\mathbb{P}_i}, m)$
3 : **return** (π, x)

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

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

Fig. 1. No-Read and No-Write security games

4 Generic Construction

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

1. An EUF-CMA-secure SPS construction, $\mathcal{SPS}.(\mathsf{Pgen}, \mathsf{KG}, \mathsf{Sign}, \mathsf{Randz}, \mathsf{Vf})$ (see full version [28] for formal definition).
2. A computational Knowledge-Sound NIZK proof, $\mathcal{ZK}.(\mathsf{K_{crs}}, \mathsf{P}, \mathsf{V}, \mathsf{Sim})$ (see full version for formal definition [28]).
3. A publicly re-randomizable CP-ABE scheme, $r\mathcal{ABE}.(\mathsf{Pgen}, \mathsf{KGen}, \mathsf{Col}, \mathsf{Enc}, \mathsf{Randz}, \mathsf{Dec})$ (see full version for formal definition [28]).

For a given predicate function PF, message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$, the generic construction consists of the following PPT algorithms:

- **RA setup** ($\mathsf{RAgen}(\mathbb{U}, \lambda)$)**:** Takes the security parameter λ and an attribute universe $\mathbb{U}$, and runs the $r\mathcal{ABE}.\mathsf{Pgen}(\lambda, \mathbb{U})$ algorithm to generate the global and CP-ABE parameters. It outputs RA's master secret key set $\mathsf{msk}_{ra} = (\mathsf{msk}_{r\mathcal{ABE}})$ and RA's public parameters $\mathsf{pp}_{ra} = (\mathsf{pp}_{r\mathcal{ABE}})$.
- **SA setup** ($\mathsf{SAgen}(\mathsf{pp}_{ra}, \mathbf{R_L})$)**:** Takes RA's public parameters pp_{ra} and relation $\mathbf{R_L}$ as inputs and runs the $\mathcal{ZK}.\mathsf{K}_{\mathbf{crs}}(\mathbf{R_L})$, $\mathcal{SPS}.\mathsf{Pgen}(\lambda)$ and $\mathcal{SPS}.\mathsf{KG}(\mathsf{pp})$ algorithms and returns $\mathsf{pp}_{sa} = (\mathsf{pp}, \mathsf{vk}, \mathbf{crs})$ and $\mathsf{msk}_{sa} = (\mathbf{ts}, \mathsf{sk})$ as outputs. The underlying relation $\mathbf{R_L}$ is defined corresponding to the NP-language $\mathbf{L}$ for the statement $\mathsf{x} = (\sigma', \mathsf{vk}', \mathsf{ek}', \mathsf{Ct})$ and witness $\mathsf{w} = (\sigma, \mathsf{ek}, m, r, t)$.
- **Decryption KGen** ($\mathsf{DecKGen}(\mathsf{msk}_{ra}, \mathbb{B})$)**:** Takes as inputs RA's master secret key msk_{ra} and a key index $\mathbb{B} \in \Sigma_k$. It generates the private decryption key $\mathsf{dk}_{\mathbb{B}}$ by executing the algorithm $r\mathcal{ABE}.\mathsf{KGen}(\mathsf{msk}_{ra}, \mathbb{B})$.
- **Encryption KGen** ($\mathsf{EncKGen}(\mathsf{pp}_{ra}, \mathsf{msk}_{sa}, \mathbb{P}, \textsc{Pf})$)**:** Takes as inputs pp_{ra}, msk_{sa} and a ciphertext index $\mathbb{P} \in \Sigma_c$ that indicates to whom the sender is allowed to talk based on predicate function $\textsc{Pf}(.,.)$. It executes the collector algorithm $r\mathcal{ABE}.\mathsf{Col}(\mathsf{pp}_{ra}, \mathbb{P})$ to obtain the aggregated value $\mathsf{ek}_{\mathbb{P}}$ and then signs it by running the algorithm $\mathcal{SPS}.\mathsf{Sign}(\mathsf{sk}, \mathsf{ek}_{\mathbb{P}})$. It returns both the encryption key and the underlying signature to the sender.
- **Encryption** ($\mathsf{Enc}\,(\mathsf{pp}_{sa}, \mathsf{pp}_{ra}, m, \mathsf{ek}_{\mathbb{P}}, \sigma, W)$)**:** Takes as inouts the secret encryption key $\mathsf{ek}_{\mathbb{P}}$ and the underlying signature σ, the public parameters and a message $m \in \mathcal{M}$. It re-randomizes σ under an initial random string μ by running $\mathcal{SPS}.\mathsf{Randz}(\mathsf{pp}_{sa}, \mathsf{ek}_{\mathbb{P}}, \sigma, W; \mu)$. Next it runs the re-randomizable CP-ABE encryption algorithm $r\mathcal{ABE}.\mathsf{Enc}(\mathsf{pp}_{ra}, m, \mathsf{ek}_{\mathbb{P}})$ and proves knowledge of hidden values by executing the $\mathcal{ZK}.\mathsf{P}(\mathbf{R_L}, \mathbf{crs}, \mathsf{w}, \mathsf{x})$ algorithm. It returns the instance and underlying proof (π, x) as outputs.
- **Sanitization** ($\mathsf{San}(\mathsf{pp}_{sa}, \mathsf{pp}_{ra}, \pi, \mathsf{x})$)**:** Takes as inputs the proof π and the instance x: if $\mathcal{SPS}.\mathsf{Vf}(\mathsf{pp}, \mathsf{vk}', \sigma', \mathsf{ek}') = 1$ and $\mathcal{ZK}.\mathsf{V}(\mathbf{R_L}, \mathbf{crs}, \pi, \mathsf{x}) = 1$, it runs the algorithm $r\mathcal{ABE}.\mathsf{Randz}(\mathsf{pp}_{ra}, \mathsf{Ct})$ and returns the sanitized ciphertext $\tilde{\mathsf{Ct}}$ as output; otherwise it rejects the link and returns $\perp$.
- **Decryption** ($\mathsf{Dec}(\mathsf{pp}_{sa}, \mathsf{pp}_{ra}, \tilde{\mathsf{Ct}}, \mathsf{dk}_{\mathbb{B}})$)**:** Takes as inputs the public parameters, a sanitized ciphertext $\tilde{\mathsf{Ct}}$ and the decryption key $\mathsf{dk}_{\mathbb{B}}$. It returns the plaintext $m \in \mathcal{M}$ by executing $r\mathcal{ABE}.\mathsf{Dec}(\mathsf{pp}_{ra}, \tilde{\mathsf{Ct}}, \mathsf{dk}_{\mathbb{B}})$ algorithm if and only if $\textsc{Pf}(\mathbb{B}, \mathbb{P}) = 1$; otherwise this algorithm returns $\perp$.

Theorem 1. *The proposed generic* CD-ABACE *construction is correct.*

The proof can be found in the full version [28].

Theorem 2. *The proposed generic* CD-ABACE *scheme satisfies the* No-Read *rule of Definition 7.*

The proof can be found in the full version [28].

Theorem 3. *No PPT adversary $\mathcal{A}$ can win the* No-Write *security game of Definition 8 for the proposed* CD-ABACE *scheme under a fixed predicate function* $\textsc{Pf}(.,.)$.

The proof can be found in the full version [28].

5 An Efficient CD-ABACE Scheme

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

Structure-Preserving Signature (SPS): In this paper, we use a variant of the selectively re-randomizable SPS scheme of Abe et al. [1] (see full version [28]) as an efficient, unified and selectively re-randomizable SPS. Since in the proposed CD-ABACE construction the generator of the first cyclic group is hidden and the message is a second group element over the Type-III bilinear groups, we need to slightly modify this scheme with the following PPT algorithms:

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

The proof of correctness is identical to that of Abe et al.'s SPS construction, where a message is part of the second rather than the first group. As the first group generator is hidden in the proposed CD-ABACE scheme, we need to take $[\alpha^2]_1$ instead of $[1]_1$ to generate and verify signatures.

Non-Interactive Zero-Knowledge (NIZK) Proofs: As discussed in full version [28], Zero-Knowledge proofs [15] allow a prover to convince the verifier about the validity of a statement without revealing any other information. We use a standard Schnorr proof [27] to prove the knowledge of exponents in the random oracle model. To convert an interactive protocol to a non-interactive framework,

we utilize the Fiat-Shamir heuristic [11]. More precisely, the prover has access to a hash function, modeled as a random function ($\mathcal{O}$), to generate the challenges instead of receiving them from the verifier. For a given cyclic group $\mathbb{G}_i$ of order p with generator g_i, we denote by $\text{PoK}\{(\mathsf{w}) : \mathbf{R_L}(\mathsf{x}, \mathsf{w}) = 1\}$, the proof of knowledge of a hidden witness w that satisfies a given relation $\mathbf{R_L}$. Figure 2 formalizes a NIZK in ROM for proof of exponentiation.

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

Fig. 2. Proof of knowledge of exponents

An Efficient Re-randomizable CP-ABE: In what follows, we define a new IND-CPA-secure CP-ABE scheme with a constant key and ciphertext size. The Boolean function of this scheme is applied in AND-gate circuits. Although Guo et al. in [17] took a similar approach and presented a constant-key size CP-ABE scheme, the ciphertext size in their scheme increases linearly with the total number of attributes. The proposed re-randomizable CP-ABE scheme consists of the following algorithms:

- $(\mathsf{pp}, \mathsf{msk}) \leftarrow \mathcal{ABE}.\mathsf{Pgen}(\mathbb{U}, \lambda)$: Takes as inputs an attribute space $\mathbb{U}$ with size n along with the security parameter λ, and runs a Type-III bilinear group generator $\mathcal{BG}(\lambda) = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathsf{p}, \hat{e})$. It also selects a standard collision-resistant hash function $\mathsf{H} \leftarrow\$\, \mathcal{H}$ that is modeled as a random oracle in the security proofs. For a randomly selected integer $\alpha \leftarrow\$\, \mathbb{Z}_p^*$, it computes $h_i = \left[\alpha^i\right]_2$ as the set of monomials in $\mathbb{G}_2$ and $g_2 = \left[\alpha^2\right]_1$. It returns the master secret key $\mathsf{msk} = \left([1]_1, \alpha\right)$ and the system's public parameters $\mathsf{pp} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathsf{p}, \hat{e}, g_2, \{h_i\}_{i=0}^n, [\alpha]_T, \mathsf{H})$.
- $(\mathsf{dk}_\mathbb{B}) \leftarrow \mathcal{ABE}.\mathsf{KGen}(\mathsf{msk}, \mathbb{B})$: Takes as inputs msk and generates a secret decryption key corresponding to attribute set $\mathbb{B} \in \Sigma_k$, such that $|\mathbb{B}| < n - 1$. It first computes the Zero-Polynomial $Z_\mathbb{B}(x) = \prod_{i=1}^n (x - k_i)^{\overline{b[i]}}$ such that $k_i = \{\mathsf{H}(U_i)\}_{U_i \in \mathbb{U}}$. It returns the secret decryption key $\mathsf{dk}_\mathbb{B} = [1/Z_\mathbb{B}(\alpha)]_1$.
- $(\mathtt{Ct}) \leftarrow \mathcal{ABE}.\mathsf{Enc}\,(\mathsf{pp}, m, \mathbb{P})$: Takes as inputs the message $m \in \mathcal{M}$, the public parameters pp and an access structure $\mathbb{P} \in \Sigma_c$. It first samples $r \leftarrow\$\, \mathbb{Z}_p^*$, calculates $Z_\mathbb{P}(x) = \sum_{j=0}^n z_j x^j$ and returns the ciphertext as a tuple $\mathtt{Ct} = (\mathbb{P}, C, C_1, C_2) = (\mathbb{P}, m\,[r\alpha]_T\,, (\prod_{j=0}^n h_{j+1}^{z_j})^r = [r\alpha Z_\mathbb{P}(\alpha)]_2\,, g_2^{-r} = \left[-r\alpha^2\right]_1)$. We define the collector algorithm as $\mathsf{Col}(\mathsf{pp}, \mathbb{P}) = [\alpha Z_\mathbb{P}(\alpha)]_2$.

– $(m', \perp) \leftarrow \mathcal{ABE}.\mathsf{Dec}\,(\mathsf{pp}, \mathtt{Ct}, \mathsf{dk}_{\mathbb{B}})$: This algorithm takes as input the public parameters pp, a ciphertext $\mathtt{Ct}$ and a secret decryption key $\mathsf{dk}_{\mathbb{B}}$. If $\mathbb{P} \subseteq \mathbb{B}$, it computes, $F_{\mathbb{B},\mathbb{P}}(x) = \prod_{i=1}^{n}(x - k_i)^{c[i]} = \sum_{j=0}^{n} f_j x^j$ for $c[i] = b[i] - p[i]$ and returns $m' = C \cdot \left(\left(C_2 \bullet \prod_{i=1}^{n}(h_{i-1})^{f_i}\right) \cdot (\mathsf{dk}_{\mathbb{B}} \bullet C_1)\right)^{\frac{-1}{f_0}}$; otherwise it responds with $\perp$.

In the full version [28], we evaluated the proposed CP-ABE scheme regarding its security properties including the correctness and IND-CPA.

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

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

Remark 2. 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 [30]. Indeed, the access policy corresponding to a ciphertext only reveals the list of receivers who satisfy a specific set of attributes, even though it never leaks any information about the identity of the receivers. Under the assumption that there is more than one user who satisfies a set of certain attributes, the adversary is unable to deduce for which specific receiver the challenge ciphertext is intended.

Related Works: The first CP-ABE scheme, which allows the data owners to implement an arbitrary and fine-grained access policy in terms of any monotonic formula for each message was proposed by Bethencourt et al. at IEEE S&P 2007 in [6]; its security was proven in the *Generic Group Model* (GGM). In a subsequent work, Cheung et al. [9] constructed a CP-ABE scheme in the standard model, which is however restricted to a single AND-gate. Waters [35] introduced an asymptotically efficient CP-ABE scheme in the standard model, which is based on a *Linear Secret Sharing Scheme* (LSSS) to establish an arbitrary access policy. Lewko and Waters [21] 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 [18,29]. While these CP-ABE schemes allow to define in an effective way the right to access data, either the key or the ciphertext size grows linearly in the number of attributes. Therefore, CP-ABE schemes based on AND-gate circuits are considered promising candidates to address this downside. In this approach the sender defines a specific Boolean AND-gate circuit such that a recipient can learn the encrypted data iff they satisfy all the attributes, otherwise the decryption algorithm returns nothing. Considering AND-gate circuits

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Fig. 3. The proposed CD-ABACE scheme

provides a constant ciphertext length; several CP-ABE schemes are proposed based on this approach [17,32].

The Proposed CD-ABACE Scheme: At this point, we can wrap up the construction described in Fig. 3 by taking a family of collision-resistant hash functions $\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$. Our CD-ABACE scheme is built under a CP-ABE scheme based on AND-gate circuits with constant key and ciphertext sizes. The primary motivation behind this circuit choice is to construct a fully constant ACE within the context of CD-ABACE schemes. Note that we can build more universal circuit levels using the generic model discussed in Sect. 4.

Remark 3. While the proposed CD-ABACE scheme achieves a weak notion of receiver anonymity, it improves Wang and Chow's weak point where recipients' identities are public. In order to resolve this issue we can use the existing CP-ABE schemes with a more universal circuit level, but this compromises the efficiency. For instance, according to Garg et al. [14], we can fully anonymize the receiver using our generic construction based on multilinear maps and *iO* assumptions. We specify in the full version [28] a CD-ABACE scheme, using Waters's CP-ABE [35], which is defined under Linear Secret Sharing Schemes; we compare it with our proposed CD-ABACE scheme in Sect. 6.

6 Performance Analysis

In this section, we examine how the performance of our proposed fully-constant CD-ABACE scheme compares to the selectively-secure ACE scheme of Wang and Chow [33], which is the only implemented ACE construction to date and a CD-ABACE variant of Waters's CP-ABE [35] that is described in detail in the full version [28].

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

– *Total number of Attributes/Users versus RA Setup time*: The top left graph displays the relationship between the total number of attributes/users and time required to generate the parameter of the Receiver Authority. As can be seen, in our scheme and [33] scheme the time required to run this algorithm grows linearly with the total number of attributes/users, and for a generous consideration of 1 000 attributes, it only requires ∼200 milliseconds (ms) and ∼300 ms, respectively. However, for an ACE variation of Waters' CP-ABE [35] construction (see full version [28]) this time is constant and less than 30 ms.

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

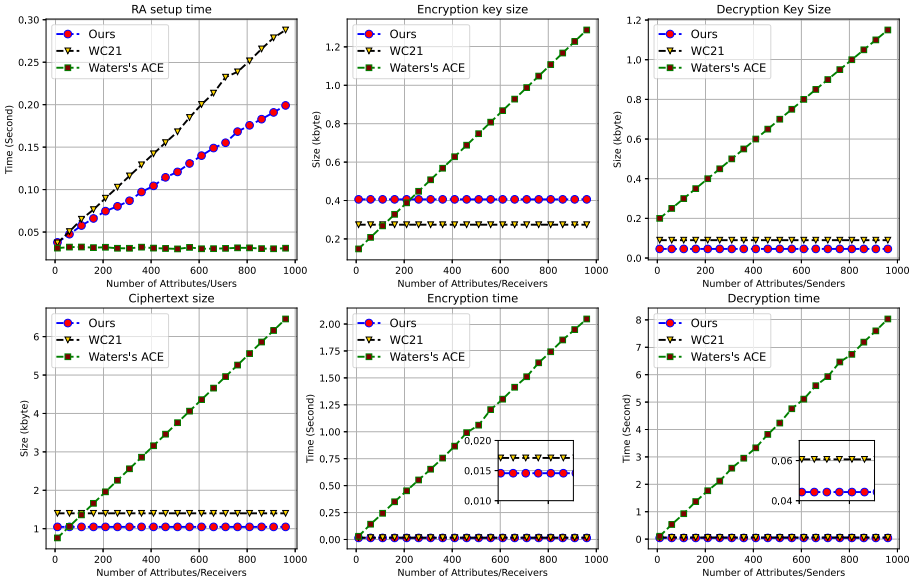

Fig. 4. Running time of attribute size dependence algorithms

- *Maximum number of Attributes/Receivers versus Encryption key size*: The top centre graph of Fig. 4 shows the relationship between the total number of attributes/receivers that a sender can send to them and the size of the stored encryption key. As can be seen, this relationship in Waters' ACE variant is linear, however the our proposed construction and [33] require a constant storage. Assuming 1 000 attributes/receivers to be the highest number used by a sender, the required memory for storing this key for [33], Waters' ACE variant and our scheme is ∼300, ∼1 200 and ∼400 bytes, respectively.
- *Maximum number of Attributes/Senders versus Decryption key size*: The top right graph of Fig. 4 shows the relationship between maximum the number of attributes/senders for each receiver and the size of the decryption key. As can be seen, in Waters' ACE variant this relationship grows linearly with number of attributes while in both our scheme and [33] the requires storage is constant independent of the number of attributes/senders; for instance, this size for a user having 1 000 attributes/senders is equal to ∼50, ∼100 bytes, while Waters' ACE variant is equal to ∼1.2 KB.
- *Number of Attributes/Receivers versus ciphertext size*: The bottom left graph of Fig. 4 depicts the relationship between the total number of attributes/receivers in the policy and the length of ciphertext. As can be seen, in Waters' ACE scheme this relationship is linear while our scheme and [33] achieve a constant ciphertext size. For instance, a ciphertext with 100 embedded attributes/receivers in the policy has a ciphertext of size ∼1, ∼1.4, ∼7 KB in our scheme, [33] and Waters' ACE scheme.

- *Number of Attributes/Receivers versus Encryption time*: The bottom centre graph of Fig. 4 shows the relationship between the total number of attributes/receivers of in the embedded policy and the encryption time. As can be seen, the time required to encrypt a ciphertext in our scheme and [33] is constant, while in Waters's ACE variation it grows linearly with the total number of attributes. For example, a sender in Waters' ACE, [33] and our scheme requires ∼2 000, ∼18, ∼15 ms to encrypt a message with 1 000 embedded attributes/receivers.
- *Number of Attributes/Senders versus Decryption time*: The bottom right graph of Fig. 4 shows the relationship between the maximum number of attributes/senders of each receivers and the decryption time. As can be seen, the time required to decrypt a ciphertext in Waters' ACE variant grows linearly with maximum number of attributes, while this overhead in our scheme and [33] is constant. For instance, a receiver in [33,35] and our proposed construction requires ∼8 000, ∼60, ∼45 ms to decrypt a ciphertext with 1 000 attributes in the policy.

Overall, our scheme has improved the receivers' key length and privacy level from identity-based to attribute-based. The ciphertext size has also been reduced, along with the number of public parameters. Since the second group generator is hidden in [33], the SA has to choose a new generator to create the SPS parameters. In contrast, the proposed variant of Abe et al.'s SPS [1] requires no new generator for the second cyclic group, and the intended NIZK proof cuts out the need for a target group proof of exponentiation.

7 Conclusion

In this work, we proposed a generic and an efficient CD-ABACE scheme based on attribute-based predicate functions. In comparison with earlier works, the length of the secret decryption keys and the ciphertext size has been substantially reduced to less than ∼50 and ∼1000 bytes as compared to Wang and Chow scheme where the size was ∼100 and ∼1400 bytes, respectively. Moreover, the computational overhead of encryption and decryption is linear in the number of the policy attributes and user attributes, respectively. Also, it is formally proved that the proposed scheme satisfies the No-Read and the No-Write rules based on standard assumptions. We leave the construction of a CD-ABACE scheme based on a Boolean circuit instead of AND-gate circuits with the same performance as an interesting open problem. As we discussed, the main downside for AND-gate circuits is that the attribute sets in plain may reveal some meaningful information about the intended constraints and consequently, applying a Boolean circuit can result in stronger anonymity guarantees for the receivers.

Acknowledgements. We would like to thank Sherman S. M. Chow, Georg Fuchsbauer, Karim Baghery, Ward Beullens, Pavel Hubáček and anonymous reviewers for their helpful discussions and valuable comments. This work was supported by Flanders Innovation & Entrepreneurship through the Spearhead Cluster Flux50 ICON project

PrivateFlex. In addition, this work was supported in part by the Research Council KU Leuven C1 on Security and Privacy for Cyber-Physical Systems and the Internet of Things with contract number C16/15/058 and by CyberSecurity Research Flanders with reference number VR20192203.

References

1. Abe, M., Groth, J., Ohkubo, M., Tibouchi, M.: Unified, minimal and selectively randomizable structure-preserving signatures. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 688–712. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54242-8_29
2. Akinyele, J.A., et al.: Charm: a framework for rapidly prototyping cryptosystems. J. Cryptogr. Eng. **3**(2), 111–128 (2013). https://doi.org/10.1007/s13389-013-0057-3
3. Attrapadung, N., Libert, B.: Functional encryption for inner product: achieving constant-size ciphertexts with adaptive security or support for negation. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 384–402. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13013-7_23
4. Beimel, A..: Secure schemes for secret sharing and key distribution. Faculty of Computer Science, Technion-Israel Institute of Technology (1996)
5. Bell, D.E., LaPadula, L.J.: Secure computer systems: mathematical foundations. Technical report, DTIC document (1973)
6. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: 2007 IEEE Symposium on Security and Privacy, SP 2007, pp. 321–334. IEEE (2007)
7. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_13
8. Brengel, M., Rossow, C.: Identifying key leakage of bitcoin users. In: Bailey, M., Holz, T., Stamatogiannakis, M., Ioannidis, S. (eds.) RAID 2018. LNCS, vol. 11050, pp. 623–643. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00470-5_29
9. Cheung, L., Newport, C.: Provably secure ciphertext policy ABE. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, pp. 456–465 (2007)
10. Damgård, I., Haagh, H., Orlandi, C.: Access control encryption: enforcing information flow with cryptography. In: Hirt, M., Smith, A. (eds.) TCC 2016. LNCS, vol. 9986, pp. 547–576. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53644-5_21
11. Fiat, A., Shamir, A.: How to prove yourself: practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987). https://doi.org/10.1007/3-540-47721-7_12
12. Fuchsbauer, G., Gay, R., Kowalczyk, L., Orlandi, C.: Access control encryption for equality, comparison, and more. In: Fehr, S. (ed.) PKC 2017. LNCS, vol. 10175, pp. 88–118. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54388-7_4
13. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. SIAM J. Comput. **45**(3), 882–929 (2016)

14. Garg, S., Gentry, C., Halevi, S., Sahai, A., Waters, B.: Attribute-based encryption for circuits from multilinear maps. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8043, pp. 479–499. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40084-1_27
15. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM J. Comput. **18**(1), 186–208 (1989)
16. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM Conference on Computer and Communications Security, pp. 89–98 (2006)
17. Guo, F., Mu, Y., Susilo, W., Wong, D.S., Varadharajan, V.: CP-ABE with constant-size keys for lightweight devices. IEEE Trans. Inf. Forensics Secur. **9**(5), 763–771 (2014)
18. Hong, H., Sun, Z.: An efficient and secure attribute based signcryption scheme with LSSS access structure. Springerplus **5**(1), 644 (2016)
19. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 146–162. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78967-3_9
20. Kim, S., Wu, D.J.: Access control encryption for general policies from standard assumptions. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017. LNCS, vol. 10624, pp. 471–501. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70694-8_17
21. Lewko, A., Waters, B.: Decentralizing attribute-based encryption. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 568–588. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20465-4_31
22. Okamoto, T., Takashima, K.: Adaptively attribute-hiding (hierarchical) inner product encryption. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 591–608. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_35
23. Osborn, S., Sandhu, R., Munawer, Q.: Configuring role-based access control to enforce mandatory and discretionary access control policies. ACM Trans. Inf. Syst. Secur. (TISSEC) **3**(2), 85–106 (2000)
24. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM **56**(6), 1–40 (2009)
25. Sabelfeld, A., Myers, A.C.: Language-based information-flow security. IEEE J. Sel. Areas Commun. **21**(1), 5–19 (2003)
26. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_27
27. Schnorr, C.P.: Efficient identification and signatures for smart cards. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 239–252. Springer, New York (1990). https://doi.org/10.1007/0-387-34805-0_22
28. Sedaghat, M., Preneel, B.: Cross-domain attribute-based access control encryption. Cryptology ePrint Archive, Report 2021/074 (2021). https://eprint.iacr.org/2021/074
29. Sedaghat, S.M., Ameri, M.H., Mohajeri, J., Aref, M.R.: An efficient and secure data sharing in Smart Grid: ciphertext-policy attribute-based signcryption. In: 2017 Iranian Conference on Electrical Engineering (ICEE), pp. 2003–2008. IEEE (2017)

30. Shahandashti, S.F., Safavi-Naini, R.: Threshold attribute-based signatures and their application to anonymous credential systems. In: Preneel, B. (ed.) AFRICACRYPT 2009. LNCS, vol. 5580, pp. 198–216. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02384-2_13
31. Tan, G., Zhang, R., Ma, H., Tao, Y.: Access control encryption based on LWE. In: Proceedings of the 4th ACM International Workshop on ASIA Public-Key Cryptography, pp. 43–50. ACM (2017)
32. Tran, P.V.X., Dinh, T.N., Miyaji, A.: Efficient ciphertext-policy ABE with constant ciphertext length. In: 2012 7th International Conference on Computing and Convergence Technology (ICCCT), pp. 543–549. IEEE (2012)
33. Wang, X., Chow, S.M.: Cross-domain access control encryption: arbitrary-policy, constant-size, efficient. In: IEEE Symposium on Security and Privacy (SP), Los Alamitos, CA, USA, pp. 388–401. IEEE Computer Society (May 2021)
34. Wang, X., Wong, H.W.H., Chow, S.S.M.: Access control encryption from group encryption. In: Sako, K., Tippenhauer, N.O. (eds.) ACNS 2021. LNCS, vol. 12726, pp. 417–441. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-78372-3_16
35. Waters, B.: Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19379-8_4