Batch-Puncturing Circuit CP-ABE (and More) from Lattices

Yongkang Lang[1], Fangguo Zhang[2], Jianghong Wei[3], Xinyi Huang[4], and Xiaofeng Chen[1(✉)]

[1] State Key Laboratory of Integrated Service Networks (ISN), Xidian University, Xi'an, 710071, China
lang_yongkang@163.com, xfchen@xidian.edu.cn

[2] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China
isszhfg@mail.sysu.edu.cn

[3] State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China
jianghong.wei.xxgc@gmail.com

[4] College of Cyber Security, Jinan University, Guangzhou 510006, China
xyhuang81@gmail.com

Abstract. Puncturable attribute-based encryption (PABE) not only supports fine-grained access control over encrypted data, but also enables users to revoke the decryption capability for specific messages by puncturing tags, thereby achieving fine-grained forward security. It finds wide applications in scenarios such as sharing government classified documents and personal health records. However, existing PABE schemes only support tag-by-tag puncturing, where each puncturing operation is done through key delegation, which causes the key size to grow with the number of punctured tags. This inefficiency makes PABE impractical in scenarios that require frequent puncturing or mass revocations. To address this limitation, it is crucial to support batch puncturing of tags, i.e., the decryption capability for messages associated with multiple tags can be revoked simultaneously via a single puncture.

In this work, we construct a ciphertext-policy attribute-based encryption (CP-ABE) scheme for circuits with batch-puncturing. Notably, the size of the punctured key in our scheme is independent of the number of punctured tags, as well as the size and depth of the circuits. This is achieved by leveraging the evasive learning with errors (LWE) and tensor LWE assumptions. In addition, we observe that puncturable CP-ABE can be re-stated by dual-policy ABE (DP-ABE) with key delegation, and generalize batch-puncturing CP-ABE to provide the first lattice-based construction of DP-ABE for circuits. Moreover, inspired by the observation of Agrawal and Yamada (Eurocrypt '20), we introduce the puncturing property into optimal broadcast encryption (BE), capturing a new primitive called puncturable BE, which allows the receiver to securely erase sensitive messages without communicating with the authority.

Keywords: Attribute-Based Encryption · Evasive and Tensor LWE · Puncturable Encryption · Circuits

1 Introduction

Attribute-based encryption (ABE) [11, 24, 34] is a powerful cryptographic primitive that supports fine-grained access control over encrypted data. Over the past two decades, the research on ABE has focused on the two primary types: ciphertext-policy ABE (CP-ABE) and key-policy ABE (KP-ABE). In a CP-ABE scheme, a message μ is encrypted as a ciphertext ct corresponding to a function (policy) $f : \mathcal{X} \to \mathcal{Y}$ that specifies which users have the ability to decrypt this data. Each secret key $\mathsf{sk}_{\mathbf{x}}$ contains an attribute vector $\mathbf{x} \in \mathcal{X}$ that describes the user's credentials. If $f(\mathbf{x}) = 0 \in \mathcal{Y}$,[1] then decryption using $\mathsf{sk}_{\mathbf{x}}$ returns μ. Conversely, KP-ABE has an inverted structure, where the secret key sk_f corresponds to a function f, and a message μ is encrypted as a ciphertext ct associated with the attributes.

Since the primitive of ABE was introduced in the seminal work of [34], it has attracted substantial attention within the research community, with different constructions achieving different trade-offs between expressiveness, efficiency, and security. In these studies, many constructions are based on bilinear groups, which are vulnerable to attacks by quantum computers. To this end, the work of Gorbunov, Vaikuntanathan, and Wee [23], along with that of Boneh et al. [12] (henceforth BGG+) revealed new advances in ABE for circuits from learning with errors (LWE). Building on this foundation, lattice-based ABE for circuits have garnered considerable attention over the past decade, especially KP-ABE for the richest function class [17, 18, 26, 42]. However, CP-ABE schemes have traditionally relied on pairings to only support circuits in NC^1 [11, 16, 28]. Until recently, Wee [41] proposed the first lattice-based CP-ABE for circuits of a-priori bounded polynomial depth based on new assumptions, namely evasive and tensor LWE. Although these assumptions are strong, they have enabled some progress in the long-stalled problem of constructing CP-ABE for the richest function class from lattices. Additionally, Wee [43] presents almost-optimal CP-ABE for circuits from succinct LWE, a falsifiable assumption following from evasive LWE.

ABE has found widespread applications in various domains such as cloud storage systems [35], electronic medical records [5] and online social networks [9]. However, directly deploying ABE in these real-world scenarios raises several practical challenges. More specifically, as the communication lifespan of a system increases, the long-term user keys pose an increasingly significant challenge. An adversary may persistently collect ciphertexts over time and wait for an opportunity to compromise a user's key (e.g., through malware extraction on mobile devices, coercion by authoritarian entities, phishing, or even accidental key leakage via cloud backups). Once the key is leaked (either forced or accidentally, which is considered inevitable in reality), all previously recorded ciphertexts encrypted under the corresponding policies (or attributes) may be subject to decryption, resulting in a complete loss of confidentiality. Unlike the situation in identity-based encryption, where the impact of key leakage is limited to a single

[1] Following the convention of lattice-based cryptography, $f(\mathbf{x}) = 0$ implies that the decryption condition is satisfied.

identity and can be mitigated by requesting a new key from the key generation center, ABE faces a particularly severe challenge: the exposure of a user's key may endanger the interests of many potentially innocent users, especially those with similar attributes or access structures, rendering the entire ABE-based group communication system unusable.

To mitigate such risks, periodic key rotation has been widely regarded as a good practice. Major cloud service providers such as Amazon[2], Google[3], and Microsoft[4] actively recommend it, and in some cases, it is even mandated by regulatory standards [10]. Ideally, it is desired to ensure that messages sent prior to the attack remain confidential even if an attacker steals a user's current key. To address these concerns, and inspired by Green and Miers [25], who utilized non-monotonic KP-ABE to construct puncturable encryption (PE), Phuong et al. [32] introduced puncturable ABE for any linear secret-sharing scheme (LSSS)-realizable access structure. Puncturable ABE is similar to ordinary ABE, except that each ciphertext is additionally associated with a set of tags. Moreover, it introduces a *puncture* algorithm that, given the current key sk and a tag t', outputs a new (punctured) key $\mathsf{sk}_{t'}$ that can decrypt all ciphertexts decryptable by sk except those encrypted under the tag t'. Unlike traditional forward-secure encryption [14], which revokes decryption capabilities for all ciphertexts within a single time period at once, puncturable ABE allows users to precisely and instantly revoke the decryption capability of a key for specific messages (or recipients or time periods) via the key puncturing process, achieving fine-grained forward security. In other words, users can selectively achieve "burn-after-reading" for certain important messages, rather than destroying all the messages.

However, following the initial PE structure [25], existing PE [36–38] and puncturable ABE [19,30,32] suffer from notable limitations: each puncture operation can only handle a single tag, and the punctured key has a size of $\Omega(\eta)$, where η is the number of punctured tags. This limitation becomes particularly problematic in scenarios that require frequent puncturing or mass revocations, such as secure handling of classified government information [44] and cloud-assisted personal health records [30]. In such settings, where sensitive data are securely erased via such tag-by-tag puncturing, the decryption cost and the key storage overhead quickly become impractical. The situation is even worse for lattice-based cryptosystems [19,38], where the punctured key size increases quadratically with the number of punctured tags. Another troublesome limitation of [19,38] is that their punctured key size is bounded by the pre-determined circuit depth, which is due to the fact that the puncturing property is achieved by appropriate circuit class setting. This means that if one trivially attempts to modify the circuit class setting to puncture multiple tags simultaneously, a larger circuit depth must be set at system initialization. As a result, even if only a single tag is punctured, an additional cost is added to the punctured key, which is equivalent to the storage overhead incurred when puncturing the maximum allowable number of tags in a

[2] https://docs.aws.amazon.com/kms/latest/developerguide/rotate-keys.html.
[3] https://cloud.google.com/kms/docs/key-rotation?hl=en.
[4] https://learn.microsoft.com/en-us/azure/storage/blobs/security-recommendations.

Table 1. Comparison with prior PE and puncturable ABE.

Reference	Type	UTP	Expression	Assumption	PQ	BP	|psk|
GM15 [25]	PE	KP-ABE	N/A	DBDH	✗	✗	$O_\lambda(\eta)^\dagger$
SSSLG20 [36]	PE	KH-IRKEM	N/A	DBDH	✗	✗	$O_\lambda(\eta)^\dagger$
SSS23 [37]	PE	EKHP-PRF	N/A	Non-uniform MLWE	✓	✗	$O_\lambda(\eta)$
SDLP20 [38]	PE	KP-ABE	N/A	LWE	✓	✗	$O_\lambda(\eta \cdot d^2)$
PNXW18 [32]	CP/KP-ABE	KP-ABE	LSSS, Monotone	DBDH	✗	✗	$O_\lambda(\eta)^\dagger$
DSDR22 [19]	KP-ABE	KP-ABE	Circuits, P	LWE	✓	✗	$O_\lambda(\eta \cdot d^2)$
This work	CP-ABE	KP-ABE	Circuits, P	Evasive LWE Tensor LWE	✓	✓	$O_\lambda(1)$

Note. The column UTP shows the underlying technology used for puncturing in the scheme. The column PQ shows whether the scheme is based on a post-quantum assumption. The column BP shows whether the scheme supports batch-puncturing. "N/A" means not applicable, as this is a PE scheme. DBDH refers to the decision bilinear Diffie-Hellman assumption. |psk| is the punctured secret key size. $O_\lambda(\cdot)^\dagger$ hides factors polynomial in λ and the group element $\mathbb{G}$ factors, whereas $O_\lambda(\cdot)$ hides factors polynomial in λ. η is the number of punctured tags. d is the circuit depth. Note that [36] and [37] are PE schemes, which are constructed from key-homomorphic identity-based revocable key encapsulation mechanism (KH-IRKEM) and extended key-homomorphic puncturable PRF (EKHP-PRF) respectively.

single operation. Moreover, to our knowledge, there is currently no lattice-based puncturable CP-ABE that supports the function class P. This appears to stem from the fact that PE is typically constructed from KP-ABE, and it is challenging to coherently combine two distinct types of circuit (KP and CP) ABE from lattices. Thus, based on the above discussion, a natural question arises:

For general circuits, can we achieve CP-ABE with batch-puncturing from lattices, ensuring that the size of punctured keys remains independent of both the circuit depth and the number of punctured tags?

1.1 Our Contributions

In this work, the above question is answered affirmatively. Our contributions are summarized as follows.

– We construct the first lattice-based batch-puncturing CP-ABE (BPCP-ABE) scheme for circuits. In particular, our scheme enables revoking access to ciphertexts under multiple tags for a certain key via a single puncture operation, where the number of tags punctured each time can be freely determined by users. Moreover, the size of the punctured key is independent of the number of punctured tags as well as the depth and size of circuits. To achieve this, we introduce novel lattice techniques referred to as *randomized dual-tier keys* and *approximate relationship* in homomorphic computation. We are optimistic that these techniques will find broader applications in the duality of lattice-based ABE and beyond. Table 1 provides a detailed comparison of our scheme with previous works on PE and puncturable ABE.

- We provide a security proof for our BPCP-ABE under the recently introduced evasive LWE [39,41] assumption and the variant of tensor LWE assumption. The latter is a consequence of our generalization of a Gaussian vector $\mathbf{r}$ to a structured Gaussian matrix $\mathbf{R}$ in the tensor LWE assumption [41].
- Two natural generalizations of BPCP-ABE are presented: puncturable broadcast encryption (PBE) and dual-policy ABE (DP-ABE) [7,8] for circuits. PBE enables secure erasure of sensitive messages without requiring communication with the authority. For circuit ABE, we show its duality from evasive lattice assumptions, thereby yielding the first lattice-based DP-ABE scheme for circuits. This supports fine-grained bilateral access control over encrypted data, ensuring the equality of permissions between data owners and recipients. We provide more details in Appendix F.

2 Preliminaries

Notations. For a positive integer $k \in \mathbb{N}$, we represent the set $\{1, \ldots, k\}$ with $[k]$. Row vectors and matrices are represented by bold lowercase letters (e.g. $\mathbf{v}$) and bold uppercase letters (e.g. $\mathbf{V}$). They are indexed using brackets, $\mathbf{V}[i, j]$ is the component in the i-th row and the j-th column of $\mathbf{V}$. For any $\mathbf{v}, \mathbf{V}$ over $\mathbb{Z}$, we use $\|\mathbf{v}\|, \|\mathbf{V}\|$ to denote the maximum absolute value over all the entries. For any two vectors $\mathbf{u}$ and $\mathbf{v}$ (resp. matrices $\mathbf{U}$, $\mathbf{V}$), we use $[\mathbf{u}|\mathbf{v}]$ (resp. $[\mathbf{U}|\mathbf{V}]$) to denote horizontal concatenation of $\mathbf{u}$ and $\mathbf{v}$ (resp. $\mathbf{U}$ and $\mathbf{V}$). For any $\mathbf{V}$, its Gram-Schmidt orthogonalization is denoted by $\widetilde{\mathbf{V}}$. If v is an entry of $\mathbf{v}$, then $v \leftarrow \mathcal{D}$ denotes a random sample from distribution $\mathcal{D}$, and $v \leftarrow S$ denotes a uniform random sample from set S. We use $\approx_s$ (resp. $\approx_c$, $\equiv$) as the abbreviation for statistically (resp. computationally, perfectly) indistinguishable. $\mathsf{Round}_\beta(v)$ denotes a function that outputs 0 if $\|v\| < \beta$ and 1 otherwise. We let $\mathsf{poly}(m)$ and $\mathsf{negl}(\cdot)$ denote a polynomial function of m and a negligible function, respectively.

Tensor Product. For matrices $\mathbf{U} \in \mathbb{Z}^{n\times m}, \mathbf{V} \in \mathbb{Z}^{k\times t}$, their tensor product (Kronecker product) is defined as

$$\mathbf{U} \otimes \mathbf{V} = \begin{pmatrix} \mathbf{U}[1,1]\mathbf{V} & \cdots & \mathbf{U}[1,m]\mathbf{V} \\ \vdots & \ddots & \vdots \\ \mathbf{U}[n,1]\mathbf{V} & \cdots & \mathbf{U}[n,m]\mathbf{V} \end{pmatrix} \in \mathbb{Z}^{nk\times mt}.$$

The mixed-product property for tensor product is

$$(\mathbf{A}\cdot\mathbf{C}) \otimes (\mathbf{B}\cdot\mathbf{D}) = (\mathbf{A}\otimes\mathbf{B})\cdot(\mathbf{C}\otimes\mathbf{D}) \in \mathbb{Z}^{nk\times wv}$$

where $\mathbf{A} \in \mathbb{Z}^{n\times m}, \mathbf{B} \in \mathbb{Z}^{k\times t}, \mathbf{C} \in \mathbb{Z}^{m\times w}, \mathbf{D} \in \mathbb{Z}^{t\times v}$. The mixed-product can be naturally generalized as follows. For $\mathbf{A} \in \mathbb{Z}^{n\times m}, \mathbf{B} \in \mathbb{Z}^{k\times t}$, $\mathbf{A}\otimes\mathbf{B} = (\mathbf{I}_n \otimes \mathbf{B})\cdot(\mathbf{A}\otimes\mathbf{I}_t)$. We follow the convention that matrix multiplication has higher precedence than the tensor product, allowing us to express $\mathbf{A}\otimes(\mathbf{BC})$ simply as $\mathbf{A}\otimes\mathbf{BC}$.

Due to space limitations, we defer the lattice background to Appendix A, including the LWE assumption and its relevant variants (e.g., small-secret LWE and non-uniform LWE), lattice trapdoors as well as lattice evaluation.

2.1 Rounding Function and Properties

Recall that the definition of the rounding function in [29].

Definition 1 (Rounding Function). *For an integer $q \geq 2$, a "rounding" function $\lfloor \cdot \rceil_p : \mathbb{Z}_q \to \mathbb{Z}_p$ is defined as*

$$\lfloor x \rceil_p = \left\lfloor \frac{p \cdot \hat{x}}{q} \right\rceil \mod p$$

where p is a small modulus and satisfies the condition $q \geq p \geq 2$, and $\hat{x} \in \mathbb{Z}$ is an arbitrary integer that is congruent to x modulo q.

Observe that the definition of $\lfloor \cdot \rceil_p$ is independent of the choice of $\hat{x}$. Specifically, for $x^* = \hat{x} + k \cdot q \in \mathbb{Z}$ where $k \in \mathbb{Z}$, namely $x^* = \hat{x} \mod q$, we have

$$\left\lfloor \frac{p \cdot x^*}{q} \right\rceil = \left\lfloor \frac{p \cdot \hat{x}}{q} \right\rceil \mod p.$$

Naturally, the function $\lfloor \cdot \rceil_p$ can be extended to the variants of matrices and vectors over $\mathbb{Z}_q$. We will use the following properties that had been established in [17, 29].

Lemma 1 (Properties of Rounding Function).

- *For any $\mathbf{u}, \mathbf{v} \in \mathbb{Z}_q^n$, $\exists \mathbf{e} \in \mathbb{Z}^n$ with $\|\mathbf{e}\| \leq 2$ such that $\lfloor \mathbf{u} + \mathbf{v} \mod q \rceil_p = \lfloor \mathbf{u} \rceil_p + \lfloor \mathbf{v} \rceil_p + \mathbf{e} \mod p$.*
- *For any $\mathbf{v} \in \mathbb{Z}^n$, $\mathbf{P} \in \mathbb{Z}_q^{n \times k}$, $\exists \mathbf{e} \in \mathbb{Z}^k$ with $\|\mathbf{e}\| \leq n \cdot \|\mathbf{v}\|$ such that $\lfloor \mathbf{v} \cdot \mathbf{P} \mod q \rceil_p = \mathbf{v} \cdot \lfloor \mathbf{P} \rceil_p + \mathbf{e} \mod p$.*

We now introduce a new property for $\lfloor \cdot \rceil_p$, which is proved in Appendix B.

Lemma 2. *For any $\mathbf{v} \in \mathbb{Z}^n$, $\mathbf{P} \in \mathbb{Z}_q^{n \times k}$, $\mathbf{Q} \in \mathbb{Z}^{k \times t}$, $\exists \mathbf{e} \in \mathbb{Z}^t$ with $\|\mathbf{e}\| \leq (n+1)\, k \cdot \|\mathbf{v}\| \cdot \|\mathbf{Q}\|$ such that*

$$\lfloor \mathbf{v} \cdot \mathbf{P} \cdot \mathbf{Q} \mod q \rceil_p = \mathbf{v} \cdot \lfloor \mathbf{P} \rceil_p \cdot \mathbf{Q} + \mathbf{e} \mod p.$$

3 Batch-Puncturing Ciphertext-Policy Attribute-Based Encryption

3.1 Syntax and Security Definition of BPCP-ABE

Below, we provide the definition of the primary focus in this work, BPCP-ABE. The syntax generally conforms to the standard notation for puncturable ABE [32], except that we consider a BatchPun algorithm that allows inputs with multiple tags.

Definition 2 (Batch-Puncturing Attribute-Based Encryption). *Let λ be a security parameter, $\ell = \ell(\lambda)$ be the number of attributes, $\mathcal{K} = \mathcal{K}(\lambda)$ be the number of tags, $\mathcal{T} = \mathcal{T}(\lambda)$ be a tags space, $\eta_{max} = \eta_{max}(\lambda)$ is the maximum number of puncture, and $\mathcal{M} = \mathcal{M}(\lambda)$ be a plaintext space. Let $\mathcal{X}_\lambda^\ell$ and $\mathcal{Y}_\lambda$ be two finite sets. Define $\{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ as a class of functions with $\mathcal{F}_\lambda = \{f : \mathcal{X}_\lambda^\ell \to \mathcal{Y}_\lambda\}$. An attribute-based encryption scheme with batch-puncturing consists of a tuple of efficient algorithms $\Pi_{\mathsf{BPCP\text{-}ABE}}$ with the following properties:*

- $\mathsf{Setup}(1^\lambda, \ell, \mathcal{K}, \eta_{max}) \to (\mathsf{mpk}, \mathsf{msk})$. *On input the security parameter λ, the number of attributes ℓ, the number of tags $\mathcal{K}$ and the maximum number of puncture η_{max}, the* Setup *algorithm outputs the master public key* mpk *and the master secret key* msk.
- $\mathsf{Enc}(\mathsf{mpk}, f, \mu, \{t_1, \ldots, t_\mathcal{K}\}) \to \mathsf{ct}$. *On input the master public key* mpk, *a function $f \in \mathcal{F}_\lambda$, a message $\mu \in \mathcal{M}$, and a list of tags $t_1, \ldots, t_\mathcal{K} \in \mathcal{T}$, the* Enc *algorithm outputs a ciphertext* ct.
- $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathbf{x}) \to \mathsf{sk}_{\mathbf{x},\emptyset}$. *On input the master public key* mpk, *the master secret key* msk, *and an attribute vector $\mathbf{x} \in \mathcal{X}_\lambda^\ell$, the* KeyGen *algorithm outputs a secret key $\mathsf{sk}_{\mathbf{x},\emptyset} := (\mathsf{ask}_\mathbf{x}, \mathsf{psk}_\emptyset)$, which consists of two parts: one related to $\mathbf{x}$ is called the attribute secret key $\mathsf{ask}_\mathbf{x}$, and the other related to the punctured set $P_{\mathbf{x},0}$ is called the punctured secret key $\mathsf{psk}_\emptyset$. Here, the punctured set $P_{\mathbf{x},0} := \emptyset$, since no puncturing has occurred.*
- $\mathsf{BatchPun}(\mathsf{mpk}, \mathsf{sk}_{\mathbf{x},\eta-1}, \{t'_{\eta,1}, \ldots, t'_{\eta,k_\eta}\}) \to \mathsf{sk}_{\mathbf{x},\eta}$. *On input the master public key* mpk, *the secret key $\mathsf{sk}_{\mathbf{x},\eta-1}$, and a list of punctured tags $t'_{\eta,1}, \ldots, t'_{\eta,k_\eta} \in \mathcal{T}$ where $\eta, k_\eta \in \mathbb{N}$, the* BatchPun *algorithm outputs a new secret key $\mathsf{sk}_{\mathbf{x},\eta} = (\mathsf{ask}_\mathbf{x}, \mathsf{psk}_\eta)$. Ciphertexts that can be decrypted by $\mathsf{sk}_{\mathbf{x},\eta-1}$ can also be decrypted by $\mathsf{sk}_{\mathbf{x},\eta}$, except for those encrypted under a list of tags whose intersection with $\{t'_{\eta,1}, \ldots, t'_{\eta,k_\eta}\}$ is non-empty. Here, update the punctured set as $P_{\mathbf{x},\eta} = P_{\mathbf{x},\eta-1} \cup \{t'_{\eta,1}, \ldots, t'_{\eta,k_\eta}\}$.*
- $\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_{\mathbf{x},\eta}, \mathsf{ct}) \to \mu$ *or* $\perp$. *On input the master public key* mpk, *the secret key $\mathsf{sk}_{\mathbf{x},\eta}$, a ciphertext* ct, *the* Dec *algorithm outputs a message μ or a failure symbol $\perp$.*

Correctness. *For all inputs $\lambda \in \mathbb{N}$, $\mathbf{x} \in \mathcal{X}_\lambda^\ell$, $f \in \mathcal{F}_\lambda$ and $\mu \in \mathcal{M}$, we require the following:*

- *If $f(\mathbf{x}) = 0$,*

$$\Pr[\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_{\mathbf{x},\emptyset}, \mathsf{Enc}(\mathsf{mpk}, f, \mu, \{t_1, \ldots, t_\mathcal{K}\})) = \mu] = 1 - \mathsf{negl}(\lambda).$$

- *If $f(\mathbf{x}) = 0$ and $P_{\mathbf{x},\eta} \cap \{t_1, \ldots, t_\mathcal{K}\} = \emptyset$,*

$$\Pr[\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_{\mathbf{x},\eta}, \mathsf{Enc}(\mathsf{mpk}, f, \mu, \{t_1, \ldots, t_\mathcal{K}\})) = \mu] = 1 - \mathsf{negl}(\lambda).$$

where $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \ell, \mathcal{K}, \eta_{max})$, $\mathsf{sk}_{\mathbf{x},\emptyset} \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathbf{x})$, $\mathsf{sk}_{\mathbf{x},\eta} \leftarrow \mathsf{BatchPun}(\mathsf{mpk}, \mathsf{sk}_{\mathbf{x},\eta-1}, \{t'_{\eta,1}, \ldots, t'_{\eta,k_\eta}\})$, and $P_{\mathbf{x},\eta} = P_{\mathbf{x},\eta-1} \cup \{t'_{\eta,1}, \ldots, t'_{\eta,k_\eta}\}$ with $P_{\mathbf{x},0} = \emptyset$.

Security Definition. For a BPCP-ABE scheme $\Pi_{\mathsf{BPCP\text{-}ABE}}$, we focus on selective security. Specifically, the adversary $\mathcal{A}$ needs to specify the challenge (policy) function and tags before receiving mpk. The following describes the experiment $\mathsf{EXP}^{\mathsf{BP\text{-}ABE}}_{\mathcal{A},\Pi}$ in detail.

1. **Initial:** The adversary $\mathcal{A}$ declares a challenge function $f^* \in \mathcal{F}_\lambda$ and a list of tags $t_1^*, \ldots, t_{\mathcal{K}}^* \in \mathcal{T}$.
2. **Setup:** The challenger $\mathcal{C}$ runs $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \ell, \mathcal{K}, \eta_{max})$, and gives mpk to $\mathcal{A}$. In addition, $\mathcal{C}$ maintains a tuple $(\mathbf{x}, \mathsf{sk}_{\mathbf{x},\eta}, P_{\mathbf{x}}, C_{\mathbf{x}})$ to record the state of the secret key for the attribute vector $\mathbf{x} \in \mathcal{X}_\lambda^\ell$, where $\mathsf{sk}_{\mathbf{x},\eta}$ represents the secret key that has been punctured by the tags in the set $P_{\mathbf{x}}$ and $C_{\mathbf{x}}$ is a set of corrupted tags. $P_{\mathbf{x}}$ and $C_{\mathbf{x}}$ are initially empty for $\mathbf{x}$.
3. **Query Phase:** Throughout the experiment, $\mathcal{A}$ adaptively performs the following queries in any order, and $\mathcal{C}$ responds to these queries as follows.
   a) **Puncture Queries:** $\mathcal{A}$ makes a punctured query for $\mathbf{x}$ and $\{t'_1, \ldots, t'_k\}$, $\mathcal{C}$ searches for the tuple $(\mathbf{x}, \mathsf{sk}_{\mathbf{x},\eta}, P_{\mathbf{x}}, C_{\mathbf{x}})$. If the tuple does not exist, $\mathcal{C}$ computes
   $$\mathsf{sk}_{\mathbf{x},\emptyset} \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathbf{x}),$$
   $$\mathsf{sk}_{\mathbf{x},1} \leftarrow \mathsf{BatchPun}(\mathsf{mpk}, \mathsf{sk}_{\mathbf{x},\emptyset}, \{t'_1, \ldots, t'_k\}),$$
   adds $\{t'_1, \ldots, t'_k\}$ to $P_{\mathbf{x}}$, and stores the tuple $(\mathbf{x}, \mathsf{sk}_{\mathbf{x},1}, P_{\mathbf{x}}, C_{\mathbf{x}})$. Otherwise, $\mathcal{C}$ directly computes $\mathsf{sk}_{\mathbf{x},\eta+1} \leftarrow \mathsf{BatchPun}(\mathsf{mpk}, \mathsf{sk}_{\mathbf{x},\eta}, \{t'_1, \ldots, t'_k\})$. Let $P_{\mathbf{x}} = P_{\mathbf{x}} \cup \{t'_1 \ldots, t'_k\}$, then update to the new tuple $(\mathbf{x}, \mathsf{sk}_{\mathbf{x},\eta+1}, P_{\mathbf{x}}, C_{\mathbf{x}})$.
   b) **Corrupt Queries:** When $\mathcal{A}$ first makes a corruption query for $\mathbf{x}$, $\mathcal{C}$ considers the following four cases:
      Case 1: If $f^*(\mathbf{x}) \neq 0$ and the tuple $(\mathbf{x}, \mathsf{sk}_{\mathbf{x},\eta}, P_{\mathbf{x}}, C_{\mathbf{x}})$ does not exist, $\mathcal{C}$ computes $\mathsf{sk}_{\mathbf{x},\emptyset} \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathbf{x})$ and returns it to $\mathcal{A}$, sets $C_{\mathbf{x}} = P_{\mathbf{x}} = \emptyset$, and creates a new tuple $(\mathbf{x}, \mathsf{sk}_{\mathbf{x},\emptyset}, P_{\mathbf{x}}, C_{\mathbf{x}})$.
      Case 2: If $f^*(\mathbf{x}) \neq 0$ and the tuple $(\mathbf{x}, \mathsf{sk}_{\mathbf{x},\eta}, P_{\mathbf{x}}, C_{\mathbf{x}})$ exists, $\mathcal{C}$ returns $\mathsf{sk}_{\mathbf{x},\eta}$ to $\mathcal{A}$ and lets $C_{\mathbf{x}} = P_{\mathbf{x}}$.
      Case 3: If $f^*(\mathbf{x}) = 0$ and the tuple $(\mathbf{x}, \mathsf{sk}_{\mathbf{x},\eta}, P_{\mathbf{x}}, C_{\mathbf{x}})$ exists, $\mathcal{C}$ returns $\perp$ when $P_{\mathbf{x}} \cap \{t_1^*, \ldots, t_{\mathcal{K}}^*\} = \emptyset$; otherwise, $\mathcal{C}$ returns the latest secret key $\mathsf{sk}_{\mathbf{x},\eta}$ to $\mathcal{A}$ and sets $C_{\mathbf{x}} = P_{\mathbf{x}}$.
      Case 4: If $f^*(\mathbf{x}) = 0$ and the tuple $(\mathbf{x}, \mathsf{sk}_{\mathbf{x},\eta}, P_{\mathbf{x}}, C_{\mathbf{x}})$ does not exist, $\mathcal{C}$ returns $\perp$.
   c) **Challenge Queries:** $\mathcal{A}$ issues a message pair $(\mu_0, \mu_1) \in \mathcal{M}^2$. $\mathcal{C}$ samples $\alpha \leftarrow \{0, 1\}$, computes $\mathsf{ct}_\alpha \leftarrow \mathsf{Enc}(\mathsf{mpk}, f^*, \mu_\alpha, \{t_1^*, \ldots, t_{\mathcal{K}}^*\})$ and returns it to $\mathcal{A}$.
4. **Guess:** The output of the experiment is a guessed bit α' generated by $\mathcal{A}$.

In the above experiment, the advantage $Adv^{\mathsf{Sel\text{-}IND}}_{\mathcal{A},\Pi}(1^\lambda)$ of $\mathcal{A}$ is defined as

$$Adv^{\mathsf{Sel\text{-}IND}}_{\mathcal{A},\Pi}(1^\lambda) := \big|\Pr[\mathsf{EXP}^{\mathsf{BP\text{-}ABE}}_{\mathcal{A},\Pi}(1^\lambda) = 1 | \alpha = 0] - \Pr[\mathsf{EXP}^{\mathsf{BP\text{-}ABE}}_{\mathcal{A},\Pi}(1^\lambda) = 1 | \alpha = 1]\big|.$$

Definition 3 (Selective Security for BPCP-ABE). *A BPCP-ABE scheme is said to satisfy selective security if for any PPT adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ for all sufficiently large $\lambda \in \mathbb{N}$ such that $Adv^{\mathsf{Sel\text{-}IND}}_{\mathcal{A},\Pi}(1^\lambda) = \mathsf{negl}(\lambda)$.*

In the following, we consider a weaker variant of the security definition, which is also of concern in the current lattice-based CP-ABE [2,4,41]. In this definition, $\mathcal{A}$ specifies the attribute set $\mathbf{x}$ and its associated set of punctured tags $P_{\mathbf{x}}$ for corrupt queries with f^* before the setup phase.

Definition 4 (Very Selective Security for BPCP-ABE). *We define very selective security for* BPCP-ABE *as the same as the above selective security, except that adversary $\mathcal{A}$ is required to submit corrupt queries $(\mathbf{x}_1, P_{\mathbf{x}_1}), ..., (\mathbf{x}_Q, P_{\mathbf{x}_Q})$ to the challenger before the setup phase, where Q is the number of corrupt queries. We define the corresponding advantage of $\mathcal{A}$ as $Adv^{\mathsf{VerSel\text{-}IND}}_{\mathcal{A},\Pi}(1^\lambda)$. If this advantage value is negligible, the* BPCP-ABE *scheme is said to satisfy very selective security.*

4 Our Assumptions

To establish the security of our construction, we present a variant of the evasive LWE assumption [26,41].

Assumption 1 (Evasive LWE). *Let* Samp *be a PPT algorithm that on input 1^λ, outputs*

$$\widehat{\mathbf{M}} \in \mathbb{Z}_p^{n\times 2m_0}, \mathbf{W} \in \mathbb{Z}_q^{(2nm_0+n)\kappa\times t}, \mathsf{aux} \in \{0,1\}^*.$$

We define the following advantage functions:

$$\begin{aligned}
\mathsf{Adv}^{\mathrm{PRE}}_{\mathcal{A}_0}(\lambda) :=& \Pr\left[\mathcal{A}_0\left(\begin{array}{l}\mathsf{aux}, \mathbf{B}, \mathbf{B}_{\mathsf{att}}, \mathbf{B}_{\mathsf{tag}}, \mathbf{A}_{\mathsf{tag}}, \widehat{\mathbf{M}}, \hat{\mathbf{s}}\mathbf{B}+\mathbf{e}_0, \hat{\mathbf{s}}\mathbf{W}+\mathbf{e}_W, \\ \boxed{\mathbf{s}_2\mathbf{B}_{\mathsf{att}}+\mathbf{e}_1, \mathbf{s}\mathbf{B}_{\mathsf{tag}}+\mathbf{e}_2, \mathbf{s}_3\mathbf{A}_{\mathsf{tag}}+\mathbf{e}_3}, \mathbf{s}_1\widehat{\mathbf{M}}+\mathbf{e}\end{array}\right)=1\right] \\
&- \Pr\left[\mathcal{A}_0\left(\mathsf{aux}, \mathbf{B}, \mathbf{B}_{\mathsf{att}}, \mathbf{B}_{\mathsf{tag}}, \mathbf{A}_{\mathsf{tag}}, \widehat{\mathbf{M}}, \mathbf{v}_0, \mathbf{v}_W, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}\right)=1\right], \\
\mathsf{Adv}^{\mathrm{POST}}_{\mathcal{A}_1}(\lambda) :=& \Pr\left[\mathcal{A}_1\left(\begin{array}{l}\mathsf{aux}, \mathbf{B}, \mathbf{B}_{\mathsf{att}}, \mathbf{B}_{\mathsf{tag}}, \mathbf{A}_{\mathsf{tag}}, \widehat{\mathbf{M}}, \hat{\mathbf{s}}\mathbf{B}+\mathbf{e}_0, \mathbf{K}, \\ \boxed{\mathbf{s}_2\mathbf{B}_{\mathsf{att}}+\mathbf{e}_1, \mathbf{s}\mathbf{B}_{\mathsf{tag}}+\mathbf{e}_2, \mathbf{s}_3\mathbf{A}_{\mathsf{tag}}+\mathbf{e}_3}, \mathbf{s}_1\widehat{\mathbf{M}}+\mathbf{e}\end{array}\right)=1\right] \\
&- \Pr\left[\mathcal{A}_1\left(\mathsf{aux}, \mathbf{B}, \mathbf{B}_{\mathsf{att}}, \mathbf{B}_{\mathsf{tag}}, \mathbf{A}_{\mathsf{tag}}, \widehat{\mathbf{M}}, \mathbf{v}_0, \mathbf{K}, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}\right)=1\right]
\end{aligned}$$

where

$$\begin{aligned}
&(\widehat{\mathbf{M}}, \mathbf{W}, \mathsf{aux}) \leftarrow \mathsf{Samp}(1^\lambda), \mathbf{B} \leftarrow \mathbb{Z}_q^{(2nm_0+n)\times(2nm_0+n)\kappa}, \\
&\mathbf{B}_{\mathsf{att}} \leftarrow \mathbb{Z}_q^{n\times 2m_0m_2}, \mathbf{B}_{\mathsf{tag}} \leftarrow \mathbb{Z}_p^{n\times m_1}, \mathbf{A}_{\mathsf{tag}} \leftarrow \mathbb{Z}_q^{n\times m_2\mathcal{K}}, \\
&\mathbf{s}_0 \leftarrow \mathbb{Z}_q^{2nm_0}, \mathbf{s}_1 \leftarrow \mathcal{D}^n_{\mathbb{Z},\chi}, \mathbf{s}_2 \leftarrow \mathcal{D}^n_{\mathbb{Z},\chi}, \mathbf{s}_3 \leftarrow \mathcal{D}^n_{\mathbb{Z},\chi}, \mathbf{s} \leftarrow \mathcal{D}^n_{\mathbb{Z},\chi}, \hat{\mathbf{s}} := [\mathbf{s}_0|\mathbf{s}_1], \\
&\mathbf{e}_0 \leftarrow \mathcal{D}^{(2nm_0+n)\kappa}_{\mathbb{Z},\chi}, \mathbf{e}_W \leftarrow \mathcal{D}^t_{\mathbb{Z},\chi}, \mathbf{e}_1 \leftarrow \mathcal{D}^{2m_0m_2}_{\mathbb{Z},\chi}, \\
&\mathbf{e}_2 \leftarrow \mathcal{D}^{m_1}_{\mathbb{Z},\chi}, \mathbf{e}_3 \leftarrow \mathcal{D}^{m_2\mathcal{K}}_{\mathbb{Z},\chi}, \mathbf{e} \leftarrow \mathcal{D}^{2m_0}_{\mathbb{Z},\chi}, \\
&\mathbf{v}_0 \leftarrow \mathbb{Z}_q^{(2nm_0+n)\kappa}, \mathbf{v}_W \leftarrow \mathbb{Z}_q^t, \mathbf{v}_1 \leftarrow \mathbb{Z}_q^{2m_0m_2}, \\
&\mathbf{v}_2 \leftarrow \mathbb{Z}_p^{m_1}, \mathbf{v}_3 \leftarrow \mathbb{Z}_q^{m_2\mathcal{K}}, \mathbf{v} \leftarrow \mathbb{Z}_p^{2m_0}, \\
&\mathbf{K} \leftarrow \mathbf{B}^{-1}(\mathbf{W}) \text{ with standard deviation } O(\sqrt{m_0 \log q}).
\end{aligned}$$

We say that the evasive LWE *assumption holds if for every PPT* Samp, $\mathcal{A}_1$*, there exists another PPT* $\mathcal{A}_0$ *and a polynomial* $Q(\cdot)$ *such that*

$$\mathsf{Adv}^{\mathrm{PRE}}_{\mathcal{A}_0}(\lambda) \geq \mathsf{Adv}^{\mathrm{POST}}_{\mathcal{A}_1}(\lambda)/Q(\lambda) - \mathsf{negl}(\lambda).$$

Remark 1 (Comparison with [41]). Our evasive LWE is the same as the evasive lattice assumption in [41], except for the boxed terms, which constitute three additional sLWE instances independent of $(\hat{\mathbf{s}}\mathbf{B} + \mathbf{e}_0, \hat{\mathbf{s}}\mathbf{W} + \mathbf{e}_W)$. Since the task of distinguishing distributions is not made easier by adding independent pseudorandom terms to the joint distributions [41], the combined hardness of the standard evasive LWE and sLWE assumptions implies that our modified evasive LWE assumption holds.

Remark 2. Consistent with [27, 41], our evasive assumptions are restricted to public-coin samplers, which is public-coin evasive LWE, a condition enforced by providing the distinguisher with the randomness of the sampler. Restricting the assumption to public coin samplers is sufficient to exclude counter-examples based on obfuscation. This differs from the versions used for witness encryption in [39], which employ private-coin samplers and provide the discriminator with auxiliary information generated by the sampler, rather than the sampler randomness.

Tensor LWE. Next, We briefly review tensor LWE introduced by [41]. For all $\mathbf{x}_1, ..., \mathbf{x}_Q$, the tensor LWE assumption asserts that

$$(\mathbf{A}, \{\underbrace{\mathbf{s}(\mathbf{I}_n \otimes \mathbf{r}_i^\top)(\mathbf{A} - \mathbf{x}_i \otimes \mathbf{G})}, \mathbf{r}_i^\top\}_{i\in[Q]}) \approx_c (\mathbf{A}, \{\$, \mathbf{r}_i^\top\}_{i\in[Q]}),$$

where $\mathbf{s}$ is a random vector, $\mathbf{r}_i$ is a low-norm Gaussian vector, and the symbol \$ represents an independent uniform sample. Here, we use the wavy underlines to denote the omission of small noise terms.

We now consider a matrix variant of tensor LWE assumption. Recall that $\mathbf{r}_i \leftarrow \mathcal{D}^m_{\mathbb{Z},\chi}$ is a low-norm vector in tensor LWE assumption. Next, we expand $\mathbf{r}_i$ to a matrix $\mathbf{R}_i \leftarrow \mathcal{D}^{k\times t}_{\mathbb{Z},\chi}$ for $k, t \in \mathbb{N}$. Accordingly, to facilitate matrix multiplication, the matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m\ell}$ is extended to $\mathbf{A}\otimes\mathbf{I}$. Thus, the tensor LWE assumption inherently asserts that the distribution of $\{\mathbf{s}(\mathbf{I}_n\otimes\mathbf{R}_i)((\mathbf{A}-\mathbf{x}_i\otimes\mathbf{G}_{n,q})\otimes\mathbf{I}_t)+\mathbf{e}_i\}_{i\in[Q]}$ is indistinguishable from random given the matrices $\mathbf{A}, \{\mathbf{R}_i\}_{i\in[Q]}$. Note that the secret $\mathbf{s}$ and the error terms $\{\mathbf{e}_i\}_{i\in[Q]}$ are chosen from their respective domains with suitable dimensions. It is straightforward to observe that the variant of tensor LWE for the case $t = 1$ implies tensor LWE. Below we provide a lemma for the tensor LWE variant, which proves that its hardness can be based on tensor LWE assumption, with the proof deferred to Appendix C.

Lemma 3 (Tensor LWE, adapted). *Let* $n, m, k, t, q, \ell, Q \in \mathbb{N}$ *be parameters and* δ, χ *be the Gaussian parameters. Then, for* $\mathbf{x}_1, \ldots, \mathbf{x}_Q \in \mathbb{Z}_q^\ell$*, the tensor* LWE *hardness assumption implies*

$$\left(\mathbf{A}, \{\mathbf{s}\left(\mathbf{I}_n \otimes \mathbf{R}_i\right)\left(\left(\mathbf{A} - \mathbf{x}_i \otimes \mathbf{G}_{n,q}\right) \otimes \mathbf{I}_t\right) + \mathbf{e}_i, \mathbf{R}_i\}_{i\in[Q]}\right) \approx_c \left(\mathbf{A}, \{\mathbf{v}_i, \mathbf{R}_i\}_{i\in[Q]}\right)$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m\ell}, \mathbf{s} \leftarrow \mathbb{Z}_q^{nk}, \mathbf{R}_i \leftarrow \mathcal{D}^{k\times t}_{\mathbb{Z},\delta}, \mathbf{e}_i \leftarrow \mathcal{D}^{mt\ell}_{\mathbb{Z},\chi}, \mathbf{v}_i \leftarrow \mathbb{Z}_q^{mt\ell}$.

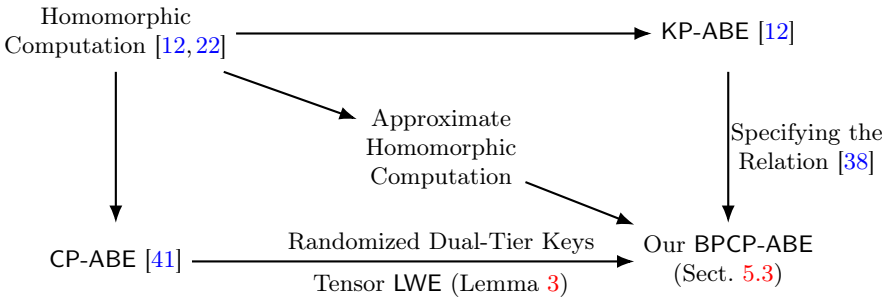


Fig. 1. Logic flow of technical overview.

5 BPCP-ABE for Circuits from Lattices

5.1 Technical Overview

Below we first give a high-level technical overview for our BPCP-ABE. Our construction starts by leveraging the techniques introduced in homomorphic computation on matrices [12,22] and Wee's CP-ABE construction [41]. Fig. 1 illustrates the logical flow of the technical overview of our construction.

We first review the BGG+ (KP-ABE) scheme proposed by Boneh et al. [12], which together with [22] laid the basis for the homomorphic computation technique. Subsequently, we briefly explain how, in [38], one can achieve the puncturing property from BGG+ by appropriately setting the circuit class. Shifting to another construction paradigm, we review the construction of Wee's CP-ABE [41], especially the essence of its randomized keys and the evasive and tensor LWE assumptions used to prove the security of this scheme. We generalize the latter and propose a variant of tensor LWE, which provides the basis to explain the extension of Wee's scheme to our BPCP-ABE. After that, we provide a high-level overview of the main techniques developed in our work, particularly the *randomized dual-tier keys* and the *approximate relationship* within the homomorphic computation for attributes and policies. Finally, by integrating the prior techniques, we sketch our BPCP-ABE construction.

Structure of BGG+. We briefly review the KP-ABE scheme for a circuit of size s and depth d in [12]. Fix some $\lambda, n \in \mathbb{Z}^+$, prime q and $m = O(n \log q)$. Given $\overline{\mathbf{A}} \in \mathbb{Z}_q^{n \times m\ell}$ and a function $g : \mathbb{Z}_q^\ell \to \{0,1\}$, we can deduce a matrix $\overline{\mathbf{A}}_g \in \mathbb{Z}_q^{n \times m}$ from [12, 22]. For any $\mathbf{t} \in \mathbb{Z}_q^\ell$, this will yield a low-norm matrix $\mathbf{H}_{\overline{\mathbf{A}},g,\mathbf{t}}$ with $\|\mathbf{H}_{\overline{\mathbf{A}},g,\mathbf{t}}\| = m^{O(d)} \cdot s$ satisfying $(\overline{\mathbf{A}} - \mathbf{t} \otimes \mathbf{G}_{n,q}) \cdot \mathbf{H}_{\overline{\mathbf{A}},g,\mathbf{t}} = \overline{\mathbf{A}}_g - g(\mathbf{t}) \cdot \mathbf{G}_{n,q}$, where $\mathbf{G}_{n,q}$ is a special gadget matrix [31]. Using this homomorphic computation between attribute $\mathbf{t}$ and policy function g, the BGG+ scheme is as follows,

$$
\begin{aligned}
\mathsf{mpk}_{\mathsf{KP\text{-}ABE}} &= (\mathbf{A}_0 \leftarrow \mathbb{Z}_q^{n \times m}, \overline{\mathbf{A}} \leftarrow \mathbb{Z}_q^{n \times m\ell}, \mathbf{P} \leftarrow \mathbb{Z}_q^{n \times \lambda}),\\
\mathsf{ct}_{\mathsf{KP\text{-}ABE}} &= (\overbrace{\mathbf{s} \cdot \mathbf{A}_0}^{\mathbf{c}_0}, \overbrace{\mathbf{s} \cdot (\overline{\mathbf{A}} - \mathbf{t} \otimes \mathbf{G}_{n,q})}^{\mathbf{c}_1}, \overbrace{\mathbf{s} \cdot \mathbf{P} + \mu \cdot \lfloor q/2 \rfloor}^{\mathbf{c}}), \mathbf{s} \leftarrow \mathbb{Z}_q^n, \mu \in \{0,1\}^\lambda,\\
\mathsf{sk}_{\mathsf{KP\text{-}ABE}} &= (\mathbf{K}_g \leftarrow \mathcal{D}_{\mathbb{Z},\tau}^{2m \times \lambda} \text{ s.t. } [\mathbf{A}_0 \mid \overline{\mathbf{A}}_g] \cdot \mathbf{K}_g = \mathbf{P}).
\end{aligned}
$$

From BGG+ to PE: Enabled by Appropriate Circuit Class Setting. Following the idea of [38], we define a family of functions $\mathcal{G} := \{g_{t'} : \mathbb{Z}_q^{\mathcal{K}} \to \{0,1\} \mid t' \in \mathbb{Z}_q\}$ for the puncturing property. In particular, for a set of tags $\mathbf{t} = \{t_1, \ldots, t_{\mathcal{K}}\}$, we have $g_{t'}(\mathbf{t}) = \mathcal{K} - \sum_{i\in[\mathcal{K}]} EQ_{t'}(t_i)$,[5] where $EQ_{t'} : \mathbb{Z}_q \to \{0,1\}$ is an equality test function indexed by a tag t', namely $EQ_{t'}(t) = \begin{cases} 0 & \text{if } t'=t \\ 1 & \text{otherwise} \end{cases}$. Assume that no two elements in $\mathbf{t}$ are equal. Then,

$$g_{t'}(\mathbf{t}) = \mathcal{K} - \sum_{i\in[\mathcal{K}]} EQ_{t'}(t_i) = \begin{cases} 0 & \text{if } \{t_1, \ldots, t_{\mathcal{K}}\} \cap t' = \emptyset \\ 1 & \text{otherwise.} \end{cases} \tag{1}$$

Plugging this restricted relation into the function family of BGG+ captures the puncturing property. In a nutshell, we replace g in BGG+ with $g_{t'}$ as described above, and the attribute $\mathbf{t}$ in BGG+ is marked as the tag. Then, puncture is achieved by delegating the policy key. For simplicity, we set $\mathcal{K} = \ell$ in the overview.

Structure of Wee. Briefly review the CP-ABE scheme for a circuit of size s and depth d in [41]. For any $\mathbf{x} \in \mathbb{Z}_q^\ell$ and $f : \mathbb{Z}_q^\ell \to \{0,1\}$, then $(\underline{\mathbf{A}} - \mathbf{x} \otimes \mathbf{G}_{n,q}) \cdot \mathbf{H}_{\underline{\mathbf{A}},f,\mathbf{x}} = \underline{\mathbf{A}}_f - f(\mathbf{x}) \cdot \mathbf{G}_{n,q}$. Wee's scheme is as follows,

$$\begin{aligned}
\mathsf{mpk}_{\mathsf{CP\text{-}ABE}} &= (\underline{\mathbf{A}} \leftarrow \mathbb{Z}_q^{n\times m\ell}, \mathbf{B} \leftarrow \mathbb{Z}_q^{(n+mn)\times(m+m^2)}, \mathbf{D} \leftarrow \mathbb{Z}_q^{n\times m}, \mathbf{u} \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^m), \\
\mathsf{ct}_{\mathsf{CP\text{-}ABE}} &= (\overbrace{[\mathbf{s}_0|\mathbf{s}_1]\cdot\mathbf{B}}^{\mathbf{c}_0}, \overbrace{\mathbf{s}_0\cdot\mathbf{D} + \mathbf{s}_1\cdot(\underline{\mathbf{A}}_f\mathbf{u}^\top \otimes \mathbf{I}_m) + \mu\cdot\mathbf{g}}^{\mathbf{c}}), \mathbf{s}_0 \leftarrow \mathbb{Z}_q^n, \mathbf{s}_1 \leftarrow \mathbb{Z}_q^{mn}, \mu \in \{0,1\}, \\
\mathsf{sk}_{\mathsf{CP\text{-}ABE}} &= \left(\mathbf{r} \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^m, \mathbf{K}_{\mathbf{x}} \leftarrow \mathcal{D}_{\mathbb{Z},\tau}^{(m+m^2)\times(m\ell+1)} \text{ s.t. } \mathbf{B}\cdot\mathbf{K}_{\mathbf{x}} = \begin{pmatrix} \mathbf{D}\mathbf{r}^\top \\ (\underline{\mathbf{A}} - \mathbf{x}\otimes\mathbf{G}_{n,q})\otimes\mathbf{r}^\top \end{pmatrix}\right).
\end{aligned}$$

A notable observation is that even when the attribute encoding $(\underline{\mathbf{A}} - \mathbf{x} \otimes \mathbf{G}_{n,q})$ undergoes a tensor product operation with a random Gaussian vector $\mathbf{r}$, the correctness of the above homomorphic computation can still be guaranteed. Specifically, using the properties of the mixed product, we have

$$((\underline{\mathbf{A}} - \mathbf{x}\otimes\mathbf{G}_{n,q})\otimes\mathbf{r}^\top)\cdot\mathbf{H}_{\underline{\mathbf{A}},f,\mathbf{x}}\mathbf{u}^\top = ((\underline{\mathbf{A}}_f\mathbf{u}^\top - f(\mathbf{x})\cdot\mathbf{G}_{n,q}\mathbf{u}^\top)\otimes\mathbf{I}_m)\cdot\mathbf{r}^\top.$$

If $f(\mathbf{x}) = 0$, the message μ is recovered by computing the following term

$$\mathbf{c}\cdot\mathbf{r}^\top - \mathbf{c}_0\cdot\mathbf{K}_{\mathbf{x}}\cdot\begin{pmatrix} 1 \\ \mathbf{H}_{\underline{\mathbf{A}},f,\mathbf{x}}\mathbf{u}^\top \end{pmatrix}.$$

Randomizing Dual-Tier Keys. In the following, we extend Wee's key [41] to a key with duality (KP & CP) for constructing our BPCP-ABE. In particular,

- observe that the correctness of the above homomorphic computation on matrices is preserved even when the attribute encoding $(\underline{\mathbf{A}} - \mathbf{x}\otimes\mathbf{G}_{n,q})$ is used to perform tensor operations with a structured Gaussian matrix $\mathbf{V} := \begin{pmatrix} \mathbf{R} \\ \mathbf{r}^\top \end{pmatrix}$,

[5] We adopt the convention of lattice construction. When $g_{t'}(\mathbf{t}) = 0$, it means that the tag t' has not been punctured, and the ciphertext related to $\mathbf{t}$ can be decrypted using the key associated with the policy $g_{t'}$.

where $\mathbf{R} \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^{m\times m}$. Specifically, by leveraging the mixed-product property, we obtain

$$((\underline{\mathbf{A}} - \mathbf{x}\otimes \mathbf{G}_{n,q})\otimes \mathbf{V})\cdot(\mathbf{H}_{\underline{\mathbf{A}},f,\mathbf{x}}\mathbf{u}^\top \otimes \mathbf{I}_{m+1}) = ((\underline{\mathbf{A}}_f\mathbf{u}^\top - f(\mathbf{x})\cdot \mathbf{G}_{n,q}\mathbf{u}^\top)\otimes \mathbf{I}_{2m})\cdot \mathbf{V};$$

- we have a special requirement for $\mathbf{R}$, that is, the low-norm matrix $\mathbf{R}$ satisfies $\mathbf{C}\cdot\mathbf{R} = \mathbf{A}$ for $\mathbf{A}, \mathbf{C} \leftarrow \mathbb{Z}_q^{n\times m}$, where the authority has trapdoors for $\mathbf{A}$;
- obtain a dual-tier key

$$\mathsf{sk} = (\mathbf{V},\quad [\mathbf{A}|\mathbf{D}\mathbf{r}^\top]^{-1},\quad \mathbf{K}_\mathbf{x})$$

where (1) $[\mathbf{A}|\mathbf{D}\mathbf{r}^\top]^{-1}$ is the trapdoor for $[\mathbf{A}|\mathbf{D}\mathbf{r}^\top]$ from $\mathbf{A}^{-1}$(Lemma 5), which can be used to derive a policy-related key via key delegation and (2) $\mathbf{K}_\mathbf{x}$ satisfies $\mathbf{B}\cdot\mathbf{K}_\mathbf{x} = (\underline{\mathbf{A}} - \mathbf{x}\otimes\mathbf{G}_{n,q})\otimes \begin{pmatrix}\mathbf{R}\\ \mathbf{r}^\top\end{pmatrix}$, which is an attribute key that can decrypt ciphertext under the policy satisfied by $\mathbf{x}$. Observe that all items in the dual-tier key are randomized similarly to $\mathsf{sk}_{\mathsf{CP\text{-}ABE}}$ of [41] via Gaussian random items.

For the ciphertext items, we replace $\mathbf{c}$ in Wee's CP-ABE with

$$\mathbf{s}_0\cdot[\mathbf{C}|\mathbf{D}] + \mathbf{s}_1\cdot(\mathbf{A}_f\mathbf{u}^\top\otimes\mathbf{I}_{2m}) + \mu\cdot\mathbf{g}.$$

Only when the key $\mathbf{K}_\mathbf{x}$, with its attribute $\mathbf{x}$ satisfying the policy f, is used can the term $\mathbf{s}_1(\mathbf{A}_f\mathbf{u}^\top\otimes\mathbf{I}_{2m})\mathbf{V}$ be eliminated from $\mathbf{c}\cdot\mathbf{V}$, and as a result, $\mathbf{s}_0\cdot[\mathbf{A}|\mathbf{D}\mathbf{r}^\top] + \mu\cdot\mathbf{g}$ can be obtained. Then, using the trapdoor $[\mathbf{A}|\mathbf{D}\mathbf{r}^\top]^{-1}$, we eliminate the masking term $\mathbf{s}_0\cdot[\mathbf{A}|\mathbf{D}\mathbf{r}^\top]$, thereby recovering the message μ. Up to this point, it is noted that by adding $\mathbf{s}_0\cdot(\overline{\mathbf{A}} - \mathbf{t}\otimes\mathbf{G}_{n,q})$ to the ciphertext, we can obtain an ABE scheme that supports fine-grained bilateral access control over encrypted data. In this scheme, the policy g in the key is defined as the restricted relation mentioned previously (just as in (1)), and the attributes $\mathbf{t}$ in the ciphertext are defined as the tags. This yields a puncturable CP-ABE, where $[\mathbf{A}|\mathbf{D}\mathbf{r}^\top]^{-1}$ in sk supports key delegation, which can be leveraged to achieve key puncturing. Taking the punctured tag t' as an example, by applying the technique in [15], we can derive $[\mathbf{A}|\mathbf{D}\mathbf{r}^\top|\overline{\mathbf{A}}_{g_{t'}}]^{-1}$ from $[\mathbf{A}|\mathbf{D}\mathbf{r}^\top]^{-1}$. As a result, even if $f(\mathbf{x}) = 0$, a user with the secret key $(\mathbf{r}, \mathbf{R}, [\mathbf{A}|\mathbf{D}\mathbf{r}^\top|\overline{\mathbf{A}}_{g_{t'}}]^{-1}, \mathbf{K}_\mathbf{x})$ is still unable to decrypt the ciphertext under the tag $\mathbf{t}$ that contains t' ($g_{t'}(\mathbf{t}) = 1$). Besides, the introduction of $\mathbf{V}$ in the key ensures that our puncturable CP-ABE prevents collusion between users who only satisfy $f(\mathbf{x}) = 0$ and users who only satisfy $g_{t'}(\mathbf{t}) = 0$. For this randomized key that connects the two types of ABE in a coherent way, we refer to it as the structured Gaussian-related trapdoors (SGR-Trapdoors) and provide its formal definition in Sect. 5.2.

Approximate Homomorphic Computation. A natural idea for achieving batch puncturing is to construct a large circuit $g_{t'_1,\ldots,t'_k}$ such that, for some $k\in\mathbb{N}$ and $\mathbf{t}\in\mathbb{Z}_q^\ell$, if the intersection of $\mathbf{t}$ and the punctured tag set $\{t'_1,\ldots,t'_k\}$ is not empty, then $g_{t'_1,\ldots,t'_k}(\mathbf{t}) = 1$, otherwise $g_{t'_1,\ldots,t'_k}(\mathbf{t}) = 0$. However, as the number of punctured tags increases, this approach would require a larger circuit depth d,

which in turn requires a larger modulus q. Since q is fixed a priori, the maximum number of tags to be punctured at each time is also predetermined. Thus, even if we puncture a single tag, an additional matrix will be added to the punctured key, which is equivalent to the storage overhead incurred when puncturing the maximum allowable number of tags in a single operation.

To address this obstacle, we observe an approximately homomorphic computation on matrices that, although accompanied by some noise terms, does not cause a noise blow-up. In other words, applying this relationship to ABE for circuits does not affect the correctness of decryption. Our starting point is the observation that the relation between attributes and policies exhibits additive homomorphism, namely

$$(\overline{\mathbf{A}} - \mathbf{t} \otimes \mathbf{G}_{n,q}) \cdot \sum_{j\in[k]} \mathbf{H}_{\overline{\mathbf{A}},g_j,\mathbf{t}} = \sum_{j\in[k]} \overline{\mathbf{A}}_{g_j} - \sum_{j\in[k]} g_j(\mathbf{t}) \cdot \mathbf{G}_{n,q} \mod q.$$

where $\{g_j\}_{j\in[k]}$ represent the functions used to puncture the tags $\{t'_j\}_{j\in[k]}$. Simply let $\mathbf{H}'_{\overline{\mathbf{A}},g,\mathbf{t}} = \sum_{j\in[k]} \mathbf{H}_{\overline{\mathbf{A}},g_j,\mathbf{t}}, \mathbf{A}'_g = \sum_{j\in[k]} \overline{\mathbf{A}}_{g_j}, g'(\mathbf{t}) = \sum_{j\in[k]} g_j(\mathbf{t})$, we have

$$(\overline{\mathbf{A}} - \mathbf{t} \otimes \mathbf{G}_{n,q}) \cdot \mathbf{H}'_{\overline{\mathbf{A}},g,\mathbf{t}} = \mathbf{A}'_g - g'(\mathbf{t}) \cdot \mathbf{G}_{n,q} \mod q. \tag{2}$$

Note that this method also requires a larger modulus q as the number of tags increases at each puncture. This is because $\|\mathbf{H}'_{\overline{\mathbf{A}},g,\mathbf{t}}\|$ grows as the number of punctured tags grows. To fix this, we modify (2) to a variant that is independent of the circuit depth d (without parameters m, q): given a small modulus $p(< q)$, $m_1 = n \cdot \lceil \log p \rceil$ and a low-norm $\mathbf{U} \leftarrow \mathcal{D}^{m\times m_1}_{\mathbb{Z},\chi}$, we let $\hat{\mathbf{H}}_{\overline{\mathbf{A}},g,\mathbf{t}} = \mathbf{H}'_{\overline{\mathbf{A}},g,\mathbf{t}} \cdot \mathbf{U}, \hat{\mathbf{A}} = \mathbf{A}'_g \cdot \mathbf{U}, \hat{\mathbf{G}}_{n,q} = \mathbf{G}_{n,q} \cdot \mathbf{U}$. This yields the following approximate relation

$$\lfloor (\overline{\mathbf{A}} - \mathbf{t} \otimes \mathbf{G}_{n,q}) \cdot \hat{\mathbf{H}}_{\overline{\mathbf{A}},g,\mathbf{t}} \rceil_p \approx \lfloor \hat{\mathbf{A}} \rceil_p - \lfloor g'(\mathbf{t}) \cdot \hat{\mathbf{G}}_{n,q} \rceil_p$$

where $\lfloor \cdot \rceil_p$ is a rounding function that rounds term by term from $\mathbb{Z}_q$ to $\mathbb{Z}_p$ (see Definition 1 for more details) and the $\approx$ ignores some noise terms that do not affect the correctness of decryption. Observe that if $g'(\mathbf{t}) = 0$, we obtain a matrix $\lfloor \hat{\mathbf{A}} \rceil_p$ associated with the punctured tags $\{t'_j\}_{j\in[k]}$, which is independent of d. Thus, the punctured keys sampled in this manner are also independent of d.

Our BPCP-ABE. Building upon the techniques outlined above, we present an informal construction of BPCP-ABE under the setting of single puncturing with two tags t'_1, t'_2 for simplicity. To enhance readability, we briefly give some parameter settings: $m_0 = O(n \log p), m_1 = n\lceil \log p \rceil, m_2 = n\lceil \log q \rceil, \kappa = O(\log q)$. Let $g_{t'_1}$ and $g_{t'_2}$ be the functions for punctured tags t'_1 and t'_2 (defined as in (1)), and the matrices associated with them be $\mathbf{A}_{t'_1}$ and $\mathbf{A}_{t'_2}$ respectively. Note that this simplified version has security flaws that are easily fixable, and serves only

to clarify our techniques. Our BPCP-ABE can be summarised as follows:

$$\mathsf{mpk} = \begin{pmatrix} \mathbf{A}, \mathbf{C}, \mathbf{D} \leftarrow \mathbb{Z}_p^{n\times m_0}, \mathbf{A}_{\mathsf{att}}, \mathbf{A}_{\mathsf{tag}} \leftarrow \mathbb{Z}_q^{n\times m_2\ell}, \mathbf{B} \leftarrow \mathbb{Z}_q^{2nm_0\times 2nm_0\kappa}, \\ \mathbf{B}_{\mathsf{att}} \leftarrow \mathbb{Z}_q^{n\times 2m_0m_2}, \mathbf{B}_{\mathsf{tag}} \leftarrow \mathbb{Z}_q^{n\times m_1}, \mathbf{U} \leftarrow \mathcal{D}_{\mathbb{Z},\tau_C}^{m_2\times m_1}, \mathbf{u} \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^{m_2} \end{pmatrix}.$$

$$\mathsf{ct} = \left(\overbrace{\mathbf{s}_0\mathbf{B}}^{\mathbf{c}_0}, \overbrace{\mathbf{s}_0\mathbf{G}_{2nm_0,q} + \mathbf{s}_1\mathbf{B}_{\mathsf{att}}}^{\mathbf{c}_1}, \overbrace{\mathbf{s}_2\mathbf{G}_{n,p} + \mathbf{s}\mathbf{B}_{\mathsf{tag}}}^{\mathbf{c}_2}, \overbrace{\mathbf{s}_2(\mathbf{A}_{\mathsf{tag}} - \mathbf{t}\otimes\mathbf{G}_{n,q})}^{\mathbf{c}_3}, \overbrace{\mathbf{s}[\mathbf{C}|\mathbf{D}] + \mathbf{s}_1\lfloor\mathbf{B}_f\rceil_p + \mu\mathbf{g}}^{\mathbf{c}} \right),$$

$$\mathbf{s}_0 \leftarrow \mathbb{Z}_q^{2nm_0}, \mathbf{s}_1, \mathbf{s}_2, \mathbf{s} \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^{n}, \mathbf{B}_f = \mathbf{B}_{\mathsf{att}} \cdot \mathbf{G}_{2nm_0,q}^{-1}(\mathbf{A}_f\mathbf{u}^\top \otimes \mathbf{I}_{2m_0}),$$

$$\mathsf{sk}_{\mathbf{x},\emptyset} = \begin{pmatrix} \mathbf{V} \text{ s.t. } [\mathbf{C}|\mathbf{D}]\cdot\mathbf{V} = [\mathbf{A}|\mathbf{D}\mathbf{r}^\top], \quad [\mathbf{A}|\mathbf{D}\mathbf{r}^\top]^{-1}, \\ \mathbf{K}_{\mathbf{x}} \text{ s.t. } \mathbf{B}\cdot\mathbf{K}_{\mathbf{x}} = (\mathbf{A}_{\mathsf{att}} - \mathbf{x}\otimes\mathbf{G}_{n,q})\otimes\mathbf{V} \end{pmatrix}.$$

$$\mathsf{sk}_{\mathbf{x},1} = \left(\mathbf{V},\ [\mathbf{A}|\mathbf{D}\mathbf{r}^\top|\hat{\mathbf{B}}]^{-1},\ \mathbf{K}_{\mathbf{x}}\right),\ \hat{\mathbf{B}} = \mathbf{B}_{\mathsf{tag}}\cdot\mathbf{G}_{n,p}^{-1}(\lfloor(\mathbf{A}_{t_1'} + \mathbf{A}_{t_2'})\cdot\mathbf{U}\rceil_p).$$

If $f(\mathbf{x}) = 0$, the decryptor computes $\mathbf{c}_0 \cdot \mathbf{K}_{\mathbf{x}} \cdot \left(\mathbf{H}_{\mathbf{A}_{\mathsf{att}},f,\mathbf{x}}\mathbf{u}^\top \otimes \mathbf{I}_{m_0+1}\right)$ to obtain $\mathbf{s}_0(\mathbf{A}_f\mathbf{u}^\top \otimes \mathbf{I}_{2m_0})\mathbf{V}$. It subtracts $\mathbf{s}_0(\mathbf{A}_f\mathbf{u}^\top \otimes \mathbf{I}_{2m_0})\mathbf{V}$ from $\mathbf{c}_1 \cdot \mathbf{G}_{2mm_0,q}^{-1}(\mathbf{A}_f\mathbf{u}^\top \otimes \mathbf{I}_{2m_0}) \cdot \mathbf{V}$ to get $\mathbf{s}_1\mathbf{B}_f\mathbf{V}$. The latter is fed into the round function $\lfloor\cdot\rceil_p$ to get $\mathbf{s}_1\lfloor\mathbf{B}_f\rceil_p\mathbf{V}$ from the new property we introduced (Lemma 2). Next, it computes $\mathbf{c}\mathbf{V} - \mathbf{s}_1\lfloor\mathbf{B}_f\rceil_p\mathbf{V}$ to get $\mathbf{s}[\mathbf{A}|\mathbf{D}\mathbf{r}^\top] + \mu\mathbf{g}\mathbf{V}$.

- If the key has not been punctured, one can use the trapdoor $[\mathbf{A}|\mathbf{D}\mathbf{r}^\top]^{-1}$ in $\mathsf{sk}_{\mathbf{x},\emptyset}$ to eliminate the masking term in $\mathbf{s}[\mathbf{A}|\mathbf{D}\mathbf{r}^\top] + \mu\mathbf{g}\mathbf{V}$, thereby recovering μ.
- If the key has been punctured, which implies that the decryptor has obtained $\mathsf{sk}_{\mathbf{x},1}$ through the puncture algorithm. Compute

$$\underbrace{\mathbf{c}_2 \cdot \mathbf{G}_{n,p}^{-1}(\lfloor(\mathbf{A}_{t_1'} + \mathbf{A}_{t_2'})\mathbf{U}\rceil_p)}_{\approx\ \mathbf{s}_2\lfloor(\mathbf{A}_{t_1'}+\mathbf{A}_{t_2'})\mathbf{U}\rceil_p + \mathbf{s}\hat{\mathbf{B}}} - \underbrace{\lfloor\mathbf{c}_3 \cdot (\mathbf{H}_{\mathbf{A}_{\mathsf{tag}},g_{t_1'},\mathbf{t}} + \mathbf{H}_{\mathbf{A}_{\mathsf{tag}},g_{t_2'},\mathbf{t}}) \cdot \mathbf{U}\rceil_p}_{\approx\ \lfloor\mathbf{s}_2(\mathbf{A}_{t_1'}+\mathbf{A}_{t_2'})\mathbf{U}\rceil_p}$$

  to obtain $\mathbf{s}\hat{\mathbf{B}}$, where we use $(\mathbf{A}_{\mathsf{tag}} - \mathbf{t}\otimes\mathbf{G}_{n,q}) \cdot \mathbf{H}_{\mathbf{A}_{\mathsf{tag}},g_{t_i'},\mathbf{t}} = \mathbf{A}_{t_i'}$ as $g_{t_i'}(\mathbf{t}) = 0$ for each $i \in [2]$ and Lemma 1. Immediately after that, concatenating $\mathbf{s}[\mathbf{A}|\mathbf{D}\mathbf{r}^\top] + \mu\mathbf{g}\mathbf{V}$ and $\mathbf{s}\hat{\mathbf{B}}$ yields $\mathbf{s}[\mathbf{A}|\mathbf{D}\mathbf{r}^\top|\hat{\mathbf{B}}] + [\mu\mathbf{g}\mathbf{V}|\mathbf{0}]$. Similarly, the decryptor uses $\mathsf{sk}_{\mathbf{x},1}$ to recover μ.

5.2 Structured Gaussian-Related Trapdoors

Before providing our BPCP-ABE, we first introduce the following technique that help complete our construction. The works in [21,31] present an algorithm that, given $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{C} \in \mathbb{Z}_q^{n\times m}$ and an associated trapdoor $\mathbf{C}_\tau^{-1}$, samples a low-norm matrix $\mathbf{R} \in \mathbb{Z}^{m\times m}$ such that $\mathbf{C}\cdot\mathbf{R} = \mathbf{A}$. To achieve our construction, we propose the structured Gaussian-related trapdoors (SGR-Trapdoors). Below we give its construction method based on [21,31].

Proposition 1. *There exists a PPT algorithm* SGR-Samp *that, on input random matrices* $\mathbf{A}, \mathbf{C}, \mathbf{D} \in \mathbb{Z}_q^{n\times m}$*, a* τ_A*-trapdoor* $\mathbf{A}_{\tau_A}^{-1}$ *for* $\mathbf{A}$ *and a* τ_C*-trapdoor* $\mathbf{C}_{\tau_C}^{-1}$ *for* $\mathbf{C}$*, a vector* $\mathbf{r}$ *sampled from* $\mathcal{D}_{\tau_C}^m$*, outputs a trapdoor* $\mathbf{K}$ *and a structured Gaussian matrix* $\begin{pmatrix}\mathbf{R}\\ \mathbf{r}^\top\end{pmatrix}$ *such that (1)* $\mathbf{K}$ *is a* τ'*-trapdoor for* $[\mathbf{A}|\mathbf{D}\mathbf{r}^\top]$*, where* $\tau' = \|\widetilde{\mathbf{A}_{\tau_A}^{-1}}\| \cdot \omega(\sqrt{\log m})$*; (2)* $\mathbf{K}$ *is a* τ'*-trapdoor for* $[\mathbf{C}|\mathbf{D}]\cdot\begin{pmatrix}\mathbf{R}\\ \mathbf{r}^\top\end{pmatrix}$*; (3)* $\mathbf{R}$ *is distributed statistically close to* $\mathcal{D}_{\Lambda_q^{\mathbf{A}}(\mathbf{C}),\tau_C}$*; (4)* $\mathbf{K}$ *is distributed statistically close to* $\mathcal{D}_{\Lambda_q^\perp([\mathbf{A}|\mathbf{D}\mathbf{r}^\top]),\tau'}$.

$\mathsf{SGR\text{-}Samp}(\mathbf{A}, \mathbf{A}_{\tau_A}^{-1}, \mathbf{C}, \mathbf{C}_{\tau_C}^{-1}, \mathbf{D}, \mathbf{r})$ *works as follows:*

1. *Sample* $\mathbf{R} \leftarrow \mathbf{C}^{-1}(\mathbf{A}, \tau_C)$ *such that* $\mathbf{C}\cdot\mathbf{R} = \mathbf{A}$*;*
2. *Run* $[\mathbf{A}|\mathbf{D}\mathbf{r}^\top]_{\tau_A}^{-1} \leftarrow \mathsf{TrapExt}(\mathbf{A}, \mathbf{A}_{\tau_A}^{-1}, \mathbf{D}\mathbf{r}^\top)$ *where* $\|\widetilde{[\mathbf{A}|\mathbf{D}\mathbf{r}^\top]_{\tau_A}^{-1}}\| = \|\widetilde{\mathbf{A}_{\tau_A}^{-1}}\|$*;*
3. *Run* $\mathbf{K} \leftarrow \mathsf{TrapRand}([\mathbf{A}|\mathbf{D}\mathbf{r}^\top], [\mathbf{A}|\mathbf{D}\mathbf{r}^\top]_{\tau_A}^{-1}, \tau_A)$ *where* $\mathbf{K}$ *is a* τ'*-trapdoor for* $[\mathbf{A}|\mathbf{D}\mathbf{r}^\top]$*;*
4. *Output a* SGR-Trapdoors $\mathbf{K}$ *and a structured Gaussian matrix* $\begin{pmatrix}\mathbf{R}\\ \mathbf{r}^\top\end{pmatrix}$.

Proof. By item 3 of Lemma 5 and the definition of lattice trapdoors, the output satisfies (1), (3) and (4). Since $\mathbf{C}\cdot\mathbf{R} = \mathbf{A}$, $[\mathbf{C}|\mathbf{D}]\cdot\begin{pmatrix}\mathbf{R}\\ \mathbf{r}^\top\end{pmatrix} = [\mathbf{A}|\mathbf{D}\mathbf{r}^\top]$. The output satisfies (2).

5.3 Construction

We first define a family of functions $\mathcal{G} := \{g_{t'} : \mathbb{Z}_q^{\mathcal{K}} \to \{0,1\} \mid t' \in \mathbb{Z}_q\}$ for the puncturing property. In particular, for a set of tags $\mathbf{t} := \{t_1, \ldots, t_{\mathcal{K}}\}$, we have $g_{t'}(\mathbf{t}) = \mathcal{K} - \sum_{i\in[\mathcal{K}]} EQ_{t'}(t_i)$, where $EQ_{t'} : \mathbb{Z}_q \to \{0,1\}$ is an equality test function indexed by a tag t', namely $EQ_{t'}(t) = \begin{cases}0 \text{ if } t'=t\\ 1 \text{ otherwise}\end{cases}$. Assume that no two elements in $\mathbf{t}$ are equal. Then, $g_{t'}(\mathbf{t}) = \begin{cases}0 \text{ if } \{t_1,\ldots,t_{\mathcal{K}}\}\cap t' = \emptyset\\ 1 \text{ otherwise}\end{cases}$. Now, we construct a BPCP-ABE scheme for circuit class $\mathcal{F} = \{f : \mathbb{Z}_q^\ell \to \{0,1\}\}$.

– $\mathsf{Setup}(1^\lambda, \ell, \mathcal{K}, \eta_{max})$. Sample

$$(\mathbf{A}, \mathbf{A}_{\tau_A}^{-1}) \leftarrow \mathsf{TrapGen}(1^n, 1^{m_0}, p), (\mathbf{B}, \mathbf{B}_{\tau_B}^{-1}) \leftarrow \mathsf{TrapGen}(1^{2nm_0+n}, 1^{(2nm_0+n)\kappa}, q),$$
$$(\mathbf{C}, \mathbf{C}_{\tau_C}^{-1}) \leftarrow \mathsf{TrapGen}(1^n, 1^{m_0}, p), \mathbf{A}_{\mathsf{att}} \leftarrow \mathbb{Z}_q^{n\times m_2\ell}, \mathbf{A}_{\mathsf{tag}} \leftarrow \mathbb{Z}_q^{n\times m_2\mathcal{K}}, \mathbf{D} \leftarrow \mathbb{Z}_p^{n\times m_0},$$
$$\mathbf{B}_{\mathsf{att}} \leftarrow \mathbb{Z}_q^{n\times 2m_0m_2}, \mathbf{B}_{\mathsf{tag}} \leftarrow \mathbb{Z}_p^{n\times m_1}, \mathbf{M} \leftarrow \mathbb{Z}_q^{n\times 2m_0}, \mathbf{U} \leftarrow \mathcal{D}_{\mathbb{Z},\tau_C}^{m_2\times m_1}, \mathbf{u} \leftarrow \mathcal{D}_{\mathbb{Z},\chi_1}^{m_2}.$$

Output

$$\mathsf{mpk} := (\mathbf{A}, \mathbf{A}_{\mathsf{att}}, \mathbf{A}_{\mathsf{tag}}, \mathbf{B}, \mathbf{B}_{\mathsf{att}}, \mathbf{B}_{\mathsf{tag}}, \mathbf{C}, \mathbf{D}, \mathbf{M}, \mathbf{U}, \mathbf{u}),$$
$$\mathsf{msk} := (\mathbf{A}_{\tau_A}^{-1}, \mathbf{B}_{\tau_B}^{-1}, \mathbf{C}_{\tau_C}^{-1}).$$

– $\mathsf{Enc}(\mathsf{mpk}, f, \mu, \{t_1, \ldots, t_{\mathcal{K}}\})$. Set $\mathbf{t} := \{t_1, \ldots, t_{\mathcal{K}}\}$. Compute $\mathbf{A}_f = \mathsf{EvalF}(\mathbf{A}_{\mathsf{att}}, f)$ and $\mathbf{B}_f = \mathbf{B}_{\mathsf{att}} \cdot \mathbf{G}_{2nm_0,q}^{-1}(\mathbf{A}_f\mathbf{u}^\top \otimes \mathbf{I}_{2m_0}) \mod q$. Sample

$$\mathbf{s}_0 \leftarrow \mathbb{Z}_q^{2nm_0}, \mathbf{s}_1 \leftarrow \mathcal{D}_{\mathbb{Z},\chi_1}^n, \mathbf{s}_2 \leftarrow \mathcal{D}_{\mathbb{Z},\chi_1}^n, \mathbf{s}_3 \leftarrow \mathcal{D}_{\mathbb{Z},\chi_0}^n, \mathbf{s} \leftarrow \mathcal{D}_{\mathbb{Z},\chi_1}^n,$$
$$\mathbf{e}_0 \leftarrow \mathcal{D}_{\mathbb{Z},\chi_0}^{(2nm_0+n)\kappa}, \mathbf{e}_1 \leftarrow \mathcal{D}_{\mathbb{Z},\chi_1}^{2m_0m_2}, \mathbf{e}_2 \leftarrow \mathcal{D}_{\mathbb{Z},\chi_1}^{m_1}, \mathbf{e}_3 \leftarrow \mathcal{D}_{\mathbb{Z},\chi_0}^{m_2\mathcal{K}}, \mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z},\chi_1}^{2m_0}.$$

Compute the following:

$$\begin{aligned}
\mathbf{c}_0 &:= [\mathbf{s}_0|\mathbf{s}_1]\mathbf{B} + \mathbf{e}_0 && \bmod q,\\
\mathbf{c}_1 &:= \mathbf{s}_0\mathbf{G}_{2nm_0,q} + \mathbf{s}_2\mathbf{B}_{\mathsf{att}} + \mathbf{e}_1 && \bmod q,\\
\mathbf{c}_2 &:= \mathbf{s}_3\mathbf{G}_{n,p} + \mathbf{s}\mathbf{B}_{\mathsf{tag}} + \mathbf{e}_2 && \bmod p,\\
\mathbf{c}_3 &:= \mathbf{s}_3(\mathbf{A}_{\mathsf{tag}} - \mathbf{t}\otimes\mathbf{G}_{n,q}) + \mathbf{e}_3 && \bmod q,\\
\mathbf{c} &:= \mathbf{s}[\mathbf{C}|\mathbf{D}] + \mathbf{s}_1\lfloor\mathbf{M}\rceil_p + \mathbf{s}_2\lfloor\mathbf{B}_f\rceil_p + \mathbf{e} + \mu\cdot\mathbf{g} && \bmod p.
\end{aligned}$$

Output $\mathsf{ct} := (\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c})$.

– $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathbf{x})$. Sample $\mathbf{r} \leftarrow D^{m_0}_{\mathbb{Z},\tau_C}$. Build a $\mathsf{SGR\text{-}Trapdoors}$ for matrix $\mathbf{F}_0 = \left[\mathbf{A}|\mathbf{D}\mathbf{r}^\top\right]$: $\left(\mathbf{K}_{1,\emptyset}, \mathbf{V}\right) \leftarrow \mathsf{SGR\text{-}Samp}\left(\mathbf{A}, \mathbf{A}^{-1}_{\tau_A}, \mathbf{C}, \mathbf{C}^{-1}_{\tau_C}, \mathbf{D}, \mathbf{r}\right)$ where $\mathbf{K}_{1,\emptyset}$ is a τ_0-trapdoor for $\mathbf{F}_0$ and $\mathbf{V} := \begin{pmatrix}\mathbf{R}\\ \mathbf{r}^\top\end{pmatrix}$. Sample

$$\mathbf{K}_0 \leftarrow \mathbf{B}^{-1}\left(\begin{pmatrix}(\mathbf{A}_{\mathsf{att}} - \mathbf{x}\otimes\mathbf{G}_{n,q})\otimes\mathbf{V}\\ \mathbf{M}\cdot\mathbf{V}\end{pmatrix}, \tau_B\right).$$

Output $\mathsf{sk}_{\mathbf{x},\emptyset} := (\mathsf{ask}_{\mathbf{x}}, \mathsf{psk}_\emptyset)$, where $\mathsf{ask}_{\mathbf{x}} := (\mathbf{V}, \mathbf{K}_0)$ and $\mathsf{psk}_\emptyset := \mathbf{K}_{1,\emptyset}$.

– $\mathsf{BatchPun}(\mathsf{mpk}, \mathsf{sk}_{\mathbf{x},\eta-1}, \{t'_{\eta,1}, \ldots, t'_{\eta,k_\eta}\})$. Parse $\mathsf{sk}_{\mathbf{x},\eta-1} := (\mathsf{ask}_{\mathbf{x}}, \mathsf{psk}_{\eta-1})$. Compute $\mathbf{A}'_{\eta,j} = \mathsf{EvalF}(\mathbf{A}_{\mathsf{tag}}, g_{t'_{\eta,j}})$ for all $j \in [k_\eta]$. Compute

$$\hat{\mathbf{A}}_\eta = \lfloor(\sum_{j\in[k_\eta]}\mathbf{A}'_{\eta,j})\cdot\mathbf{U}\rceil_p \in \mathbb{Z}_p^{n\times m_1},\ \hat{\mathbf{B}}_\eta = \mathbf{B}_{\mathsf{tag}}\cdot\mathbf{G}^{-1}_{n,p}(\hat{\mathbf{A}}_\eta) \in \mathbb{Z}_p^{n\times m_1}.$$

Let $\mathbf{F}_\eta := [\mathbf{F}_{\eta-1}|\hat{\mathbf{B}}_\eta]$, we will set $\mathbf{F}_0$ as before: $\mathbf{F}_0 = [\mathbf{A}|\mathbf{D}\mathbf{r}^\top]$. Sample

$$\mathbf{K}_{1,\eta} \leftarrow \mathsf{TrapRand}(\mathbf{F}_\eta, \mathsf{TrapExt}(\mathbf{F}_{\eta-1}, \mathbf{K}_{1,\eta-1}, \hat{\mathbf{B}}_\eta), \tau_{\eta-1}).$$

Output $\mathsf{sk}_{\mathbf{x},\eta} := (\mathsf{ask}_{\mathbf{x}}, \mathsf{psk}_\eta)$, where $\mathsf{psk}_\eta := \mathbf{K}_{1,\eta}$.

– $\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_{\mathbf{x},\eta}, \mathsf{ct})$. If $f(\mathbf{x}) = 0$ and $P_\eta \cap \{t_1, \ldots, t_\mathcal{K}\} = \emptyset$, parse $\mathsf{sk}_{\mathbf{x},\eta}$ as $(\mathbf{V}, \mathbf{K}_0, \mathbf{K}_{1,\eta})$ and ct as $(\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c})$. Compute

$$\begin{aligned}
\mathbf{A}_f &= \mathsf{EvalF}(\mathbf{A}_{\mathsf{att}}, f),\ \mathbf{H}_{\mathbf{A}_{\mathsf{att}},f,\mathbf{x}} = \mathsf{EvalFX}(\mathbf{A}_{\mathsf{att}}, f, \mathbf{x}),\\
(\mathbf{d}_0, \mathbf{d}_1) &= \mathbf{c}_0\cdot\mathbf{K}_0,\\
\mathbf{d}_2 &= \mathbf{d}_0\cdot\left(\mathbf{H}_{\mathbf{A}_{\mathsf{att}},f,\mathbf{x}}\mathbf{u}^\top\otimes\mathbf{I}_{m_0+1}\right),\\
\mathbf{d}_3 &= \lfloor\mathbf{c}_1\cdot\mathbf{G}^{-1}_{2nm_0,q}(\mathbf{A}_f\mathbf{u}^\top\otimes\mathbf{I}_{2m_0})\cdot\mathbf{V} - \mathbf{d}_2\rceil_p,\\
\mathbf{d}_4 &= \mathbf{c}\cdot\mathbf{V} - \mathbf{d}_3 - \lfloor\mathbf{d}_1\rceil_p.
\end{aligned}$$

- If $\eta = \emptyset$, sample $\mathbf{d}_5^\top \leftarrow \mathbf{F}_0^{-1}(\mathbf{0}^\top, \tau_0)$. Output $\mathsf{Round}_\beta(\mathbf{d}_4\cdot\mathbf{d}_5^\top)$.
- If $\eta \neq \emptyset$, sample $\mathbf{d}_5^\top \leftarrow \mathbf{F}_\eta^{-1}(\mathbf{0}^\top, \tau_\eta)$. Compute

$$\begin{aligned}
\mathbf{H}_{\mathbf{A}_{\mathsf{tag}},g_{t'_{i,j}},\mathbf{t}} &= \mathsf{EvalFX}(\mathbf{A}_{\mathsf{tag}}, g_{t'_{i,j}}, \mathbf{t}) && \text{for } i\in[\eta], j\in[k_i],\\
\mathbf{d}_{6,i} &= \mathbf{c}_3\cdot\sum_{j\in[k_\eta]}\mathbf{H}_{\mathbf{A}_{\mathsf{tag}},g_{t'_{\eta,j}},\mathbf{t}}\mathbf{U} && \text{for } i\in[\eta].
\end{aligned}$$

Output $\mathsf{Round}_\beta\left(\left[\mathbf{d}_4\middle|\left(\mathbf{c}_2\cdot\mathbf{G}^{-1}_{n,p}([\hat{\mathbf{A}}_1|\ldots|\hat{\mathbf{A}}_\eta]) - \lfloor\mathbf{d}_{6,1}|\ldots|\mathbf{d}_{6,\eta}\rceil_p\right)\right]\cdot\mathbf{d}_5^\top\right)$.

We refer to Appendix D for the concrete parameters, correctness, and error bound of our construction.

5.4 Security Proof

We prove the security of our construction via the following theorem, with the full proof deferred to Appendix E.

Theorem 1. *Let* $\Pi_{\mathsf{BPCP\text{-}ABE}}$ *be the* BPCP-ABE *construction described above, with parameters set as in (3). Then,* $\Pi_{\mathsf{BPCP\text{-}ABE}}$ *is very selective secure (Definition 4) assuming* LWE, sLWE, NLWE, *evasive and tensor* LWE.

6 Conclusion

In this work, we present a lattice-based construction of BPCP-ABE for circuits, where the size of the punctured key is independent of the number of punctured tags, as well as the depth and size of the circuits. Our construction is proven to be very selectively secure under the evasive and tensor LWE assumptions. Moreover, we propose two natural generalizations of BPCP-ABE, named PBE and DP-ABE for circuits. Looking ahead, we believe that our batch-puncturing technique developed in this work can also be extended to lattice-based PE and puncturable KP-ABE for circuits.

Acknowledgements We would like to thank the anonymous reviewers for their invaluable comments. This work was supported in part by National Cryptologic Science Fund of China under Grant (2025NCSF01011), in part by the National Natural Science Foundation of China under Grant (62432012, 62572483), and in part by the Key Research and Development Program of Shaanxi under Grant (2024GX-ZDCYL-01-08).

References

1. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010)
2. Agrawal, S., Kumari, S., Yamada, S.: Attribute based encryption for turing machines from lattices. In: Reyzin, L., Stebila, D. (eds.) CRYPTO 2024, Part III. LNCS, vol. 14922, pp. 352–386. Springer, Cham (2024)
3. Agrawal, S., Rossi, M., Yadav, A., Yamada, S.: Constant input attribute based (and predicate) encryption from evasive and tensor LWE. In: Handschuh, H., Lysyanskaya, A. (eds.) CRYPTO 2023, Part IV. LNCS, vol. 14084, pp. 532–564. Springer, Cham (2023)
4. Agrawal, S., Wichs, D., Yamada, S.: Optimal broadcast encryption from LWE and pairings in the standard model. In: Pass, R., Pietrzak, K. (eds.) TCC 2020, Part I. LNCS, vol. 12550, pp. 149–178. Springer (2020)
5. Akinyele, J.A., Pagano, M.W., Green, M.D., Lehmann, C.U., Peterson, Z.N.J., Rubin, A.D.: Securing electronic medical records using attribute-based encryption on mobile devices. In: Jiang, X., Bhattacharya, A., Dasgupta, P., Enck, W. (eds.) SPSM'11. pp. 75–86. ACM (2011)

6. Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 595–618. Springer, Heidelberg (2009)
7. Attrapadung, N., Imai, H.: Dual-policy attribute based encryption. In: Abdalla, M., Pointcheval, D., Fouque, P., Vergnaud, D. (eds.) ACNS 2009. LNCS, vol. 5536, pp. 168–185. Springer, Heidelberg (2009)
8. Attrapadung, N., Yamada, S.: Duality in ABE: converting attribute based encryption for dual predicate and dual policy via computational encodings. In: Nyberg, K. (ed.) CT-RSA 2015. LNCS, vol. 9048, pp. 87–105. Springer, Cham (2015)
9. Baden, R., Bender, A., Spring, N., Bhattacharjee, B., Starin, D.: Persona: an online social network with user-defined privacy. In: Rodriguez, P., Biersack, E.W., Papagiannaki, K., Rizzo, L. (eds.) ACM SIGCOMM 2009. pp. 135–146. ACM (2009)
10. Barker, E.: Recommendation for key management. NIST Special Publication 800-57 Part 1, Revision 4 (2016)
11. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE S&P 2007. pp. 321–334. IEEE Computer Society (2007)
12. Boneh, D., Gentry, C., Gorbunov, S., Halevi, S., Nikolaenko, V., Segev, G., Vaikuntanathan, V., Vinayagamurthy, D.: Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 533–556. Springer, Heidelberg (2014)
13. Boneh, D., Lewi, K., Montgomery, H.W., Raghunathan, A.: Key homomorphic prfs and their applications. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 410–428. Springer, Heidelberg (2013)
14. Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 255–271. Springer, Heidelberg (2003)
15. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 523–552. Springer, Heidelberg (2010)
16. Chen, J., Gay, R., Wee, H.: Improved dual system ABE in prime-order groups via predicate encodings. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 595–624. Springer, Heidelberg (2015)
17. Cini, V., Wee, H.: ABE for circuits with poly (λ) -sized keys from LWE. In: FOCS. pp. 435–446. IEEE (2023)
18. Cini, V., Wee, H.: Unbounded ABE for circuits from lwe, revisited. In: Chung, K., Sasaki, Y. (eds.) ASIACRYPT 2024, Part IV. LNCS, vol. 15487, pp. 238–267. Springer (2024)
19. Dutta, P., Susilo, W., Duong, D.H., Roy, P.S.: Puncturable identity-based and attribute-based encryption from lattices. Theor. Comput. Sci. **929**, 18–38 (2022)
20. Fiat, A., Naor, M.: Broadcast encryption. In: Stinson, D.R. (ed.) CRYPTO 93. LNCS, vol. 773, pp. 480–491. Springer, Heidelberg (1993)
21. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Dwork, C. (ed.) STOC. pp. 197–206. ACM (2008)
22. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 75–92. Springer, Heidelberg (2013)
23. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Attribute-based encryption for circuits. In: Boneh, D., Roughgarden, T., Feigenbaum, J. (eds.) STOC. pp. 545–554. ACM (2013)

24. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Juels, A., Wright, R.N., di Vimercati, S.D.C. (eds.) CCS 2006. pp. 89–98. ACM (2006)
25. Green, M.D., Miers, I.: Forward secure asynchronous messaging from puncturable encryption. In: IEEE S&P 2015. pp. 305–320. IEEE (2015)
26. Hsieh, Y., Lin, H., Luo, J.: Attribute-based encryption for circuits of unbounded depth from lattices. In: FOCS 2023. pp. 415–434. IEEE (2023)
27. Hsieh, Y., Lin, H., Luo, J.: A general framework for lattice-based ABE using evasive inner-product functional encryption. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024, Part II. LNCS, Cham, vol. 14652, pp. 433–464. Springer (2024)
28. Kowalczyk, L., Wee, H.: Compact adaptively secure ABE for nc^1 from k-lin. J. Cryptol. **33**(3), 954–1002 (2020)
29. Li, H., Lin, H., Luo, J.: ABE for circuits with constant-size secret keys and adaptive security. In: Kiltz, E., Vaikuntanathan, V. (eds.) TCC 2022, Part I. LNCS, vol. 13747, pp. 680–710. Springer, Cham (2022)
30. Mei, Q., Yang, M., Chen, J., Wang, L., Xiong, H.: Expressive data sharing and self-controlled fine-grained data deletion in cloud-assisted iot. IEEE Trans. Dependable Secur. Comput. **20**(3), 2625–2640 (2023)
31. Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012)
32. Phuong, T.V.X., Ning, R., Xin, C., Wu, H.: Puncturable attribute-based encryption for secure data delivery in internet of things. In: INFOCOM 2018. pp. 1511–1519. IEEE (2018)
33. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) STOC. pp. 84–93. ACM (2005)
34. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
35. Santos, N., Rodrigues, R., Gummadi, K.P., Saroiu, S.: Policy-sealed data: A new abstraction for building trusted cloud services. In: Kohno, T. (ed.) USENIX Security Symposium 2012. pp. 175–188. USENIX Association (2012)
36. Sun, S., Sakzad, A., Steinfeld, R., Liu, J.K., Gu, D.: Public-key puncturable encryption: Modular and compact constructions. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) PKC 2020, Part I. LNCS, vol. 12110, pp. 309–338. Springer, Cham (2020)
37. Sun, S., Steinfeld, R., Sakzad, A.: Incremental symmetric puncturable encryption with support for unbounded number of punctures. Des. Codes Cryptogr. **91**(4), 1401–1426 (2023)
38. Susilo, W., Duong, D.H., Le, H.Q., Pieprzyk, J.: Puncturable encryption: A generic construction from delegatable fully key-homomorphic encryption. In: Chen, L., Li, N., Liang, K., Schneider, S.A. (eds.) ESORICS 2020, Part II. LNCS, vol. 12309, pp. 107–127. Springer, Cham (2020)
39. Tsabary, R.: Candidate witness encryption from lattice techniques. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022, Part I. LNCS, vol. 13507, pp. 535–559. Springer,Cham (2022)
40. Waters, B., Wee, H., Wu, D.J.: Multi-authority ABE from lattices without random oracles. In: Kiltz, E., Vaikuntanathan, V. (eds.) TCC 2022, Part I. LNCS, vol. 13747, pp. 651–679. Springer, Cham (2022)
41. Wee, H.: Optimal broadcast encryption and CP-ABE from evasive lattice assumptions. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part II. LNCS, vol. 13276, pp. 217–241. Springer, Cham (2022)

42. Wee, H.: Circuit ABE with poly(depth,λ)-sized ciphertexts and keys from lattices. In: Reyzin, L., Stebila, D. (eds.) CRYPTO 2024, Part III. LNCS, vol. 14922, pp. 178–209. Springer, Cham (2024)
43. Wee, H.: Almost optimal KP and CP-ABE for circuits from succinct LWE. In: Fehr, S., Fouque, P. (eds.) EUROCRYPT 2025, Part III. LNCS, vol. 15603, pp. 34–62. Springer (2025)
44. Yang, M., Wang, H., He, D.: Puncturable attribute-based encryption from lattices for classified document sharing. IEEE Trans. Inf. Forensics Secur. **19**, 4028–4042 (2024)

Supplementary Material

A Lattices Background

Lattice. Let $\mathbf{b}_1, \ldots, \mathbf{b}_n$ be a set of linearly independent m-dimensional row vectors, a lattice $\mathcal{L}$ is the set of all integral combinations of $\mathbf{b}_1, \ldots, \mathbf{b}_n$, namely $\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) = \{\sum_{i\in[n]} x_i\mathbf{b}_i : x_i \in \mathbb{Z}\}$. The matrix $\mathbf{B} \in \mathbb{Z}^{n\times m}$ is called a basis of lattice $\mathcal{L}$ if $\mathbf{B}$ consists of vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$. For $\mathbf{B} \in \mathbb{Z}^{n\times m}$ and $\mathbf{u} \in \mathbb{Z}_q^m$, we consider the following lattices:

$$\Lambda_q^{\perp}(\mathbf{B}) := \{\mathbf{e} \in \mathbb{Z}^n | \mathbf{e}\cdot\mathbf{B} = \mathbf{0} \bmod q\} \text{ and } \Lambda_q^{\mathbf{u}}(\mathbf{B}) := \{\mathbf{e} \in \mathbb{Z}^n | \mathbf{e}\cdot\mathbf{B} = \mathbf{u} \bmod q\}.$$

Discrete Gaussians. We recall the discrete Gaussian distributions over $\mathbb{Z}$.

Definition 5 (Discrete Gaussians Distributions [21]). *The discrete Gaussian function over $\mathbb{R}$ with parameter $\chi > 0$ is given by $\rho_\chi(z) := \exp\left(-\pi\frac{|z|^2}{\chi^2}\right)$ with support $\mathbb{R}$. The discrete Gaussian distribution over $\mathbb{Z}$ with parameter χ is given by $\mathcal{D}_{\mathbb{Z},\chi}(z) := \frac{\rho_\chi(z)}{\sum_{z'\in\mathbb{Z}}\rho_\chi(z')}$ with support $\mathbb{Z}$.*

Learning with Errors. We rely on the decision variant of the LWE problem [33]. Unless otherwise noted, LWE refers to this decision variant.

Assumption 2 (LWE). *For any probabilistic polynomial-time (PPT) adversary $\mathcal{A}$, given integers $n = n(\lambda), m = m(\lambda), q = q(\lambda) > 2, \chi = \chi(\lambda)$, the $\mathsf{LWE}_{n,m,q,\chi}$ assumption states that*

$$\left|\Pr\left[\mathcal{A}(\mathbf{A},\mathbf{c}) = 1 \,\middle|\, \begin{array}{l}\mathbf{A}\leftarrow\mathbb{Z}_q^{n\times m},\\ \mathbf{s}\leftarrow\mathbb{Z}_q^n,\\ \mathbf{e}\leftarrow\mathcal{D}_{\mathbb{Z},\chi}^m,\\ \mathbf{c} = \mathbf{s}\mathbf{A}+\mathbf{e}\end{array}\right] - \Pr\left[\mathcal{A}(\mathbf{A},\mathbf{c}) = 1 \,\middle|\, \begin{array}{l}\mathbf{A}\leftarrow\mathbb{Z}_q^{n\times m},\\ \mathbf{c}\leftarrow\mathbb{Z}_q^m\end{array}\right]\right| \le \mathsf{negl}(\lambda).$$

The Normal Form of LWE. We use the normal form of the LWE problem, where the secret $\mathbf{s}$ is sampled from the error distribution $\mathcal{D}_{\mathbb{Z},\chi}^n$. Applebaum et al. [6] showed that the small-secret version, in its decision variant, is at least as hard as LWE, with only minor differences in the number of samples m. We refer to such problem as sLWE.

Assumption 3 (sLWE). *Given integers $n = n(\lambda), m = m(\lambda), q = q(\lambda) > 2, \chi = \chi(\lambda)$, for any PPT adversary $\mathcal{A}$, the $\mathsf{sLWE}_{n,m,q,\chi}$ assumption states that*

$$\left|\Pr\left[\mathcal{A}(\mathbf{A},\mathbf{c}) = 1 \,\middle|\, \begin{array}{l}\mathbf{A}\leftarrow\mathbb{Z}_q^{n\times m},\\ \mathbf{s}\leftarrow\mathcal{D}_{\mathbb{Z},\chi}^n,\\ \mathbf{e}\leftarrow\mathcal{D}_{\mathbb{Z},\chi}^m,\\ \mathbf{c} = \mathbf{s}\mathbf{A}+\mathbf{e}\end{array}\right] - \Pr\left[\mathcal{A}(\mathbf{A},\mathbf{c}) = 1 \,\middle|\, \begin{array}{l}\mathbf{A}\leftarrow\mathbb{Z}_q^{n\times m},\\ \mathbf{c}\leftarrow\mathbb{Z}_q^m\end{array}\right]\right| \le \mathsf{negl}(\lambda).$$

Non-uniform LWE. We review a variant of LWE, called non-uniform learning with errors (NLWE) [13], where the public matrix $\mathbf{A}$ is sampled from low-norm Gaussian distribution.

Assumption 4 (NLWE). *For any PPT adversary $\mathcal{A}$, given integers $n = n(\lambda), m = m(\lambda), q = q(\lambda) > 2, \chi = \chi(\lambda)$, the $\mathsf{NLWE}_{n,m,q,\gamma}$ assumption states that*

$$\left| \Pr\left[\mathcal{A}(\mathbf{A}, \mathbf{c}) = 1 \;\middle|\; \begin{array}{l} \mathbf{A} \leftarrow \mathcal{D}_{\mathbb{Z},\gamma}^{n \times m}, \\ \mathbf{s} \leftarrow \mathbb{Z}_q^n, \\ \mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^{m}, \\ \mathbf{c} = \mathbf{s}\mathbf{A} + \mathbf{e} \end{array} \right] - \Pr\left[\mathcal{A}(\mathbf{A}, \mathbf{c}) = 1 \;\middle|\; \begin{array}{l} \mathbf{A} \leftarrow \mathcal{D}_{\mathbb{Z},\gamma}^{n \times m}, \\ \mathbf{c} \leftarrow \mathbb{Z}_q^m \end{array} \right] \right| \leq \mathsf{negl}(\lambda)$$

where $\gamma = \Omega(\sqrt{n \log q})$.

The Gadget Matrix [31]. Let $n, q \in \mathbb{Z}^+$ and $m = n \cdot \lceil \log q \rceil$. The gadget vector and the gadget matrix are defined as

$$\mathbf{g}_q = (2^0, 2^1, \ldots, 2^{\lceil \log q \rceil - 1}) \in \mathbb{Z}_q^{\lceil \log q \rceil} \quad \text{and} \quad \mathbf{G}_{n,q} = \mathbf{I}_n \otimes \mathbf{g}_q \in \mathbb{Z}_q^{n \times m}.$$

Given $\mathbf{p} \in \mathbb{Z}_q^n$, $\mathbf{G}_{n,q}^{-1}(\mathbf{p})$ is defined as the m-bit vector $(\mathsf{bin}_q(\mathbf{p}[1]), ..., \mathsf{bin}_q(\mathbf{p}[n]))$, where $\mathsf{bin}_q(a)$ represents converting a value $a \in \mathbb{Z}_q$ into its binary representation. It holds that $\mathbf{G}_{n,q} \cdot \mathbf{G}_{n,q}^{-1}(\mathbf{p}) = \mathbf{p} \mod q$. This notation applies column-wise when extended to matrices, ensuring that $\mathbf{G}_{n,q} \cdot \mathbf{G}_{n,q}^{-1}(\mathbf{P}) = \mathbf{P} \mod q$ for any $\mathbf{P} \in \mathbb{Z}_q^{n \times k}$ and $k \in \mathbb{N}$.

Lemma 4 (Noise Smudging [40]). *Let λ be a security parameter. Choose any $\varepsilon \in \mathbb{Z}$ such that $\|\varepsilon\| \leq B$. Assume that $\chi \geq B \cdot \lambda^{\omega(1)}$. Then the statistical distance between the distributions $\{e : e \leftarrow \mathcal{D}_{\mathbb{Z},\chi}\}$ and $\{e + \varepsilon : e \leftarrow \mathcal{D}_{\mathbb{Z},\chi}\}$ is $\mathsf{negl}(\lambda)$.*

Lattice Trapdoors. Let $n, m, q \in \mathbb{N}$ and consider a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$. Given any $\mathbf{P} \in \mathbb{Z}_q^{n \times k}$, we use $\mathbf{A}^{-1}(\mathbf{P}, \tau)$ to denote the random variable whose distribution is the discrete Gaussian $\mathcal{D}_{\mathbb{Z},\tau}^{m \times k}$ conditioned on $\mathbf{A} \cdot \mathbf{A}^{-1}(\mathbf{P}, \tau) = \mathbf{P} \mod q$. If $\mathbf{K} \leftarrow \mathbf{A}^{-1}(\mathbf{P}, \tau)$, then $\|\mathbf{K}\| \leq c\tau\sqrt{mk}$ with probability at least $1 - e^{-\Omega(c^2)}$. For any $\mathbf{P}$, a τ-trapdoor for $\mathbf{A}$ is a procedure that can sample from the distribution $\mathbf{A}^{-1}(\mathbf{P}, \tau)$ in time $\mathsf{poly}(n, m, k, \log q)$. We denote a τ-trapdoor for $\mathbf{A}$ as $\mathbf{A}_\tau^{-1}$ for simplicity. In a long sequence of works [1,15,21,31], the following properties have been established.

Lemma 5 (Properties of Trapdoors). *There exists the following PPT algorithms:*

- $\mathsf{TrapGen}(1^n, 1^m, q)$ *that outputs* $(\mathbf{A}, \mathbf{A}_\tau^{-1})$ *where* $\tau = \omega(\sqrt{n \log q \log m})$ *and* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *is* 2^{-n}*-close to uniform for some* $m = O(n \log q)$.
- $\mathsf{TrapExt}(\mathbf{A}, \mathbf{A}_\tau^{-1}, \mathbf{B})$ *that for any* $\mathbf{B} \in \mathbb{Z}_q^{n \times k}$, *outputs a* τ*-trapdoor* $[\mathbf{A}|\mathbf{B}]_\tau^{-1}$ *for* $[\mathbf{A}|\mathbf{B}]_\tau$, *where* $\|\widetilde{\mathbf{A}_\tau^{-1}}\| = \|\widetilde{[\mathbf{A}|\mathbf{B}]_\tau^{-1}}\|$.
- $\mathsf{TrapRand}(\mathbf{A}, \mathbf{A}_\tau^{-1}, \tau)$ *that outputs a new* τ'*-trapdoor* $\mathbf{A}_{\tau'}^{-1}$ *for* $\mathbf{A}$, *where* $\tau' = \|\widetilde{\mathbf{A}_\tau^{-1}}\| \cdot \omega(\sqrt{\log m})$.

Lattice Evaluation. We review two fundamental algorithms for homomorphic computation on matrices [12,22].

Lemma 6 ($\mathsf{EvalF}, \mathsf{EvalFX}$)**.** *Given inputs $\ell, n, q \in \mathbb{N}$, $m = O(n \log q)$ and a function $f : \mathbb{Z}_q^\ell \to \{0,1\}$, where f is computable by circuits of depth d and size s, there exists a pair of efficient algorithms ($\mathsf{EvalF}, \mathsf{EvalFX}$) with the following properties.*

- $\mathsf{EvalF}(\mathbf{A}, f) \to \mathbf{A}_f$: *Taking a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m\ell}$ and a function f as inputs, it outputs $\mathbf{A}_f \in \mathbb{Z}_q^{n \times m}$.*
- $\mathsf{EvalFX}(\mathbf{A}, f, \mathbf{x}) \to \mathbf{H}_{\mathbf{A},f,\mathbf{x}}$: *Taking a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m\ell}$, a function f and $\mathbf{x} \in \mathbb{Z}^\ell$ as inputs, it outputs $\mathbf{H}_{\mathbf{A},f,\mathbf{x}} \in \mathbb{Z}^{m\ell \times m}$ with $\|\mathbf{H}_{\mathbf{A},f,\mathbf{x}}\| = m^{O(d)}s$.*
- *It holds that $(\mathbf{A} - \mathbf{x} \otimes \mathbf{G}_{n,q}) \cdot \mathbf{H}_{\mathbf{A},f,\mathbf{x}} = \mathbf{A}_f - f(x) \cdot \mathbf{G}_{n,q} \mod q$.*

B Proof of Lemma 2

Proof. Let $\hat{\mathbf{P}}$ denote the integer matrix whose entries are equal to those of $\mathbf{P}$. According to the definition of the round function $\lfloor \cdot \rceil_p$, we have

$$\lfloor \mathbf{v} \cdot \mathbf{P} \cdot \mathbf{Q} \mod q \rceil_p = \left\lfloor \frac{p}{q} \cdot \mathbf{v} \cdot \hat{\mathbf{P}} \cdot \mathbf{Q} \right\rceil = \frac{p}{q} \cdot \mathbf{v} \cdot \hat{\mathbf{P}} \cdot \mathbf{Q} + \mathbf{w} \mod p,$$

where $\mathbf{w} \in \mathbb{Q}^t$ and $\|\mathbf{w}\| \leq 1/2$. On the other hand

$$\begin{aligned} \lfloor \mathbf{v} \cdot \mathbf{P} \mod q \rceil_p \cdot \mathbf{Q} &= \left\lfloor \frac{p}{q} \cdot \mathbf{v} \cdot \hat{\mathbf{P}} \right\rceil \cdot \mathbf{Q} \\ &= \left(\frac{p}{q} \cdot \mathbf{v} \cdot \hat{\mathbf{P}} + \mathbf{f} \right) \cdot \mathbf{Q} \\ &= \frac{p}{q} \cdot \mathbf{v} \cdot \hat{\mathbf{P}} \cdot \mathbf{Q} + \mathbf{f} \cdot \mathbf{Q} \mod p, \end{aligned}$$

where $\mathbf{f} \in \mathbb{Q}^k$ and $\|\mathbf{f}\| \leq 1/2$. Observe that

$$\begin{aligned} & \lfloor \mathbf{v} \cdot \mathbf{P} \cdot \mathbf{Q} \mod q \rceil_p \\ =\ & \frac{p}{q} \cdot \mathbf{v} \cdot \hat{\mathbf{P}} \cdot \mathbf{Q} + \mathbf{w} \\ =\ & \frac{p}{q} \cdot \mathbf{v} \cdot \hat{\mathbf{P}} \cdot \mathbf{Q} + \mathbf{w} + \lfloor \mathbf{v} \cdot \mathbf{P} \mod q \rceil_p \cdot \mathbf{Q} - \left(\frac{p}{q} \cdot \mathbf{v} \cdot \hat{\mathbf{P}} \cdot \mathbf{Q} + \mathbf{f} \cdot \mathbf{Q} \right) \\ =\ & \lfloor \mathbf{v} \cdot \mathbf{P} \mod q \rceil_p \cdot \mathbf{Q} + \mathbf{w} - \mathbf{f} \cdot \mathbf{Q} \\ =\ & (\mathbf{v} \cdot \lfloor \mathbf{P} \rceil_p + \mathbf{e}') \cdot \mathbf{Q} + \mathbf{w} - \mathbf{f} \cdot \mathbf{Q} \\ =\ & \mathbf{v} \cdot \lfloor \mathbf{P} \rceil_p \cdot \mathbf{Q} + \underbrace{\mathbf{e}' \cdot \mathbf{Q} + \mathbf{w} - \mathbf{f} \cdot \mathbf{Q}}_{:=\mathbf{e}} \mod p, \end{aligned}$$

where the fourth equality uses $\lfloor \mathbf{v} \cdot \mathbf{P} \mod q \rceil_p = \mathbf{v} \cdot \lfloor \mathbf{P} \rceil_p + \mathbf{e}' \mod p$. By item 2 of Lemma 1, we have $\|\mathbf{e}'\| \leq n \cdot \|\mathbf{v}\|$. Note that $\mathbf{e}$ is a t-dimensional row vector

with a bound of

$$\begin{aligned}\|\mathbf{e}\| &= \|\mathbf{e}' \cdot \mathbf{Q} + \mathbf{w} - \mathbf{f} \cdot \mathbf{Q}\| \\ &\leq \|\mathbf{e}' \cdot \mathbf{Q}\| + \|\mathbf{w}\| + \|\mathbf{f} \cdot \mathbf{Q}\| \\ &\leq nk \cdot \|\mathbf{v}\| \cdot \|\mathbf{Q}\| + \frac{1}{2} + \frac{1}{2} \cdot k \cdot \|\mathbf{Q}\| \\ &\leq (n+1)k \cdot \|\mathbf{v}\| \cdot \|\mathbf{Q}\|,\end{aligned}$$

where the last inequality holds under the condition $\|\mathbf{v}\| \geq 1$, $\|\mathbf{Q}\| \geq 1$. Suppose $\|\mathbf{v}\| < 1$ or $\|\mathbf{Q}\| < 1$. This implies that $\mathbf{v}$ and $\mathbf{Q}$ are vectors and matrices with each entry equal to zero, respectively, which makes the claim trivial.

C Proof of Lemma 3

Proof. Let us recall a vector decomposition for tensor product [3], it can be observed that any $\mathbf{e}_i \leftarrow \mathcal{D}_\chi^{mt\ell}$ can be written as $\mathbf{e}_i = \sum_{j\in[t]} \mathbf{e}_{i,j} \otimes \boldsymbol{\xi}_j$ using $\mathbf{e}'_{i,j} \leftarrow \mathcal{D}_\chi^{m\ell}$ where $\boldsymbol{\xi}_j$ is the j-th canonical unit vector of dimension t. We extend a vector decomposition to matrix decomposition for tensor products. For any $\mathbf{R}_i \leftarrow \mathcal{D}_\delta^{k\times t}$, it can be written as

$$\mathbf{R}_i = \sum_{j\in[t]} \hat{\mathbf{r}}_{i,j}^\top \otimes \boldsymbol{\xi}_j$$

using $\hat{\mathbf{r}}_{i,j} \in \mathcal{D}_{\mathbb{Z},\delta}^k$ where $\boldsymbol{\xi}_i$ is the j-th canonical unit vector of dimension t and $\mathbf{R}_i = [\hat{\mathbf{r}}_{i,1}^\top | \ldots | \hat{\mathbf{r}}_{i,t}^\top]$. By the mixed-product property for tensor product, we have

$$\mathbf{s}\left(\mathbf{I}_n \otimes \mathbf{R}_i\right)\left(\left(\mathbf{A} - \mathbf{x}_i \otimes \mathbf{G}_{n,q}\right) \otimes \mathbf{I}_t\right) + \mathbf{e}_i = \mathbf{s}\left(\left(\mathbf{A} - \mathbf{x}_i \otimes \mathbf{G}_{n,q}\right) \otimes \mathbf{R}_i\right) + \mathbf{e}_i.$$

Next, we rewrite $\mathbf{s}\left(\left(\mathbf{A} - \mathbf{x}_i \otimes \mathbf{G}_{n,q}\right) \otimes \mathbf{R}_i\right) + \mathbf{e}_i$ using the tensor decomposition of $\mathbf{R}_i$ and $\mathbf{e}_i$, namely

$$\begin{aligned}&\mathbf{s}\left(\left(\mathbf{A} - \mathbf{x}_i \otimes \mathbf{G}_{n,q}\right) \otimes \mathbf{R}_i\right) + \mathbf{e}_i \\ &= \mathbf{s}\left(\left(\mathbf{A} - \mathbf{x}_i \otimes \mathbf{G}_{n,q}\right) \otimes \left(\sum_{j\in[t]} \hat{\mathbf{r}}_{i,j}^\top \otimes \boldsymbol{\xi}_j\right)\right) + \sum_{j\in[t]} \mathbf{e}_{i,j} \otimes \boldsymbol{\xi}_j \\ &= \mathbf{s}\left(\sum_{j\in[t]} \left(\mathbf{A} - \mathbf{x}_i \otimes \mathbf{G}_{n,q}\right) \otimes \hat{\mathbf{r}}_{i,j}^\top \otimes \boldsymbol{\xi}_j\right) + \sum_{j\in[t]} \mathbf{e}_{i,j} \otimes \boldsymbol{\xi}_j \\ &= \sum_{j\in[t]} \left(\mathbf{s}\left(\left(\mathbf{A} - \mathbf{x}_i \otimes \mathbf{G}_{n,q}\right) \otimes \hat{\mathbf{r}}_{i,j}^\top\right) + \mathbf{e}_{i,j}\right) \otimes \boldsymbol{\xi}_j \\ &= \sum_{j\in[t]} \left(\mathbf{s}\left(\mathbf{I}_n \otimes \hat{\mathbf{r}}_{i,j}^\top\right)\left(\mathbf{A} - \mathbf{x}_i \otimes \mathbf{G}_{n,q}\right) + \mathbf{e}_{i,j}\right) \otimes \boldsymbol{\xi}_j \\ &\approx_c \sum_{j\in[t]} \mathbf{v}'_{i,j} \otimes \boldsymbol{\xi}_j \\ &\equiv \mathbf{v}_i\end{aligned}$$

where $\mathbf{v}'_{i,j} \leftarrow \mathbb{Z}_q^{m\ell}$ and the $\approx_c$ relies on the tensor LWE assumption.

D Parameters, Correctness, and Error Bound

Parameters. Fix $\lambda_1 = \lambda^{\omega(1)}$. Following [17], we set $(p/q) \cdot B \leq \lambda_1$ for the relationship between p and q, where $B \leq m_2^{O(d)} s$. Let $\eta_{\mathsf{max}} = \eta_{\mathsf{max}}(\lambda) \in \mathbb{N}$ be the maximum number of variables that can be delegated. Let be $k_{\mathsf{max}} = k_{\mathsf{max}}(\lambda) < \lambda^{\omega(1)}$ the maximum number of tags that can be punctured per time. The specific parameters are as follows.

$$\begin{gathered} n = \mathsf{poly}(\lambda), m_0 = O(n \log p), m_1 = n \cdot \lceil \log p \rceil, m_2 = n \cdot \lceil \log q \rceil, \\ \kappa = O(\log q), \tau_A = O(\sqrt{n \log p}), \tau_B = O(\sqrt{2nm_0 \log q}), \tau_C = O(\sqrt{n \log p}), \\ \tau_{\mathsf{max}} = \tau_A \cdot (\sqrt{m_1 \log m_1})^{\eta_{\mathsf{max}}}, \chi_0 = \lambda_1, \chi_1 = \chi_0 \tau_C \lambda^{\omega(1)}, \chi_2 = \tau_C, \\ \chi_3 = \chi_1 \tau_C \lambda^{\omega(1)}, \beta = \chi_0 \chi_1 \tau_B \tau_C \tau_{\mathsf{max}} m_2^{O(d)} s \cdot \lambda^{\omega(1)}, p = \beta \cdot \lambda^{\omega(1)}, q = pB/\lambda^{\omega(1)}. \end{gathered} \tag{3}$$

This results in the below parameter sizes for our BPCP-ABE scheme:

$$|\mathsf{mpk}| = O_\lambda((\ell + \mathcal{K}) \cdot d^2),\ |\mathsf{ct}| = O_\lambda(\mathcal{K} \cdot d^2),\ |\mathsf{ask}_{\mathbf{x}}| = O_\lambda(\ell \cdot d^2),\ \left|\mathsf{psk}_\eta\right| = O_\lambda(\eta)$$

where $O_\lambda(\cdot)$ hides factors polynomial in λ. Here, $n = O(\lambda)$, $\log p = O(\lambda)$, $\log q = O(d \cdot \lambda)$ for ease of exposition.

Correctness. The following is a step-by-step process to verify the correctness of the scheme. First, we have

- $(\mathbf{d}_0, \mathbf{d}_1)$ are computed as $\mathbf{d}_0 = \mathbf{s}_0 \left((\mathbf{A}_{\mathsf{att}} - \mathbf{x} \otimes \mathbf{G}_{n,q}) \otimes \mathbf{V} \right) + \mathbf{e}'_0 \mod q$ and $\mathbf{d}_1 = \mathbf{s}_1 \mathbf{M} \mathbf{V} + \mathbf{e}''_0 \mod q$, where $(\mathbf{e}'_0, \mathbf{e}''_0) := \mathbf{e}_0 \mathbf{K}_0$.
- $\mathbf{d}_2$ is computed as $\mathbf{d}_2 = \mathbf{s}_0 (\mathbf{A}_f \mathbf{u}^\top \otimes \mathbf{I}_{2m_0}) \mathbf{V} + \boldsymbol{\varepsilon}_0 \mod q$, where $\boldsymbol{\varepsilon}_0 := \mathbf{e}'_0 \left(\mathbf{H}_{\mathbf{A}_{\mathsf{att}}, f, \mathbf{x}} \mathbf{u}^\top \otimes \mathbf{I}_{m_0+1} \right)$. Here, we use the key equation

$$(\mathbf{A}_{\mathsf{att}} - \mathbf{x} \otimes \mathbf{G}_{n,q}) \cdot \mathbf{H}_{\mathbf{A}_{\mathrm{att}}, f, \mathbf{x}} = \mathbf{A}_f - f(\mathbf{x}) \cdot \mathbf{G}_{n,q} \text{ with } f(\mathbf{x}) = 0.$$

- $\mathbf{d}_3$ is computed as

$$\begin{aligned} \mathbf{d}_3 &= \left\lfloor \mathbf{s}_0 (\mathbf{A}_f \mathbf{u}^\top \otimes \mathbf{I}_{2m_0}) \mathbf{V} + \mathbf{s}_2 \mathbf{B}_f \mathbf{V} - \mathbf{s}_0 \left(\mathbf{A}_f \mathbf{u}^\top \otimes \mathbf{I}_{2m_0} \right) \mathbf{V} + \boldsymbol{\varepsilon}_1 \right\rceil_p \\ &= \left\lfloor \mathbf{s}_2 \mathbf{B}_f \mathbf{V} \right\rceil_p + \lfloor \boldsymbol{\varepsilon}_1 \rceil_p + \mathbf{e}_{\varepsilon_0} \qquad \text{//from Lemma 1} \\ &= \mathbf{s}_2 \lfloor \mathbf{B}_f \rceil_p \mathbf{V} + \underbrace{\lfloor \boldsymbol{\varepsilon}_1 \rceil_p + \mathbf{e}_{\varepsilon_0} + \mathbf{e}_{\varepsilon_1}}_{\boldsymbol{\varepsilon}_2} \mod p \quad \text{//from Lemma 2} \end{aligned}$$

  where $\boldsymbol{\varepsilon}_1 = \mathbf{e}_1 \mathbf{G}^{-1}_{2mm_0,q} (\mathbf{A}_f \mathbf{u}^\top \otimes \mathbf{I}_{2m_0}) \mathbf{V} - \boldsymbol{\varepsilon}_0$. Here, the errors $\mathbf{e}_{\varepsilon_0}, \mathbf{e}_{\varepsilon_1}$ are introduced according to Lemma 1 and 2, respectively.
- $\mathbf{d}_4$ is computed as $\mathbf{d}_4 = \mathbf{s}\mathbf{F}_0 + \mu \mathbf{g} \mathbf{V} + \boldsymbol{\varepsilon}_3 \mod p$, where $\boldsymbol{\varepsilon}_3 = \mathbf{e}\mathbf{V} - \boldsymbol{\varepsilon}_2 - \mathbf{e}''_0 - \mathbf{e}_{\varepsilon''}$. Here, the error $\mathbf{e}_{\varepsilon''}$ is introduced by Lemma 2.

Next, the correctness will be analyzed from the following two cases:
Case 1: If $\eta = 0$, sample $\mathbf{d}_5^\top \leftarrow \mathbf{F}_0^{-1} \left(\mathbf{0}^\top, \tau_0 \right)$ from $\mathbf{K}_{1,\emptyset}$. We deduce that $\mathbf{d}_4 \cdot \mathbf{d}_5^\top \approx \mu \cdot \mathbf{g} \mathbf{V} \mathbf{d}_5^\top$. We put everything together and the total error term is $\boldsymbol{\varepsilon}_3 \cdot \mathbf{d}_5^\top$.

Case 2: If $\eta \neq 0$, sample $\mathbf{d}_5^\top \leftarrow \mathbf{F}_\eta^{-1}\left(\mathbf{0}^\top, \tau_\eta\right)$ from $\mathbf{K}_{1,\eta}$, where $\mathbf{F}_\eta$ is defined in BatchPun and is public. Then, we have

$$\mathbf{c}_2 \cdot \mathbf{G}_{n,p}^{-1}([\hat{\mathbf{A}}_1|...|\hat{\mathbf{A}}_\eta]) = \mathbf{s}_3[\hat{\mathbf{A}}_1|...|\hat{\mathbf{A}}_\eta] + \mathbf{s}[\hat{\mathbf{B}}_1|...|\hat{\mathbf{B}}_\eta] + \boldsymbol{\varepsilon}_4 \mod p$$

where $\boldsymbol{\varepsilon}_4 = \mathbf{e}_2\mathbf{G}_{n,p}^{-1}([\hat{\mathbf{A}}_1|\cdots|\hat{\mathbf{A}}_\eta])$. For any $i \in [\eta]$, $\mathbf{d}_{6,i}$ is computed as

$$\mathbf{d}_{6,i} = \mathbf{s}_3\Big(\sum_{j\in[k_i]} \mathbf{A}'_{i,j}\mathbf{U}\Big) + \boldsymbol{\varepsilon}_{5,i} \mod p, \text{ where } \boldsymbol{\varepsilon}_{5,i} = \mathbf{e}_3\Big(\sum_{j\in[k_i]} \mathbf{H}_{\mathbf{A}_{\mathsf{tag}},g_{t'_{i,j}},\mathbf{t}}\mathbf{U}\Big).$$

Here, we use $(\mathbf{A}_{\mathsf{tag}} - \mathbf{t}\otimes\mathbf{G}_{n,q})\cdot\mathbf{H}_{\mathbf{A}_{\mathsf{tag}},g_{t'_{i,j}},\mathbf{t}} = \mathbf{A}'_{i,j} - g_{t'_{i,j}}(\mathbf{t})\cdot\mathbf{G}_{n,q}$ as $g_{t'_{i,j}}(\mathbf{t}) = 0$ for $i \in [\eta], j \in [k_\eta]$. Further, we have

$$\begin{aligned}\lfloor \mathbf{d}_{6,1}|...|\mathbf{d}_{6,\eta}\rceil_p &= \Big\lfloor \mathbf{s}_3\Big[\sum_{j\in[k_1]}\mathbf{A}'_{1,j}\mathbf{U}\Big|...\Big|\sum_{j\in[k_\eta]}\mathbf{A}'_{\eta,j}\mathbf{U}\Big]\Big\rceil_p + \lfloor \boldsymbol{\varepsilon}_{5,1}|...|\boldsymbol{\varepsilon}_{5,\eta}\rceil_p + \mathbf{e}_{\varepsilon_2} \\ &= \mathbf{s}_3[\hat{\mathbf{A}}_1|...|\hat{\mathbf{A}}_\eta] + \lfloor \boldsymbol{\varepsilon}_{5,1}|...|\boldsymbol{\varepsilon}_{5,\eta}\rceil_p + \mathbf{e}_{\varepsilon_2} + \mathbf{e}_{\varepsilon_3} \mod p\end{aligned}$$

where the errors $\mathbf{e}_{\varepsilon_2}, \mathbf{e}_{\varepsilon_3}$ are introduced by Lemma 1. We deduce that

$$\big[\mathbf{d}_4\big|(\mathbf{c}_2 \cdot \mathbf{G}_{n,p}^{-1}([\hat{\mathbf{A}}_1|...|\hat{\mathbf{A}}_\eta]) - \lfloor \mathbf{d}_{6,1}|...|\mathbf{d}_{6,\eta}\rceil_p)\big] \cdot \mathbf{d}_5^\top = \hat{e} + [\mu\mathbf{g}\mathbf{V}|\mathbf{0}]\cdot\mathbf{d}_5^\top$$

where $\hat{e} = [\boldsymbol{\varepsilon}_3|(\boldsymbol{\varepsilon}_4 - \lfloor \boldsymbol{\varepsilon}_{5,1}|...|\boldsymbol{\varepsilon}_{5,\eta}\rceil_p - \mathbf{e}_{\varepsilon_2} - \mathbf{e}_{\varepsilon_3})]\cdot\mathbf{d}_5^\top$.

Error Bound. The error term is bounded as follows. By the three properties of the round function (Lemmas 1 and 2), we have

$$\begin{gathered}\|\mathbf{e}_{\varepsilon_0}\| \leq 2, \|\mathbf{e}_{\varepsilon_1}\| \leq 2(n+1)m_0\|\mathbf{s}_2\|\cdot\|\mathbf{V}\|, \|\mathbf{e}_{\varepsilon_2}\| \leq 2, \\ \|\mathbf{e}_{\varepsilon_3}\| \leq n_0\|\mathbf{s}_0\|, \|\mathbf{e}_{\varepsilon''}\| \leq 2(n+1)m_0\|\mathbf{s}_1\|\cdot\|\mathbf{V}\|.\end{gathered}$$

Further, we provide the bound of the error term at each step of the decryption process as follows:

$$\begin{aligned}&\|\mathbf{e}'_0\| \leq \chi_0\tau_B\cdot\mathsf{poly}(m_0),\ \|\mathbf{e}''_0\| \leq \chi_0\tau_B\cdot\mathsf{poly}(m_0) && \text{//w.r.t } \mathbf{d}_1 \\ &\|\boldsymbol{\varepsilon}_0\| \leq \chi_0\chi_1\tau_B m_2^{O(d)} s\cdot\mathsf{poly}(m_0) && \text{//w.r.t } \mathbf{d}_2 \\ &\|\boldsymbol{\varepsilon}_1\| \leq \chi_0\chi_1\tau_B\tau_C m_2^{O(d)} s\cdot\mathsf{poly}(m_0) && \text{//w.r.t } \mathbf{d}_2 \\ &\|\boldsymbol{\varepsilon}_2\| \leq \chi_0\chi_1\tau_B\tau_C m_2^{O(d)} s\cdot\mathsf{poly}(m_0) && \text{//w.r.t } \mathbf{d}_3 \\ &\|\boldsymbol{\varepsilon}_3\| \leq \chi_0\chi_1\tau_B\tau_C m_2^{O(d)} s\cdot\mathsf{poly}(m_0) && \text{//w.r.t } \mathbf{d}_4 \\ &\|\boldsymbol{\varepsilon}_4\| \leq \chi_1\cdot\mathsf{poly}(m_0) && \\ &\|\mathbf{d}_5\| \leq \tau_{\max}\cdot\mathsf{poly}(m_0) && \text{//w.r.t } \mathbf{d}_5 \\ &\|\boldsymbol{\varepsilon}_{5,i}\| \leq \chi_0\tau_C m_2^{O(d)} s\cdot\mathsf{poly}(m_0). && \text{//w.r.t } \mathbf{d}_{6,i}\end{aligned}$$

Here, fix $p < q$, $\|\lfloor\mathbf{v}\rceil_p\| < \|\mathbf{v}\|$ holds for any small term $\mathbf{v}$. On the other hand, we obtain a bound for the error term as β, namely

$$\begin{aligned}\|\hat{e}\| &= \left\|[\boldsymbol{\varepsilon}_3|(\boldsymbol{\varepsilon}_4 - \lfloor \boldsymbol{\varepsilon}_{5,1}|...|\boldsymbol{\varepsilon}_{5,\eta}\rceil_p - \mathbf{e}_{\varepsilon_2} - \mathbf{e}_{\varepsilon_3})]\cdot\mathbf{d}_5^\top\right\| \\ &\leq \chi_0\chi_1\tau_B\tau_C\tau_{\max} m_2^{O(d)} s\cdot\mathsf{poly}(m_0) \\ &\leq \beta.\end{aligned}$$

E Complete Proof of Theorem 1

Proof. Let Q be the number of corrupt queries in each case. After making adaptively corrupt queries, the (very selective) ABE adversary $\mathcal{A}$ will receive three types of keys, with others returning a termination symbol.

Case 1. First, we consider $f^*(\mathbf{x}_i) = 1$ and the absence of the tuple $(\mathbf{x}_i, \mathsf{sk}_{\mathbf{x}_i,\eta}, P_{\mathbf{x}_i}, C_{\mathbf{x}_i})$ where $i \in [Q]$. To establish the security of our scheme, it suffices to show the computational indistinguishability of the following distributions, as D_0 captures all information available to the adversary in its view:
Distribution D_0:

$$\begin{pmatrix} \mathsf{mpk},\ \mathbf{c}_0 = [\mathbf{s}_0|\mathbf{s}_1]\mathbf{B} + \mathbf{e}_0 \bmod q, & \mathbf{c}_1 = \mathbf{s}_0\mathbf{G}_{2nm_0,q} + \mathbf{s}_2\mathbf{B}_{\mathsf{att}} + \mathbf{e}_1 \bmod q, \\ \mathbf{c}_2 = \mathbf{s}_3\mathbf{G}_{n,p} + \mathbf{s}\mathbf{B}_{\mathsf{tag}} + \mathbf{e}_2 \bmod p, & \mathbf{c}_3 = \mathbf{s}_3(\mathbf{A}_{\mathsf{tag}} - \mathbf{t}^* \otimes \mathbf{G}_{n,q}) + \mathbf{e}_3 \bmod q, \\ \mathbf{c} = \mathbf{s}[\mathbf{C}|\mathbf{D}] + \mathbf{s}_1\lfloor\mathbf{M}\rceil_p + \mathbf{s}_2\lfloor\mathbf{B}_{f^*}\rceil_p + \mathbf{e} \bmod p, & \{\mathsf{sk}_{\mathbf{x}_i,\emptyset} = (\mathbf{V}_i, \mathbf{K}_{0,i}, \mathbf{K}_{1,\emptyset,i})\}_{i\in[Q]} \end{pmatrix}$$

Distribution D_1:

$$\begin{pmatrix} \mathsf{mpk}, & \mathbf{v}_0 \leftarrow \mathbb{Z}_q^{(2nm_0+n)\kappa}, & \mathbf{v}_1 \leftarrow \mathbb{Z}_q^{2m_0m_2}, & \mathbf{v}_2 \leftarrow \mathbb{Z}_p^{m_1}, \\ \mathbf{v}_3 \leftarrow \mathbb{Z}_q^{m_2\mathcal{K}}, & \mathbf{v} \leftarrow \mathbb{Z}_p^{2m_0}, & \{\mathsf{sk}_{\mathbf{x}_i,\emptyset} = (\mathbf{V}_i, \mathbf{K}_{0,i}, \mathbf{K}_{1,\emptyset,i})\}_{i\in[Q]} \end{pmatrix}$$

We invoke our variant of evasive LWE assumption, and the outputs of its sampler are defined as follows:

$$\begin{aligned} \mathsf{aux} &= \begin{pmatrix} \overbrace{\mathbf{x}_1, \ldots, \mathbf{x}_Q, f^*, \mathbf{t}^*, P_{\mathbf{x}_1}, \ldots, P_{\mathbf{x}_Q}, \mathsf{coins}_{\mathcal{A}}}^{\mathsf{aux}_0}, \{\mathbf{V}_i\}_{i\in[Q]}, \\ \{\mathbf{K}_{1,\emptyset,i}\}_{i\in[Q]}, \mathbf{A}, \mathbf{A}_{\mathsf{att}}, \mathbf{A}_{\mathsf{tag}}, \mathbf{B}_{\mathsf{att}}, \mathbf{B}_{\mathsf{tag}}, \mathbf{C}, \mathbf{D}, \mathbf{M}, \mathbf{U}, \mathbf{u} \end{pmatrix} \\ \mathbf{W}_0 &= \left[(\mathbf{A}_{\mathsf{att}} - \mathbf{x}_1 \otimes \mathbf{G}_{n,q}) \otimes \mathbf{V}_1 | \ldots | (\mathbf{A}_{\mathsf{att}} - \mathbf{x}_Q \otimes \mathbf{G}_{n,q}) \otimes \mathbf{V}_Q\right], \\ \mathbf{W}_1 &= [\mathbf{M}\mathbf{V}_1 | \ldots | \mathbf{M}\mathbf{V}_Q], \\ \mathbf{W} &= \begin{pmatrix} \mathbf{W}_0 & \\ & \mathbf{W}_1 \end{pmatrix}, \widehat{\mathbf{M}} = \lfloor\mathbf{M}\rceil_p \end{aligned}$$

where $\mathsf{coin}_{\mathcal{A}}$ is adversary's coin, and aux_0 is additionally given for the distinguisher. By evasive LWE, it is sufficient to establish that $D_0 \approx_c D_1$ if we show the computational indistinguishability of the following distributions:
Distribution D'_0:

$$\begin{pmatrix} \mathbf{c}_0 = [\mathbf{s}_0|\mathbf{s}_1]\mathbf{B} + \mathbf{e}_0 \bmod q,\ \mathbf{c}_1 = \mathbf{s}_0\mathbf{G}_{2nm_0,q} + \boxed{\mathbf{s}_2\mathbf{B}_{\mathsf{att}} + \mathbf{e}_1} \bmod q, \\ \mathbf{c}_2 = \mathbf{s}_3\mathbf{G}_{n,p} + \boxed{\mathbf{s}\mathbf{B}_{\mathsf{tag}} + \mathbf{e}_2} \bmod p,\ \mathbf{c}_3 = \mathbf{s}_3(\mathbf{A}_{\mathsf{tag}} - \mathbf{t}^* \otimes \mathbf{G}_{n,q}) + \mathbf{e}_3 \bmod q, \\ \mathbf{c} = \mathbf{s}[\mathbf{C}|\mathbf{D}] + \mathbf{s}_1\lfloor\mathbf{M}\rceil_p + \mathbf{s}_2\lfloor\mathbf{B}_{f^*}\rceil_p + \mathbf{e} \bmod p, \\ \left\{\mathbf{c}'_{0,i} = \mathbf{s}_0((\mathbf{A}_{\mathsf{att}} - \mathbf{x}_i \otimes \mathbf{G}_{n,q}) \otimes \mathbf{V}_i) + \mathbf{e}'_{0,i}, \mathbf{c}''_{0,i} = \mathbf{s}_1\mathbf{M}\mathbf{V}_i + \mathbf{e}''_{0,i}\right\}_{i\in[Q]} \end{pmatrix}$$

Distribution D'_1:

$$\begin{pmatrix} \mathbf{v}_0 \leftarrow \mathbb{Z}_q^{(2nm_0+n)\kappa},\ \mathbf{v}_1 \leftarrow \mathbb{Z}_q^{2m_0m_2},\ \mathbf{v}_2 \leftarrow \mathbb{Z}_p^{m_1}, \mathbf{v}_3 \leftarrow \mathbb{Z}_q^{m_2\mathcal{K}}, \\ \mathbf{v} \leftarrow \mathbb{Z}_p^{2m_0}, \{\mathbf{v}'_{0,i} \leftarrow \mathbb{Z}_q^{(m_0+1)m_2\ell},\ \mathbf{v}''_{0,i} \leftarrow \mathbb{Z}_q^{m_0+1}, \}_{i\in[Q]} \end{pmatrix}$$

where $\mathbf{e}'_{0,i} \leftarrow \mathcal{D}_{\mathbb{Z},\chi_2}^{(m_0+1)m_2\ell}$ and $\mathbf{e}''_{0,i} \leftarrow \mathcal{D}_{\mathbb{Z},\chi_3}^{m_0+1}$. Below we prove pseudorandomness of D'_0 via a hybrid sequence.

- H_0: This is the same as D'_0, which is the pre-condition of the evasive LWE assumption.
- H_1: This is the same as H_0, except we sample $(\mathbf{v}_0 \leftarrow \mathbb{Z}_q^{(2nm_0+n)\kappa}, \mathbf{v}''_{0,i} \leftarrow \mathbb{Z}_q^{m_0+1}, \mathbf{v} \leftarrow \mathbb{Z}_p^{2m_0})$ to replace $(\mathbf{c}_0, \mathbf{c}''_{0,i}, \mathbf{c})$. Let us first write $\mathbf{B}$ as $\binom{\mathbf{B}_{\mathsf{top}}}{\mathbf{B}_{\mathsf{bot}}}$, where $\mathbf{B}_{\mathsf{bot}}$ represents the bottom n rows of $\mathbf{B}$. Then, we observe that the indistinguishability follows from sLWE, which implies

$$(\mathbf{B}_{\mathsf{bot}}, \mathbf{M}, \mathbf{s}_1\mathbf{B}_{\mathsf{bot}} + \mathbf{e}_0, \mathbf{s}_1\mathbf{M} + \mathbf{e}_M) \approx_c (\mathbf{B}_{\mathsf{bot}}, \mathbf{M}, \bar{\mathbf{v}}_0, \bar{\mathbf{v}}'')$$

  where $\mathbf{e}_M \leftarrow \mathcal{D}_{\mathbb{Z},\chi_1}^{2m_0}, \bar{\mathbf{v}}_0 \leftarrow \mathbb{Z}_q^{(2nm_0+n)\kappa}$ and $\bar{\mathbf{v}}'' \leftarrow \mathbb{Z}_q^{2m_0}$. We have

$$\begin{aligned}
(\mathbf{c}_0, \mathbf{c}''_{0,i}) = \ & (\mathbf{s}_0\mathbf{B}_{\mathsf{top}} + \mathbf{s}_1\mathbf{B}_{\mathsf{bot}} + \mathbf{e}_0, \ \mathbf{s}_1\mathbf{M}\mathbf{V}_i + \mathbf{e}''_{0,i} &&) \\
= \ & (\mathbf{s}_0\mathbf{B}_{\mathsf{top}} + \mathbf{s}_1\mathbf{B}_{\mathsf{bot}} + \mathbf{e}_0, \ (\mathbf{s}_1\mathbf{M} + \mathbf{e}_M)\mathbf{V}_i - \mathbf{e}_M\mathbf{V}_i + \mathbf{e}''_{0,i} &&) \\
\approx_c \ & (\mathbf{s}_0\mathbf{B}_{\mathsf{top}} + \bar{\mathbf{v}}_0, \qquad \bar{\mathbf{v}}''\mathbf{V}_i - \mathbf{e}_M\mathbf{V}_i + \mathbf{e}''_{0,i} &&) \\
\approx_s \ & (\mathbf{v}_0, \qquad \bar{\mathbf{v}}''\mathbf{V}_i + \mathbf{e}''_{0,i} &&) \\
\approx_c \ & (\mathbf{v}_0, \qquad \mathbf{v}''_{0,i} &&)
\end{aligned}$$

  Here, the $\approx_s$ relies on noise smudging, namely $\mathbf{e}''_{0,i} \approx_s \mathbf{e}''_{0,i} - \mathbf{e}_M\mathbf{V}_i$, which in turn follows from $\chi_3 \geq \chi_1\tau_C\lambda^{\omega(1)}$. In addition, the last $\approx_c$ relies on NLWE, which tell us $(\mathbf{V}_i, \bar{\mathbf{v}}''\mathbf{V}_i + \mathbf{e}''_{0,i}) \approx_c (\mathbf{V}_i, \mathbf{v}''_{0,i})$. Similarly, we have $\mathbf{c} \approx_c \mathbf{v}$ over $\mathbb{Z}_p$. Note that the secret $\mathbf{s}_1$ does not appear in any other components. Thus, we have $\mathsf{H}_0 \approx_c \mathsf{H}_1$.
- H_2: This is the same as H_1, except we sample $\mathbf{v}_1 \leftarrow \mathbb{Z}_q^{2m_0m_2}$ to replace $\mathbf{c}_1$. We have $\mathsf{H}_1 \approx_c \mathsf{H}_2$, via the sLWE assumption. Clearly, it tells us $(\mathbf{B}_{\mathsf{att}}, \mathbf{s}_2\mathbf{B}_{\mathsf{att}} + \mathbf{e}_1) \approx_c (\mathbf{B}_{\mathsf{att}}, \bar{\mathbf{v}}_1)$ where $\bar{\mathbf{v}}_1 \leftarrow \mathbb{Z}_q^{2m_0m_2}$. Note that the secret $\mathbf{s}_2$ does not appear in any other component. Thus, we have

$$\mathbf{c}_1 \approx_c \mathbf{s}_0\mathbf{G}_{2nm_0,q} + \bar{\mathbf{v}}_1 \approx_s \mathbf{v}_1.$$

- H_3: This is the same as H_2, except we sample $\mathbf{v}_2 \leftarrow \mathbb{Z}_p^{m_1}$ to replace $\mathbf{c}_2$. We observe that the indistinguishability follows from sLWE, which implies

$$(\mathbf{B}_{\mathsf{tag}}, \mathbf{s}\mathbf{B}_{\mathsf{tag}} + \mathbf{e}_2) \approx_c (\mathbf{B}_{\mathsf{tag}}, \bar{\mathbf{v}}_2)$$

  where $\bar{\mathbf{v}}_2 \leftarrow \mathbb{Z}_p^{m_1}$. Then, $\mathbf{c}_2 \approx_c \bar{\mathbf{v}}_2 + \mathbf{s}_3\mathbf{G}_{n,p} \approx_s \mathbf{v}_2$. Note that the secret $\mathbf{s}$ does not appear in any other components. Thus, we have $\mathsf{H}_2 \approx_c \mathsf{H}_3$ via the sLWE assumption.
- H_4: This is the same as H_3, except we sample $\mathbf{v}_3 \leftarrow \mathbb{Z}_q^{m_2\mathcal{K}}$ to replace $\mathbf{c}_3$. We observe that the indistinguishability follows from sLWE, which implies

$$(\mathbf{A}_{\mathsf{tag}} - \mathbf{t}^* \otimes \mathbf{G}_{n,q}, \mathbf{s}_3(\mathbf{A}_{\mathsf{tag}} - \mathbf{t}^* \otimes \mathbf{G}_{n,q}) + \mathbf{e}_3) \approx_c (\mathbf{A}_{\mathsf{tag}} - \mathbf{t}^* \otimes \mathbf{G}_{n,q}, \mathbf{v}_3).$$

  Note that the secret $\mathbf{s}_3$ does not appear in any other components. Thus, we have $\mathsf{H}_3 \approx_c \mathsf{H}_4$ via the sLWE assumption.

– H_5: This is the same as H_4, except that we sample $\mathbf{v}'_{0,i} \leftarrow \mathbb{Z}_q^{(m_0+1)m_2\ell}$ to replace $\mathbf{c}'_{0,i}$. To show that $\mathsf{H}_4 \approx_c \mathsf{H}_5$, it is sufficient to prove that $\mathbf{c}'_{0,i}$ is pseudorandom, namely

$$\mathbf{c}'_{0,i} = \mathbf{s}_0 \left((\mathbf{A}_{\mathsf{att}} - \mathbf{x}_i \otimes \mathbf{G}_{n,q}) \otimes \mathbf{V}_i \right) + \mathbf{e}'_{0,i} \approx_c \mathbf{v}'_{0,i}.$$

In particular, the $\approx_c$ in $\mathbf{c}'_{0,i}$ relies on tensor LWE (Lemma 3) which tells us

$$\begin{aligned} &\left(\mathbf{A}_{\mathsf{att}}, \{\mathbf{s}_0 \left(\mathbf{I}_n \otimes \mathbf{V}_i\right) \left((\mathbf{A}_{\mathsf{att}} - \mathbf{x}_i \otimes \mathbf{G}_{n,q}) \otimes \mathbf{I}_{m_0+1} \right) + \mathbf{e}'_{0,i}, \mathbf{V}_i\}_{i\in[Q]}\right) \\ \approx_c &\left(\mathbf{A}_{\mathsf{att}}, \{\mathbf{v}'_{0,i}, \mathbf{V}_i\}_{i\in[Q]}\right). \end{aligned}$$

The analysis for the remaining two cases proceeds analogously to the first case. Specifically, the simulation strategy and the sequence of hybrid games require only symmetric modifications to accommodate the different query structures. For the sake of brevity, we omit the exhaustive details here.

F Implications to PBE and DP-ABE

In this section, through appropriately setting the circuit class supported by our BPCP-ABE in Sect. 5.3, two novel schemes, dubbed PBE and DP-ABE, can be obtained. Below we present the syntax and security definition of PBE.

F.1 Syntax and Security Definition of PBE

Before giving the definition of PBE, let us recall BE [20]. In a BE scheme, the broadcaster is allowed to encrypt messages to a specified set of users $F \subseteq [N]$, where N is the number of users in the system. Each user i has an index and a corresponding secret key sk_i. Only users in the set F can successfully decrypt by using their secret keys sk_i where $i \in F$, while unauthorized users $i \notin F$ cannot decrypt even if they collude. The PBE syntax is canonical for BE, except that each user is allowed to update a punctured key through an additional algorithm PBE.BatchPun. This key puncture mechanism allows users in the PBE system to erase selected data, aiming to achieve fine-grained forward security. Additionally, we require that PBE supports batch-puncturing, which means that running the PBE.BatchPun algorithm once can revoke access permissions for ciphertexts under multiple tags. A formal definition is given below.

Definition 6 (Puncturable Broadcast Encryption). *Let λ be a security parameter, $N = N(\lambda)$ be the number of users, $\mathcal{K} = \mathcal{K}(\lambda)$ be the number of tags, $\mathcal{T} = \mathcal{T}(\lambda)$ be a tags space, and $\mathcal{M} = \mathcal{M}(\lambda)$ be a plaintext space. A* PBE *scheme consists of a tuple of efficient algorithms Π_{PBE} with the following properties:*

– $\mathsf{PBE.Setup}(1^\lambda, N, \mathcal{K}) \to (\mathsf{mpk}, \mathsf{msk})$. *On input the security parameter λ, the number of users N and the number of tags $\mathcal{K}$ in each ciphertext, the* PBE.Setup *algorithm outputs the master public key* mpk *and the master secret key* msk.

- $\mathsf{PBE.Enc}(\mathsf{mpk}, F, \mu, \{t_1, \ldots, t_\mathcal{K}\}) \to \mathsf{ct}$. *On input the master public key* mpk*, a subset* $F \subseteq [N]$*, a message* $\mu \in \mathcal{M}$*, and a list of tags* $t_1, \ldots, t_\mathcal{K} \in \mathcal{T}$*, the* $\mathsf{PBE.Enc}$ *algorithm outputs a ciphertext* ct.
- $\mathsf{PBE.KeyGen}(\mathsf{mpk}, \mathsf{msk}, i) \to \mathsf{sk}_{i,\emptyset}$. *On input the master public key* mpk*, the master secret key* msk*, and a user* $i \in N$*, the* $\mathsf{PBE.KeyGen}$ *algorithm outputs a secret key* $\mathsf{sk}_{i,\emptyset} = (\mathsf{isk}_i, \mathsf{psk}_\emptyset)$*, where* isk_i *is the index secret key and* $\mathsf{psk}_\emptyset$ *is the punctured secret key.*
- $\mathsf{PBE.BatchPun}(\mathsf{mpk}, \mathsf{sk}_{i,\eta-1}, \{t'_{\eta,1}, \ldots, t'_{\eta,k_\eta}\}) \to \mathsf{sk}_{i,\eta}$. *On input the master public key* mpk*, the secret key* $\mathsf{sk}_{i,\eta-1}$*, and a list of punctured tags* $\{t'_{\eta,1}, \ldots, t'_{\eta,k_\eta}\}$ *where* $\eta, k_\eta \in \mathbb{N}$*, the* $\mathsf{PBE.BatchPun}$ *algorithm outputs a new secret key* $\mathsf{sk}_{i,\eta} = (\mathsf{isk}_i, \mathsf{psk}_\eta)$.
- $\mathsf{PBE.Dec}(\mathsf{mpk}, F, \mathsf{sk}_{i,\eta}, \mathsf{ct}) \to \mu$ *or* $\perp$. *On input the master public key* mpk*, a subset* F*, the punctured secret key* $\mathsf{sk}_{i,\eta}$*, a ciphertext* ct *with tags* $\{t_1, ..., t_\mathcal{K}\}$*, the* $\mathsf{PBE.Dec}$ *algorithm outputs a message* μ *or a failure symbol* $\perp$.

Correctness. *For all inputs* $\lambda \in \mathbb{N}, i \in [N], F \subseteq [N]$ *and* $\mu \in \mathcal{M}$*, we require the following:*

- *If* $i \in F$*,*

$$\Pr[\mathsf{PBE.Dec}(\mathsf{mpk}, F, \mathsf{sk}_{i,\emptyset}, \mathsf{PBE.Enc}(\mathsf{mpk}, F, \mu, \{t_1, \ldots, t_\mathcal{K}\})) = \mu] = 1 - \mathsf{negl}(\lambda).$$

- *If* $i \in F$ *and* $P_\eta \cap \{t_1, \ldots, t_\mathcal{K}\} = \emptyset$*,*

$$\Pr[\mathsf{PBE.Dec}(\mathsf{mpk}, F, \mathsf{sk}_{i,\eta}, \mathsf{PBE.Enc}(\mathsf{mpk}, F, \mu, \{t_1, \ldots, t_\mathcal{K}\})) = \mu] = 1 - \mathsf{negl}(\lambda).$$

where $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{PBE.Setup}(1^\lambda, N, \mathcal{K})$, $\mathsf{sk}_{i,\emptyset} \leftarrow \mathsf{PBE.KeyGen}(\mathsf{mpk}, \mathsf{msk}, i)$, $\mathsf{sk}_{i,\eta} \leftarrow \mathsf{PBE.BatchPun}(\mathsf{mpk}, \mathsf{sk}_{i,\eta-1}, \{t'_{\eta,1}, \ldots, t'_{\eta,k_\eta}\})$.

Security Definition. Since PBE is captured as a special case of BPCP-ABE by specifying the relation, the security experiment $\mathsf{EXP}^{\mathsf{PBE}}_{\mathcal{A},\Pi}$ and the advantage $Adv^{\mathsf{Sel\text{-}IND}}_{\mathcal{A},\Pi_{\mathsf{PBE}}}(1^\lambda)$ of adversaries for PBE follow from $\mathsf{EXP}^{\mathsf{BP\text{-}ABE}}_{\mathcal{A},\Pi}$ (see Sect. 3.1) and will not be elaborated further.

Definition 7 (Selective Security for PBE). *A* PBE *scheme is said to satisfy selective security if for any PPT adversary* $\mathcal{A}$*, there exists a negligible function* $\mathsf{negl}(\cdot)$ *for all sufficiently large* $\lambda \in \mathbb{N}$ *such that* $Adv^{\mathsf{Sel\text{-}IND}}_{\mathcal{A},\Pi_{\mathsf{PBE}}}(1^\lambda) = \mathsf{negl}(\lambda)$.

Remark 3. Selective security and very selective security are equivalent for PBE, as the game for the selective adversary can be simulated by the very selective adversary using the secret keys provided by the challenger [4, 41].

F.2 Obtaining PBE from BPCP-ABE

PBE can be regarded as a special case of BPCP-ABE for circuits. Here, we consider our BPCP-ABE for the Boolean circuit class $\mathcal{F} = \{f_Y : \{0,1\}^{\lceil \log N \rceil} \to$

$\{0,1\}\}_{Y\subseteq[N]}$, which results in a PBE scheme. Define the equation testing function indexed by the tag $\mathbf{y} \in \{0,1\}^{\lceil \log N \rceil}$ as

$$EQ_{\mathbf{y}} : \{0,1\}^{\lceil \log N \rceil} \to \{0,1\}, \text{ where } EQ_{\mathbf{y}}(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} = \mathbf{y} \\ 1 & \text{if } \mathbf{x} \neq \mathbf{y}. \end{cases}$$

We associate each set $Y \subseteq [N]$ with the circuit f_Y, defined by

$$\mathbf{x} \mapsto 1 + \sum_{\mathbf{y}\in Y} EQ_{\mathbf{y}}(\mathbf{x}) - |Y| = \begin{cases} 0 & \text{if } \mathbf{x} \in Y \\ 1 & \text{if } \mathbf{x} \notin Y. \end{cases}$$

Plugging the above construction into BPCP-ABE (Sect. 5.3), we obtain a PBE scheme for N users with parameters

$$|\mathsf{mpk}| = \mathsf{poly}(\log N, \lambda, \mathcal{K}), \quad |\mathsf{ct}| = \mathsf{poly}(\log N, \lambda, \mathcal{K}),$$
$$|\mathsf{isk}| = \mathsf{poly}(\log N, \lambda), \quad |\mathsf{psk}_\eta| = \mathsf{poly}(\lambda, \eta)$$

where $\mathcal{K}$ and η are the number of tags in the ciphertext and the number of punctures, respectively.

F.3 Obtaining DP-ABE for Circuits from BPCP-ABE

The idea behind our construction is to observe that a general PE framework [38] can be obtained from BGG+ [12] and key delegation mechanisms [15]. In turn, PE can also serve as a foundation for building a delegatable KP-ABE. As a generalization, we modify the part of BPCP-ABE that captures the puncture property to obtain the first post-quantum DP-ABE for circuits[6]. To be specific, we generalize the function $g_{t'}$ associated with the punctured tag t' in BPCP-ABE to a general function (objective policy) $g \in \mathcal{G}$, which is no longer tied to the previous formulation (see (1)). Here, t' serves solely as an index that references the function g, while the tag $\mathbf{t}$ in BPCP-ABE is an objective attribute for DP-ABE. Subsequently, the single index t' is plugged into BPCP-ABE as a punctured tag, while f (resp. $\mathbf{x}$) serves as a subjective policy (resp. subjective attribute) for DP-ABE. With this approach, we can obtain a DP-ABE construction from $\Pi_{\mathsf{BPCP\text{-}ABE}} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{BatchPun}, \mathsf{Dec})$. Our DP-ABE for circuit classes $\mathcal{F} = \{f : \mathbb{Z}_q^\ell \to \{0,1\}\}$ and $\mathcal{G} = \{g : \mathbb{Z}_q^\mathcal{K} \to \{0,1\}\}$ works as follows.

- $\mathsf{DP.Setup}(1^\lambda, \ell, \mathcal{K})$. On input the security parameter λ, the number of subjective attributes ℓ and the number of objective attributes $\mathcal{K}$, the setup algorithm does the following:

[6] In a DP-ABE scheme [7,8], a message μ is encrypted under a subjective policy f and an objective attribute vector $\mathbf{t}$ chosen by the sender. The secret key corresponds to a subjective attribute vector $\mathbf{x}$ and an objective policy g. Successful decryption occurs if and only if $f(\mathbf{x}) = 0$ and $g(\mathbf{t}) = 0$. In other words, both the sender and the receiver can specify fine-grained access policies to encrypted data, and the receiver can only recover the message μ when a match occurs.

  - Run $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \ell, \mathcal{K}, \eta_{max})$.
  - Output $\mathsf{mpk}^{\mathsf{DP\text{-}ABE}} := \mathsf{mpk}$, $\mathsf{msk}^{\mathsf{DP\text{-}ABE}} := \mathsf{msk}$.
- $\mathsf{DP.Enc}(\mathsf{mpk}^{\mathsf{DP\text{-}ABE}}, \mu, (f, \mathbf{t}))$. On input the master public key $\mathsf{mpk}^{\mathsf{DP\text{-}ABE}}$, a message $\mu \in \mathcal{M}$, a subjective function $f \in \mathcal{F}$ and an objective attribute vector $\mathbf{t} \in \mathbb{Z}_q^{\mathcal{K}}$, the encryption algorithm does the following:
  - Parse $\mathbf{t}$ as $\{t_1, ..., t_{\mathcal{K}}\}$.
  - Run $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}^{\mathsf{DP\text{-}ABE}}, f, \mu, \{t_1, \ldots t_{\mathcal{K}}\})$.
  - Output $\mathsf{ct}_{f,\mathbf{t}}^{\mathsf{DP\text{-}ABE}} := \mathsf{ct}$.
- $\mathsf{DP.KeyGen}(\mathsf{mpk}^{\mathsf{DP\text{-}ABE}}, \mathsf{msk}^{\mathsf{DP\text{-}ABE}}, (\mathbf{x}, g))$. On input the master public key $\mathsf{mpk}^{\mathsf{DP\text{-}ABE}}$, the master secret key $\mathsf{msk}^{\mathsf{DP\text{-}ABE}}$, a subjective attribute vector $\mathbf{x} \in \mathbb{Z}_q^{\ell}$ and an objective function $g \in \mathcal{G}$, the keygen algorithm does the following:
  - Run $\mathsf{sk}_{\mathbf{x},\emptyset} \leftarrow \mathsf{KeyGen}(\mathsf{mpk}^{\mathsf{DP\text{-}ABE}}, \mathsf{msk}^{\mathsf{DP\text{-}ABE}}, \mathbf{x})$.
  - Run $\mathsf{sk}_{\mathbf{x},1} \leftarrow \mathsf{BatchPun}(\mathsf{mpk}^{\mathsf{DP\text{-}ABE}}, \mathsf{sk}_{\mathbf{x},\emptyset}, t')$. Here, the function $g \in \mathcal{G}$ is indexed only by t'.
  - Output $\mathsf{sk}_{\mathbf{x},g}^{\mathsf{DP\text{-}ABE}} := \mathsf{sk}_{\mathbf{x},1}$.
- $\mathsf{DP.Dec}(\mathsf{mpk}^{\mathsf{DP\text{-}ABE}}, \mathsf{sk}_{\mathbf{x},g}^{\mathsf{DP\text{-}ABE}}, \mathsf{ct}_{f,\mathbf{t}}^{\mathsf{DP\text{-}ABE}}, (f, \mathbf{t}), (\mathbf{x}, g))$. On input the master public key $\mathsf{mpk}^{\mathsf{DP\text{-}ABE}}$, a decryption key $\mathsf{sk}_{\mathbf{x},g}^{\mathsf{DP\text{-}ABE}}$ along with its associated pair consisting of a subjective attribute vector $\mathbf{x}$ and an objective function g, a ciphertext $\mathsf{ct}_{f,\mathbf{t}}^{\mathsf{DP\text{-}ABE}}$ along with its associated pair consisting of a subjective function f and an objective attribute vector $\mathbf{t}$, the decryption algorithm does the following:
  - If $f(\mathbf{x}) = 0$ and $g(\mathbf{t}) = 0$, output $\mu \leftarrow \mathsf{Dec}(\mathsf{mpk}^{\mathsf{DP\text{-}ABE}}, \mathsf{sk}_{\mathbf{x},g}^{\mathsf{DP\text{-}ABE}}, \mathsf{ct}_{f,\mathbf{t}}^{\mathsf{DP\text{-}ABE}})$.
  - Otherwise, output $\perp$.

Efficiency. The parameter settings of our DP-ABE scheme are the same as those of BPCP-ABE (Appendix D), which achieves:

$$\left|\mathsf{mpk}^{\mathsf{DP\text{-}ABE}}\right| = O_\lambda((\ell + \mathcal{K}) \cdot d^2),\ \left|\mathsf{ct}_{f,\mathbf{t}}^{\mathsf{DP\text{-}ABE}}\right| = O_\lambda(\mathcal{K} \cdot d^2),\ \left|\mathsf{sk}_{\mathbf{x},g}^{\mathsf{DP\text{-}ABE}}\right| = O_\lambda(\ell \cdot d^2)$$

where $O_\lambda(\cdot)$ hides factors polynomial in λ.

Theorem 2 (Correctness). *Assuming that our* BPCP-ABE *(Sect. 5.3) satisfies correctness, our* DP-ABE *construction is correct.*

Proof. The correctness can be proved from BPCP-ABE, just constrain the number of punctures $\eta = 1$.

Theorem 3 (Security). *Assuming that our* BPCP-ABE *(Sect. 5.3) satisfies very selective security, our* DP-ABE *scheme for circuits is very selective secure.*

Proof. The proof is based on the idea from [7]. Let Q be the number of key queries made by the adversary $\mathcal{A}$. For the challenge function (policy) f^* and the challenge attribute vector $\mathbf{t}^*$, we consider two cases: $f^*(\mathbf{x}_i) \neq 0$ and $g_i(\mathbf{t}^*) \neq 0$, where $i \in [Q]$. Without loss of generality, we first analyze $f^*(\mathbf{x}_i) \neq 0$.

- For $f^*(\mathbf{x}_i) \neq 0$:

- ▷ If $g_i(\mathbf{t}^*) = 0$, we prove that our DP-ABE is secure for this type of key through a series of hybrids in Case 1 of BPCP-ABE (see Appendix E). This stems from the absence of the tuple $(\mathbf{x}_i, \mathsf{sk}_{\mathbf{x}_i,\eta}, P_{\mathbf{x}_i}, C_{\mathbf{x}_i})$ in Case 1 of BPCP-ABE, indicating that $\mathcal{A}$ obtains an initial punctured key. With such a key, $\mathcal{A}$ can derive any new punctured secret key for $\mathbf{x}_i$. Plugging it into the setting of DP-ABE, even if $\mathcal{A}$ obtains the decryption key satisfying $f^*(\mathbf{x}_i) \neq 0$ and $g_i(\mathbf{t}^*) = 0$, our DP-ABE ciphertext is indistinguishable from random.
- ▷ If $g_i(\mathbf{t}^*) \neq 0$, this corresponds to Case 2 of BPCP-ABE. It suffices to prove the indistinguishability of our construction for the key with the number of punctures $\eta = 1$. Plugging it into the setting of DP-ABE, through a series of hybrids in Case 2 of BPCP-ABE, we show that even if the adversary has the decryption key satisfying $f^*(\mathbf{x}_i) \neq 0$ and $g_i(\mathbf{t}^*) \neq 0$, the ciphertext remains indistinguishable from random.

– For $g_i(\mathbf{t}^*) \neq 0$:
- ▷ If $f^*(\mathbf{x}_i) \neq 0$, this case has been discussed above.
- ▷ If $f^*(\mathbf{x}_i) = 0$, this clearly corresponds to Case 3 of BPCP-ABE. Plugging it into the setting of DP-ABE, through a series of hybrids in Case 3 of BPCP-ABE, the security proof for this case can be completed.