Optimal Broadcast Encryption from Pairings and LWE

Shweta Agrawal[1](✉) and Shota Yamada[2](✉)

[1] IIT Madras, Chennai, India
shweta.a@cse.iitm.ac.in
[2] National Institute of Advanced Industrial Science and Technology (AIST),
Tokyo, Japan
yamada-shota@aist.go.jp

Abstract. Boneh, Waters and Zhandry (CRYPTO 2014) used multilinear maps to provide a solution to the long-standing problem of public-key broadcast encryption (BE) where all parameters in the system are small. In this work, we improve their result by providing a solution that uses only *bilinear* maps and Learning With Errors (LWE). Our scheme is fully collusion-resistant against any number of colluders, and can be generalized to an identity-based broadcast system with short parameters. Thus, we reclaim the problem of optimal broadcast encryption from the land of "Obfustopia".

Our main technical contribution is a ciphertext policy attribute based encryption (CP-ABE) scheme which achieves special efficiency properties – its ciphertext size, secret key size, and public key size are all independent of the size of the circuits supported by the scheme. We show that this special CP-ABE scheme implies BE with optimal parameters; but it may also be of independent interest. Our constructions rely on a novel interplay of bilinear maps and LWE, and are proven secure in the generic group model.

1 Introduction

Broadcast Encryption (BE) [30] enables a sender to encrypt a message for a subset of users who are listening on a broadcast channel. In more detail, in a BE system, a sender can encrypt to any set S of its choice, and any user in S can decrypt the broadcast using its secret key. The system is said to be fully collusion resistant if no collection of users outside S can learn anything about the plaintext.

Introduced in a seminal work by Fiat and Naor [30], the primitive of broadcast encryption has received significant attention, with diverse constructions achieving different tradeoffs in the sizes of ciphertext, secret key and public parameters. Of particular importance is the size of the ciphertext overhead: namely, the size of the ciphertext beyond what is necessary for the description of the recipient set S and the symmetric encryption of the plaintext message. A BE scheme is said to have low overhead if the ciphertext overhead depends at most logarithmically

A. Canteaut and Y. Ishai (Eds.): EUROCRYPT 2020, LNCS 12105, pp. 13–43, 2020.
https://doi.org/10.1007/978-3-030-45721-1_2

on the number of users in the system (N, say). In this work, we focus on BE systems that are public key, have low ciphertext overhead and are fully collusion resistant.

The first work to satisfy the above desiderata was by Boneh, Gentry, and Waters [13], and was based on hardness assumptions on bilinear maps. This construction achieved optimal (constant) ciphertext overhead and short secret keys, but suffered from public parameter size which is linear in the number of users N. Follow-ups based on bilinear maps improved some aspects of this construction [8,28,29,33,37,45], but could not improve the public key size. Indeed, even relying on the existence of the powerful indistinguishability obfuscation [31], BE with short public key remained elusive (though it achieved other remarkable properties) [16].

This state of affairs was improved considerably by the work of Boneh, Waters and Zhandry [15] who provided the first construction of broadcast encryption, achieving optimal parameters including short public key, by relying on multilinear maps. This marked the first solution to a long standing open problem. However, the constructions suggested by [15] also have some limitations. In more detail, the [15] provide three broadcast encryption systems that use an $O(\log N)$ way multilinear map – this necessitates the degree of the map to be *polynomial* when N is exponential. More importantly, existing candidates of multilinear maps have been subject to many attacks [7,23–27,38,42] and their security is poorly understood. Thus, the question of broadcast encryption with optimal parameter size has so far, remained squarely in the land of "Obfustopia".

Our Results. In this work, we reclaim broadcast encryption from Obfustopia by providing a solution that uses only *bilinear* maps and Learning With Errors (LWE). Our scheme is public key, fully collusion-resistant against any number of colluders, and can be generalized to an identity-based broadcast system with short parameters. Along the way, we provide the first ciphertext policy attribute based encryption scheme whose ciphertext size, secret key size, and public key size are all independent of the size of the circuits supported by the scheme. This construction may be of independent interest. Our constructions rely on a novel interplay between bilinear maps and LWE and are proven secure in the generic group model.

1.1 Our Techniques

Recasting BE as CP-ABE: Our starting observation is that the question of broadcast encryption can be re-stated in terms of the notion of *ciphertext policy attribute based encryption* (CP-ABE). In a CP-ABE scheme, a ciphertext for a message m is labelled with a function (policy) f, and secret keys are labelled with public attributes $\mathbf{x}$ from the domain of f. Decryption succeeds to yield the hidden message m if and only if the attribute satisfies the policy, namely $f(\mathbf{x}) = 1$. To see BE as a special case of CP-ABE, note that the ciphertext may encode a circuit F_S that checks membership of a given user index in a set of recipients S, and the attributes $\mathbf{x}$ may encode user index in the set N.

Thus, a user i can use her CP-ABE secret key to test whether i is a member of the set S encoded in the ciphertext, and recover the message m if and only if this is true. Then, a natural approach to construct BE is to leverage CP-ABE schemes. However, unsurprisingly, constructions of CP-ABE achieving optimal parameters that suffice for BE, has been elusive.

From Pairings to LWE: All known constructions of BE from standard assumptions (i.e. without relying on the existence of multilinear maps or indistinguishability obfuscation) are based on various assumptions on bilinear groups. Since the question of optimal BE from pairings has met with little progress for over a decade, it is evidently meaningful to look at assumptions on other mathematical structures to seek a way forward. The most obvious candidate that presents itself is the versatile Learning With Errors (LWE) assumption, which has led to breakthroughs in similar primitives, notably in fully homomorphic encryption [17,18,20].

Let us then examine what is known from LWE in this context. The dual notion of key-policy ABE has met with fantastic success from LWE – the works of Gorbunov et al. [34] and Boneh et al. [12] show how to construct KP-ABE for all circuits (on the other hand, constructions based on pairings could only support the much weaker circuit class NC_1). KP-ABE is the same as CP-ABE with the roles of circuit and attributes swapped. Additionally, the KP-ABE construction of Boneh et al. [12], henceforth denoted as BGG^+, manages to encode the circuit very succinctly – in more detail, the size of the public and secret keys in the BGG^+ construction are independent of the circuit size and depend only on the *depth* of the circuit. Additionally, the size of the ciphertext is also independent of the circuit size and depends only on input length. Since the input length for the circuit F_S that checks membership in S is an encoding of a user index, it is of size $O(\log N)$. Moreover, it is easy to check that the depth of F_S is also $O(\log N)$. Therefore, if we have a CP-ABE with analogous efficiency, namely, so that the public key size, secret key size, and ciphertext size do not depend on the size but only input length and depth of the circuit, it follows that we can obtain BE with optimal parameters.

Constructing CP-ABE from LWE: Thus, it suffices to ask whether we can have a CP-ABE scheme, denoted by cpABE, with the desired efficiency. To leverage the succinctness of the circuit encoding of BGG^+, a naive idea is to set $\mathsf{cpABE.CT}(F_S) = \mathsf{BGG}^+.\mathsf{SK}(F_S)$. Two immediate problems present themselves: (i) Where to embed the message m[1], and (ii) Computing $\mathsf{BGG}^+.\mathsf{SK}(F_S)$ requires the master secret but encryption is a public key algorithm.

To address these challenges, a first idea is to exploit the *decomposability* of BGG^+. In more detail, decomposability means that the ciphertext for attribute $\mathbf{x}$ and message m may be decomposed into $|\mathbf{x}| + 1$ encodings, one for each bit x_i of the attribute string and message m – these are tied together using common randomness used during their generation. Let us denote the encoding corresponding to bit x_i as ψ_{i,x_i}. Then, a natural idea is to let the encryptor sample a fresh instance of the BGG^+ scheme, generate $\mathsf{BGG}^+.\mathsf{SK}(F_S)$ and encrypt each

[1] This question is surprisingly non-trivial even in the symmetric key setting.

$\psi_{i,b}$ using a different public key encryption scheme, say with PKE key $\mathsf{PK}_{i,b}$. This yields a CP-ABE with the desired efficiency, inherited directly from the succinctness of the BGG^+ key and the decomposability of the BGG^+ ciphertext.

Constraining the Information Leaked (Or, Back to Pairings): However, this scheme is obviously not collusion resistant: a user with keys for $\mathbf{x}$ and $\bar{\mathbf{x}}$ can decrypt every ciphertext. To make the scheme collusion resistant, we would like to replace the naive use of public key encryption above into a more sophisticated scheme, which hides all but the output of the BGG^+ decryption algorithm. This description bears close resemblance to a functional encryption scheme for some restricted functionality, for which we turn to—pairings! In particular, we isolate the $\psi_{i,b}$ by randomizing and lifting them to the exponent of a bilinear group. The hope is that we may provide a secret key for some attribute $\mathbf{x}$ such that it only allows the appropriate ψ_{i,x_i} to be selected and combined so that only the output of the BGG^+ decryption is revealed, and that the randomization, which will be unique to every cpABE ciphertext and secret key pair, prevents collusion attacks.

Evaluation of NC_1 Circuit in the Exponent: Several questions arise. First, we discussed above that the circuit for checking membership in set S is in NC_1 – however, pairings are only capable of supporting at most quadratic operations. How then, do we hope to compute an NC_1 circuit in the exponent of a bilinear group? The answer lies in the specific structure of the BGG^+ evaluation algorithm, which, even for a circuit in P is *linear* in the encodings and the secret key, followed by a final *rounding* step to remove the noise. Indeed, the knowledgeable reader may observe that this very linearity of the BGG^+ evaluation procedure has been the cause of attacks in other contexts [1] – what is a "curse" there is a "blessing" here! However, the rounding step remains – this is in NC_1 and clearly cannot be performed in the exponent.

An approach is to perform the linear computation (which represents the circuit F_S) in the exponent, recover the output via discrete log, and then compute the rounding in the clear. Again, it is unclear this satisfies either correctness or security. For the former, note that recovering the encoded output from the exponent requires that the output be polynomially bounded. In this case, the output is the message bit plus some noise that resulted from the homomorphic evaluation. While the noise in this context may be superpolynomial in general, we can convert our NC_1 circuit into a branching program and leverage the asymmetric noise growth for BP evaluation of BGG^+ encodings to ensure that the noise is bounded by a polynomial [35].

The more worrisome issue is that of security. It is well known that the noise that results from homomorphic evaluation of encodings leaks the noise in the original encodings and is often a security threat –in fact, the savvy reader may have observed that this leakage is one of the main barriers in constructing iO from standard assumptions [2,6]. However, here we are rescued by the serendipitous fact that what we are trying to build here is a kind of attribute based encryption, not functional encryption! In more detail, the leakage caused by the noise is a security threat in the context of functional encryption, as formalized in [1,2,6] – this is because a decryptor who possesses some secret keys for a functional encryption scheme must still not be able to learn anything about the encrypted message beyond what the keys reveal. On the other hand, attribute based encryption is a much simpler "all or nothing" primitive – if the adversary possesses a single key that decrypts a ciphertext, there are no more secrets the scheme withholds from her. Hence, if the adversary has a key that lets her recover the value encoded in the exponent, the additional leakage created by the noise terms do not pose a security threat.

Preventing Mix and Match Attacks: To prevent collusions, we design the decryption algorithm so that the decryptor obtains a randomized version of the ciphertext components in the exponents as $g_T^{\delta\psi_{i,x_i}}$, where g_T is the target group, ψ_{i,x_i} are BGG^+ encodings as defined above and δ is user specific randomness. Since δ is user specific, the attacker cannot combine partial decryption results of multiple users, preventing mix and match attacks.

Hence, it suffices to restrict our attention to the single user case. Here, we must ensure that the adversary only gets components $\{g_T^{\delta\psi_{i,x_i}}\}_i$ corresponding to the particular key $\mathbf{x}$ issued to her, instead of all the components $\{g_T^{\delta\psi_{i,b}}\}_{i,b}$ for $b \in \{0,1\}$. Furthermore, we must ensure that the attribute vector $\mathbf{x}$ is processed in the correct sequence – i.e., its bits are not permuted. To prevent these attacks, we bind each entry of the ciphertext and each bit of the secret key attribute $\mathbf{x}$ to the corresponding positions. This is possible by setting the master public key to be $\{g^{w_{i,b}}\}_{i,b}$ where $w_{i,b}$ are randomly chosen for each position i and $b \in \{0,1\}$ and setting the secret key and ciphertext as $\{g_2^{\delta/w_{i,x_i}}\}_i$ and $\{g_1^{\psi_{i,b}w_{i,b}}\}_{i,b}$ respectively. We remark that we need to use asymmetric pairings to prevent ciphertext (respectively key) components from being paired between themselves to leak information. By tying element values to their positions, we ensure that pairing of the ciphertext and secret key components corresponding to different positions result in a term which looks like $g_T^{\delta\psi_{i,b}w_{i,b}/w_{i',b'}}$ for $(i,b) \neq (i',b')$. Now, we claim that a term of the form $g^{\delta w_{i,b}/w_{i',b'}}$ is useless to the attacker – to see this, note that in the generic group model, an attacker cannot obtain any information about a value encoded in the exponent unless she finds a non-trivial linear relation that contains that term. However, since the term $\delta w_{i,b}/w_{i',b'}$ appears only when we pair the ciphertext component with position (i,b) with the secret key component with position (i',b') and cannot appear anywhere else, it follows that it cannot appear as a term in a linear combination that results in 0 (except with negligible probability). Thus, by using $\{w_{i,b}\}_{i,b}$, we enforce that the computation follows the desired path.

Combining the above ideas, we obtain our final CP-ABE scheme. By setting the circuit class appropriately, this yields BE and even Identity Based BE (or IBBE). Please see Sect. 3 for the CP-ABE and Sect. 5 for the construction of BE and IBBE.

Security in the Generic Group Model: We prove security in the generic group model, which closely follows the intuition we explained so far. Specifically, we will show that the adversary cannot find any non-trivial linear relation among the partial decryption results of the ciphertext components. The main challenge in the security proof is that the partial decryption results obtained by using different secret keys are correlated – in more detail, they can contain terms $g_1^{\delta\psi_{i,0}}$ and $g_1^{\delta'\psi_{i,1}}$ where $\psi_{i,0}$ and $\psi_{i,1}$, if learned simultaneously, lead to a complete break of security. Simulating these in the standard model using the security of BGG^+ appears difficult.

To address the issue we first observe that the adversary cannot take a linear combination among partial decryption results obtained by two different secret keys in a meaningful way, since they are randomized by the user specific randomness introduced for preventing collusions. This implies that if the adversary manages to find a non-trivial linear relation among the partial decryption results, all the terms involved should be obtained from the same secret key. We also observe that until the point when the adversary finds the *first* non-trivial linear relation, the simulator can simulate the generic group oracles without knowing the corresponding encodings. This can be done by simply pretending that there is no non-trivial linear relation among the terms.

The above observations allow us to concentrate on the security proof for the single-key case without worrying about the partial decryption results by other keys. We can then conclude by using the security of the BGG^+ scheme. In more detail, an adversary who can find a non-trivial linear relation among the partial decryption results can be used to distinguish a BGG^+ ciphertexts from random ones, since the partial decryption result by a single key essentially corresponds to a BGG^+ ciphertext in exponent and it cannot find any non-trivial linear relation among the random ciphertext components as long as the modulus size is exponential.

1.2 Related Works

In an independent work (that predates ours), Brakerski and Vaikuntanathan [21] also construct broadcast encryption achieving optimal parameters. Their techniques as well as final result are very different from ours – while our work crucially uses pairings in conjunction with LWE, they rely entirely on LWE and new assumptions in the regime of lattices. Both works can be seen as following the broad approach of starting with a succinct single-key CP-ABE from LWE[2], and adding collusion resistance using pairings (ours) or new techniques in the lattice regime (theirs).

The techniques in our work are similar in spirit to a growing line of work that uses "the best of both" of pairings and LWE [2,6,36,39], but quite different in details. Closest to our work are techniques used to construct *key policy functional encryption* [2,6,39], which use FHE (based on LWE) for encrypted evaluation and pairings for performing FHE decryption in the exponent. While a major challenge in these constructions is the leakage caused by FHE decryption noise,

[2] The single key CP-ABE with succinct CT was also discovered by Boneh and Kim [14].

we sidestep this issue altogether because BE is an "all or nothing primitive" with no secrets from a legitimate key holder. On the other hand, we need new tricks to handle the functionality and security of a *ciphertext-policy* scheme – for instance, we need to use position-wise randomness on the exponent to prevent ciphertext and secret key components from being paired in illegitimate positions to leak information.

2 Preliminaries

In this section, we define some preliminaries that we require.

2.1 Attribute Based Encryption

Let $R = \{R_\lambda : A_\lambda \times B_\lambda \to \{0,1\}\}_\lambda$ be a relation where A_λ and B_λ denote "ciphertext attribute" and "key attribute" spaces. An attribute-based encryption (ABE) scheme for R is defined by the following PPT algorithms:

$\mathsf{Setup}(1^\lambda) \to (\mathsf{mpk}, \mathsf{msk})$: The setup algorithm takes as input the unary representation of the security parameter λ and outputs a master public key mpk and a master secret key msk.

$\mathsf{Enc}(\mathsf{mpk}, X, \mu) \to \mathsf{ct}$: The encryption algorithm takes as input a master public key mpk, a ciphertext attribute $X \in A_\lambda$, and a message bit μ. It outputs a ciphertext ct.

$\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, Y) \to \mathsf{sk}_Y$: The key generation algorithm takes as input the master public key mpk, the master secret key msk, and a key attribute $Y \in B_\lambda$. It outputs a private key sk_Y.

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{ct}, X, \mathsf{sk}_Y, Y) \to \mu$ or $\perp$: We assume that the decryption algorithm is deterministic. The decryption algorithm takes as input the master public key mpk, a ciphertext ct, ciphertext attribute $X \in A_\lambda$, a private key sk_Y, and private key attribute $Y \in B_\lambda$. It outputs the message μ or $\perp$ which represents that the ciphertext is not in a valid form.

Definition 2.1 (Correctness). *An ABE scheme for relation family R is correct if for all $\lambda \in \mathbb{N}$, $X \in A_\lambda, Y \in B_\lambda$ such that $R(X,Y) = 1$, and for all messages $\mu \in \mathcal{M}$,*

$$\Pr\left[\begin{array}{l}(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda),\ \mathsf{sk}_Y \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, Y),\\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, X, \mu):\ \mathsf{Dec}\Big(\mathsf{mpk}, \mathsf{sk}_Y, Y, \mathsf{ct}, X\Big) \neq \mu\end{array}\right] = \mathrm{negl}(\lambda)$$

where the probability is taken over the coins of Setup*,* KeyGen*, and* Enc*.*

Definition 2.2 ($\mathsf{Ada\text{-}IND}$ security for ABE). *For an ABE scheme* $\mathsf{ABE} = \{\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec}\}$ *for a relation family $R = \{R_\lambda : A_\lambda \times B_\lambda \to \{0,1\}\}_\lambda$ and a message space $\{\mathcal{M}_\lambda\}_{\lambda \in \mathbb{N}}$ and an adversary* A*, let us define* $\mathsf{Ada\text{-}IND}$ *security game as follows.*

1. **Setup phase:** *On input* 1^λ*, the challenger samples* $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ *and gives* mpk *to* A*.*
2. **Query phase:** *During the game,* A *adaptively makes the following queries, in an arbitrary order.* A *can make unbounded many key queries, but can make only single challenge query.*
   (a) **Key Queries:** A *chooses an input* $Y \in B_\lambda$*. For each such query, the challenger replies with* $\mathsf{sk}_Y \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, Y)$*.*
   (b) **Challenge Query:** *At some point,* A *submits a pair of equal length messages* $(\mu_0, \mu_1) \in (\mathcal{M})^2$ *and the target* $X^\star \in A_\lambda$ *to the challenger. The challenger samples a random bit* $b \leftarrow \{0,1\}$ *and replies to* A *with* $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, X^\star, \mu_b)$*.*

   We require that $R(X^\star, Y) = 0$ *holds for any* Y *such that* A *makes a key query for* Y *in order to avoid trivial attacks.*
3. **Output phase:** A *outputs a guess bit* b' *as the output of the experiment.*

We define the advantage $\mathsf{Adv}^{\mathsf{Ada\text{-}IND}}_{\mathsf{ABE},\mathsf{A}}(1^\lambda)$ *of* A *in the above game as*

$$\mathsf{Adv}^{\mathsf{Ada\text{-}IND}}_{\mathsf{ABE},\mathsf{A}}(1^\lambda) := \left|\Pr[\mathsf{Exp}_{\mathsf{ABE},\mathsf{A}}(1^\lambda) = 1 | b = 0] - \Pr[\mathsf{Exp}_{\mathsf{ABE},\mathsf{A}}(1^\lambda) = 1 | b = 1]\right|.$$

The ABE scheme ABE *is said to satisfy* $\mathsf{Ada\text{-}IND}$ *security (or simply adaptive security) if for any stateful PPT adversary* A*, there exists a negligible function* $\mathrm{negl}(\cdot)$ *such that* $\mathsf{Adv}^{\mathsf{Ada\text{-}IND}}_{\mathsf{ABE},\mathsf{A}}(1^\lambda) \neq \mathrm{negl}(\lambda)$*.*

We can consider the following stronger version of the security where we require the ciphertext to be pseudorandom.

Definition 2.3 ($\mathsf{Ada\text{-}INDr}$ **security for ABE).** *We define* $\mathsf{Ada\text{-}INDr}$ *security game similarly to* $\mathsf{Ada\text{-}IND}$ *security game except that the adversary* A *chooses single message* μ *instead of* (μ_0, μ_1) *at the challenge phase and the challenger returns* $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, X^\star, \mu)$ *if* $b = 0$ *and a random ciphertext* $\mathsf{ct} \leftarrow \mathcal{CT}$ *from a ciphertext space* $\mathcal{CT}$ *if* $b = 1$*. We define the advantage* $\mathsf{Adv}^{\mathsf{Ada\text{-}INDr}}_{\mathsf{ABE},\mathsf{A}}(1^\lambda)$ *of the adversary* A *accordingly and say that the scheme satisfies* $\mathsf{Ada\text{-}INDr}$ *security if the quantity is negligible.*

We also consider (weaker) selective versions of the above notions, where A specifies its target $X^\star$ at the beginning of the game.

Definition 2.4 ($\mathsf{Sel\text{-}IND}$ **security for ABE).** *We define* $\mathsf{Sel\text{-}IND}$ *security game as* $\mathsf{Ada\text{-}IND}$ *security game with the exception that the adversary* A *has to choose the challenge ciphertext attribute* $X^\star$ *before the setup phase but key queries* $Y_1, Y_2, \ldots$ *and choice of* (μ_0, μ_1) *can still be adaptive. We define the advantage* $\mathsf{Adv}^{\mathsf{Sel\text{-}IND}}_{\mathsf{ABE},\mathsf{A}}(1^\lambda)$ *of the adversary* A *accordingly and say that the scheme satisfies* $\mathsf{Sel\text{-}INDr}$ *security (or simply selective security) if the quantity is negligible.*

Definition 2.5 ($\mathsf{Sel\text{-}INDr}$ **security for ABE).** *We define* $\mathsf{Sel\text{-}INDr}$ *security game as* $\mathsf{Ada\text{-}INDr}$ *security game with the exception that the adversary* A *has to choose the challenge ciphertext attribute* $X^\star$ *before the setup phase but key queries* $Y_1, Y_2, \ldots$ *and choice of* μ *can still be adaptive. We define the advantage* $\mathsf{Adv}^{\mathsf{Sel\text{-}INDr}}_{\mathsf{ABE},\mathsf{A}}(1^\lambda)$ *of the adversary* A *accordingly and say that the scheme satisfies* $\mathsf{Sel\text{-}INDr}$ *security if the quantity is negligible.*

In the following, we recall definitions of various ABEs by specifying the relation. We start with the standard notions of ciphertext-policy attribute-based encryption (CP-ABE) and key-policy attribute-based encryption (KP-ABE).

CP-ABE for circuits. We define CP-ABE for circuit class $\{\mathcal{C}_\lambda\}_\lambda$ by specifying the relation. Here, $\mathcal{C}_\lambda$ is a set of circuits with input length $\ell(\lambda)$ and binary output. We define $A_\lambda^{\mathsf{CP}} = \mathcal{C}_\lambda$ and $B_\lambda^{\mathsf{CP}} = \{0,1\}^\ell$. Furthermore, we define the relation R_λ^{CP} as $R_\lambda^{\mathsf{CP}}(C, \mathbf{x}) = \neg C(\mathbf{x})$.[3]

KP-ABE for circuits. To define KP-ABE for circuits, we simply swap key and ciphertext attributes in CP-ABE for circuits. More formally, to define KP-ABE for circuits, we define $A_\lambda^{\mathsf{KP}} = \{0,1\}^\ell$ and $B_\lambda^{\mathsf{KP}} = \mathcal{C}_\lambda$. We also define $R_\lambda^{\mathsf{KP}} : A_\lambda^{\mathsf{KP}} \times B_\lambda^{\mathsf{KP}} \to \{0,1\}$ as $R_\lambda^{\mathsf{KP}}(\mathbf{x}, C) = \neg C(\mathbf{x})$.

We can also capture identity-based broadcast encryption (IBBE) and broadcast encryption (BE) as special cases of ABE by specifying the relations.

IBBE. To define IBBE, we define $A_\lambda^{\mathsf{IBBE}} = \mathcal{ID}(\lambda)^{\le t}$ and $B_\lambda^{\mathsf{IBBE}} = \mathcal{ID}(\lambda)$, where $\mathcal{ID}(\lambda)$ is the identity space and $\mathcal{ID}(\lambda)^{\le t}$ denotes all subsets of $\mathcal{ID}(\lambda)$ with size at most t. We also define $R_\lambda^{\mathsf{IBBE}} : A_\lambda^{\mathsf{IBBE}} \times B_\lambda^{\mathsf{IBBE}} \to \{0,1\}$ as $R_\lambda^{\mathsf{IBBE}}(S, \mathsf{id}) = \begin{cases} 1 & \text{if } \mathsf{id} \in S \\ 0 & \text{if } \mathsf{id} \notin S \end{cases}$. For IBBE, we typically require that the ciphertext size should be $o(t) \cdot \mathrm{poly}(\lambda)$, since otherwise we have a trivial construction from IBE.

BE. To define BE, we define $A_\lambda^{\mathsf{BE}} = 2^{[N(\lambda)]}$ and $B_\lambda^{\mathsf{BE}} = [N(\lambda)]$, where $N(\lambda) = \mathrm{poly}(\lambda)$ is the number of users in the system and $2^{[N(\lambda)]}$ denotes all subsets of $[N]$. We also define $R_\lambda^{\mathsf{BE}} : A_\lambda^{\mathsf{BE}} \times B_\lambda^{\mathsf{BE}} \to \{0,1\}$ as $R_\lambda^{\mathsf{BE}}(S, i) = 1$ when $i \in S$ and $R_\lambda^{\mathsf{BE}}(S, i) = 0$ otherwise. For BE, we typically require that the ciphertext size should be $o(N) \cdot \mathrm{poly}(\lambda)$, since otherwise we have a trivial construction from plain public key encryption.

We also define dual versions of BE and IBBE where the ciphertext and secret key attributes are swapped.

Dual IBBE (DIBBE). To define DIBBE, we define $A_\lambda^{\mathsf{DIBBE}} = \mathcal{ID}(\lambda)$ and $B_\lambda^{\mathsf{DIBBE}} = \mathcal{ID}(\lambda)^{\le t}$, where $\mathcal{ID}(\lambda)$ is the identity space. We define $R_\lambda^{\mathsf{DIBBE}} : A_\lambda^{\mathsf{DIBBE}} \times B_\lambda^{\mathsf{DIBBE}} \to \{0,1\}$ as $R_\lambda^{\mathsf{IBBE}}(\mathsf{id}, S) = 1$ if $\mathsf{id} \in S$ and $R_\lambda^{\mathsf{IBBE}}(\mathsf{id}, S) = 0$ otherwise.

Dual BE (DBE). To define DBE, we define $A_\lambda^{\mathsf{DBE}} = [N(\lambda)]$ and $B_\lambda^{\mathsf{DBE}} = 2^{[N(\lambda)]}$, where $N(\lambda) = \mathrm{poly}(\lambda)$ is the number of users in the system. We also define $R_\lambda^{\mathsf{DBE}} : A_\lambda^{\mathsf{DBE}} \times B_\lambda^{\mathsf{DBE}} \to \{0,1\}$ as $R_\lambda^{\mathsf{DBE}}(i, S) = 1$ when $i \in S$ and $R_\lambda^{\mathsf{DBE}}(i, S) = 0$ otherwise.

2.2 Lattice Preliminaries

Here, we recall some facts on lattices that are needed for the exposition of our construction. Throughout this section, n, m, and q are integers such that

[3] Here, we follow the standard convention in lattice-based cryptography where the decryption succeeds when $C(\mathbf{x}) = 0$ rather than $C(\mathbf{x}) = 1$.

$n = \mathrm{poly}(\lambda)$ and $m \geq n\lceil \log q \rceil$. In the following, let $\mathsf{SampZ}(\gamma)$ be a sampling algorithm for the truncated discrete Gaussian distribution over $\mathbb{Z}$ with parameter $\gamma > 0$ whose support is restricted to $z \in \mathbb{Z}$ such that $|z| \leq \sqrt{n}\gamma$.

Learning with Errors. We the introduce then learning with errors (LWE) problem.

Definition 2.6 (The LWE Assumption). *Let $n = n(\lambda)$, $m = m(\lambda)$, and $q = q(\lambda) > 2$ be integers and $\chi = \chi(\lambda)$ be a distribution over $\mathbb{Z}_q$. We say that the $\mathsf{LWE}(n, m, q, \chi)$ hardness assumption holds if for any PPT adversary A we have*

$$|\Pr[\mathsf{A}(\mathbf{A}, \mathbf{s}^\top \mathbf{A} + \mathbf{x}^\top) \to 1] - \Pr[\mathsf{A}(\mathbf{A}, \mathbf{v}^\top) \to 1]| \leq \mathrm{negl}(\lambda)$$

where the probability is taken over the choice of the random coins by the adversary A and $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$, $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $\mathbf{x} \leftarrow \chi^m$, and $\mathbf{v} \leftarrow \mathbb{Z}_q^m$. We also say that $\mathsf{LWE}(n, m, q, \chi)$ problem is subexponentially hard if the above probability is bounded by $2^{-n^\epsilon} \cdot \mathrm{negl}(\lambda)$ for some constant $0 < \epsilon < 1$ for all PPT A.

As shown by previous works [19,43], if we set $\chi = \mathsf{SampZ}(\gamma)$, the $\mathsf{LWE}(n, m, q, \chi)$ problem is as hard as solving worst case lattice problems such as gapSVP and SIVP with approximation factor $\mathrm{poly}(n) \cdot (q/\gamma)$ for some $\mathrm{poly}(n)$. Since the best known algorithms for 2^k-approximation of gapSVP and SIVP run in time $2^{\tilde{O}(n/k)}$, it follows that the above $\mathsf{LWE}(n, m, q, \chi)$ with noise-to-modulus ratio 2^{-n^ϵ} is likely to be (subexponentially) hard for some constant ϵ.

Trapdoors. Let us consider a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$. For all $\mathbf{V} \in \mathbb{Z}_q^{n\times m'}$, we let $\mathbf{A}_\gamma^{-1}(\mathbf{V})$ be an output distribution of $\mathsf{SampZ}(\gamma)^{m\times m'}$ conditioned on $\mathbf{A} \cdot \mathbf{A}_\gamma^{-1}(\mathbf{V}) = \mathbf{V}$. A γ-trapdoor for $\mathbf{A}$ is a trapdoor that enables one to sample from the distribution $\mathbf{A}_\gamma^{-1}(\mathbf{V})$ in time $\mathrm{poly}(n, m, m', \log q)$ for any $\mathbf{V}$. We slightly overload notation and denote a γ-trapdoor for $\mathbf{A}$ by $\mathbf{A}_\gamma^{-1}$. We also define the special gadget matrix $\mathbf{G} \in \mathbb{Z}_q^{n\times m}$ as the matrix obtained by padding $\mathbf{I}_n \otimes (1, 2, 4, 8, \ldots, 2^{\lceil \log q \rceil})$ with zero-columns. The following properties had been established in a long sequence of works [3,4,19,22,32,41].

Lemma 2.7 (Properties of Trapdoors). *Lattice trapdoors exhibit the following properties.*

1. *Given $\mathbf{A}_\tau^{-1}$, one can obtain $\mathbf{A}_{\tau'}^{-1}$ for any $\tau' \geq \tau$.*
2. *Given $\mathbf{A}_\tau^{-1}$, one can obtain $[\mathbf{A}\|\mathbf{B}]_\tau^{-1}$ and $[\mathbf{B}\|\mathbf{A}]_\tau^{-1}$ for any $\mathbf{B}$.*
3. *There exists an efficient procedure $\mathsf{TrapGen}(1^n, 1^m, q)$ that outputs $(\mathbf{A}, \mathbf{A}_{\tau_0}^{-1})$ where $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ for some $m = O(n \log q)$ and is 2^{-n}-close to uniform, where $\tau_0 = \omega(\sqrt{n \log q \log m})$.*

Lattice Evaluation. The following is an abstraction of the evaluation procedure in previous LWE based FHE and ABE schemes. We follow the presentation by Tsabary [47], but with different parameters.

Lemma 2.8 (Fully Homomorphic Computation [35]**).** *There exists a pair of deterministic algorithms* $(\mathsf{EvalF}, \mathsf{EvalFX})$ *with the following properties.*

- $\mathsf{EvalF}(\mathbf{B}, F) \to \mathbf{H}_F$. *Here,* $\mathbf{B} \in \mathbb{Z}_q^{n\times m\ell}$ *and* $F : \{0,1\}^\ell \to \{0,1\}$ *is a circuit.*
- $\mathsf{EvalFX}(F, \mathbf{x}, \mathbf{B}) \to \widehat{\mathbf{H}}_{F,\mathbf{x}}$. *Here,* $\mathbf{x} \in \{0,1\}^\ell$ *with* $x_1 = 1$[4] *and* $F : \{0,1\}^\ell \to \{0,1\}$ *is a circuit with depth* d. *We have* $[\mathbf{B} - \mathbf{x} \otimes \mathbf{G}]\widehat{\mathbf{H}}_{F,\mathbf{x}} = \mathbf{B}\mathbf{H}_F - F(\mathbf{x})\mathbf{G} \bmod q$, *where we denote* $[x_1\mathbf{G}\| \cdots \|x_k\mathbf{G}]$ *by* $\mathbf{x} \otimes \mathbf{G}$. *Furthermore, we have* $\|\mathbf{H}_F\|_\infty \leq m \cdot 2^{O(d)}, \quad \|\widehat{\mathbf{H}}_{F,\mathbf{x}}\|_\infty \leq m \cdot 2^{O(d)}$.
- *The running time of* $(\mathsf{EvalF}, \mathsf{EvalFX})$ *is bounded by* $\mathrm{poly}(n, m, \log q, 2^d)$.

The above algorithms are taken from [35], which is a variant of similar algorithms proposed by Boneh et al. [12]. The algorithms in [12] work for any polynomial-sized circuit F, but $\|\mathbf{H}_F\|_\infty$ and $\|\mathbf{H}_{F,\mathbf{x}}\|_\infty$ become super-polynomial even if the depth of the circuit is shallow (i.e., logarithmic depth). On the other hand, the above algorithms run in polynomial time only when F is of logarithmic depth, but $\|\mathbf{H}_F\|_\infty$ and $\|\mathbf{H}_{F,\mathbf{x}}\|_\infty$ can be polynomially bounded. The latter property is crucial for our purpose.

2.3 KP-ABE Scheme by Boneh et al. [12]

We will use a variant of the KP-ABE scheme proposed by Boneh et al. [12] as a building block of our construction of CP-ABE. We call the scheme BGG^+ and provide the description of the scheme in the following. We focus on the case where the policies associated with secret keys are limited to circuits with logarithmic depth rather than arbitrary polynomially bounded depth, so that we can use the evaluation algorithm due to Gorbunov and Vinayagamurthy [35] (see Lemma 2.8). This allows us to bound the noise growth during the decryption by a polynomial factor, which is crucial for our application.

The scheme supports the circuit class $\mathcal{C}_{\ell(\lambda),d(\lambda)}$, which is a set of all circuits with input length $\ell(\lambda)$ and depth at most $d(\lambda)$ with arbitrary $\ell(\lambda) = \mathrm{poly}(\lambda)$ and $d(\lambda) = O(\log \lambda)$.

$\mathsf{Setup}(1^\lambda)$: On input 1^λ, the setup algorithm defines the parameters $n = n(\lambda)$, $m = m(\lambda)$, noise distribution χ over $\mathbb{Z}$, τ_0, τ, and $B = B(\lambda)$ as specified later. It then proceeds as follows.

1. Sample $(\mathbf{A}, \mathbf{A}_{\tau_0}^{-1}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$ such that $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$.
2. Sample random matrix $\mathbf{B} = (\mathbf{B}_1, \ldots, \mathbf{B}_\ell) \leftarrow (\mathbb{Z}_q^{n\times m})^\ell$ and a random vector $\mathbf{u} \leftarrow \mathbb{Z}_q^n$.
3. Output the master public key $\mathsf{mpk} = (\mathbf{A}, \mathbf{B}, \mathbf{u})$ and the master secret key $\mathsf{msk} = \mathbf{A}_{\tau_0}^{-1}$.

$\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, F)$: The key generation algorithm takes as input the master public key mpk, the master secret key msk, and a circuit $F \in \mathcal{F}_\lambda$ and proceeds as follows.

1. Compute $\mathbf{H}_F = \mathsf{EvalF}(\mathbf{B}, F)$ and $\mathbf{B}_F = \mathbf{B}\mathbf{H}_F$.

[4] This condition may be necessary for the lemma to hold for arbitrary F.

2. Compute $[\mathbf{A} \| \mathbf{B}_F]_\tau^{-1}$ from $\mathbf{A}_{\tau_0}^{-1}$ and sample $\mathbf{r} \in \mathbb{Z}^{2m}$ as $\mathbf{r} \leftarrow [\mathbf{A} \| \mathbf{B}_F]_\tau^{-1}(\mathbf{u})$.
3. Output the secret key $\mathsf{sk}_F := \mathbf{r}$.

$\mathsf{Enc}(\mathsf{mpk}, \mathbf{x}, \mu)$: The encryption algorithm takes as input the master public key mpk, an attribute $\mathbf{x} \in \{0,1\}^\ell$ with $x_1 = 1$,[5] and a message $\mu \in \{0,1\}$ and proceeds as follows.

1. Sample $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $e_1 \leftarrow \chi$, $\mathbf{e}_2 \leftarrow \chi^m$, and $\mathbf{S}_{i,b} \leftarrow \{-1,1\}^{m \times m}$ for $i \in [\ell]$ and $b \in \{0,1\}$. Then, set $\mathbf{e}_{i,b} := \mathbf{S}_{i,b}^\top \mathbf{e}_2$ for $i \in [\ell]$ and $b \in \{0,1\}$.
2. Compute
$$\psi_1 := \mathbf{s}^\top \mathbf{u} + e_1 + \mu \lceil q/2 \rceil \in \mathbb{Z}_q, \quad \psi_2^\top := \mathbf{s}^\top \mathbf{A} + \mathbf{e}_2^\top \in \mathbb{Z}_q^m,$$
$$\psi_{i,b}^\top := \mathbf{s}^\top (\mathbf{B} - x_i \mathbf{G}) + \mathbf{e}_{i,b}^\top \in \mathbb{Z}_q^m \quad \text{for all } i \in [\ell] \text{ and } b \in \{0,1\}.$$
3. Output the ciphertext $\mathsf{ct}_\mathbf{x} := (\psi_1, \psi_2, \{\psi_{i,x_i}\}_{i \in [\ell]})$, where x_i is the i-th bit of $\mathbf{x}$.

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_\mathbf{x}, \mathbf{x}, F, \mathsf{ct}_F)$: The decryption algorithm takes as input the master public key mpk, a secret key sk_F for a circuit F, and a ciphertext $\mathsf{ct}_\mathbf{x}$ for an attribute $\mathbf{x}$ and proceeds as follows.

1. Parse $\mathsf{ct}_\mathbf{x} \to (\psi_1 \in \mathbb{Z}_q, \psi_2 \in \mathbb{Z}_q^m, \{\psi_{i,x_i} \in \mathbb{Z}_q^m\}_{i \in [\ell]})$, and $\mathsf{sk}_F \in \mathbb{Z}^{2m}$. If any of the component is not in the corresponding domain or $F(\mathbf{x}) = 1$, output $\perp$.
2. Concatenate $\{\psi_{i,x_i}\}_{i \in [\ell]}$ to form $\psi_3^\top = (\psi_{1,x_1}^\top, \ldots, \psi_{\ell,x_\ell}^\top)$.
3. Compute $\psi' := \psi_1 - [\psi_2^\top \| \psi_3^\top] \mathbf{r}$.
4. Output 0 if $\psi' \in [-B, B]$ and 1 if $[-B + \lceil q/2 \rceil, B + \lceil q/2 \rceil]$.

Remark 2.9. We note that the encryption algorithm above computes redundant components $\{\psi_{i,\neg x_i}\}_{i \in [\ell]}$ in the second step, which are discarded in the third step. However, due to this redundancy, the scheme has the following special structure that will be useful for us. Namely, the first and the second steps of the encryption algorithm can be executed without knowing $\mathbf{x}$. Only the third step of the encryption algorithm needs the information of $\mathbf{x}$, where it chooses $\{\psi_{i,x_i}\}_{i \in [\ell]}$ from $\{\psi_{i,b}\}_{i \in [\ell], b \in \{0,1\}}$ depending on each bit of $\mathbf{x}$ and then output the former terms along with ψ_1 and ψ_2. Looking ahead, our construction of CP-ABE in Sect. 3 crucially relies on this special structure. There, the encryption algorithm, who takes as input a circuit C that specifies the policy and does not know the corresponding input $\mathbf{x}$, executes the first two steps of the above encryption algorithm. This is possible since these two steps do not need the knowledge of $\mathbf{x}$.

Parameters and Security. We choose the parameters for the scheme as follows:

$$m = n^{1.1} \log q, \qquad q = 2^{\Theta(\lambda)}, \qquad \chi = \mathsf{SampZ}(3\sqrt{n}),$$
$$\tau_0 = n \log q \log m, \qquad \tau = m^{3.1} \ell \cdot 2^{O(d)} \qquad B = n^2 m^2 \tau \cdot 2^{O(d)}.$$

[5] This restriction is required to apply Lemma 2.8. We can remove the condition by increasing the dimension of $\mathbf{x}$ by 1 and considering function F that ignores the first bit.

The parameter n will be chosen depending on whether we need $\mathsf{Sel\text{-}INDr}$ security or $\mathsf{Ada\text{-}INDr}$ security for the scheme. If it suffices to have $\mathsf{Sel\text{-}INDr}$ security, we set $n = \lambda^c$ for some constant $c > 1$. If we need $\mathsf{Ada\text{-}INDr}$ security, we have to enlarge the parameter to be $n = (\ell\lambda)^c$ in order to compensate for the security loss caused by the complexity leveraging.

We remark that if we were to use the above ABE scheme stand-alone, we would have been able to set q polynomially bounded as in [35]. The reason why we set q exponentially large is that we combine the scheme with bilinear maps of order q to lift the ciphertext components to the exponent so that they are "hidden" in some sense (See Sect. 4). In order to use the security of the bilinear map, we set the group order q to be exponentially large.

The following theorem summarizes the security and efficiency properties of the construction. There are two parameter settings depending on whether we assume subexponential hardness of LWE or not.

Theorem 2.10 (Adapted from [12,35]). *Assuming hardness of* $\mathsf{LWE}(n, m, q, \chi)$ *with* $\chi = \mathsf{SampZ}(3\sqrt{n})$ *and* $q = O(2^{n^{1/\epsilon}})$ *for some constant* $\epsilon > 1$*, the above scheme satisfies* $\mathsf{Sel\text{-}INDr}$ *security. Assuming* subexponential *hardness of* $\mathsf{LWE}(n, m, q, \chi)$ *with the same parameters, the above scheme satisfies* $\mathsf{Ada\text{-}INDr}$ *security with respect to the ciphertext space* $\mathcal{CT} := \mathbb{Z}_q^{m(\ell+1)+1}$

2.4 Bilinear Map Preliminaries

Here, we introduce our notation for bilinear maps and the bilinear generic group model following Baltico et al. [9], who specializes the framework by Barthe [10] for defining generic k-linear groups to the bilinear group settings. The definition closely follows that of Maurer [40], which is equivalent to the alternative formulation by Shoup [46].

Notation on Bilinear Maps. A bilinear group generator takes as input 1^λ and outputs a group description $\mathbb{G} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$, where q is a prime of $\Theta(\lambda)$ bits, $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are cyclic groups of order q, $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is a non-degenerate bilinear map, and g_1 and g_2 are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. We require that the group operations in $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ as well as the bilinear map e can be efficiently computed. We employ the implicit representation of group elements: for a matrix $\mathbf{A}$ over $\mathbb{Z}_q$, we define $[\mathbf{A}]_1 := g_1^{\mathbf{A}}$, $[\mathbf{A}]_2 := g_2^{\mathbf{A}}$, $[\mathbf{A}]_T := g_T^{\mathbf{A}}$, where exponentiation is carried out component-wise.

We also use the following less standard notations. For vectors $\mathbf{w} = (w_1, \ldots, w_\ell)^\top \in \mathbb{Z}_q^\ell$ and $\mathbf{v} = (w_1, \ldots, w_\ell)^\top \in \mathbb{Z}_q^\ell$ of the same length, $\mathbf{w} \odot \mathbf{v}$ denotes the vector that is obtained by component-wise multiplications. Namely, $\mathbf{v} \odot \mathbf{w} = (v_1 w_1, \ldots, v_\ell w_\ell)^\top$. When $\mathbf{w} \in (\mathbb{Z}_q^*)^\ell$, $\mathbf{v} \oslash \mathbf{w}$ denotes the vector $\mathbf{v} \oslash \mathbf{w} = (v_1/w_1, \ldots, v_\ell/w_\ell)^\top$. It is easy to verify that for vectors $\mathbf{c}, \mathbf{d} \in \mathbb{Z}_q^\ell$

and $\mathbf{w} \in (\mathbb{Z}_q^*)^\ell$, we have $(\mathbf{c} \odot \mathbf{w}) \odot (\mathbf{d} \oslash \mathbf{w}) = \mathbf{c} \odot \mathbf{d}$. For group elements $[\mathbf{v}]_1 \in \mathbb{G}_1^\ell$ and $[\mathbf{w}]_1 \in \mathbb{G}_2^\ell$, $[\mathbf{v}]_1 \odot [\mathbf{w}]_2$ denotes $([v_1 w_1]_T, \ldots, [v_\ell w_\ell]_T)^\top$, which is efficiently computable from $[\mathbf{v}]_1$ and $[\mathbf{w}]_2$ using the bilinear map e.

Generic Bilinear Group Model. Let $\mathbb{G} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$ be a bilinear group setting, L_1, L_2, and L_T be lists of group elements in $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ respectively, and let $\mathcal{D}$ be a distribution over L_1, L_2, and L_T. The generic group model for a bilinear group setting $\mathbb{G}$ and a distribution $\mathcal{D}$ is described in Fig. 1. In this model, the challenger first initializes the lists L_1, L_2, and L_T by sampling the group elements according to $\mathcal{D}$, and the adversary receives handles for the elements in the lists. For $s \in \{1, 2, T\}$, $L_s[h]$ denotes the h-th element in the list L_s. The handle to this element is simply the pair (s, h). An adversary running in the generic bilinear group model can apply group operations and bilinear maps to the elements in the lists. To do this, the adversary has to call the appropriate oracle specifying handles for the input elements. The challenger computes the result of a query, stores it in the corresponding list, and returns to the adversary its (newly created) handle. Handles are not unique (i.e., the same group element may appear more than once in a list under different handles).

We remark that we slightly simplify the definition of the generic group model by Baltico et al. [9]. Whereas they allow the adversary to access the equality test oracle, which is given two handles (s, h_1) and (s, h_2) and returns 1 if $L_s[h_1] = L_s[h_2]$ and 0 otherwise for all $s \in \{1, 2, T\}$, we replace this oracle with the zero-test oracle, which is given a handle (s, h) and returns 1 if $L_s[h] = 0$ and 0 otherwise only for the case of $s = T$. We claim that even with this modification, the model is equivalent to the original one. This is because we can perform the equality test for (s, h_1) and (s, h_2) using our restricted oracles as follows. Let us first consider the case of $s = T$. In this case, we can get the handle (T, h') corresponding to $L_T[h_1] - L_T[h_2]$ by calling neg_T and add_T. We then make a zero-test query for (T, h'). Clearly, we get 1 if $L_s[h_1] = L_s[h_2]$ and 0 otherwise. We next consider the case of $s \in \{1, 2\}$. This case can be reduced to the case of $s = T$ by lifting the group elements corresponding to h_1 and h_2 to the group elements in $\mathbb{G}_T$ by taking bilinear maps with an arbitrary non-unit group element in $\mathbb{G}_{3-s}$, which is possible by calling map_e.

Symbolic Group Model. The symbolic group model for a bilinear group setting $\mathbb{G}$ and a distribution $\mathcal{D}_P$ gives to the adversary the same interface as the corresponding generic group model, except that internally the challenger stores lists of element in the field $\mathbb{Z}_p(X_1, \ldots, X_n)$ instead of lists of group elements. The oracles add_s, neg_s, map, and zt computes addition, negation, multiplication, and equality in the field. In our work, we will use the subring $\mathbb{Z}_p[X_1, \ldots, X_n, 1/X_1, \ldots, 1/X_n]$ of the entire field $\mathbb{Z}_p(X_1, \ldots, X_n)$. Note that any element f in $\mathbb{Z}_p[X_1, \ldots, X_n, 1/X_1, \ldots, 1/X_n]$ can be represented as $f(X_1, \ldots, X_n) = \sum_{(c_1, \ldots, c_n) \in \mathbb{Z}^n} a_{c_1, \ldots, c_n} X_1^{c_1} \cdots X_n^{c_n}$ using $\{a_{c_1, \ldots, c_n} \in \mathbb{Z}_p\}_{(c_1, \ldots, c_n) \in \mathbb{Z}^n}$, where we have $a_{c_1, \ldots, c_n} = 0$ for all but finite $(c_1, \ldots, c_n) \in \mathbb{Z}^n$. Note that this expression is unique.

State: Lists L_1, L_2, L_T over $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ respectively.
Initializations: Lists L_1, L_2, L_T sampled according to distribution $\mathcal{D}$.
Oracles: The oracles provide black-box access to the group operations, the bilinear map, and equalities.

- For all $s \in \{1, 2, T\}$: $\mathsf{add}_s(h_1, h_2)$ appends $L_s[h_1] + L_s[h_2]$ to L_s and returns its handle $(s, |L_s|)$.
- For all $s \in \{1, 2, T\}$: $\mathsf{neg}_s(h_1, h_2)$ appends $-L_s[h_1]$ to L_s and returns its handle $(s, |L_s|)$.
- $\mathsf{map}_e(h_1, h_2)$ appends $e(L_1[h_1], L_2[h_2])$ to L_T and returns its handle $(T, |L_T|)$.
- $\mathsf{zt}_T(h)$ returns 1 if $L_T[h] = 0$ and 0 otherwise.

All oracles return $\perp$ when given invalid indices.

Fig. 1. Generic group model for bilinear group setting $\mathbb{G} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$ and distribution $\mathcal{D}$.

3 Our Construction of CP-ABE

Here, we describe our new construction of CP-ABE scheme. Our construction can deal with any circuit class $\mathcal{F} = \{\mathcal{F}_\lambda\}_\lambda$ that is subclass of $\{\mathcal{C}_{\ell(\lambda),d(\lambda)}\}_\lambda$ with arbitrary $\ell(\lambda) \leq \mathrm{poly}(\lambda)$ and $d(\lambda) = O(\log \lambda)$, where $\mathcal{C}_{\ell(\lambda),d(\lambda)}$ is a set of circuits with input length $\ell(\lambda)$ and depth at most $d(\lambda)$. As we will see in Sect. 5, we can obtain new constructions of BE, IBBE, CP-ABE by setting the circuit class $\mathcal{F}$ appropriately. In order to get the scheme, we use the KP-ABE scheme BGG^+ for the circuit class $\mathcal{F} = \{\mathcal{F}_\lambda\}_\lambda$ that is described in Sect. 2.3 as an ingredient. Our construction below can be seen as a conversion from an ABE scheme to another ABE scheme with dual predicate.

$\mathsf{Setup}(1^\lambda)$: On input 1^λ, the setup algorithm defines the parameters $n = n(\lambda)$, $m = m(\lambda)$, noise distribution χ over $\mathbb{Z}$, τ_0, τ, and $B = B(\lambda)$ as specified in Sect. 2.3. It samples a group description $\mathbb{G} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, [1]_1, [1]_2)$. It then sets $L := (2\ell + 1)m + 2$ and proceeds as follows.

1. Sample $\mathbf{w} \leftarrow (\mathbb{Z}_q^*)^L$ and compute $[\mathbf{w}]_1$.
2. Output $\mathsf{mpk} = ([1]_1, [1]_2, [\mathbf{w}]_1)$ and $\mathsf{msk} = \mathbf{w}$.

$\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathbf{x})$: The key generation algorithm takes as input the master public key mpk, the master secret key msk, and an attribute $\mathbf{x} \in \{0,1\}^\ell$ with $x_1 = 1$ and proceeds as follows.

1. Let $\mathbf{1} := (1, \ldots, 1)^\top \in \mathbb{Z}_q^m$ and $\mathbf{0} := (0, \ldots, 0)^\top \in \mathbb{Z}_q^m$. Set

$$\phi_0 = 1 \in \mathbb{Z}_q, \quad \phi_1 = 1 \in \mathbb{Z}_q, \quad \phi_2 := \mathbf{1} \in \mathbb{Z}_q^m,$$

$$\phi_{i,b} := \begin{cases} \mathbf{1} \in \mathbb{Z}_q^m & \text{if } b = x_i \\ \mathbf{0} \in \mathbb{Z}_q^m & \text{if } b \neq x_i \end{cases} \quad \text{for } i \in [\ell] \text{ and } b \in \{0,1\}. \tag{3.1}$$

2. Vectorize $(\phi_0, \phi_1, \phi_2, \{\phi_{i,b}\}_{i,b})$ to form a vector $\mathbf{d} \in \mathbb{Z}_q^L$ by concatenating each entry of the vectors in a predetermined order.

3. Sample $\delta \leftarrow \mathbb{Z}_q^*$.
4. Compute $[\delta \mathbf{d} \oslash \mathbf{w}]_2 \in \mathbb{G}_2^L$ from $\mathsf{msk} = \mathbf{w}$ in msk.
5. Output $\mathsf{sk}_{\mathbf{x}} = [\delta \mathbf{d} \oslash \mathbf{w}]_2$.

$\mathsf{Enc}(\mathsf{mpk}, F, \mu)$: The encryption algorithm takes as input the master public key mpk, the circuit F, and a message $\mu \in \{0,1\}$ and proceeds as follows.

1. Sample fresh BGG^+ scheme:
   (a) Sample $(\mathbf{A}, \mathbf{A}_{\tau_0}^{-1}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$ such that $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$.
   (b) Sample random matrix $\mathbf{B} = (\mathbf{B}_1, \ldots, \mathbf{B}_\ell) \leftarrow (\mathbb{Z}_q^{n \times m})^\ell$ and a random vector $\mathbf{u} \leftarrow \mathbb{Z}_q^n$.
2. Compute BGG^+ function key for circuit F:
   (a) Compute $\mathbf{H}_F = \mathsf{EvalF}(\mathbf{B}, F)$ and $\mathbf{B}_F = \mathbf{B}\mathbf{H}_F$.
   (b) Compute $[\mathbf{A} \| \mathbf{B}_F]_\tau^{-1}$ from $\mathbf{A}_{\tau_0}^{-1}$ and sample $\mathbf{r} \in \mathbb{Z}^{2m}$ as $\mathbf{r} \leftarrow [\mathbf{A} \| \mathbf{B}_F]_\tau^{-1}(\mathbf{u})$.
3. Compute BGG^+ ciphertext for all possible inputs:
   (a) Sample $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $e_1 \leftarrow \chi$, $\mathbf{e}_2 \leftarrow \chi^m$, and $\mathbf{S}_{i,b} \leftarrow \{-1,1\}^{m \times m}$ for $i \in [\ell]$ and $b \in \{0,1\}$. Then, set $\mathbf{e}_{i,b} := \mathbf{S}_{i,b}^\top \mathbf{e}_2$ for $i \in [\ell]$ and $b \in \{0,1\}$.
   (b) Compute
$$\begin{aligned} \psi_0 &:= 1 \in \mathbb{Z}_q, \quad \psi_1 := \mathbf{s}^\top \mathbf{u} + e_1 + \mu \lceil q/2 \rceil \in \mathbb{Z}_q, \\ \psi_2^\top &:= \mathbf{s}^\top \mathbf{A} + \mathbf{e}_2^\top \in \mathbb{Z}_q^m, \\ \psi_{i,b}^\top &:= \mathbf{s}^\top (\mathbf{B}_i - b\mathbf{G}) + \mathbf{e}_{i,b}^\top \in \mathbb{Z}_q^m \quad \text{for } i \in [\ell] \text{ and } b \in \{0,1\}. \end{aligned} \tag{3.2}$$
4. Encode BGG^+ ciphertexts in exponent of bilinear group:
   (a) Vectorize $(\psi_0, \psi_1, \psi_2, \{\psi_{i,b}\}_{i,b})$ to form a vector $\mathbf{c} \in \mathbb{Z}_q^L$ by concatenating each entry of the vectors in a predetermined order (that aligns with the one used in the key generation algorithm).
   (b) Sample $\gamma \leftarrow \mathbb{Z}_q^*$.
   (c) Compute $[\gamma \mathbf{c} \odot \mathbf{w}]_1 \in \mathbb{G}_2^L$ from γ, $\mathbf{c}$, and $[\mathbf{w}]_1$ in mpk.
5. Output $\mathsf{ct}_F = (\mathsf{ct}_0 = (\mathbf{A}, \mathbf{B}), \mathsf{ct}_1 = [\gamma \mathbf{c} \odot \mathbf{w}]_1, \mathsf{ct}_2 = \mathbf{r})$.

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_{\mathbf{x}}, \mathbf{x}, F, \mathsf{ct}_F)$: The decryption algorithm takes as input the master public key mpk, the secret key $\mathsf{sk}_{\mathbf{x}}$ for an attribute $\mathbf{x}$, and the ciphertext ct_F for a circuit F and proceeds as follows.

1. Parse $\mathsf{ct}_F \to (\mathsf{ct}_0 = (\mathbf{A} \in \mathbb{Z}_q^{n \times m}, \mathbf{B} \in \mathbb{Z}_q^{n \times m\ell}), \mathsf{ct}_1 \in \mathbb{G}_1^L, \mathsf{ct}_2 \in \mathbb{Z}^{2m})$ and $\mathsf{sk}_{\mathbf{x}} \in \mathbb{G}_2^L$. If any of the component is not in the corresponding domain or $F(\mathbf{x}) = 1$, output $\perp$.
2. Unmask BGG^+ ciphertexts corresponding to $\mathbf{x}$ by using secret key: Compute $[\mathbf{v}]_T := \mathsf{ct}_1 \odot \mathsf{sk}_{\mathbf{x}}$ and de-vectorize $[\mathbf{v}]_T$ to obtain
$$[v_0]_T \in \mathbb{G}_T, \ [v_1]_T \in \mathbb{G}_T, \ [\mathbf{v}_2]_T \in \mathbb{G}_T^m, \ [\mathbf{v}_{i,b}]_T \in \mathbb{G}_T^m, \quad \text{for } i \in [\ell], b \in \{0,1\}.$$
3. Evaluate circuit F on BGG^+ ciphertexts in the exponent: Compute $\widehat{\mathbf{H}}_{F,\mathbf{x}} = \mathsf{EvalF}(F, \mathbf{x}, \mathbf{B})$.

4. Perform BGG^+ decryption in the exponent:
Form $[\mathbf{v}_{\mathbf{x}}^\top]_T = [\mathbf{v}_{1,x_1}^\top, \ldots, \mathbf{v}_{\ell,x_\ell}^\top]_T$ and $\mathsf{ct}_2^\top = (\mathbf{r}_1^\top \in \mathbb{Z}_q^m, \mathbf{r}_2^\top \in \mathbb{Z}_q^m)$. Then compute
$$[v']_T := [v_1 - (\mathbf{v}_2^\top \mathbf{r}_1 + \mathbf{v}_{\mathbf{x}}^\top \widehat{\mathbf{H}}_{F,\mathbf{x}} \mathbf{r}_2)]_T$$
from $[v_1]_T$, $[\mathbf{v}_2]_T$, $[\mathbf{v}_{\mathbf{x}}]_T$, $\mathbf{r}_1$, $\mathbf{r}_2$, and $\widehat{\mathbf{H}}_{F,\mathbf{x}}$.
5. Recover exponent via brute force if $F(\mathbf{x}) = 0$:
Find $\eta \in [-B, B] \cup [-B + \lceil q/2 \rceil, B + \lceil q/2 \rceil]$ such that $[v_0]_T^\eta = [v']_T$ by brute-force search. If there is no such η, output $\perp$. To speed up the operation, one can employ the baby-step giant-step algorithm.
6. Output 0 if $\eta \in [-B, B]$ and 1 if $[-B + \lceil q/2 \rceil, B + \lceil q/2 \rceil]$.

Correctness. To see correctness of the scheme, we first observe that we have $\mathsf{ct}_1 \odot \mathsf{sk}_{\mathbf{x}} = [\gamma\delta \cdot \mathbf{c} \odot \mathbf{d}]_T$ and thus

$$v_0 = \gamma\delta, \quad v_1 = \gamma\delta \left(\mathbf{s}^\intercal \mathbf{u} + e_1 + \mu \lceil q/2 \rceil\right), \quad \mathbf{v}_2^\top = \gamma\delta \left(\mathbf{s}^\top \mathbf{A} + \mathbf{e}_2^\top\right),$$
$$\mathbf{v}_{i,b}^\top = \begin{cases} \gamma\delta \left(\mathbf{s}^\top (\mathbf{B}_i - x_i \mathbf{G}) + \mathbf{e}_{i,x_i}^\top\right) & \text{if } b = x_i \\ \mathbf{0} & \text{if } b = 1 - x_i \end{cases}.$$

From the above, we have $\mathbf{v}_{\mathbf{x}}^\top = \mathbf{s}^\top (\mathbf{B} - \mathbf{x} \otimes \mathbf{G}) + \mathbf{e}_{\mathbf{x}}^\top$ for $\mathbf{e}_{\mathbf{x}}^\top := (\mathbf{e}_{1,x_1}^\top, \cdots, \mathbf{e}_{\ell,x_\ell}^\top)$. We then have

$$\begin{aligned} \mathbf{v}_2^\top \mathbf{r}_1 + \mathbf{v}_{\mathbf{x}}^\top \widehat{\mathbf{H}}_{F,\mathbf{x}} \mathbf{r}_2 &= \gamma\delta \left(\mathbf{s}^\top \mathbf{A} + \mathbf{e}_2^\top\right) \mathbf{r}_1 + \gamma\delta \left(\mathbf{s}^\top (\mathbf{B} - \mathbf{x} \otimes \mathbf{G}) + \mathbf{e}_{\mathbf{x}}^\top\right) \widehat{\mathbf{H}}_{F,\mathbf{x}} \mathbf{r}_2 \\ &= \gamma\delta \left(\mathbf{s}^\top (\mathbf{A}\mathbf{r}_1 + \mathbf{B}_F \mathbf{r}_2) + \mathbf{e}_2^\top \mathbf{r}_1 + \mathbf{e}_{\mathbf{x}}^\top \widehat{\mathbf{H}}_{F,\mathbf{x}} \mathbf{r}_2\right) \\ &= \gamma\delta \left(\mathbf{s}^\top \mathbf{u} + \mathbf{e}_2^\top \mathbf{r}_1 + \mathbf{e}_{\mathbf{x}}^\top \widehat{\mathbf{H}}_{F,\mathbf{x}} \mathbf{r}_2\right) \end{aligned}$$

where the second equation follows from $(\mathbf{B} - \mathbf{x} \otimes \mathbf{G}) \widehat{\mathbf{H}}_{F,\mathbf{x}} = \mathbf{B}_F$ and the third equation follows form $[\mathbf{A} \| \mathbf{B}_F] \mathbf{r} = \mathbf{u}$. This implies

$$v' = \gamma\delta \left(\mu \lceil q/2 \rceil + e_1 - \mathbf{e}_2^\top \mathbf{r}_1 - \mathbf{e}_{\mathbf{x}}^\top \widehat{\mathbf{H}}_{F,\mathbf{x}} \mathbf{r}_2\right).$$

Recall that we set $\chi = \mathsf{SampZ}(3\sqrt{n})$. By the definition of SampZ, we have $\|e_1\|_\infty \le 3n$ and $\|\mathbf{e}_2\|_\infty \le 3n$. Furthermore, we have $\|\mathbf{e}_{i,b}\|_\infty = \|\mathbf{S}_{i,b}^\top \mathbf{e}_2\|_\infty \le 3mn$ for $i \in [\ell]$ and $b \in \{0,1\}$, $\|\mathbf{r}\|_\infty \le \sqrt{n}\tau$, and $\|\widehat{\mathbf{H}}_{F,\mathbf{x}}\|_\infty \le m \cdot 2^{O(d)}$, where the last inequality follows from Lemma 2.8. Thus, we have

$$\|e_1 - \mathbf{e}_2^\top \mathbf{r}_1 - \mathbf{e}_{\mathbf{x}}^\top \widehat{\mathbf{H}}_{F,\mathbf{x}} \mathbf{r}_2\|_\infty \le O(n^{1.5} m^2 \tau \cdot 2^{O(d)}) \le B$$

by our choice of B. The correctness therefore follows. Note that since $B = \mathrm{poly}(n, \ell) \cdot 2^{O(d)} = \mathrm{poly}(\lambda)$, the decryption algorithm runs in polynomial time.

Efficiency of the Scheme. Here, we evaluate the efficiency of the above scheme. In particular, we measure the sizes of the parameters. The master public key of the scheme consists of $L+2$ group elements. Since $L = O(m\ell)$, we have that the master public key can be represented by a binary string of length $\ell \cdot \mathrm{poly}(\lambda)$. Next, we observe that a secret key in the scheme consists of L group elements, which can be represented by a binary string of length $\ell \cdot \mathrm{poly}(\lambda)$. Finally, a ciphertext in the scheme consists of $O(nm)$ elements of $\mathbb{Z}_q$ and L group elements. The former elements are represented by a binary string of length $\mathrm{poly}(\lambda)$ if we only need $\mathsf{Sel\text{-}INDr}$ security for the underlying KP-ABE scheme. If we need $\mathsf{Ada\text{-}INDr}$ security, the length of the binary string is $\mathrm{poly}(\ell, \lambda)$. Therefore, the length of the whole ciphertext is $\ell \cdot \mathrm{poly}(\lambda)$ if we only need $\mathsf{Sel\text{-}INDr}$ security for the underlying KP-ABE scheme and $\mathrm{poly}(\ell, \lambda)$ if we need $\mathsf{Ada\text{-}INDr}$ security. In any case, the sizes of all parameters in the system are independent of the size of the circuits being supported by the scheme, which is a notable feature of the scheme.

4 Security Proof for Our CP-ABE

This section is devoted to prove the following theorem that asserts the security of our CP-ABE scheme in Sect. 3.

Theorem 4.1. *Our CP-ABE scheme for function class $\mathcal{F}$ satisfies* $\mathsf{Ada\text{-}IND}$ *security in the generic group model assuming that the KP-ABE* BGG^+ *for function class $\mathcal{F}$ satisfies* $\mathsf{Ada\text{-}INDr}$ *security.*

Overview of the Proof. Before going to the formal proof, we give its overview. The proof is done by considering a sequence of games and consists of two parts. In the first part of the proof, which is captured by a series of game hops from $\mathbf{Game}_0$ through $\mathbf{Game}_5$ defined below, we prove that it is pointless for the adversary to take pairing products between unmatching positions of the ciphertext and secret key components and then take linear combinations among them. Therefore, the only possible strategy for the adversary is to take linear combination among "partial decryption results" obtained by taking pairing products between matching positions of the ciphertext and secret key components and infer information of the message being encrypted. In the second step of the proof, which is captured by the game hop from $\mathbf{Game}_5$ to $\mathbf{Game}_6$, we show that this type of attack does not work either. To do so, we further consider a sequence of subgames from $\mathbf{Game}_{5,0}$ through $\mathbf{Game}_{5,8}$. We first prove that taking linear combinations among partial decryption results from different secret keys is useless. This is the key step that excludes the collusion attack and is captured by the game hop from $\mathbf{Game}_{5.3}$ to $\mathbf{Game}_{5.4}$. At this point, the only strategy for the adversary is to take linear combination among partial decryption result obtained by single secret key. Finally, in the step from $\mathbf{Game}_{5.7}$ to $\mathbf{Game}_{5.8}$, we use the security of the BGG^+ ABE to conclude that this strategy does not work either. To invoke the security of BGG^+ ABE, we use the fact that the partial decryption result obtained by secret key for x forms randomized version of BGG^+ ABE ciphertext for attribute x in the exponent.

Proof. To prove the theorem, we fix a PPT adversary A that makes at most $Q_{\mathsf{kq}}(\lambda)$ key queries and $Q_{\mathsf{zt}}(\lambda)$ zero-test queries during the game. Furthermore, we assume that A always chooses $(\mu_0, \mu_1) = (0, 1)$ as its target message at the challenge phase. This can be assumed without loss of generality since our scheme is a single-bit scheme. In order to prove the security, we consider following sequence of games. Let us denote the event that A outputs correct guess for b at the end of $\mathbf{Game}_{\mathrm{x}}$ as E_{x}.

$\mathbf{Game}_0$: This is the real game in the generic group model. To fix the notation and for the sake of concreteness, we briefly describe the game here. Without loss of generality, we assume that the challenger simulates the generic group oracle for A. At the beginning of the game, the challenger picks $\mathbf{w} \leftarrow (\mathbb{Z}_q^*)^L$ and sets the master public key $\mathsf{mpk} = ([1]_1, [1]_2, [\mathbf{w}]_1)$ and the master secret key $\mathsf{msk} = \mathbf{w}$. Then, it gives handles to the group elements in mpk to A. To respond to the j-th key query $\mathbf{x}^{(j)}$ made by A, the challenger samples $\delta_j \leftarrow \mathbb{Z}_q^*$, sets $\mathbf{d}^{(j)} \in \mathbb{Z}_q^L$ as specified in the key generation algorithm, and sets $\mathsf{sk}^{(j)} = [\delta_j \mathbf{d}^{(j)} \oslash \mathbf{w}]_2$. It then gives the handles corresponding to the group elements in $\mathsf{sk}^{(j)}$ to A. To answer the challenge query for a circuit F, the challenger first picks the message $b \leftarrow \{0, 1\}$ to be encrypted, chooses $\gamma \leftarrow \mathbb{Z}_q^*$, computes $\mathbf{A}$, $\mathbf{B}$, $\mathbf{r}$, $\mathbf{c}$ as specified in the encryption algorithm (where b is encrypted), and forms the challenge ciphertext as $\mathsf{ct}_F = (\mathsf{ct}_0 = (\mathbf{A}, \mathbf{B}), \mathsf{ct}_1 = [\gamma \mathbf{c} \odot \mathbf{w}]_1, \mathsf{ct}_2 = \mathbf{r})$. It then returns $\mathsf{ct}_0 = (\mathbf{A}, \mathbf{B})$, handles to $\mathsf{ct}_1 = [\gamma \mathbf{c} \odot \mathbf{w}]_1$, and ct_2 to A. By definition, the advantage of A against the scheme is $\left|\Pr[\mathsf{E}_0] - \frac{1}{2}\right|$.

$\mathbf{Game}_1$: This game is the same as the previous game except that the challenger samples $\mathbf{w} = (w_1, \ldots, w_L)^\top$, $\delta_1, \ldots, \delta_{Q_{\mathsf{kq}}}$, $\mathbf{A}$, $\mathbf{B}$, $\mathbf{u}$, γ, b, and $\mathbf{c} = (c_1, \ldots, c_L)^\top$ at the beginning of the game. Note that $\mathbf{c}$ is sampled from the distribution that is only dependent on the bit b being encrypted, and is *independent* of the circuit F that is specified by A later in the game. Therefore, this game is well-defined. As we prove in Lemma 4.2, we have $\Pr[\mathsf{E}_0] = \Pr[\mathsf{E}_1]$.

$\mathbf{Game}_2$: In this game, we (partially) switch to the symbolic group model and replace $\{w_i\}_{i \in [L]}$, $\{\delta_j\}_{j \in [Q_{\mathsf{kq}}]}$, γ, and $\{c_i\}_{i \in [L]}$ in $\mathbb{Z}_q$ with the formal variables $\{W_i\}_{i \in [L]}$, $\{\Delta_j\}_{j \in [Q_{\mathsf{kq}}]}$, Γ, and $\{C_i\}_{i \in [L]}$ respectively. As a result, all handles given to A refer to elements in the ring

$$\mathbb{T} := \mathbb{Z}_q[W_1, \ldots, W_L, 1/W_1, \ldots, 1/W_L, \Delta_1, \ldots, \Delta_{Q_{\mathsf{kq}}}, \Gamma, C_1, \ldots, C_L],$$

where $\{1/W_i\}_i$ are needed to represent the components in the secret keys. However, when the challenger answers the zero-test queries, it substitutes the formal variables with corresponding elements in $\mathbb{Z}_q$. Namely, in this game, the challenger picks $\{w_i\}_i$, $\{\delta_j\}_j$, γ, and $\{c_i\}_i$ at the beginning of the game as specified in the previous game and when A makes a zero-test query for a handle corresponding to $f(W_1, \ldots, W_L, \Delta_1, \ldots, \Delta_{Q_{\mathsf{kq}}}, \Gamma, C_1, \ldots, C_L) \in \mathbb{T}$, the challenger returns 1 if

$$f(w_1, \ldots, w_L, \delta_1, \ldots, \delta_{Q_{\mathsf{kq}}}, \gamma, c_1, \ldots, c_L) = 0$$

holds over $\mathbb{Z}_q$ and 0 otherwise. As we prove in Lemma 4.3, we have $\Pr[\mathsf{E}_1] = \Pr[\mathsf{E}_2]$.

Here, we list all the components in $\mathbb{T}$ for which corresponding handles are given to A in **Game**$_2$ as either handles to the group elements in mpk, the challenge ciphertext, or secret keys:

$$S_1 := \left\{1,\ W_i,\ \{C_i \Gamma W_i\}_{i\in[L]}\right\},\quad S_2 := \left\{1,\ \{d_i^{(j)} \Delta_j / W_i\}_{i\in[L], j\in[Q_{\mathsf{kq}}]}\right\}$$

where $d_i^{(j)} \in \{0,1\}$ is the i-th entry of $\mathbf{d}^{(j)}$. Note that S_1 and S_2 correspond to handles for elements in $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. We then define S_T as $S_T := \{X \cdot Y : X \in S_1, Y \in S_2, X \cdot Y \neq 0\}$. If we explicitly write down S_T, we have $S_T = S_{T,1} \cup S_{T,2}$ where

$$S_{T,1} := \left\{\begin{array}{ll} 1, & \\ W_i,\ C_i \Gamma W_i, & \text{for } i \in [L], \\ \Delta_j, & \text{for } j \in [Q_{\mathsf{kq}}], \\ \Delta_j / W_i, & \text{for } i \in [L], j \in [Q_{\mathsf{kq}}] \text{ such that } d_i^{(j)} = 1, \\ \Delta_j W_{i'} / W_i, & \text{for } i, i' \in [L], j \in [Q_{\mathsf{kq}}] \text{ such that } i \neq i' \text{ and } d_i^{(j)} = 1 \\ C_{i'} \Gamma \Delta_j W_{i'} / W_i & \text{for } i, i' \in [L], j \in [Q_{\mathsf{kq}}] \text{ such that } i \neq i' \text{ and } d_i^{(j)} = 1 \end{array}\right\}$$

and $S_{T,2} = \{\, C_i \Gamma \Delta_j \text{ for } i \in [L], j \in [Q_{\mathsf{kq}}] \text{ such that } d_i^{(j)} = 1 \,\}$. Here, $S_{T,2}$ consists of terms that are obtained by taking product between matching positions of the ciphertext and secret keys, whereas $S_{T,1}$ consists of terms that are obtained by taking product between unmatching positions of the ciphertext and secret keys or between master public key and the ciphertext or secret keys. Note that any handle submitted to the zero-test oracle by A during the game refers to an element f in $\mathbb{T}$ that can be represented as

$$f(W_1, \ldots, W_L, \Delta_1, \ldots, \Delta_{Q_{\mathsf{kq}}}, \Gamma, C_1, \ldots, C_L) = \sum_{Z \in S_T} a_Z Z \tag{4.1}$$

where the coefficients $\{a_Z \in \mathbb{Z}_q\}_{Z \in S_T}$ can be efficiently computed. Furthermore, $\{a_Z \in \mathbb{Z}_q\}_{Z \in S_T}$ satisfying the above equation is unique since all monomials in S_T are distinct.

Game$_3$: In this game, we change the game so that $\{W_i\}_{i\in[L]}$, $\{\Delta_j\}_{j\in[Q_{\mathsf{kq}}]}, \Gamma$ are treated as formal variables rather than elements in $\mathbb{Z}_q$ even when answering zero-test queries. Namely, the challenger no longer samples $\{w_i\}_{i\in[L]}$, $\{\delta_j\}_{j\in[Q_{\mathsf{kq}}]}$, and γ at the beginning of the game and when A makes a zero-test query for a handle corresponding to $f(W_1, \ldots, W_L, \Delta_1, \ldots, \Delta_{Q_{\mathsf{kq}}}, \Gamma, C_1, \ldots, C_L) \in \mathbb{T}$, the challenger returns 1 if

$$f(W_1, \ldots, W_L, \Delta_1, \ldots, \Delta_{Q_{\mathsf{kq}}}, \Gamma, c_1, \ldots, c_L) = 0 \tag{4.2}$$

holds over $\mathbb{T}$ and 0 otherwise, where $\{c_i\}_{i\in[L]}$ are sampled at the beginning of the game as specified in the previous game. As we prove in Lemma 4.4, we have $|\Pr[\mathsf{E}_2] - \Pr[\mathsf{E}_3]| \leq Q_{\mathsf{zt}}(L+3)^2/q$.

Game$_4$: This game is the same as the previous game except that the challenger aborts the game and enforces the adversary to output a random bit when there exists $i \in [L]$ such that $c_i = 0$, where $\mathbf{c} = (c_1, \ldots, c_L)^\top$ is sampled as in the previous game. As we prove in Lemma 4.5, we have $|\Pr[\mathsf{E}_3] - \Pr[\mathsf{E}_4]| \le L/q$.

Game$_5$: In this game, we further change the way zero-test queries are answered. In particular, when A makes a zero-test query for a handle corresponding to $f \in \mathbb{T}$ that can be represented as

$$f(W_1, \ldots, W_L, \varDelta_1, \ldots, \varDelta_{Q_{\mathsf{kq}}}, \varGamma, C_1, \ldots, C_L) = \sum_{Z \in S_{T,1}} a_Z Z + \sum_{Z \in S_{T,2}} a_Z Z, \tag{4.3}$$

the challenger returns 0 if there exists $Z \in S_{T,1}$ such that $a_Z \neq 0$. Otherwise, the challenger answers the query as in the previous game. As we prove in Lemma 4.6, we have $\Pr[\mathsf{E}_4] = \Pr[\mathsf{E}_5]$.

Game$_6$: In this game, we change the game so that zero-test queries are performed over the ring $\mathbb{T}$. Namely, when A makes a zero-test query for a handle corresponding to $f \in \mathbb{T}$ the challenger returns 0 if $f \neq 0$ over $\mathbb{T}$. Equivalently, the challenger returns 0 if there exists $Z \in S_T$ such that $a_Z = 0$ when A makes a zero-test query for a handle corresponding to $f \in \mathbb{T}$ that is represented as Eq. (4.1). Note that $(c_1, \ldots, c_L)$ is not used in this game and the challenger does not have to sample it any more. As we prove in Lemma 4.7, there exists a PPT adversary B such that $|\Pr[\mathsf{E}_5] - \Pr[\mathsf{E}_6]| \le Q_{\mathsf{kq}} Q_{\mathsf{zt}} \cdot (\mathsf{Adv}^{\mathsf{Ada\text{-}INDr}}_{\mathsf{BGG}^+, \mathsf{B}}(1^\lambda) + 1/q)$.

We can see that the adversary cannot obtain any information about the encrypted message b in **Game**$_6$ since the challenge ciphertext is replaced by formal variables $(C_1, \ldots, C_L)$ that does not contain any information of b and the answers to the zero test queries do not depend on b neither. Therefore, we have $\Pr[\mathsf{E}_6] = 1/2$. Thus, there exists a PPT adversary B against $\mathsf{Ada\text{-}INDr}$ security of BGG^+ such that

$$\left| \Pr[\mathsf{E}_0] - \frac{1}{2} \right| \le Q_{\mathsf{kq}} Q_{\mathsf{zt}} \cdot \left(\mathsf{Adv}^{\mathsf{Ada\text{-}INDr}}_{\mathsf{BGG}^+, \mathsf{B}}(1^\lambda) + \frac{1}{q} \right) + \frac{Q_{\mathsf{zt}}(L+3)^2 + L}{q}.$$

In particular, assuming BGG^+ satisfies $\mathsf{Ada\text{-}INDr}$ security, the above quantity is negligible as desired.

To finish the proof of Theorem 4.1, it remains to prove Lemmas 4.2, 4.3, 4.4, 4.5, 4.6, and 4.7 in the following.

Lemma 4.2 (Game$_\mathbf{0}$ ≡ Game$_\mathbf{1}$). *We have* $\Pr[\mathsf{E}_0] = \Pr[\mathsf{E}_1]$.

Proof. Since this is only a conceptual change, the lemma immediately follows.

Lemma 4.3 (Game$_\mathbf{1}$ ≡ Game$_\mathbf{2}$). *We have* $\Pr[\mathsf{E}_1] = \Pr[\mathsf{E}_2]$.

Proof. Since zero-test queries in **Game**$_2$ are answered by using $\{w_i\}_i$, $\{\delta_j\}_j$, γ, and $\{c_i\}_i$ that are sampled from exactly the same distribution as that in **Game**$_1$, the view of A in **Game**$_2$ is not altered from that in **Game**$_1$. The lemma therefore follows.

Lemma 4.4 (Game$_2$ $\approx_s$ Game$_3$). *We have* $|\Pr[\mathsf{E}_2]-\Pr[\mathsf{E}_3]| \leq Q_{\mathsf{zt}}(L+3)^2/q$.

Proof. Let us observe that **Game**$_2$ and **Game**$_3$ differ only when A submits a handle corresponding to a polynomial $f(W_1,\ldots,W_L,\Delta_1,\ldots,\Delta_{Q_{\mathsf{kq}}},\Gamma, C_1,\ldots,C_L) \in \mathbb{T}$ satisfying $f(w_1,\ldots,w_L,\delta_1,\ldots,\delta_{Q_{\mathsf{kq}}},\gamma,c_1,\ldots,c_L) = 0$ and $f(W_1,\ldots,W_L,\Delta_1,\ldots,\Delta_{Q_{\mathsf{kq}}},\Gamma,c_1,\ldots,c_L) \neq 0$ to the zero-test oracle. Let F denote the event. It suffices to bound the probability of F occurring in **Game**$_2$. To do so, let us fix an element f in $\mathbb{T}$ and $c_1,\ldots,c_L$ in $\mathbb{Z}_q$. We then define a polynomial $g(W_1,\ldots,W_L,\Delta_1,\ldots,\Delta_{Q_{\mathsf{kq}}},\Gamma) \in \mathbb{Z}_q[W_1,\ldots,W_L,\Delta_1,\ldots,\Delta_{Q_{\mathsf{kq}}},\Gamma]$ as

$$\begin{aligned} &g(W_1,\ldots,W_L,\Delta_1,\ldots,\Delta_{Q_{\mathsf{kq}}},\Gamma) \\ &:= \left(\prod_{i\in[L]} W_i\right) \cdot f(W_1,\ldots,W_L,\Delta_1,\ldots,\Delta_{Q_{\mathsf{kq}}},\Gamma,c_1,\ldots,c_L). \end{aligned}$$

Note that in the above, the term $(\prod_i W_i)$ is introduced in order to clear the denominators that possibly appear in f and to make sure that g is in the ring $\mathbb{Z}_q[W_1,\ldots,W_L,\Delta_1,\ldots,\Delta_{Q_{\mathsf{kq}}},\Gamma]$ rather than in $\mathbb{T}$. We observe that F occurs if and only if $g(w_1,\ldots,w_L,\delta_1,\ldots,\delta_{Q_{\mathsf{kq}}},\gamma) = 0$ and $g(W_1,\ldots,W_L,\Delta_1,\ldots,\Delta_{Q_{\mathsf{kq}}},\Gamma) \neq 0$ since we have $w_i \neq 0$ for all $i \in [L]$. We can bound this probability by $(L+3)^2/q$ using Schwartz-Zippel lemma since g is a polynomial in $\mathbb{Z}_q[W_1,\ldots,W_L,\Delta_1,\ldots,\Delta_{Q_{\mathsf{kq}}},\Gamma]$ with degree at most $L+3$. (Recall that f can be represented as a linear combination of the terms in S_T.) Since A makes at most Q_{zt} zero-test queries, the lemma follows by the union bound.

Lemma 4.5 (Game$_3$ $\approx_s$ Game$_4$). *We have* $|\Pr[\mathsf{E}_3]-\Pr[\mathsf{E}_4]| \leq L/q$.

Proof. We observe that each entry of $\mathbf{c} = (c_1,\ldots,c_L)$ is either fixed to be 1 or distributed uniformly at random over $\mathbb{Z}_q$. Therefore, by the union bound, the probability that there is $i \in [L]$ such that $c_i = 0$ can be bounded by L/q. The lemma therefore follows.

Lemma 4.6 (Game$_4$ $\equiv$ Game$_5$). *We have* $\Pr[\mathsf{E}_4] = \Pr[\mathsf{E}_5]$.

Proof. We observe that **Game**$_4$ and **Game**$_5$ differ only when A makes a zero-test query for a handle corresponding to $f \in \mathbb{T}$ that satisfies Eq. (4.2) and there exists $Z \in S_{T,1}$ such that $a_Z \neq 0$ when we express f as Eq. (4.3). We claim that such f does not exist and two games are actually equivalent. For the sake of contradiction, assume that such f exists. Then Eq. (4.2) implies

$$\sum_{Z\in S_{T,1}} a_Z Z(c_1,\ldots,c_L) + \sum_{Z\in S_{T,2}} a_Z Z(c_1,\ldots,c_L) = 0,$$

where $Z(c_1,\ldots,c_L)$ denotes $Z(W_1,\ldots,W_L,\Delta_1,\ldots,\Delta_{Q_{\mathsf{kq}}},\Gamma,c_1,\ldots,c_L) \in \mathbb{T}$ in the above. We can see that $\sum_{Z\in S_{T,1}} a_Z Z(c_1,\ldots,c_L) = 0$ holds since we have

$$\left\{ \sum_{Z\in S_{T,1}} a'_Z Z(c_1,\ldots,c_L) : a'_Z \in \mathbb{Z}_q \right\} \cap \left\{ \sum_{Z\in S_{T,2}} a''_Z Z(c_1,\ldots,c_L) : a''_Z \in \mathbb{Z}_q \right\} = \{0\},$$

which follows from the fact that monomials in $S_{T,1}$ and $S_{T,2}$ are distinct even if we substitute $\{C_i\}_i$ in $S_{T,1}$ and $S_{T,2}$ with $\{c_i\}_i$ and ignore the difference between the coefficients of the monomials. Furthermore, $\sum_{Z \in S_{T,1}} a_Z Z(c_1, \ldots, c_L) = 0$ implies $a_Z = 0$ for all $Z \in S_{T,1}$, which follows from $\mathbf{c} \in (\mathbb{Z}_q^*)^L$ and from the fact that all monomials in $S_{T,1}$ are distinct even if we substitute $\{C_i\}_i$ with $\{c_i\}_i$ and ignore the difference between the coefficients of the monomials. However, this contradicts the assumption that there exists $Z \in S_{T,1}$ such that $a_Z \neq 0$. This completes the proof of the lemma.

Lemma 4.7 (Game$_\mathbf{5}$ $\approx_c$ Game$_\mathbf{6}$). *There exists a PPT adversary* B *such that* $|\Pr[\mathsf{E}_5] - \Pr[\mathsf{E}_6]| \leq Q_{\mathsf{kq}} Q_{\mathsf{zt}} \cdot \left(\mathsf{Adv}^{\mathsf{Ada\text{-}INDr}}_{\mathsf{BGG^+,B}}(1^\lambda) + 1/q\right)$.

Proof. We first observe that **Game**$_5$ and **Game**$_6$ differ only when A makes a zero-test query for a handle corresponding to $f \in \mathbb{T}$ that can be represented as

$$f(W_1, \ldots, W_L, \Delta_1, \ldots, \Delta_{Q_{\mathsf{kq}}}, \Gamma, C_1, \ldots, C_L) = \sum_{Z \in S_{T,2}} a_Z Z \tag{4.4}$$

and satisfies $f \neq 0$ over $\mathbb{T}$ and Eq. (4.2). We call such a query *bad*. In the following, we prove that the probability that A makes a bad query in **Game**$_5$ is negligible. To do so, we consider following sequence of games. We define F_{x} as the event that A makes a bad query in **Game**$_{5,\mathrm{x}}$ and the challenger does not abort.

Game$_{5.0}$: This game is the same as **Game**$_5$. By definition, the probability that A makes a bad query in **Game**$_5$ is $\Pr[\mathsf{F}_0]$.

Game$_{5.1}$: In this game, we change the previous game so that the challenger picks a random guess k^* for the first bad query as $k^* \leftarrow [Q_{\mathsf{zt}}]$ at the beginning of the game. Furthermore, we change the game so that the challenger aborts if the k^*-th zero-test query is not the first bad query. Since k^* is chosen uniformly at random and independent from the view of A, the guess is correct with probability $1/Q_{\mathsf{zt}}$ conditioned on F_0. Therefore, we have $\Pr[\mathsf{F}_1] = \Pr[\mathsf{F}_0]/Q_{\mathsf{zt}}$.

Game$_{5.2}$: This game is the same as the previous game except that the challenger aborts the game immediately after A makes the k^*-th zero-test query. Since whether F_1 occurs or not is irrelevant to how the game proceeds after the k^*-th zero-test query is made by A, we clearly have $\Pr[\mathsf{F}_2] = \Pr[\mathsf{F}_1]$.

Game$_{5.3}$: In this game, we change the game so that the challenger answers the first $k^* - 1$ zero-test queries by performing zero tests over $\mathbb{T}$. Furthermore, we change the game so that the sampling of $\mathbf{c}$ is deferred until the k^*-th zero-test query is made by A. We first observe that the game is well-defined since $\mathbf{c}$ is used only for the k^*-th zero-test query. Furthermore, since the first $k^* - 1$ zero-test queries that refer to $f \in \mathbb{T}$ such that $f \neq 0$ are answered by 0 whenever F_2 happens, we have $\Pr[\mathsf{F}_3] \geq \Pr[\mathsf{F}_2]$.

Game$_{5.4}$: To define the game, we first define the set $S_{T,2,j} := \{C_i \Gamma \Delta_j\}_{i \in [L] \text{ s.t. } d_i^{(j)}=1}$. By definition, we have $S_{T,2} = \cup_{j \in [Q_{\mathsf{kq}}]} S_{T,2,j}$. Using this notation, any $f \in \mathbb{T}$ referred by a bad query can be represented as

$$f(W_1, \ldots, W_L, \Delta_1, \ldots, \Delta_{Q_{\mathsf{kq}}}, \Gamma, C_1, \ldots, C_L) = \sum_{j \in [Q_{\mathsf{kq}}]} \sum_{Z \in S_{T,2,j}} a_Z Z. \quad (4.5)$$

In this game, we change the game so that the challenger aborts the game if the bad query made by A refers to f such that there *does not* exist $j \in [Q_{\mathsf{kq}}]$ satisfying

$$\sum_{Z \in S_{T,2,j}} a_Z Z \neq 0 \quad \text{and} \quad \sum_{Z \in S_{T,2,j}} a_Z Z(c_1, \ldots, c_L) = 0, \quad (4.6)$$

where $Z(c_1, \ldots, c_L)$ denotes $Z(W_1, \ldots, W_L, \Delta_1, \ldots, \Delta_{Q_{\mathsf{kq}}}, \Gamma, c_1, \ldots, c_L) \in \mathbb{T}$ above. We claim that this actually cannot happen. To see this, we first observe that since we have $f \neq 0$ for a bad query, there exists $j \in [Q_{\mathsf{kq}}]$ satisfying $\sum_{Z \in S_{T,2,j}} a_Z Z \neq 0$. Furthermore, we have

$$\sum_{Z \in S_{T,2,j}} a_Z Z(c_1, \ldots, c_L) = - \sum_{j' \neq j} \sum_{Z \in S_{T,2,j'}} a_Z Z(c_1, \ldots, c_L)$$

from Eq. (4.2). However, the above is impossible unless the left hand side equals to 0 since any monomial in $S_{T,2,j}$ never appears in $S_{T,2,j'}$ for $j' \neq j$ even if we replace $\{C_i\}_i$ with $\{c_i\}_i$ and ignore the difference between the coefficients of the monomials. Therefore, the change made in this game is only conceptual and we have $\Pr[\mathsf{F}_4] = \Pr[\mathsf{F}_3]$.

Game$_{5.5}$: In this game, we change the previous game so that the challenger picks $j^* \leftarrow [Q_{\mathsf{kq}}]$ uniformly at random at the beginning of the game. Furthermore, we add the abort condition that the challenger aborts if Eq. (4.6) does not hold with respect to $j = j^*$ for f that is referred by the k^*-th zero-test query. Since there exists $j' \in [Q_{\mathsf{kq}}]$ that satisfies Eq. (4.6) as long as F_4 occurs and j^* is chosen uniformly at random and independent from the view of A, we have $\Pr[\mathsf{F}_5] \geq \Pr[\mathsf{F}_4]/Q_{\mathsf{kq}}$.

Game$_{5.6}$: In this game, we further change the game so that the challenger aborts the game if the j^*-th key query has not been made yet at the point when the k^*-th zero-test query is made. We claim that conditioned on F_5 happens, the challenger never aborts. To see this, we observe that if the j^*-th key query has not been made then terms that contain Δ_{j^*} has not been given to A and there is no way to make a zero-test query for f such that $\sum_{Z \in S_{T,2,j^*}} a_Z Z \neq 0$, since all terms in $S_{T,2,j^*}$ are multiples of Δ_{j^*}. We therefore have $\Pr[\mathsf{F}_6] = \Pr[\mathsf{F}_5]$.

Game$_{5.7}$: In this game, we further change the game so that the challenger samples c_i only for $i \in [L]$ such that $d_i^{(j^*)} = 1$, where j^* is chosen at the beginning of the game as in **Game**$_{5.5}$. The game is still well-defined since the only place in the game where we need the information of $\mathbf{c}$ is when checking Eq. (4.6) and we only need $\{c_i\}_{i \in [L] \text{ s.t. } d_i^{(j^*)}=1}$ there. (Recall that we have

$S_{T,2,j} = \{C_i \Gamma \Delta_j\}_{i\in[L] \text{ s.t. } d_i^{(j)}=1}$.) Clearly, this does not change the view of A. We therefore have $\Pr[\mathsf{F}_7] = \Pr[\mathsf{F}_6]$.

From Eqs. (3.1) and (3.2), we can see that $\{c_i\}_{i\in[L] \text{ s.t. } d_i^{(j^*)}=1}$ consists of the following components:

$$\psi_0 = 1, \quad \psi_1 := \mathbf{s}^\top \mathbf{u} + e_1 + \mu\lceil q/2 \rceil, \quad \psi_2^\top := \mathbf{s}^\top \mathbf{A} + \mathbf{e}_2^\top,$$
$$\psi_{i,x_i^{(j^*)}}^\top := \mathbf{s}^\top(\mathbf{B}_i - x_i^{(j^*)}\mathbf{G}) + \mathbf{e}_{i,x_i^{(j^*)}}^\top \quad \text{for } i \in [\ell],$$

where $x_i^{(j^*)}$ is the i-th entry of $\mathbf{x}^{(j^*)}$.

Game$_{5.8}$: In this game, we further change the game so that the challenger samples

$$\psi_0 := 1 \in \mathbb{Z}_q, \quad \psi_1 \leftarrow \mathbb{Z}_q, \quad \psi_2 \leftarrow \mathbb{Z}_q^m, \quad \psi_{i,b} \leftarrow \mathbb{Z}_q^m \quad \text{for } i \in [\ell] \text{ and } b \in \{0,1\}$$

and sets $\{c_i\}_{i\in[L] \text{ s.t. } d_i^{(j^*)}=1}$ from the above components.[6] As we prove in Lemma 4.8, there exists a PPT adversary B such that $\mathsf{Adv}^{\mathsf{Ada\text{-}INDr}}_{\mathsf{BGG}^+,\mathsf{B}}(1^\lambda) \geq |\Pr[\mathsf{F}_7] - \Pr[\mathsf{F}_8]|$.

As we will prove in Lemma 4.9, we have $\Pr[\mathsf{F}_8] \leq 1/q$. This allows us to bound $\Pr[\mathsf{F}_0]$ as $\Pr[\mathsf{F}_0] \leq Q_{\mathsf{kq}} Q_{\mathsf{zt}} \cdot (\mathsf{Adv}^{\mathsf{Ada\text{-}INDr}}_{\mathsf{BGG}^+,\mathsf{B}}(1^\lambda) + 1/q)$, where B is a PPT adversary. This completes the proof of Lemma 4.7.

It remains to prove Lemmas 4.8 and 4.9 in the following.

Lemma 4.8 (Game$_{5.7} \approx_c$ Game$_{5.8}$). *There exists a PPT adversary* B *such that* $\mathsf{Adv}^{\mathsf{Ada\text{-}INDr}}_{\mathsf{BGG}^+,\mathsf{B}}(1^\lambda) \geq |\Pr[\mathsf{F}_7] - \Pr[\mathsf{F}_8]|$.

Proof. We show that if A can distinguish **Game**$_{5.7}$ from **Game**$_{5.8}$, we can build another adversary B against $\mathsf{Ada\text{-}INDr}$ security of BGG^+. The adversary B acts as the challenger and simulates the game for A. Looking ahead, setup phase and key queries are trivial to handle since they do not need any parameter of BGG^+. The only steps we need care are the simulation of the challenge phase and the k^*-th zero-test query, where B needs to interact with its challenger in order to handle them. We describe how B proceeds in the following.

Setup phase. At the beginning of the game, B is given 1^λ and the master public key of BGG^+ $(\mathbf{A}, \mathbf{B}, \mathbf{u})$. It then gives the handles to $1, W_1, \ldots, W_L$ corresponding to $\mathbb{G}_1$ and the handle to 1 corresponding to $\mathbb{G}_2$ to A. These handles correspond to the master public key. B also samples $j^* \leftarrow [Q_{\mathsf{kq}}]$, $k^* \leftarrow [Q_{\mathsf{zt}}]$, and $b \leftarrow \{0,1\}$ and keeps them secret.

Key Queries. Given the j-th secret key query for $\mathbf{x}^{(j)}$ made by A, B proceeds as follows. B first forms $\mathbf{d}^{(j)} \in \mathbb{Z}_q^L$ as specified in the key generation algorithm and returns the handles corresponding to $(d_1^{(j)}\Delta_j/W_1, \ldots, d_L^{(j)}\Delta_j/W_L)$ in $\mathbb{G}_2$ to A.

[6] Note that until this step, we have not changed the distribution of $\{c_i\}_{i\in[L]}$ except that we stop sampling c_i for i such that $d_i^{(j^*)} = 1$ in **Game**$_{5,7}$.

Challenge Query. When A makes the challenge query for a circuit F, B makes a *secret key query* for F to its challenger and is given $\mathbf{r}$ sampled as $\mathbf{r} \leftarrow [\mathbf{A}\|\mathbf{B}_F]_\tau^{-1}(\mathbf{u})$. B then sets $\mathsf{ct}_0 = (\mathbf{A}, \mathbf{B})$, $\mathsf{ct}_2 := \mathbf{r}$, and ct_1 as the handles corresponding to the formal variables $(C_1, \ldots, C_L)$ and gives $\mathsf{ct} = (\mathsf{ct}_0, \mathsf{ct}_1, \mathsf{ct}_2)$ to A as the challenge ciphertext.

Generic Group Queries. B honestly handles the queries for the generic group oracle corresponding to addition, negation, and multiplication (bilinear map) made by A by keeping track of the underlying encodings in $\mathbb{T}$ associated with the handles. For the k-th zero-test query that refers to an element f in $\mathbb{T}$, B returns 1 if $f = 0$ over $\mathbb{T}$ and 0 otherwise if $k < k^*$. If $k = k^*$, B first checks whether the j^*-th key query has already been made and aborts otherwise, as specified in $\mathbf{Game}_{5.6}$. It then makes the challenge query for the attribute $\mathbf{x}^{(j^*)}$, where $\mathbf{x}^{(j^*)}$ is the attribute for which A has made the j^*-th key query, and the message b to its challenger. Then B obtains its challenge ciphertext $(\psi_1, \psi_2, \{\psi_{i,x_i^{(j^*)}}\}_{i\in[\ell]})$. It then sets $\psi_0 = 1$ and forms $\{c_i\}_{i\in[L] \text{ s.t. } d_i^{(j^*)}=1}$ by vectorizing the terms appropriately. Finally, it checks whether Eq. (4.6) holds or not using $\{c_i\}_{i\in[L] \text{ s.t. } d_i^{(j^*)}=1}$ as specified in $\mathbf{Game}_{5.7}$ and outputs 1 if it holds and 0 otherwise.

Analysis. It is easy to see that B simulates $\mathbf{Game}_{5.7}$ if the challenge ciphertext for B is the real one and $\mathbf{Game}_{5.8}$ if it is chosen uniformly at random from the ciphertext space. Therefore, it can be seen that B outputs 1 with probability $\Pr[\mathsf{F}_7]$ if the challenge bit for B is 0 and $\Pr[\mathsf{F}_8]$ otherwise. Therefore, B's advantage against BGG^+ is $|\Pr[\mathsf{F}_7] - \Pr[\mathsf{F}_8]|$. This completes the proof of the lemma.

Lemma 4.9. *We have* $\Pr[\mathsf{F}_8] = 1/q$.

Proof. We observe that F_8 occurs only when A makes a zero-test query that refers to a handle $f \neq 0$ that can be represented as Eq. (4.4) and satisfies Eq. (4.6) with respect to j^* where $\{c_i\}_{i\in[L] \text{ s.t. } d_i^{(j^*)}=1}$ are chosen as $\mathbf{Game}_{5.8}$. However, Eq. (4.6) can happen only with probability at most $1/q$ since f is represented as a linear combination of $\{C_i \Gamma \Delta_j\}_{i,j}$ and all entries of $\{c_i\}_{i\in[L] \text{ s.t. } d_i^{(j^*)}=1}$ are chosen uniformly at random except for the entry that is fixed to be 1.

5 Implications to CP-ABE, BE, and IBBE

In this section, we show that by setting the circuit class supported by our CP-ABE scheme in Sect. 3 appropriately, we can obtain various new schemes with different security and efficiency tradeoffs. In particular, we obtain new CP-ABE, BE, and IBBE schemes from the LWE assumption in the bilinear generic group model. Our CP-ABE scheme achieves the notable efficiency property that the sizes of all the parameters in the system do not depend on the size of the circuits supported by the scheme. Similarly, our BE (resp., IBBE) schemes achieve optimal parameter size, in the sense that the sizes of all parameters in the system are bounded by a fixed polynomial that is independent from the number of users (resp., upper bound on the number of recipients). These efficiency properties have never been achieved without using indistinguishability obfuscation or multilinear maps.

5.1 New CP-ABE Scheme

By setting $\mathcal{F}_{\mathsf{CP}} := \{\mathcal{C}_{\ell(\lambda),d(\lambda)}\}_\lambda$ in the construction in Sect. 3, we obtain a CP-ABE scheme that can deal with the set of circuits whose input length and depth are $\ell(\lambda)$ and $d(\lambda)$, respectively. In order to prove $\mathsf{Ada\text{-}IND}$ security for the resulting scheme, we need to be able to prove $\mathsf{Ada\text{-}INDr}$ security for the KP-ABE scheme BGG^+ for the same circuit class as stated in Theorem 4.1. This is possible by assuming subexponential hardness of LWE as we see in Theorem 2.10. The notable feature of the resulting scheme is that the sizes of the master public key, ciphertexts, and secret keys are independent from the size of the circuits supported by the scheme. The sizes of these parameters are only dependant on the input length and the depth of the circuits.

Summarizing the above discussion, we get the following theorem.

Theorem 5.1. *Assuming the subexponential hardness of LWE, we have a CP-ABE scheme for circuit class $\mathcal{C}_{\ell,d}$ for arbitrary $\ell = \mathrm{poly}(\lambda)$ and $d = O(\log \lambda)$ that satisfies $\mathsf{Ada\text{-}IND}$ security in the bilinear generic group model. The sizes of the master public key, ciphertexts, and secret keys are bounded by $\mathrm{poly}(\lambda, \ell, d)$.*

We note that in all previous CP-ABE scheme (e.g., [11,44,48]) for NC_1, either the ciphertext or secret key size depends on the circuit size supported by the scheme.

5.2 New BE Scheme with Optimal Parameter Size

Here, we show that we can obtain a BE scheme with optimal parameter size by setting the circuit class $\mathcal{F}$ supported by the CP-ABE scheme in Sect. 3 appropriately.

Obtaining DBE from KP-ABE. In order to get the BE scheme, we first observe that we can implement a DBE scheme by a KP-ABE scheme for the following circuit class $\mathcal{F}_{\mathsf{BE}}$ defined as $\mathcal{F}_{\mathsf{BE}} = \left\{F_S : \{0,1\}^{\lceil \log N \rceil} \to \{0,1\}\right\}_{S \subseteq [N]}$ where $F_S(i) = \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{if } i \notin S \end{cases}$. Here, we identify a user index $i \in [N]$ and elements in S with binary strings in $\{0,1\}^{\lceil \log N \rceil}$ by a natural bijection map between $\{0,1\}^{\lceil \log N \rceil}$ and $[2^{\lceil \log N \rceil}] \supseteq [N]$. Since the depth of F_S affects the efficiency of the DBE scheme, we want F_S to be as shallow as possible. For this purpose, we compute F_S by first computing $b_j := (i \overset{?}{=} j)$ for all $j \in S$ *in parallel* and then computing $\vee_{j \in S} b_j$. The first step can be implemented with depth $O(\log \log N)$ and the second step with $O(\log N)$. This allows us to implement F_S with depth $O(\log |S|) \leq O(\log N)$. By the definition of F_S, one can see that this KP-ABE scheme implements the functionality of DBE.

Plugging the DBE into Our Construction in Sect. 3. We then instantiate the KP-ABE for the circuit class $\mathcal{F}_{\mathsf{BE}}$ with BGG^+ and plug this scheme into our CP-ABE construction in Sect. 3. Since the ciphertext and key attributes of the CP-ABE scheme are swapped from the underlying KP-ABE scheme, we obtain a BE scheme as a result. This instantiation is possible since the depth of the circuits is bounded by $O(\log N) \leq O(\log \lambda)$ and we can take the upper bound on the depth $d(\lambda)$ to be larger than this. The sizes of the master public key, ciphertexts, and secret keys in the resulting BE scheme are bounded by $\mathrm{poly}(\log N, \lambda) = \mathrm{poly}(\lambda)$, which is independent of the number of users, since the depth and input length of the circuits in $\mathcal{F}_{\mathsf{BE}}$ is bounded by $O(\log N)$. Note that we crucially rely on the efficiency property of our CP-ABE scheme that the sizes of all parameters in the system are independent of the size of the circuits being supported, where the latter can be as large as $O(N)$ for $\mathcal{F}_{\mathsf{BE}}$.

Security of the Resulting BE Scheme. In order for the resulting BE scheme to have $\mathsf{Ada\text{-}IND}$ security, we need the underlying KP-ABE scheme BGG^+ to have $\mathsf{Ada\text{-}INDr}$ security as stated in Theorem 4.1. In the general case where the input length for the circuits is of $\mathrm{poly}(\lambda)$, we need to assume subexponential hardness of LWE to prove $\mathsf{Ada\text{-}INDr}$ security for BGG^+ as we see in Theorem 2.10. However, since we restrict the circuit class for BGG^+ to be $\mathcal{F}_{\mathsf{BE}}$ here, we can avoid assuming subexponential hardness of LWE and base the security of our scheme on polynomial hardness of LWE. To see this, we first recall that for proving $\mathsf{Sel\text{-}INDr}$ security for BGG^+, polynomial hardness of LWE is enough (Theorem 2.10). We then observe that in the special case of DBE, $\mathsf{Sel\text{-}INDr}$ and $\mathsf{Ada\text{-}INDr}$ are actually equivalent, since one can guess the target attribute $i^\star \in [N]$ chosen by the adversary in the security game with only polynomial security loss.

Summarizing the above discussion, we get the following theorem.

Theorem 5.2. *Assuming the LWE assumption, we have a BE scheme that satisfies* $\mathsf{Ada\text{-}IND}$ *security in the bilinear generic group model. The sizes of the master public key, ciphertexts, and secret keys are bounded by a fixed polynomial* $\mathrm{poly}(\lambda)$ *that is independent of* N.

In the full version of our paper [5], we show that we can obtain an IBBE scheme with optimal parameter size by setting the circuit class $\mathcal{F}$ supported by the CP-ABE scheme in Sect. 3 appropriately.

Acknowledgement. We thank anonymous reviewers for helpful comments. Shota Yamada is supported by JST CREST Grant Number JPMJCR19F6 and JSPS KAKENHI Grant Number 16K16068.

References

1. Agrawal, S.: Stronger security for reusable garbled circuits, general definitions and attacks. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10401, pp. 3–35. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63688-7_1

2. Agrawal, S.: Indistinguishability obfuscation without multilinear maps: new methods for bootstrapping and instantiation. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019. LNCS, vol. 11476, pp. 191–225. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17653-2_7
3. 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). https://doi.org/10.1007/978-3-642-13190-5_28
4. Agrawal, S., Boneh, D., Boyen, X.: Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 98–115. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14623-7_6
5. Agrawal, S., Yamada, S.: Optimal broadcast encryption from pairings and LWE. Eprint 2020/228
6. Ananth, P., Jain, A., Lin, H., Matt, C., Sahai, A.: Indistinguishability obfuscation without multilinear maps: IO from LWE, bilinear maps, and weak pseudorandomness. In: Crypto (2019)
7. Apon, D., Döttling, N., Garg, S., Mukherjee, P.: Cryptanalysis of indistinguishability obfuscations of circuits over GGH13. Eprint 2016 (2016)
8. Attrapadung, N., Libert, B.: Functional encryption for inner product: achieving constant-size ciphertexts with adaptive security or support for negation. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 384–402. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13013-7_23
9. Baltico, C.E.Z., Catalano, D., Fiore, D., Gay, R.: Practical functional encryption for quadratic functions with applications to predicate encryption. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10401, pp. 67–98. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63688-7_3
10. Barthe, G., Fagerholm, E., Fiore, D., Mitchell, J., Scedrov, A., Schmidt, B.: Automated analysis of cryptographic assumptions in generic group models. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8616, pp. 95–112. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44371-2_6
11. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy, pp. 321–334 (2007)
12. Boneh, D., et al.: 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). https://doi.org/10.1007/978-3-642-55220-5_30
13. Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer, Heidelberg (2005). https://doi.org/10.1007/11535218_16
14. Boneh, D., Kim, S.: Single key CP-ABE. Personal Communication (2016)
15. Boneh, D., Waters, B., Zhandry, M.: Low overhead broadcast encryption from multilinear maps. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8616, pp. 206–223. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44371-2_12
16. Boneh, D., Zhandry, M.: Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. Algorithmica **79**(4), 1233–1285 (2016). https://doi.org/10.1007/s00453-016-0242-8

17. Brakerski, Z.: Fully homomorphic encryption without modulus switching from classical GapSVP. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 868–886. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_50
18. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) fully homomorphic encryption without bootstrapping. In: ITCS, pp. 309–325 (2012)
19. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing, STOC 2013. ACM (2013)
20. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In: FOCS (2011)
21. Brakerski, Z., Vaikuntanathan, V.: Lattice-inspired broadcast encryption and succinct ciphertext policy ABE. Personal communication (2020)
22. 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). https://doi.org/10.1007/978-3-642-13190-5_27
23. Cheon, J.H., Fouque, P.-A., Lee, C., Minaud, B., Ryu, H.: Cryptanalysis of the new CLT multilinear map over the integers. Eprint 2016/135
24. Cheon, J.H., Han, K., Lee, C., Ryu, H., Stehlé, D.: Cryptanalysis of the multilinear map over the integers. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 3–12. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46800-5_1
25. Cheon, J.H., Jeong, J., Lee, C.: An algorithm for NTRU problems and cryptanalysis of the GGH multilinear map without a low level encoding of zero. Eprint 2016/139
26. Coron, J.-S., et al.: Zeroizing without low-level zeroes: new MMAP attacks and their limitations. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9215, pp. 247–266. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-47989-6_12
27. Coron, J.-S., Lee, M.S., Lepoint, T., Tibouchi, M.: Zeroizing attacks on indistinguishability obfuscation over CLT13. In: Fehr, S. (ed.) PKC 2017. LNCS, vol. 10174, pp. 41–58. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54365-8_3
28. Delerablée, C.: Identity-based broadcast encryption with constant size ciphertexts and private keys. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 200–215. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76900-2_12
29. Delerablée, C., Paillier, P., Pointcheval, D.: Fully collusion secure dynamic broadcast encryption with constant-size ciphertexts or decryption keys. In: Takagi, T., Okamoto, T., Okamoto, E., Okamoto, T. (eds.) Pairing 2007. LNCS, vol. 4575, pp. 39–59. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73489-5_4
30. Fiat, A., Naor, M.: Broadcast encryption. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 480–491. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48329-2_40
31. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: FOCS (2013). http://eprint.iacr.org/
32. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC, pp. 197–206 (2008)

33. Gentry, C., Waters, B.: Adaptive security in broadcast encryption systems (with short ciphertexts). In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 171–188. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01001-9_10
34. Gorbunov, S., Vaikuntanathan, V., Wee, H.: Attribute based encryption for circuits. In: STOC (2013)
35. Gorbunov, S., Vinayagamurthy, D.: Riding on asymmetry: efficient abe for branching programs. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9452, pp. 550–574. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48797-6_23
36. Goyal, R., Quach, W., Waters, B., Wichs, D.: Broadcast and trace with N^ε ciphertext size from standard assumptions. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019. LNCS, vol. 11694, pp. 826–855. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26954-8_27. https://eprint.iacr.org/2019/636
37. He, K., Weng, J., Liu, J.-N., Liu, J.K., Liu, W., Deng, R.H.: Anonymous identity-based broadcast encryption with chosen-ciphertext security. In: Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, ASIA CCS 2016 (2016)
38. Hu, Y., Jia, H.: Cryptanalysis of GGH map. Cryptology ePrint Archive: Report 2015/301 (2015)
39. Jain, A., Lin, H., Sahai, A.: Simplifying constructions and assumptions for $i\mathcal{O}$. Cryptology ePrint Archive, Report 2019/1252 (2019). https://eprint.iacr.org/2019/1252
40. Maurer, U.: Abstract models of computation in cryptography. In: Smart, N.P. (ed.) Cryptography and Coding 2005. LNCS, vol. 3796, pp. 1–12. Springer, Heidelberg (2005). https://doi.org/10.1007/11586821_1
41. 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). https://doi.org/10.1007/978-3-642-29011-4_41
42. Miles, E., Sahai, A., Zhandry, M.: Annihilation attacks for multilinear maps: cryptanalysis of indistinguishability obfuscation over GGH13. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9815, pp. 629–658. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53008-5_22
43. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM **56**(6), 84–93 (2009). Extended abstract in STOC 2005
44. Rouselakis, Y., Waters, B.: Practical constructions and new proof methods for large universe attribute-based encryption. In: ACM-CCS, pp. 463–474 (2013)
45. Sakai, R., Furukawa, J.: Identity-based broadcast encryption. IACR Cryptology ePrint Archive (2007)
46. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 256–266. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-69053-0_18
47. Tsabary, R.: Fully secure attribute-based encryption for t-CNF from LWE. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019. LNCS, vol. 11692, pp. 62–85. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26948-7_3
48. Waters, B.: Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19379-8_4