Constant Input Attribute Based (and Predicate) Encryption from Evasive and Tensor LWE

Shweta Agrawal[*] Mélisa Rossi[†] Anshu Yadav[‡] Shota Yamada[§]

Abstract

Constructing advanced cryptographic primitives such as obfuscation or broadcast encryption from standard hardness assumptions in the post quantum regime is an important area of research, which has met with limited success despite significant effort. It is therefore extremely important to find new, simple to state assumptions in this regime which can be used to fill this gap. An important step was taken recently by Wee (Eurocrypt '22) who identified two new assumptions from lattices, namely evasive LWE and tensor LWE, and used these to construct broadcast encryption and ciphertext policy attribute based encryption for P with optimal parameters. Independently, Tsabary formulated a similar assumption and used it to construct witness encryption (Crypto '22). Following Wee's work, Vaikuntanathan, Wee and Wichs independently provided a construction of witness encryption (Asiacrypt '22).

In this work, we advance this line of research by providing the first construction of multi-input attribute based encryption (miABE) for the function class NC_1 for *any* constant arity from evasive LWE. Our construction can be extended to support the function class P by using evasive and a suitable strengthening of tensor LWE. In more detail, our construction supports k encryptors, for any constant k, where each encryptor uses the master secret key msk to encode its input $(\mathbf{x}_i, m_i)$, the key generator computes a key sk_f for a function $f \in \mathsf{NC}_1$ and the decryptor can recover $(m_1, \ldots, m_k)$ if and only if $f(\mathbf{x}_1, \ldots, \mathbf{x}_k) = 1$. The only known construction for miABE for NC_1 by Agrawal, Yadav and Yamada (Crypto '22) supports arity 2 and relies on pairings in the generic group model (or with a non-standard knowledge assumption) in addition to LWE. Furthermore, it is completely unclear how to go beyond arity 2 using this approach due to the reliance on pairings.

Using a compiler from Agrawal, Yadav and Yamada (Crypto '22), our miABE can be upgraded to multi-input *predicate* encryption for the same arity and function class. Thus, we obtain the first constructions for constant-arity predicate and attribute based encryption for a generalized class such as NC_1 or P from simple assumptions that may be conjectured post-quantum secure. Along the way, we show that the tensor LWE assumption can be reduced to standard LWE in an important special case which was not known before. This adds confidence to the plausibility of the assumption and may be of wider interest.

[*] IIT Madras, Chennai, shweta@cse.iitm.ac.in

[†] ANSSI, Paris, melissa.rossi@ssi.gouv.fr

[‡] IIT Madras, Chennai, anshu.yadav06@gmail.com

[§] AIST, Tokyo, yamada-shota@aist.go.jp

Contents

1 Introduction

Attribute Based Encryption. Attribute based encryption (ABE) [SW05, GPSW06] enables fine grained access control on encrypted data. In this notion, an encryptor computes a ciphertext encoding a secret message m and a public *attribute* vector $\mathbf{x}$, a key generator computes a secret key associated with a function f, and decryption outputs m if and only if $f(\mathbf{x}) = 1$. Security is formalized using an *indistinguishability* style game, where an adversary is asked to distinguish between an encryption of $(m_0, \mathbf{x}_0)$ and $(m_1, \mathbf{x}_1)$ given secret keys that do not decrypt the challenge. A further strengthening of this notion, traditionally referred to as *predicate* encryption (PE), additionally enables hiding of the attributes $\mathbf{x}$ which are public in ABE.

The Multi-Input Setting. The recent work of Agrawal, Yadav and Yamada [AYY22] (henceforth by AYY) proposed decentralizing these notions to the multi-input setting, where the attribute $\mathbf{x}$ and the message m may be distributed among multiple parties, who must encrypt their inputs independently using uncorrelated random coins. In more detail, we now have k encryptors, who each encrypt their input $\{\mathbf{x}_i, m_i\}_{i\in[k]}$ using a master secret key msk[1], the key generator provides a key sk_f for an arity k function f, and decryption recovers $(m_1, \ldots, m_k)$ if and only if $f(\mathbf{x}_1, \ldots, \mathbf{x}_k) = 1$.

While the notion of multi-input ABE, denoted by miABE, had been studied before [BJK+18], this was as a stepping stone to constructing *Witness Encryption*. AYY argue that miABE is an important primitive in its own right and not just as a stepping stone to witness encryption, since it captures the demands of real world data more realistically than single input ABE. At the heart of miABE is the idea that though data may be generated in different places, it may be correlated in meaningful ways and natural access control policies are likely to embed constraints that pertain to the entire data. Hence, all information related to any self-contained unit, such as an individual or organization, should be considered together for the purpose of access control.

As a simple example, consider a dental care facility that has multiple branches in different geographical locations. A patient Alice (say) may visit the branch near her home if she needs a consultation on a Saturday but the branch near her work place if the appointment is on a weekday. Indeed, she may visit a branch in another city if she is travelling and needs dental assistance. In this scenario, each branch will contain some subset of the data pertaining to Alice's dental history. Yet, all this data must be considered together in order to make decisions about future treatments. To enable this, each branch might encrypt their patient data everyday and upload it to a central repository. Ideally, a doctor should be able access all the information related to Alice's dental history if the doctor's key satisfies the relevant access control policy (for instance if she is one of Alice's designated doctors and all the records correspond to the Alice).

As another example, consider a businessman Bob (say) whose job involves frequent travelling. To stay healthy, he may become a member of a fitness center which has branches in several cities and visit the nearest one in his current location. The fitness center could have reduced rates or other promotional offers for clients depending on their usage, and Bob would wish to benefit from these though his usage is split across locations. As in the example above, each branch can encrypt their local data and upload it to a central location and secret keys could be provided to compute eligibility for the offer by collating this data. If eligible, the personal information can be decrypted and the offer can be extended. Finally, consider a research project which spans multiple universities. Each university could encrypt their findings and upload it to a central server, and keys could be provided for accessing the joint data based on some policy that spans the entire dataset. Please see [AYY22] for several other examples.

[1] As in multi-input functional encryption, the notion of miABE is primarily meaningful in the secret key setting, due to excessive leakage that occurs in the public key setting.

1.1 Prior Work

AYY provided the first constructions for multi-input attribute based (and predicate) encryption. Specifically, they provided the first construction for two-input *key-policy* ABE for NC_1 from LWE and pairings by leveraging a surprising connection between the algebraic properties required to build two input ABE and the techniques developed in the context of broadcast encryption [AY20, AWY20]. They also provided heuristic constructions for 2 input ABE for P and 3 input ABE for NC_1 – we will not discuss these here since our focus will be on constructions that admit a proof. Additionally, they gave a compiler that "lifts" any constant arity ABE scheme to a PE scheme of the same arity using the power of lockable obfuscation, which can be constructed from the Learning With Errors (LWE) assumption. Independently, Francati et al. [FFMV23] provided multi-input PE (hence also ABE) schemes for the restricted functionality of conjunctions of (bounded) polynomial depth from LWE. Notably, one of their constructions can support polynomial arity unlike AYY, which is a plus. On the other hand, their security model does not support collusions, which is typically the main technical challenge in constructing ABE and PE even in the single input setting. As another plus, when restricted to constant (though not polynomial) arity, their constructions can support user corruption, which AYY cannot – indeed AYY cannot even support arity for any constant though they support a much more expressive function class which is not restricted to conjunctions.

We briefly mention the stronger notion of multi-input *functional encryption* (miFE) [GGG+14], which generalizes multi-input ABE and PE. In contrast to miABE and miPE, miFE has been studied extensively, and admits constructions for various functionalities from a variety of assumptions [GGG+14, AJ15, AGRW17, DOT18, ACF+18, CDG+18, Tom19, ABKW19, ABG19, LT19, AGT21b, AGT21a, AGT22]. However, since multi-input FE for NC_1 implies indistinguishability obfuscation (iO) [BGI+01, GGH+13], it remains an important area of study to instantiate weaker notions such as miABE and miPE from assumptions not known to imply iO. This is particularly important in the post quantum regime, where constructions of iO are still based on strong, ill-understood assumptions which are often broken [Agr19, APM20, WW21, GP21, DQV+21, AJS23]. Several prior works therefore focus on instantiating iO based constructions from weaker assumptions [AY20, AWY20, Wee22, Tsa22, VWW22, AKYY23], a direction also followed by the present work.

1.2 Our Results

As seen above, current known results for miABE schemes are quite restricted – the result of AYY appears to be fundamentally stuck at arity 2, while the result of Francati et al. [FFMV23] is tailored to the restricted functionality of conjunctions, offering no avenue for generalization to arbitrary NC_1 circuits.

In this work, we significantly extend the reach of multi-input ABE schemes by providing the first construction of miABE for the function class NC_1 for *any* constant arity from the recently introduced evasive LWE assumption [Wee22, Tsa22]. Our construction can be extended to support the function class P by using evasive and a suitable strengthening of tensor LWE. For the special case of arity 2, we need only the assumptions introduced by Wee, i.e. evasive LWE for NC_1 and evasive plus tensor LWE for P (i.e. we do not need to strengthen tensor LWE).[2]

In more detail, our construction supports k encryptors, for any constant k, where each encryptor uses the master secret key msk to encode its input $(\mathbf{x}_i, m_i)$, the key generator

[2] Actually, our definition of evasive LWE is slightly different from that defined in [Wee22]. Please refer to Assumption 3.1 and the related discussion.

Paper	**Arity**	**Functionality**	**Corruption**	**Collusion**	**Assumption**
[FFMV23]	Poly	Conjunctions in P	No	No	LWE
[FFMV23]	Constant	Conjunctions in P	Yes	No	LWE
[AYY22]	2	NC_1	No	Yes	Koala and LWE
[AYY22]	2	P	No	Yes	Heuristic
This	2	P	No	Yes	Evasive and Tensor LWE
This	Constant	NC_1	No	Yes	Evasive LWE
This	Constant	P	No	Yes	Evasive and strong Tensor LWE

Table 1: Comparison with Prior Work in miPE. Note that KOALA is a non-standard knowledge type assumption and "heuristic" means that there is no proof of security.

computes a key sk_f for a function $f \in \mathsf{NC}_1$ (or P at the cost of a stronger assumption) and the decryptor can recover $(m_1, \ldots, m_k)$ if and only if $f(\mathbf{x}_1, \ldots, \mathbf{x}_k) = 1$. We prove security in the standard indistinguishability game defined by AYY from the aforementioned assumptions. Using the compiler from AYY, our miABE schemes can be upgraded to multi-input predicate encryption schemes for the same arity and function class. Along the way, we show that the tensor LWE assumption can be reduced to standard LWE in a special case which was not known before. This adds confidence to the plausibility of the assumption and may be of wider interest.

We defer details about our strengthening of tensor LWE for P as well as the new implication discussed above to the technical overview (Section 1.3) since stating them formally will require heavy notation which we do not want to introduce here. We provide a comparison with known results in Table 1.

Perspective: Connection to Witness Encryption. Witness encryption (WE) is defined for some NP language L with a corresponding witness relation R. In WE, an encryptor encrypts a message m to a particular problem instance x. The decryptor can recover m if $x \in L$ and it knows a witness w such that $R(x, w) = 1$. Security posits that a ciphertext hides the message m so long as $x \notin L$. Brakerski et al. [BJK+18] showed that miABE for polynomial arity implies witness encryption – this may explain in part why constructions of miABE have been so elusive. Even for smaller arity, there are nontrivial implications – for instance, the arity 2 miABE for NC_1 by AYY implies a compression factor of $1/3$ for witness encryption, which may be considered surprising. In the other direction, it is well known that in the single input setting, witness encryption implies attribute based encryption [GGH+13]. It is completely unclear however, how to generalize this implication to the multi-input setting – in the setting of single input, the ABE ciphertext contains a WE ciphertext for an NP statement that embeds the attribute. If the attributes are distributed amongst multiple parties, the above approach fails and appears challenging to extend. Thus, miABE implies new results in WE but not the other way around – indeed, in miABE, all encryptors must choose their randomness independently to construct a ciphertext for their respective slot, whereas in WE, there is only one encryptor who constructs the ciphertexts for all slots, making it possible to choose correlated randomness across slots. As we will see, this creates a major technical hurdle in designing miABE, which is not present in WE. Also note that miABE can subsequently be strengthened to miPE using lockable obfuscation, as discussed above.

We also note that single input ABE is the strongest application of the stated definition of WE in [GGH+13]. Since the definition of WE given in [GGH+13] only hides the message in the ciphertext when the statement is not in the language, the notion is insufficient to give any meaningful security guarantee when the statement is actually believed to be true but the witness is not known, such as solutions to some of the Clay Institute Millennium Prize Problems, as discussed in [GGH+13]. Hence, we believe that the primitives of miABE and miPE deserve to be studied even from assumptions that are already known to imply WE, such as evasive LWE [Tsa22, VWW22].

1.3 Technical Overview

Recap of AYY. As observed by AYY, the main difficulty in building an miABE scheme is simultaneously fulfilling two opposing requirements: (1) each encryptor should be able to generate its own ciphertexts independently, (2) these independently generated ciphertexts should permit some kind of "joining" that lets them be viewed as multiple components of a single ABE ciphertext, such that decryption can proceed as in the single input setting. To achieve joining of ciphertext components, existing single input schemes generate multiple ciphertext components using common randomness. However, evidently, two independent sources, each generating an unbounded number of ciphertexts (say Q_1 and Q_2 respectively) cannot even store, much less embed, $Q_1 \cdot Q_2$ random strings in the ciphertexts they compute (even if they share a common PRF key).

In the two-input setting, AYY solve this conundrum by using the beautiful synergy between the algebraic structure offered by lattice based single input ABE schemes and pairing based constructions. This synergy was first discovered and harnessed by Agrawal and Yamada [AY20] in the context of broadcast encryption (a.k.a succinct single input ciphertext policy ABE for NC_1). The work of AYY noticed that the same synergy can be beneficial for the two-input *key* policy ABE setting, albeit for different reasons.

In more detail, AYY achieve the joining of ciphertexts via common randomness by letting each party embed fresh randomness in the exponent of a pairing based group for each ciphertext it computes. Now, party 1 (respectively 2) has Q_1 (respectively Q_2) random elements embedded in its Q_1 (respectively Q_2) ciphertexts. Using the pairing operation, the dercryptor can compute $Q_1 \cdot Q_2$ elements by pairwise multiplication in the exponent. In more detail, for each input, party 1 samples randomness t_1 and encodes it in $\mathbb{G}_1$, party 2 samples randomness t_2 and encodes it in $\mathbb{G}_2$, where $\mathbb{G} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a pairing group with prime order q. Now these ciphertexts may be combined to form a new ciphertext with respect to the randomness $t_1 t_2$ on $\mathbb{G}_T$. This allows to uniquely separate every pair of ciphertexts, since each pair (i, j) where $i \in [Q_1]$ and $j \in [Q_2]$, will have unique randomness $t_1^i t_2^j$. We have by security of pairings that these $Q_1 \cdot Q_2$ correlated terms are indistinguishable from random in the exponent. This allows for generating the requisite randomness and solving the difficulty described above.

Fruitful interplay of pairings and lattices. However, generating joint randomness was not the final goal – the ciphertexts generated using the above joining procedure must behave like an ABE! Note that, having relied on a pairing, whatever we have obtained must live in the exponent of a group. Also note that, pairing based ABE schemes have been rendered unhelpful by this point, since the single multiplication afforded by the pairing has been used up and can no longer participate in the design of the ABE. Here, AYY, similarly to [AY20, AWY20] are rescued by the serendipitously well-fitting structure of a lattice based ABE scheme constructed by Boneh et al [BGG+14]. In [BGG+14] (henceforth BGG+), decryption works as follows: (i) homomorphically compute the circuit f on ciphertext encodings – this step is *linear* even for $f \in \mathsf{P}$, (ii) perform a product of the ciphertext matrix and secret key vector, (iii) round the recovered value to recover the message. Hence, the first two steps can be performed "upstairs" in the exponent and the last step may be performed "downstairs" by recovering the exponent brute force.

Structure of BGG^+*.* Let us recall the structure of the BGG^+ scheme, since this forms the starting point of our construction. As observed in multiple works, in BGG^+, the ciphertext for an attribute $\mathbf{x} \in [\ell]$ in BGG^+ is computed by first generating LWE encodings for all possible values of the attribute $\mathbf{x}$, namely, $\{\psi_{i,b}\}_{i\in[\ell],b\in\{0,1\}}$ and then choosing $\{\psi_{i,x_i}\}_{i\in[\ell]}$ where x_i is the i-th bit of attribute $\mathbf{x}$. Here, $\psi_{i,b} = \mathbf{s}(\mathbf{A}_i - x_{i,b} \cdot \mathbf{G}) + \mathsf{noise}$ where $\mathbf{A}_i \in \mathbb{Z}_q^{n\times m}$ are public matrices, $\mathbf{s} \in \mathbb{Z}_q^n$ is freshly chosen randomness, and $\mathbf{G} \in \mathbb{Z}_q^{n\times m}$ is the special "gadget" matrix which admits a public trapdoor (details not important here). Here, and in the remainder of this overview, we use noise to denote freshly and independently sampled noise terms of appropriate dimension, for each sample. Choosing components based on $\mathbf{x}$ and concatenating the samples yields $\mathbf{s}(\mathbf{A} - \mathbf{x} \otimes \mathbf{G}) + \mathsf{noise}$, where $\mathbf{A} \in \mathbb{Z}_q^{n\times \ell m}$ denotes the concatenation of $\{\mathbf{A}_i\}_{i\in[\ell]}$.

To evaluate a circuit $f \in \mathsf{P}$, BGG^+ observe that there exists an efficiently computable low norm matrix, denoted by $\widehat{\mathbf{H}}_{\mathbf{A},f,\mathbf{x}}$, so that the right multiplication of $(\mathbf{A} - \mathbf{x} \otimes \mathbf{G})$ by $\widehat{\mathbf{H}}_{\mathbf{A},f,\mathbf{x}}$ yields a quantity of the form $\mathbf{A}_f - f(\mathbf{x})\mathbf{G}$ – since the matrix is low norm, this can be right multiplied to $\mathbf{s}(\mathbf{A} - \mathbf{x} \otimes \mathbf{G}) + \mathsf{noise}$ to obtain approximately $\mathbf{s}(\mathbf{A}_f - f(\mathbf{x})\mathbf{G})$ without blowing up the noise. The decryption key for a function f is a low norm vector which, loosely speaking, is used in a matrix vector product that allows to cancel the masking term $\mathbf{s}\mathbf{A}_f$ when $f(\mathbf{x}) = 0$, and this in turn allows to recover the message.

Circling back to AYY, the first encryptor can (roughly speaking) compute $[t_1 \cdot \psi_{\mathbf{x}}]_1$, $[t_1]_1$, the second encryptor can compute $[t_2 \cdot \psi_{\mathbf{y}}]_2$, $[t_2]_2$ and the decryptor can compute $[t_1 t_2 \psi_{\mathbf{x}\|\mathbf{y}}]_T$, $[t_1 t_2]_T$. Note that randomization by $t_1 t_2$ is absolutely essential for security, else the adversary can potentially recover terms like $\mathbf{s}(\mathbf{A} - \mathbf{x} \otimes \mathbf{G}) + \mathsf{noise}$ and $\mathbf{s}(\mathbf{A} - \overline{\mathbf{x}} \otimes \mathbf{G}) + \mathsf{noise}$ in the exponent, which allows to cancel $\mathbf{s}\mathbf{A}$ by subtraction, and leads to a complete break of security. Next, the circuit f can be evaluated in the exponent as described above by right multiplication with a low norm matrix and the secret key can be applied by the matrix vector product to obtain the (scaled) message plus some noise in the exponent. The noise growth can be suitably bounded for the circuit class NC_1, and given $[t_1 t_2]_T$, one can recover the message using brute force discrete log computation.

While AYY takes an important first step towards constructing miABE schemes, it is evident that going beyond degree two is difficult while relying on pairings. Indeed, they do consider arity 3 by additionally relying on ideas from a clever lattice based scheme by Brakerski and Vaikuntanathan [BV22] but this scheme is heuristic, i.e. does not have a proof based on any clean assumption. Thus, it is completely unclear how to go beyond arity 2 using the techniques of AYY, even for NC_1. A natural idea to overcome the barrier of 2 is to rely on lattices in lieu of pairings.

Towards Lattice Based Constructions. Taking a step back, a promising direction would be to consider the lattice adaptation of the Agrawal-Yamada broadcast encryption scheme [AY20] recently proposed by Wee [Wee22]. This construction makes important progress in identifying a clean assumption in the lattice regime that captures the functionality provided by the pairing without relying on bilinear groups, and can be used to construct advanced primitives like broadcast encryption and witness encryption without relying on iO (or the messy assumptions needed to build iO in the post quantum regime). In more detail, Wee [Wee22] suggested two new assumptions – the evasive LWE and tensor LWE and used these to construct ciphertext polict ABE schemes with optimal parameters. We describe his approach next.

Overview of Wee's approach. The main idea of Wee is to cleverly replace the randomization in the exponent by tensoring on the ground. In more detail, Wee observes that the transformation of $(\mathbf{A} - \mathbf{x} \otimes \mathbf{G})$ to $(\mathbf{A}_f - f(\mathbf{x})\mathbf{G})$ via right multiplication by $\widehat{\mathbf{H}}_{\mathbf{A},f,\mathbf{x}}$ is preserved under tensoring with random low norm vectors $\mathbf{r}$. To see this, note that

$$\mathbf{s}(\mathbf{A} \otimes \mathbf{r}^\top) + \mathsf{noise} = \underbrace{\mathbf{s}(\mathbf{I} \otimes \mathbf{r}^\top)}_{\text{Randomized secret}} \mathbf{A} + \mathsf{noise}$$

where the latter quantity can be seen as BGG^+ ciphertext with a tensored LWE secret. This easily implies that homomorphism is preserved even with tensoring as desired. Hence, one can homomorphically evaluate f on $(\mathbf{A} - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{r}^\top$ to obtain $(\mathbf{A}_f - f(\mathbf{x})\mathbf{G}) \otimes \mathbf{r}^\top$ via right multiplication by $\widehat{\mathbf{H}}_{\mathbf{A},f,\mathbf{x}}$.

Importantly, Wee shows that a very natural adaptation of [AY20], obtained by replacing randomization in the exponent by tensoring can be shown secure under a new and elegant assumption, which he calls *evasive* LWE. To support NC_1, he shows that evasive LWE suffices, while to support P, one additionally needs another new assumption, which he calls *tensor* LWE. The formulation of a relatively simple and general assumption in the lattice regime that allows to give a proof for a very natural construction of succint ciphertext policy ABE is a very important contribution which is likely to influence many future lattice constructions, including ours. We describe these assumptions next.

Evasive LWE. The evasive LWE assumption, introduced by Wee [Wee22] (and independently Tsabary [Tsa22]), is a strengthening of the LWE assumption which says that certain extra information, namely Gaussian preimages to LWE public matrices, can only be used in a "semi-honest" way. Recall that the LWE assumption says that

$$(\mathbf{B}, \mathbf{s}\mathbf{B} + \mathbf{e}) \approx_c (\mathbf{B}, \mathbf{c})$$

where $\mathbf{B} \leftarrow \mathbb{Z}_q^{n \times m}$, $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow \chi^m$ for some low norm "noise" distribution χ and $\mathbf{c} \leftarrow \mathbb{Z}_q^m$. Intuitively, the evasive LWE assumption says that if the adversary is additionally given some low norm matrix $\mathbf{K}$ such that $\mathbf{BK} = \mathbf{P}$, which we denote as $\mathbf{B}^{-1}(\mathbf{P})$ (as in the literature, see for instance [Wee22]), for some efficiently sampleable matrix $\mathbf{P}$, then the adversary can exploit this extra information only via the limited means of computing the product $(\mathbf{s}\mathbf{B} + \mathbf{e}) \cdot \mathbf{B}^{-1}(\mathbf{P}) \approx \mathbf{s}\mathbf{P}$ and trying to distinguish this from uniform. The assumption says that this is the only additional capability that the adversary obtains, besides its existing strategies for breaking LWE.

Evidently, the distribution of $\mathbf{P}$ here is of crucial importance – for instance, if $\mathbf{P} = \mathbf{0}$, then $\mathbf{B}^{-1}(\mathbf{P})$ is a trapdoor for $\mathbf{B}$ and can be used to easily break LWE. On the other extreme, if $\mathbf{P}$ is chosen uniformly, then this assumption reduces to standard LWE. The "playing ground" of evasive LWE is in the middle – namely, when it holds that

$$(\mathbf{B}, \mathbf{P}, \mathbf{s}\mathbf{B} + \mathbf{e}, \mathbf{s}\mathbf{P} + \mathbf{e}') \approx_c (\mathbf{B}, \mathbf{P}, \mathbf{c}, \mathbf{c}')$$

then

$$(\mathbf{B}, \mathbf{s}\mathbf{B} + \mathbf{e}, \mathbf{B}^{-1}(\mathbf{P})) \approx_c (\mathbf{B}, \mathbf{c}, \mathbf{B}^{-1}(\mathbf{P})).$$

Here, the former condition is referred to as the PRE condition and the latter as POST. The actual assumption used by the scheme is more complex and includes more LWE samples that use the same secret $\mathbf{s}$ as well as some (carefully chosen) auxiliary information aux. To formalize the PRE condition, the assumption must specify an efficient sampler Samp which outputs the correlated LWE matrices. We defer the formalization to Section 3; here we only remark that the assumption captures in the lattice setting, the guarantees provided by the generic group model for pairings, namely the intuition that an adversary can only use legitimate operations to learn anything. It is therefore very natural (in hidsight) that this assumption should be able to replace the reliance on the generic group model in the constructions of [AY20, AWY20].

Tensor LWE. The tensor LWE assumption states that correlated BGG^+ samples tensored with different random vectors remain pseudorandom. In more detail, for all $\mathbf{x}_1, \cdots, \mathbf{x}_Q \in \{0,1\}^\ell$, it posits that

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

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times \ell m}, \mathbf{s} \leftarrow \mathbb{Z}_q^{mn}, \mathbf{e}_i \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^{\ell m}, \mathbf{r}_i \leftarrow \mathcal{D}_{\mathbb{Z},\gamma}^{m}, \mathbf{c}_i \leftarrow \mathbb{Z}_q^{\ell m}$.

Note that there are no Gaussian preimages in the above assumption. In our work, we show that for the special case where $\mathbf{x}_i = 0\ \forall i \in [Q]$, tensor LWE reduces to standard LWE (Lemma 3.7). In more detail, let $\mathcal{A}$ be an attacker for Tensor LWE with $\mathbf{x}_i = \mathbf{0}$ for all $i \in [Q]$. $\mathcal{A}$ is given either $\mathbf{A}, \{\mathbf{s}(\mathbf{I}_n \otimes \mathbf{r}_i^\top)\mathbf{A} + \mathbf{e}_i, \mathbf{r}_i^\top\}_{i\in[Q]}$ or $\mathbf{A}, \{\mathbf{c}_i, \mathbf{r}_i^\top\}_{i\in[Q]}$. We prove that under the LWE assumption, $\mathcal{A}$ has a negligible probability of distinguishing the left hand side from the right hand side. This implication was not known before, and increases our confidence in the assumption, which is new and not so well studied. Please see Lemma 3.7 for details.

Generalizing Tensor LWE. While tensor LWE as stated by Wee suffices for our construction of 2-ABE for P, for extending the arity to any constant k, we require a strengthening of this assumption. In more detail, we require that for all $\mathbf{x}_{j_1,\ldots,j_k} \in \{0,1\}^\ell$ indexed by $j_1, \ldots, j_k \in [Q]$, it holds that:

$$\begin{aligned}&\mathbf{A}, \left\{\mathbf{s}(\mathbf{I}_n \otimes \mathbf{r}_{1,j_1}^\top \otimes \cdots \otimes \mathbf{r}_{k,j_k}^\top)(\mathbf{A} - \mathbf{x}_{j_1,\ldots,j_k} \otimes \mathbf{G}) + \mathbf{e}_{j_1,\ldots,j_k}, \mathbf{r}_{i,j_i}\right\}_{i\in[k],j_1,\ldots,j_k\in[Q]}\\ \approx_c\quad &\mathbf{A}, \{\mathbf{c}_{i,j_i}, \mathbf{r}_{i,j_i}\}_{i\in[k],j_1,\ldots,j_k\in[Q]}\end{aligned}$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times \ell m}, \mathbf{s} \leftarrow \mathbb{Z}_q^{nm^k}, \mathbf{e}_{j_1,\ldots,j_k} \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^{\ell m}, \mathbf{r}_{i,j_i} \leftarrow \mathcal{D}_{\mathbb{Z},\gamma}^{m}, \mathbf{c}_{i,j_i} \leftarrow \mathbb{Z}_q^{\ell m}$.

It is easy to see that the generalized tensor LWE yields Wee's version of tensor LWE for $k = 1$.

Two Input ABE from evasive and tensor LWE. As a warmup, we first describe our construction of miABE for arity 2. For NC_1, our construction can be proven secure by relying solely on evasive LWE while for P, we additionally need tensor LWE. We will show subsequently how to generalize this to any constant arity. In this work, we consider a modified syntax of miABE where there is only a single encryption slot which is public key, and multiple key generation slots, which require the master secret key. This syntax better fits our construction and easily implies the standard definition of miABE which has multiple encryptors that have as input the master secret key, and a single key generator who also requires the master secret key – please see Section 2.1 for details.

Given the above discussion, a natural approach to construct miABE schemes from lattices is to try adapting the ideas in AYY by replacing the use of pairings with tensoring, analogously to Wee's approach of adapting the Agrawal-Yamada broadcast encryption scheme to lattices in Wee. We show that in the end, this approach indeed can be made to work, but via several failed attempts which require new techniques to overcome, and a complex security proof, which requires proving several new lemmas. Below, we outline the pathway to our final construction, detailing the hurdles we encounter and the ideas towards their resolution.

Attempt 1. We attempt to design a scheme using tensor based randomization from Wee to instantiate the template of AYY. We sketch the construction at a high level below. We suppress dimensions for ease of readability in this overview.

1. The master public key is $(\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, \mathbf{u})$ where $\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}$ are sampled uniformly and $\mathbf{u}$ is sampled from the discrete Gaussian distribution. The master key is a trapdoor for $\mathbf{A}_0$ and a trapdoor for $\mathbf{B}$.

2. The encryptor, given input $(\mathbf{x}, \mu)$ where $\mathbf{x}$ is the attribute and μ is the message, samples randomness $\mathbf{s}$ along with requisite noise terms and computes

$$\underbrace{\mathbf{s}\mathbf{A}_0 + \mathsf{noise}}_{\mathbf{c}_0},\ \underbrace{\mathbf{s}\big((\mathbf{A}_1 - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{I}\big) + \mathsf{noise}}_{\mathbf{c}_1},\ \underbrace{\mathbf{s}(\mathbf{G}\mathbf{u}^\top \otimes \mathbf{I}) + \mathsf{noise}}_{\mathbf{c}_2},\ \underbrace{\mathbf{s}\mathbf{B} + \mathsf{noise}}_{\mathbf{c}_3}$$

   if $\mu = 0$ and else samples random elements of appropriate dimensions if $\mu = 1$. Note that the encryption procedure is public key.

3. The first key generator (to be interpreted as the second encryptor), given input msk and attribute $\mathbf{y}$ samples Gaussian random vector $\mathbf{r}$ and computes

$$\mathsf{sk}_\mathbf{y} = \mathbf{B}^{-1}\big((\mathbf{A}_2 - \mathbf{y} \otimes \mathbf{G}) \otimes \mathbf{r}^\top\big), \mathbf{r}^\top$$

   It outputs this as the secret key for $\mathbf{y}$. Note that the randomizer $\mathbf{r}$ is used to prevent collusion attacks – in its absence, an attacker can obtain samples corresponding to $\mathbf{y}$ and $\overline{\mathbf{y}}$ (i.e. complement of $\mathbf{y}$) and launch attack as discussed earlier.

4. The second key generator, given msk and function f as input computes $\mathsf{sk}_f = (\mathbf{A}_0 \| \mathbf{A}_f)^{-1}(\mathbf{G}\mathbf{u}^\top)$ and outputs this as the secret key for f.

5. The decryptor does the following:

   (a) *Computing ciphertext component for second attribute:* It combines the ciphertext $\mathbf{c}_3$ with the first secret key $\mathsf{sk}_\mathbf{y}$ to obtain $\mathbf{s}\big((\mathbf{A}_2 - \mathbf{y} \otimes \mathbf{G}) \otimes \mathbf{r}^\top\big) + \mathsf{noise}$.

   (b) *Randomizing ciphertext component for first attribute:* From $\mathbf{c}_1$ and $\mathsf{sk}_\mathbf{y}$, it computes $\big(\mathbf{s}\big((\mathbf{A}_1 - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{I}\big) + \mathsf{noise}\big)(\mathbf{I} \otimes \mathbf{r}^\top) = \mathbf{s}\big((\mathbf{A}_1 - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{r}^\top\big) + \mathsf{noise}$

   (c) *Producing a complete* BGG^+ *ciphertext:* Concatenating the results of the previous two steps, we get

$$\mathbf{s}\big((\mathbf{A}_1 \| \mathbf{A}_2) - (\mathbf{x} \| \mathbf{y}) \otimes \mathbf{G}) \otimes \mathbf{r}^\top\big) + \mathsf{noise}$$

   Note that this looks exactly like a BGG^+ sample except for the tensoring with $\mathbf{r}^\top$. As discussed above, Wee shows that homomorphic computation is preserved under right tensoring with $\mathbf{r}^\top$.

   (d) BGG^+ *Homomorphic evaluation:* Computing the circuit f homomorphically on this BGG^+ sample, we obtain

$$\mathbf{s}\big((\mathbf{A}_f - f(\mathbf{x}, \mathbf{y})\mathbf{G}) \otimes \mathbf{r}^\top\big) + \mathsf{noise}$$

   If $f(\mathbf{x}, \mathbf{y}) = 0$, then we get $\mathbf{s}\big(\mathbf{A}_f \otimes \mathbf{r}^\top\big) + \mathsf{noise}$. Concatenating with the ciphertext component $\mathbf{c}_0$, we get

$$\mathbf{s}\big(\mathbf{A}_0 \| \mathbf{A}_f) \otimes \mathbf{r}^\top\big) + \mathsf{noise}$$

   (e) *Applying* BGG^+ *secret key.* By right multiplying the second slot secret key $(\mathbf{A}_0 \| \mathbf{A}_f)^{-1}(\mathbf{G}\mathbf{u}^\top) \otimes \mathbf{I}$ to this, we get

$$\mathbf{s}\big(\mathbf{G}\mathbf{u}^\top \otimes \mathbf{r}^\top\big) + \mathsf{noise}$$

(f) BGG^+ *decryption with tensoring.* Multiplying $\mathbf{c}_2$ with $\mathbf{I} \otimes \mathbf{r}^\top$, we get $\mathbf{s}\big(\mathbf{G}\mathbf{u}^\top \otimes \mathbf{r}^\top\big) + \mathsf{noise}$. Subtracting from the output of the previous step, we get a small value when $\mu = 0$. Thus, we recover μ when $f(\mathbf{x}, \mathbf{y}) = 0$.

The above scheme provides functionality and does not appear to have any immediate attacks. However, we are unable to prove security of this scheme based on the evasive/tensor LWE assumption. This is because the evasive LWE assumption accommodates Gaussian preimages for fixed matrices, namely terms of the form $\mathbf{B}^{-1}(\mathbf{P})$, where $\mathbf{B}$ is a random matrix and $\mathbf{P}$ is structured, but does not know how to handle terms such as $(\mathbf{A}_0 \| \mathbf{A}_f)^{-1}(\mathbf{G}\mathbf{u}^\top)$. Since $\mathbf{A}_f$ is highly structured, this is incompatible with the assumption.

Attempt 2. To handle this barrier, in our next attempt, we use an idea by Wee to remove the problematic term $(\mathbf{A}_0 \| \mathbf{A}_f)^{-1}(\mathbf{G}\mathbf{u}^\top)$. Note that the purpose of this term is to create an LWE sample with secret $\mathbf{s}$ and matrix $\mathbf{A}_f$. In more detail, as shown in step 5e, the term $\mathbf{s}\big((\mathbf{A}_0 \| \mathbf{A}_f) \otimes \mathbf{r}^\top\big) + \mathsf{noise}$ obtained by homomorphic evaluation is combined together with the secret key in the second slot $(\mathbf{A}_0 \| \mathbf{A}_f)^{-1}(\mathbf{G}\mathbf{u}^\top)$ to obtain $\mathbf{s}(\mathbf{G}\mathbf{u}^\top \otimes \mathbf{r}^\top) + \mathsf{noise}$. As shown in step 5f, this term is then used to unmask the ramdomized $\mathbf{c}_2$, i.e. $\mathbf{s}\big(\mathbf{G}\mathbf{u}^\top \otimes \mathbf{r}^\top\big) + \mathsf{noise}$ by subtraction to recover μ.

So as to do away with the requirement of revealing $(\mathbf{A}_0 \| \mathbf{A}_f)^{-1}(\mathbf{G}\mathbf{u}^\top)$, we provide an alternate route to recover μ. We change the second slot secret key sk_f to $\mathbf{B}^{-1}(\mathbf{A}_f\mathbf{u} \otimes \mathbf{I})$, and use this together with the term $\mathbf{s}\mathbf{B} + \mathsf{noise}$ provided in the ciphertext to obtain $\mathbf{s}(\mathbf{A}_f\mathbf{u} \otimes \mathbf{I}) + \mathsf{noise}$. This allows us to cancel the mask $\mathbf{s}\mathbf{A}_f$ obtained via homomorphic evaluation and brings us closer to relying only on evasive and tensor LWE.

Below, we detail only the modifications we make to our previous attempt:

1. The encryptor, given input $(\mathbf{x}, \mu)$ where $\mathbf{x}$ is the attribute and μ is the message, samples randomness $\mathbf{s}$ along with requisite noise terms and computes

$$\underbrace{\cancel{\mathbf{s}\mathbf{A}_0 + \mathsf{noise}}}_{\mathbf{c}_0},\ \underbrace{\mathbf{s}\big((\mathbf{A}_1 - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{I}\big) + \mathsf{noise}}_{\mathbf{c}_1},\ \underbrace{\cancel{\mathbf{s}(\mathbf{G}\mathbf{u}^\top \otimes \mathbf{I}) + \mathsf{noise}}}_{\mathbf{c}_2},\ \underbrace{\mathbf{s}\mathbf{B} + \mathsf{noise}}_{\mathbf{c}_3}$$

   if $\mu = 0$ else samples random elements of appropriate dimensions if $\mu = 1$.

2. The second key generator, given msk and function f computes $\mathsf{sk}_f = \mathbf{B}^{-1}(\mathbf{A}_f\mathbf{u}^\top \otimes \mathbf{I})$. At this junction, we let $\mathbf{u}$ be chosen independently by each user instead of fixing it in the public parameters to prevent the adversary from requesting keys for correlated functions and obtaining correlated LWE samples of the form $\mathbf{s}\mathbf{A}_f\mathbf{u}^\top + \mathsf{noise}$ with the same $\mathbf{u}$ and same $\mathbf{s}$.

3. During decryption,

   (a) BGG^+ homomorphic evaluation is simplified. We only compute the circuit f homomorphically on this BGG^+ sample, to obtain

$$\mathbf{s}\big((\mathbf{A}_f - f(\mathbf{x}, \mathbf{y})\mathbf{G}) \otimes \mathbf{r}^\top\big) + \mathsf{noise}$$

   If $f(\mathbf{x}, \mathbf{y}) = 0$, then we get $\mathbf{s}\big(\mathbf{A}_f \otimes \mathbf{r}^\top\big) + \mathsf{noise}$. There is no need to concatenate with $\mathbf{c}_0$ (this is no longer even provided) but we must right multiply by $(\mathbf{u}^\top \otimes \mathbf{I})$ to obtain $\mathbf{s}\big(\mathbf{A}_f\mathbf{u}^\top \otimes \mathbf{r}^\top\big) + \mathsf{noise}$. Recall that $\mathbf{u}$ is low norm, hence does not blow up the noise.

   (b) The second slot key $\mathbf{B}^{-1}(\mathbf{A}_f\mathbf{u}^\top \otimes \mathbf{I})$ is right multiplied to $\mathbf{c}_3$ to get $\mathbf{s}(\mathbf{A}_f\mathbf{u}^\top \otimes \mathbf{I}) + \mathsf{noise}$. By right multiplying with $(\mathbf{I} \otimes \mathbf{r}^\top)$, we now recover the masking term $\mathbf{s}\big(\mathbf{A}_f\mathbf{u} \otimes \mathbf{r}^\top\big) + \mathsf{noise}$ which can be subtracted from the output of the previous step. If this is small, learn that $\mu = 0$.

Importantly, at this point, we can hope to use evasive LWE to “get rid” of the preimages $\mathbf{B}^{-1}\big((\mathbf{A}_2 - \mathbf{y} \otimes \mathbf{G}) \otimes \mathbf{r}^\top\big)$ and $\mathbf{B}^{-1}(\mathbf{A}_f\mathbf{u}^\top \otimes \mathbf{I})$ from the distribution seen by the adversary. This essentially reduces the task of proving the security of the scheme to that of proving the pseudorandomness of the terms

$$\mathbf{s}((\mathbf{A}_1 - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{I}) + \mathsf{noise}, \;\; \mathbf{s}((\mathbf{A}_2 - \mathbf{y} \otimes \mathbf{G}) \otimes \mathbf{r}^\top) + \mathsf{noise}, \;\; \mathbf{s}(\mathbf{A}_f\mathbf{u}^\top \otimes \mathbf{I}) + \mathsf{noise}$$

Unfortunately, we are still not done, even by relying additionally on tensor LWE. This is because tensor LWE only posits pseudorandomness of LWE samples with respect to secret $\mathbf{s}(\mathbf{I} \otimes \mathbf{r})$. In particular, the presence of the terms $\mathbf{s}((\mathbf{A}_1 - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{I}) + \mathsf{noise}$ and $\mathbf{s}(\mathbf{A}_f\mathbf{u}^\top \otimes \mathbf{I}) + \mathsf{noise}$ cannot be handled by invoking tensor LWE since they do not have the right form (in particular no $\mathbf{r}$ term appears in these). Therefore, we must handle these next.

Attempt 3. Let us first explain how to deal with the first term $\mathbf{s}((\mathbf{A}_1 - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{I}) + \mathsf{noise}$. As in Wee, the idea is to “mask” the problematic term, in this case, $\mathbf{s}((\mathbf{A}_1 - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{I}) + \mathsf{noise}$, with a pseudorandom term $\mathbf{s}_0(\mathbf{A}_0 \otimes \mathbf{I}) + \mathsf{noise}$ such that there is a way to provide an “unmasking” term using which, we can recover a simulatable term $\mathbf{s}((\mathbf{A}_1 - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{r}^\top) + \mathsf{noise}$ but nothing else is revealed [3].

In more detail, we make the following changes:

1. We replace $\mathbf{s}((\mathbf{A}_1 - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{I}) + \mathsf{noise}$ by $\mathbf{c} = \mathbf{s}((\mathbf{A}_1 - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{I}) + \mathbf{s}_0(\mathbf{A}_0 \otimes \mathbf{I}) + \mathsf{noise}$.
2. Next, we put some terms so that the ciphertext along with the first slot of the secret key jointly generates $\mathbf{d} := \mathbf{s}_0(\mathbf{A}_0 \otimes \mathbf{r}^\top) + \mathsf{noise}$, which is an “unmasking” term.
3. To obtain the desired term, we compute $\mathbf{c}(\mathbf{I} \otimes \mathbf{r}^\top) - \mathbf{d} = \mathbf{s}((\mathbf{A}_1 - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{r}) + \mathsf{noise}$.

Furthermore, it is easy to show that $\mathbf{s}_0(\mathbf{A}_0 \otimes \mathbf{I}) + \mathsf{noise}$ is pseudorandom by LWE (since $\mathbf{s}_0$ is a fresh randomness introduced only for this specific purpose), which implies that $\mathbf{c}$ is also pseudorandom. This allows us to conclude that $\mathbf{d}$ does not reveal anything more than the desired term, since $\mathbf{c}$ and the desired term determine $\mathbf{d}$.

At this stage, the scheme looks like the following, where for brevity we again omit to mention components that are unchanged.

1. The encryptor computes $\mathbf{c}_1 = (\mathbf{s}, \mathbf{s}_0)\begin{pmatrix} (\mathbf{A}_1 - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{I} \\ \mathbf{A}_0 \otimes \mathbf{I} \end{pmatrix} + \mathsf{noise}$ and $\mathbf{c}_2 = (\mathbf{s}, \mathbf{s}_0)\mathbf{B} + \mathsf{noise}$ for $\mu = 0$ (and random elements for $\mu = 1$).
2. The first slot key is $\mathsf{sk}_\mathbf{y} \leftarrow \mathbf{B}^{-1}\begin{pmatrix} (\mathbf{A}_2 - \mathbf{y} \otimes \mathbf{G}) \otimes \mathbf{r}^\top & \\ & \mathbf{A}_0 \otimes \mathbf{r}^\top \end{pmatrix}$
3. The second slot key is $\mathsf{sk}_f \leftarrow \mathbf{B}^{-1}\begin{pmatrix} \mathbf{A}_f\mathbf{u}^\top \otimes \mathbf{I} \\ \mathbf{0} \end{pmatrix}$ and $\mathbf{u}$. This key is essentially unchanged except padding the inner matrix with zeroes to account for the longer secret.
4. Now, from the ciphertext component $\mathbf{c}_2$ and the first slot key, we get terms $\mathbf{s}(\mathbf{A}_2 - \mathbf{y} \otimes \mathbf{G}) \otimes \mathbf{r}^\top) + \mathsf{noise}$ and $\mathbf{d} = \mathbf{s}_0(\mathbf{A}_0 \otimes \mathbf{r}^\top) + \mathsf{noise}$. The second term $\mathbf{d}$ is the new term that we will make use of as described above.

[3] The informed reader may notice the similarity with randomized encodings [AIK04] and pair/predicate encodings [Att14, Wee14].

5. Now, we compute $\mathbf{c}(\mathbf{I} \otimes \mathbf{r}^\top) - \mathbf{d} = \mathbf{s}((\mathbf{A}_1 - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{r}) + \mathsf{noise}$. Using pseudorandomness of $\mathbf{c}$, we can argue that $\mathbf{d}$ did not reveal anything except $\mathbf{s}((\mathbf{A}_1 - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{r}) + \mathsf{noise}$.

At this stage, we obtained a term that tensor LWE can handle, namely $\mathbf{s}((\mathbf{A}_1 - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{r}) + \mathsf{noise}$.

Attempt 4. Next, we must deal with the second problematic term $\mathbf{s}(\mathbf{A}_f \mathbf{u}^\top \otimes \mathbf{I}) + \mathsf{noise}$. It is tempting to try the same strategy as above but unfortunately, this does not work. To see why, let us try to replace $\mathbf{s}(\mathbf{A}_f \mathbf{u}^\top \otimes \mathbf{I}) + \mathsf{noise}$ with $\mathbf{s}(\mathbf{A}_f \mathbf{u}^\top \otimes \mathbf{I}) + \mathbf{s}_1(\mathbf{D} \otimes \mathbf{I}) + \mathsf{noise}$, where $\mathbf{D}$ is some fixed matrix. We can then modify the scheme so that the ciphertext along with the first slot secret key generate the unmasking term $\mathbf{s}_1(\mathbf{D} \otimes \mathbf{r}^\top) + \mathsf{noise}$. Similarly to the above, this allows us to derive the desired term $\mathbf{s}(\mathbf{A}_f \mathbf{u}^\top \otimes \mathbf{r}^\top) + \mathsf{noise}$ which can be handled by tensor LWE. One may hope that this suffices to prove security.

However, we run into another problem, namely, that of collusion resistance. In particular, an adversary may make multiple key queries for the second slot and use the same ciphertext and first slot key for decryption. These allow her to recover $\mathbf{s}(\mathbf{A}_f \mathbf{u}^\top \otimes \mathbf{I}) + \mathbf{s}_1(\mathbf{D} \otimes \mathbf{r}^\top) + \mathsf{noise}$ and $\mathbf{s}(\mathbf{A}_{f'} \mathbf{u'}^\top \otimes \mathbf{I}) + \mathbf{s}_1(\mathbf{D} \otimes \mathbf{r}^\top) + \mathsf{noise}$ for different f and f'. Even though we want to hide two terms $\mathbf{s}(\mathbf{A}_f \mathbf{u}^\top \otimes \mathbf{I})$ and $\mathbf{s}(\mathbf{A}_{f'} \mathbf{u'}^\top \otimes \mathbf{I})$, there is only a single masking term $\mathbf{s}_1(\mathbf{D} \otimes \mathbf{r}^\top) + \mathsf{noise}$, since $\mathbf{s}_1$ would be chosen by the encryptor and $\mathbf{r}$ by the first slot key – this is clearly problematic.

To fix this, we ensure that the masking term is randomized by a user specific randomness corresponding to the second slot key. Namely, we replace $\mathbf{s}(\mathbf{A}_f \mathbf{u}^\top \otimes \mathbf{I}) + \mathbf{s}_1(\mathbf{D} \otimes \mathbf{I}) + \mathsf{noise}$ with $\mathbf{s}(\mathbf{A}_f \mathbf{u}^\top \otimes \mathbf{I}) + \mathbf{s}_1(\mathbf{D} \otimes \mathbf{t}^\top) + \mathsf{noise}$, where $\mathbf{t}$ is user specific randomness. We then use the ideas discussed previously to ensure that the ciphertext and second slot key generate $\mathbf{s}_1(\mathbf{D} \otimes \mathbf{t}^\top) + \mathsf{noise}$. This mask is removed similarly to the previous case and we may obtain $\mathbf{s}(\mathbf{A}_f \mathbf{u}^\top \otimes \mathbf{I}) + \mathsf{noise}$.

Attempt 5. Unfortunately, this still does not suffice. Recall that we wanted to generate the term $\mathbf{s}(\mathbf{A}_f \mathbf{u}^\top \otimes \mathbf{r}^\top) + \mathsf{noise}$ in order to invoke tensor LWE, which the above term does not let us do. To achieve this, we replace $\mathbf{s}(\mathbf{A}_f \mathbf{u}^\top \otimes \mathbf{I}) + \mathbf{s}_1(\mathbf{D} \otimes \mathbf{t}) + \mathsf{noise}$ with $\mathbf{s}(\mathbf{A}_f \mathbf{u}^\top \otimes \mathbf{I}) + \mathbf{s}_1(\mathbf{D} \otimes \mathbf{t} \otimes \mathbf{I}) + \mathsf{noise}$, i.e., we added some space to further randomize the masking term with $\mathbf{r}^\top$. We then let the ciphertext and secret keys for both slots jointly generate $\mathbf{s}_1(\mathbf{D} \otimes \mathbf{t}^\top \otimes \mathbf{r}^\top) + \mathsf{noise}$.

To do so, we do the following:

1. Include $\mathbf{s}_1 \mathbf{B} + \mathsf{noise}$ in the ciphertext and $\mathbf{B}^{-1}(\mathbf{C} \otimes \mathbf{r}^\top)$ in the first slot key. Multiplying them yields $\mathbf{s}_1(\mathbf{C} \otimes \mathbf{r}^\top) + \mathsf{noise}$.
2. Include $\mathbf{C}^{-1}(\mathbf{D} \otimes \mathbf{t}^\top)$ in the second slot key.

Putting these together enables us to recover the masking term as:

$$\begin{aligned}
(\mathbf{s}_1(\mathbf{C} \otimes \mathbf{r}^\top) + \mathsf{noise}) \cdot \mathbf{C}^{-1}(\mathbf{D} \otimes \mathbf{t}^\top) &= \mathbf{s}_1(\mathbf{I} \otimes \mathbf{r}^\top)\mathbf{C} \cdot \mathbf{C}^{-1}(\mathbf{D} \otimes \mathbf{t}^\top) + \mathsf{noise} \\
&= \mathbf{s}_1(\mathbf{I} \otimes \mathbf{r}^\top)(\mathbf{D} \otimes \mathbf{t}^\top) + \mathsf{noise} \\
&= \mathbf{s}_1(\mathbf{D} \otimes \mathbf{t}^\top \otimes \mathbf{r}^\top) + \mathsf{noise}
\end{aligned}$$

The above term contains randomness $\mathbf{s}_1$ chosen by the encryptor, $\mathbf{r}$ chosen by the first slot key and $\mathbf{t}$ chosen by the second slot key. Intuitively, this randomness triple separates the triple of ciphertext, first key and second key, from any other triple even if some components of the triple are reused. This allows to separate the "thread" of computation corresponding to a given triple, from all other threads, and hopefully allows us to prove security. This brings us to our final scheme.

We provide the complete construction below. The vector $\mathbf{u}$ above is now changed to a matrix $\mathbf{U}$ for syntactic reasons.

1. Set $\mathsf{mpk} = (\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, \mathbf{C}, \mathbf{D})$, and msk as trapdoors for $\mathbf{B}$ and $\mathbf{C}$.
2. To encrypt a message μ against attribute $\mathbf{x}$, do the following. If $\mu = 0$, do:
   (a) Compute $\mathbf{c}_1 = (\mathbf{s}, \mathbf{s}_0) \begin{pmatrix} (\mathbf{A}_1 - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{I} \\ \mathbf{A}_0 \otimes \mathbf{I} \end{pmatrix} + \mathsf{noise}$

   (b) Compute $\mathbf{c}_2 = (\mathbf{s}, \mathbf{s}_0, \mathbf{s}_1)\mathbf{B} + \mathsf{noise}$

   (c) Output $\mathsf{ct}_{\mathbf{x}} = (\mathbf{c}_1, \mathbf{c}_2)$

   If $\mu = 1$, output random elements in the appropriate space.
3. To compute the first slot key for attribute $\mathbf{y}$, sample
$$\mathsf{sk}_{\mathbf{y}} \leftarrow \mathbf{B}^{-1} \begin{pmatrix} (\mathbf{A}_2 - \mathbf{y} \otimes \mathbf{G}) \otimes \mathbf{r}^\top & & \\ & \mathbf{A}_0 \otimes \mathbf{r}^\top & \\ & & \mathbf{C} \otimes \mathbf{r}^\top \end{pmatrix}$$
4. To compute the second slot key for function f, sample $\mathbf{U}$, $\mathbf{t}$ and compute
$$\mathsf{sk}_f \leftarrow \mathbf{B}^{-1} \begin{pmatrix} \mathbf{A}_f \mathbf{U} \otimes \mathbf{I} \\ \mathbf{0} \\ \mathbf{D} \otimes \mathbf{t}^\top \otimes \mathbf{I} \end{pmatrix}, \quad \mathbf{C}^{-1}\left(\mathbf{D} \otimes \mathbf{t}^\top\right), \quad \mathbf{U}, \quad \mathbf{t}$$

To decrypt, first compute $\mathbf{d}_1 = \mathbf{c}_1(\mathbf{I} \otimes \mathbf{r}^\top)$, $(\mathbf{d}_2,\ \mathbf{d}_3,\ \mathbf{d}_4) = \mathbf{c}_2 \cdot \mathsf{sk}_{\mathbf{y}}$, $\mathbf{d}_5 = \mathbf{c}_2 \cdot \mathsf{sk}_{f,1}$, $\mathbf{d}_6 = \mathbf{d}_5(\mathbf{I} \otimes \mathbf{r}^\top)$ and $\mathbf{d}_7 = \mathbf{d}_4 \cdot \mathsf{sk}_{f,2}$. Then compute $\mathbf{d}_8 = \mathbf{d}_1 - \mathbf{d}_3$, $\mathbf{d}_9 = (\mathbf{d}_8 \| \mathbf{d}_2)\widehat{\mathbf{H}}_{(\mathbf{A}_1 \| \mathbf{A}_2), f, (\mathbf{x} \| \mathbf{y})}\mathbf{U}$, $\mathbf{d}_{10} = \mathbf{d}_6 - \mathbf{d}_7$. Finally, if $\mathbf{d}_{10} - \mathbf{d}_9 \approx 0$, then output 0, else 1. To see the correctness, observe:

$$\mathbf{d}_1 = \mathbf{s}((\mathbf{A}_1 - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{r}^\top) + \mathbf{s}_0(\mathbf{A}_0 \otimes \mathbf{r}^\top) + \mathsf{noise},$$
$$\mathbf{d}_2 = \mathbf{s}((\mathbf{A}_2 - \mathbf{y} \otimes \mathbf{G}) \otimes \mathbf{r}^\top) + \mathsf{noise},$$
$$\mathbf{d}_3 = \mathbf{s}_0(\mathbf{A}_0 \otimes \mathbf{r}^\top) + \mathsf{noise}, \quad \mathbf{d}_4 = \mathbf{s}_1(\mathbf{C} \otimes \mathbf{r}^\top) + \mathsf{noise},$$
$$\mathbf{d}_5 = \mathbf{s}(\mathbf{A}_f \mathbf{U} \otimes \mathbf{I}) + \mathbf{s}_1(\mathbf{D} \otimes \mathbf{t}^\top \otimes \mathbf{I}) + \mathsf{noise}, \quad \mathbf{d}_6 = \mathbf{s}(\mathbf{A}_f \mathbf{U} \otimes \mathbf{r}^\top) + \mathbf{s}_1(\mathbf{D} \otimes \mathbf{t}^\top \otimes \mathbf{r}^\top) + \mathsf{noise},$$
$$\mathbf{d}_7 = \mathbf{d}_4 \cdot \mathbf{C}^{-1}(\mathbf{D} \otimes \mathbf{t}^\top) = \mathbf{s}_1(\mathbf{D} \otimes \mathbf{t}^\top \otimes \mathbf{r}^\top) + \mathsf{noise}, \quad \mathbf{d}_8 = \mathbf{s}((\mathbf{A}_1 - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{r}^\top) + \mathsf{noise}$$
$$\mathbf{d}_9 = \mathbf{s}((\mathbf{A}_f - f(\mathbf{x}, \mathbf{y})\mathbf{G}) \otimes \mathbf{r}^\top)\mathbf{U} + \mathsf{noise}, \quad \mathbf{d}_{10} = \mathbf{s}(\mathbf{A}_f \mathbf{U} \otimes \mathbf{r}^\top) + \mathsf{noise}$$

If $f(\mathbf{x}, \mathbf{y}) = 0$, then $\mathbf{d}_{10} - \mathbf{d}_9 = \mathsf{noise}$ when $\mu = 0$, else it is large. Above, the terms $\mathbf{d}_2, \mathbf{d}_8$ in blue mimic the ciphertext components of single input BGG^+, computed as if with shared randomness by a single party holding both $\mathbf{x}$ and $\mathbf{y}$. Note that all the machinery developed above was to be able to simulate the single party setting in the two party setting, where the ciphertexts are produced using independent randomness.

Proof Sketch. For ease of exposition, we sketch the proof for the case where only a single key is generated for both the slots. First, we observe that we need to invoke evasive LWE twice, once to handle terms $\mathbf{B}^{-1}(\cdot)$ and once for $\mathbf{C}^{-1}(\cdot)$. Of these, the first application is standard, following Wee while the second one requires more care as it uses a structured LWE, as in [VWW22].

Having removed Gaussian preimages with respect to $\mathbf{B}$ and $\mathbf{C}$, we are required to show pseudorandomness of the following terms:

$$\mathbf{c}_1 = \mathbf{s}((\mathbf{A}_1 - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{I}) + \mathbf{s}_0(\mathbf{A}_0 \otimes \mathbf{I}) + \mathsf{noise}, \quad \mathbf{c}_2 = (\mathbf{s}, \mathbf{s}_0, \mathbf{s}_1)\mathbf{B} + \mathsf{noise},$$
$$\mathbf{c}_3 = \mathbf{s}((\mathbf{A}_2 - \mathbf{y} \otimes \mathbf{G}) \otimes \mathbf{r}^\top) + \mathsf{noise} \quad \mathbf{c}_4 = \mathbf{s}_0(\mathbf{A}_0 \otimes \mathbf{r}^\top) + \mathsf{noise},$$
$$\mathbf{c}_5 = \mathbf{s}_1(\mathbf{C} \otimes \mathbf{r}^\top) + \mathsf{noise}, \quad \mathbf{c}_6 = \mathbf{s}(\mathbf{A}_f \mathbf{U} \otimes \mathbf{I}) + \mathbf{s}_1(\mathbf{D} \otimes \mathbf{t}^\top \otimes \mathbf{I}) + \mathsf{noise}$$
$$\mathbf{c}_7 = \mathbf{s}_1(\mathbf{D} \otimes \mathbf{t}^\top \otimes \mathbf{r}^\top) + \mathsf{noise}$$

Above, note that $\mathbf{c}_3, \mathbf{c}_4, \mathbf{c}_5$ are generated using the secret key for the first slot and the ciphertext, $\mathbf{c}_6$ is generated using the ciphertext and secret key of the second slot, and $\mathbf{c}_7$ is generated using evasive LWE with structured secret, namely by combining $\mathbf{C}^{-1}(\mathbf{D} \otimes \mathbf{t}^\top)$ and $\mathbf{c}_5 = \mathbf{s}_1(\mathbf{I} \otimes \mathbf{r}^\top)\mathbf{C} + \mathsf{noise}$. This yields $\mathbf{s}_1(\mathbf{I} \otimes \mathbf{r}^\top)(\mathbf{D} \otimes \mathbf{t}^\top) + \mathsf{noise}$ which is equal to $\mathbf{c}_7$.

We now proceed to sketch the hybrid structure of the proof.

Game 0: This is the real game.

Game 1: Express $\mathbf{c}_4$ in terms of $\mathbf{c}_1$ and a term that tensor LWE can handle:

$$\mathbf{c}_4 = \mathbf{c}_1(\mathbf{I} \otimes \mathbf{r}^\top) - \underbrace{\left(\mathbf{s}((\mathbf{A}_1 - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{r}^\top) + \mathsf{noise}\right)}_{\mathbf{c}4'}$$

The only difference between Game 0 and Game 1 is the distribution of the noise term which can be handled by noting that $\mathbf{c}_1(\mathbf{I} \otimes \mathbf{r}^\top) \approx \mathbf{s}((\mathbf{A}_1 - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{r}^\top) + \mathbf{s}_0(\mathbf{A}_0 \otimes \mathbf{r}^\top)$ and using the standard smudging lemma(Lemma 2.10).

Game 2: We now change $\mathbf{c}_1$ and $\mathbf{c}_2$ to random by using the power of LWE with secret $\mathbf{s}_0$.

Game 3: Now, we express $\mathbf{c}_7$ in terms of $\mathbf{c}_6$ and a term which is friendly with tensor LWE:

$$\mathbf{c}_7 = \mathbf{c}_6(\mathbf{I} \otimes \mathbf{r}^\top) - \underbrace{\mathbf{s}(\mathbf{A}_f\mathbf{U} \otimes \mathbf{r}^\top) + \mathsf{noise}}_{\mathbf{c}_7'}$$

Again, the change follows using the smudging lemma.

Game 4: Change $\mathbf{c}_5$ and $\mathbf{c}_6$ to random. Note that $\mathbf{c}_6(\mathbf{I} \otimes \mathbf{r}^\top) \approx \mathbf{s}(\mathbf{A}_f\mathbf{U} \otimes \mathbf{r}^\top) + \mathbf{s}_1(\mathbf{D} \otimes \mathbf{t}^\top \otimes \mathbf{r}^\top) + \mathsf{noise}$ and $\mathbf{c}_5 = \mathbf{s}_1(\mathbf{C} \otimes \mathbf{r}^\top) + \mathsf{noise}$. Hence, it suffices to show pseudorandomness of

$$\mathbf{s}_1(\mathbf{C} \otimes \mathbf{r}^\top) + \mathsf{noise}, \quad \mathbf{s}_1(\mathbf{D} \otimes \mathbf{t}^\top \otimes \mathbf{I}) + \mathsf{noise})$$

We argue this via a new lemma by using only (standard) LWE.

Game 5: At this point it remains to argue that $\mathbf{c}_3$, $\mathbf{c}_4'$ and $\mathbf{c}_7'$ are pseudorandom. These constitute:

$$\mathbf{s}(\mathbf{I} \otimes \mathbf{r}^\top)\big((\mathbf{A}_1 \| \mathbf{A}_2) - (\mathbf{x} \| \mathbf{y}) \otimes \mathbf{G}\big) + \mathsf{noise}, \;\; \mathbf{s}(\mathbf{I} \otimes \mathbf{r}^\top)(\mathbf{A}_f\mathbf{U}) + \mathsf{noise}$$

and we can directly plug in the tensor LWE assumption to argue this.

Please see Section 4 for the detailed proof.

Extension to Constant Arity. Next, we outline how to extend the above idea to the setting of constant arity. The basic idea is to let the secret key for slot $i \in [k]$ generate

$$\mathbf{s}((\mathbf{A} - \mathbf{x}_i \otimes \mathbf{G}) \otimes \mathbf{I} \otimes \mathbf{r}_i^\top \otimes \mathbf{I}) + \underbrace{\mathbf{s}_i(\mathbf{D} \otimes \mathbf{I} \otimes \mathbf{r}_i^\top \otimes \mathbf{I}) + \mathsf{noise}}_{\text{masking term}}$$

where $\mathbf{r}_i$ is the user specific randomness associated with the secret key for the i-th slot.

In addition, we also prepare other terms so that the ciphertext and secret keys can collaboratively generate the unmasking terms as:

$$\mathbf{s}_i(\mathbf{D} \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top) + \mathsf{noise} \;\; \forall i \in [k]$$

Given the unmasking term, the decryptor can obtain

$$\mathbf{s}((\mathbf{A}_i - \mathbf{x}_i \otimes \mathbf{G}) \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top) + \mathsf{noise}$$

A similar strategy also works for masking $\mathbf{s}(\mathbf{A}_f \mathbf{U} \otimes \mathbf{I})$ and we can show that the adversary can only obtain

$$\mathbf{s}((\mathbf{A} - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top) + \mathsf{noise}, \quad \mathbf{s}(\mathbf{A}_f \mathbf{U} \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top) + \mathsf{noise}$$

which are LWE samples w.r.t randomness $\mathbf{s}(\mathbf{I} \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top)$. We refer the reader to Section 5 for the complete construction.

On Circuit Depth. As discussed above, for our miABE for NC_1, we rely only on evasive LWE, even for constant arity. For our miABE for P, we require evasive and tensor LWE for arity 2, but for general k, we need to generalize tensor LWE as discussed above.

To remove the need for (any) tensor LWE in the restricted case of NC_1 circuits, we use low norm $\mathbf{A}_i$ and switch out $\mathbf{G}$ for $\mathbf{I}$, as suggested by Wee. We also leverage the observation by Wee, that a weaker version of homomorphic computation is still possible in this setting. In addition, we show that when $\mathbf{A}_i$ and $\mathbf{G}$ are changed as above, LWE samples w.r.t $\mathbf{x}$ obtained by combining ciphertexts and secret keys are indistinguishable from those that are computed using fresh randomness for all combinations of ciphertexts and secret keys.

In more detail, let $\mathbf{i} = (i_1, \ldots, i_k)$ denote the ciphertext queries in the k slots which are being combined for decryption. Then, we show that

$$\left\{\mathbf{s}\big((\mathbf{A} - \mathbf{x}^{\mathbf{i}} \otimes \mathbf{I}) \otimes \mathbf{r}_1^{i_1\top} \otimes \ldots \otimes \mathbf{r}_1^{i_k\top}\big) + \mathsf{noise}\right\}_{i_1,\ldots,i_k \in [Q]} \approx_c \left\{\mathbf{s}_{i_1,\ldots,i_k}\big(\mathbf{A} - \mathbf{x}^{\mathbf{i}} \otimes \mathbf{I}\big) + \mathsf{noise}\right\}_{i_1,\ldots,i_k \in [Q]}$$

where $\mathbf{s}_{i_1,\ldots,i_k}$ is a unique, freshly sampled secret for the combination $\mathbf{i} = (i_1, \ldots, i_k)$. Intuitively, the shortness of $\mathbf{A}$ and $\mathbf{I}$ is used to argue that:

$$\mathbf{s}\big((\mathbf{A} - \mathbf{x}^{\mathbf{i}} \otimes \mathbf{I}) \otimes \mathbf{r}_1^{i_1\top} \otimes \ldots \otimes \mathbf{r}_k^{i_k\top}\big) + \mathsf{noise} \approx_c \big(\mathbf{s}(\mathbf{I} \otimes \mathbf{r}_1^{i_1\top} \otimes \ldots \otimes \mathbf{r}_k^{i_k\top}) + \mathsf{noise}\big)\big(\mathbf{A} - \mathbf{x}^{\mathbf{i}} \otimes \mathbf{I}\big) + \mathsf{noise}$$

which in turn allows to express $\mathbf{s}(\mathbf{I} \otimes \mathbf{r}_1^{i_1\top} \otimes \ldots \otimes \mathbf{r}_k^{i_k\top}) + \mathsf{noise}$ as $\mathbf{s}_{i_1,\ldots,i_k}$ by iteratively separating out $\mathbf{r}_j^{i_j\top}$, and adding noise to obtain a fresh secret[4]. Please see Section 5 for details.

2 Preliminaries

Notation. We begin by defining the notation that we will use throughout this work. We use bold letters to denote vectors and the notation $[a, b]$ to denote the set of integers $\{k \in \mathbb{N} \mid a \leq k \leq b\}$. We use $[n]$ to denote the set $[1, n]$. By default, we treat a vector as a row vector. For any vector $\mathbf{x}$ of length ℓ, we let x_i denote the i-th coordinate of $\mathbf{x}$, for $i \in [\ell]$. For any two vectors $\mathbf{x}$ and $\mathbf{y}$ (resp. matrices $\mathbf{X}$, $\mathbf{Y}$), $\mathbf{x}\|\mathbf{y}$ (resp. $\mathbf{X}\|\mathbf{Y}$) represents horizontal concatenation of vectors $\mathbf{x}$ and $\mathbf{y}$ (resp. matrices $\mathbf{X}$ and $\mathbf{Y}$). We use $\mathbf{1}_{\ell \times m}$ (resp. $\mathbf{0}_{\ell \times m}$) to represent a matrix of dimensions $\ell \times m$ having each entry as 1 (resp. 0). Similarly, we write $\mathbf{1}_a$ (resp. $\mathbf{0}_a$) to represent $(1, \ldots, 1) \in \mathbb{Z}_q^a$ (resp. $(0, \ldots, 0) \in \mathbb{Z}_q^a$). For any $n > 0$, $\mathbf{I}_n$ represents an identity matrix of size n. When $n = m$, we denote $\mathbf{I}_m$ by only $\mathbf{I}$ and $\mathbf{I}^{\otimes i}$ denotes $\mathbf{I}_{m^i} = \underbrace{\mathbf{I} \otimes \cdots \otimes \mathbf{I}}_{i \text{ times}}$ for any integer i.

[4] The informed reader may be reminded of the Naor-Reingold argument [NR97] used to construct a PRF from DDH or its lattice analogue [BPR12].

We say a function $f(n)$ is *negligible* if it is $O(n^{-c})$ for all $c > 0$, and we use $\mathrm{negl}(n)$ to denote a negligible function of n. We say $f(n)$ is *polynomial* if it is $O(n^c)$ for some constant $c > 0$, and we use $\mathrm{poly}(n)$ to denote a polynomial function of n. We use the abbreviation PPT for probabilistic polynomial-time. The function $\log x$ is the base 2 logarithm of x. For two distributions $\mathcal{D}_1$, $\mathcal{D}_2$ we use the notation $\mathcal{D}_1 \approx_c \mathcal{D}_2$ (resp. $\mathcal{D}_1 \approx_s \mathcal{D}_2$) to denote that a PPT adversary cannot distinguish between the distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ except only with a computational (resp. satistical) negligible distinguishing advantage. In addition, we write $\mathcal{D}_1 \equiv \mathcal{D}_2$ when $\mathcal{D}_1$ and $\mathcal{D}_2$ are perfectly indistinguishable.

2.1 Multi-Input Attribute Based Encryption

Following [AYY22], we define multi-input Attribute Based Encryption (ABE) below. A k-input ABE scheme is parametrized over an attribute space $\{(A_\lambda)^k\}_{\lambda\in\mathbb{N}}$ and function space $\{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$, where each function maps $\{(A_\lambda)^k\}_{\lambda\in\mathbb{N}}$ to $\{0,1\}$. Such a scheme is described by procedures $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}_1, \ldots, \mathsf{KeyGen}_{k-1}, \mathsf{KeyGen}_k, \mathsf{Dec})$ with the following syntax:

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

$\mathsf{Enc}(\mathsf{mpk}, \mathbf{x}_0, \mu) \to \mathsf{ct}_{\mathbf{x}_0,\mu}$: The encryption algorithm takes as input the master public key mpk, an attribute $\mathbf{x}_0 \in A_\lambda$, and message $\mu \in \{0,1\}$, and outputs a ciphertext $\mathsf{ct}_{\mathbf{x}_0,\mu}$. The attribute string $\mathbf{x}_0$ is also included as part of the ciphertext.

$\mathsf{KeyGen}_i(\mathsf{msk}, \mathbf{x}_i) \to \mathsf{sk}_{i,\mathbf{x}_i}$ for $1 \le i \le k-1$: The KeyGen algorithm for the i^{th} slot where $i \in [k-1]$, takes as input the master secret key msk, and an attribute $\mathbf{x}_i \in A_\lambda$ and outputs a key for slot i, $\mathsf{sk}_{i,\mathbf{x}_i}$. Again, we assume that the attribute string $\mathbf{x}_i$ is included as part of the secret key.

$\mathsf{KeyGen}_k(\mathsf{msk}, f) \to \mathsf{sk}_{k,f}$: The KeyGen algorithm for slot k takes as input the master secret key msk and a function $f \in \mathcal{F}_\lambda$ and outputs a key $\mathsf{sk}_{k,f}$.

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{ct}_{\mathbf{x}_0,\mu}, \mathsf{sk}_{1,\mathbf{x}_1}, \ldots, \mathsf{sk}_{k-1,\mathbf{x}_{k-1}}, \mathsf{sk}_{k,f}) \to \mu'$: The decryption algorithm takes as input a ciphertext $\mathsf{ct}_{\mathbf{x}_0,\mu}$, k keys $\mathsf{sk}_{1,\mathbf{x}_1}, \ldots, \mathsf{sk}_{k-1,\mathbf{x}_{k-1}}$, and $\mathsf{sk}_{k,f}$ and outputs a string μ'.

Next, we define correctness and security. For ease of notation, we drop the subscript λ in what follows.

Correctness: For every $\lambda \in \mathbb{N}, \mu \in \{0,1\}$, $\mathbf{x}_0, \ldots, \mathbf{x}_{k-1} \in A$, $f \in \mathcal{F}$, it holds that if $f(\mathbf{x}_0, \ldots, \mathbf{x}_{k-1}) = 0$,[5] then

$$\Pr\left[\mathsf{Dec}\begin{pmatrix} \mathsf{mpk}, \quad \mathsf{Enc}(\mathsf{mpk}, \mathbf{x}_0, \mu), \\ \mathsf{KeyGen}(\mathsf{msk}, \mathbf{x}_1), \ldots, \mathsf{KeyGen}_{k-1}(\mathsf{msk}, \mathbf{x}_{k-1}), \mathsf{KeyGen}_k(\mathsf{msk}, f) \end{pmatrix} = \mu\right] \\ = 1 - \mathrm{negl}(\lambda)$$

where the probability is over the choice of $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and over the internal randomness of Enc and $\mathsf{KeyGen}_1, \ldots, \mathsf{KeyGen}_k$.

Definition 2.1 ($\mathsf{Ada\text{-}IND}$ security for $\mathsf{k\text{-}ABE}$)**.** For a $\mathsf{k\text{-}ABE}$ scheme $\mathsf{k\text{-}ABE} = \{\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}_1, \ldots, \mathsf{KeyGen}_{k-1}, \mathsf{KeyGen}_k, \mathsf{Dec}\}$, for an attribute space $\{(A_\lambda)^k\}_{\lambda\in\mathbb{N}}$, function space $\{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ and an adversary $\mathcal{A}$, we define the $\mathsf{Ada\text{-}IND}$ security game as follows.

[5] We follow the convention in lattice based cryptography where the decryption condition is reversed with respect to the output of the function.

1. **Setup phase:** On input 1^λ, the challenger samples $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and gives mpk to $\mathcal{A}$.
2. **Query phase:** During the game, $\mathcal{A}$ adaptively makes the following queries, in an arbitrary order.
   (a) **Key Queries:** $\mathcal{A}$ makes polynomial number of key queries for each slot, say $p = p(\lambda)$. As a j-th query for slot i, $\mathcal{A}$ chooses
   $$\begin{cases} \mathbf{x}_{i,j} & \text{if } i \in [k-1] \\ f_j & \text{if } i = k, \end{cases}$$
   where $\mathbf{x}_{i,j} \in A_\lambda$ and $f_j \in \mathcal{F}_\lambda$. The challenger computes
   $$\begin{cases} \mathsf{sk}_{i,\mathbf{x}_{i,j}} = \mathsf{KeyGen}_i(\mathsf{msk}, \mathbf{x}_{i,j}) & \text{if } i \in [k-1] \\ \mathsf{sk}_{f_j} = \mathsf{KeyGen}_k(\mathsf{msk}, f_j) & \text{if } i = k \end{cases}$$
   and returns it to $\mathcal{A}$.
   (b) **Challenge Query:** $\mathcal{A}$ issues a challenge query for encryption. $\mathcal{A}$ declares $(\mathbf{x}_0, (\mu_0, \mu_1))$ to the challenger, where $\mathbf{x}_0 \in A_\lambda$ is an attribute and $(\mu_0, \mu_1) \in \{0,1\} \times \{0,1\}$ is the pair of messages. Then, the challenger samples $\beta \leftarrow \{0,1\}$, computes $\mathsf{ct}_\beta = \mathsf{Enc}(\mathsf{mpk}, \mathbf{x}_0, \mu_\beta)$ and returns it to $\mathcal{A}$.
3. **Output phase:** $\mathcal{A}$ outputs a guess bit β' as the output of the experiment.

For the adversary to be *admissible*, we require that for every $f_1, \ldots, f_p \in \mathcal{F}$, it holds that $f_{j_k}(\mathbf{x}_0, \mathbf{x}_{1,j_1}, \ldots, \mathbf{x}_{k-1,j_{k-1}}) = 1$ for every $j_1, \ldots, j_k \in [p]$.

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

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

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

Definition 2.2 (VerSel-IND security for k-ABE)**.** The definitions for VerSel-IND security for k-ABE is the same as Ada-IND security above except that the adversary $\mathcal{A}$ is required to submit the challenge query and key queries to the challenger before it samples the public key.

Comparing with the miABE Definition in [AYY22]: We note that our definition of kABE is equivalent to the one in [AYY22], except that the encryption algorithm that encrypts the message with an attribute is a public algorithm in our definition, while it is a secret algorithm in [AYY22]. Note that both in our definition as well as [AYY22], the message is associated with only a single attribute, which was shown to be sufficient in [AYY22]. In more detail, $\mathsf{Enc}(\mathsf{mpk}, \mathbf{x}, \mu)$ above is same as $\mathsf{Enc}_1(\mathsf{msk}, \mathbf{x}, \mu)$ in [AYY22], except that Enc_1 is a secret algorithm while Enc is a public algorithm. $\mathsf{KeyGen}_i(\mathsf{msk}, \mathbf{x}_i)$ is same as $\mathsf{Enc}_{i+1}(\mathsf{msk}, \mathbf{x}_i)$ in [AYY22], $\mathsf{KeyGen}_k(\mathsf{msk}, f)$ is same as $\mathsf{KeyGen}(\mathsf{msk}, f)$ in [AYY22]. Further, note that since the encryption algorithm in our definition is a public algorithm, it suffices to consider that the adversary issues only one challenge query of the form $(\mathbf{x}_0, (\mu_0, \mu_1))$, while it can issue polynomially many key queries for each slot $i \in [k]$ similar to [AYY22], where the adversary can issue polynomially many key queries and encryption queries for each slot. Finally, note that since the challenge bit β is encoded only in the ciphertext returned by the (public) encryption algorithm, the distinction between the stronger and weaker security notions in [AYY22] disappears in our definition. Thus, the security definition given above is same as the stronger security defined in [AYY22].

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 $n = \mathrm{poly}(\lambda)$ 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 introduce the learning with errors (LWE) problem.

Definition 2.3 (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 $\mathcal{A}$ we have

$$|\Pr[\mathcal{A}(\mathbf{A}, \mathbf{sA} + \mathbf{x}) \to 1] - \Pr[\mathcal{A}(\mathbf{A}, \mathbf{v}) \to 1]| \leq \mathrm{negl}(\lambda)$$

where the probability is taken over the choice of the random coins by the adversary $\mathcal{A}$ 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 $\mathcal{A}$.

As shown by previous works [Reg09, BLP+13], 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.

LWE with Low-Norm Samples. The following lemma states that the LWE problem is hard even when the public matrix is chosen from low norm Gaussian distribution.

Lemma 2.4. *[BLMR13] Let $k = k(\lambda)$, $m = m(\lambda)$ and $q = q(\lambda) > 2$ be integers. Then if* $\mathsf{LWE}(n, m, q, \gamma)$ *hardness assumption holds then for any* PPT *adversary $\mathcal{A}$ we have*

$$|\Pr[\mathcal{A}(\mathbf{A}, \mathbf{sA} + \mathbf{x}) \to 1] - \Pr[\mathcal{A}(\mathbf{A}, \mathbf{u}) \to 1]| \leq \mathrm{negl}(\lambda)$$

where $\mathbf{A} \leftarrow \mathcal{D}_{\mathbb{Z},\sigma}^{k\times m}$, $\mathbf{s} \leftarrow \mathbb{Z}_q^k$, $\mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z},\gamma}^m$, $\mathbf{u} \leftarrow \mathbb{Z}_q^m$, $k \geq 6n \log q$ and $\sigma = \Omega(\sqrt{n \log q})$.

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 $\mathbf{A}^{-1}(\mathbf{V})$ be an output distribution of $\mathsf{SampZ}(\gamma)^{m\times m'}$ conditioned on $\mathbf{A} \cdot \mathbf{A}^{-1}(\mathbf{V}, \gamma) = \mathbf{V}$. A γ-trapdoor for $\mathbf{A}$ is a trapdoor that enables one to sample from the distribution $\mathbf{A}^{-1}(\mathbf{V}, \gamma)$ 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 [GPV08, CHKP10, ABB10a, ABB10b, MP12, BLP+13].

Lemma 2.5 (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 [Tsa19].

Lemma 2.6 (Fully Homomorphic Computation [BGG+14])**.** *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.*
- $\mathsf{EvalFX}(\mathbf{B}, F, \mathbf{x}) \to \widehat{\mathbf{H}}_{\mathbf{B},F,\mathbf{x}}$*. Here,* $\mathbf{x} \in \{0,1\}^\ell$ *is a binary string whose first bit is* 0 *and the second bit is* 1 *and* $F : \{0,1\}^\ell \to \{0,1\}$ *is a circuit with depth* d *that ignores the first and the second bit of the input. Then, we have*

  $$[\mathbf{B} - \mathbf{x} \otimes \mathbf{G}]\widehat{\mathbf{H}}_{\mathbf{B},F,\mathbf{x}} = \mathbf{B}\mathbf{H}_F - F(\mathbf{x})\mathbf{G} \mod q,$$

  where we denote $[x_1\mathbf{G}\|\cdots\|x_k\mathbf{G}]$ *by* $\mathbf{x} \otimes \mathbf{G}$*. Furthermore, we have*

  $$\|\mathbf{H}_F\|_\infty \le m \cdot 2^{O(d)}, \quad \|\widehat{\mathbf{H}}_{\mathbf{B},F,\mathbf{x}}\|_\infty \le m \cdot 2^{O(d)}.$$

  Finally, we have that the topmost m *rows of* $\widehat{\mathbf{H}}_{\mathbf{B},F,\mathbf{x}}$ *constitutes an identity matrix.*
- *The running time of* $(\mathsf{EvalF}, \mathsf{EvalFX})$ *is bounded by* $\mathrm{poly}(n, m, \log q, 2^d)$.

Remark 2.7. As pointed out in [KNYY20] (See also [BV15]), we need some entry of $\mathbf{x}$ to be 1 to support arbitrary F. We therefore assume that the second bit of $\mathbf{x}$ is 1. Furthermore, we assume the first bit of $\mathbf{x}$ is 0. This assumption is introduced to make sure that the topmost m rows of $\widehat{\mathbf{H}}_{\mathbf{B},F,\mathbf{x}}$ constitutes an identity matrix, which is not guaranteed for the evaluation algorithms in [BGG+14]. As we explain below, this can be ensured easily by modifying the evaluation algorithms in [BGG+14]. Suppose that we have EvalF' and EvalFX' without this property. Denoting $\mathbf{x} = (0, \mathbf{x}')$ and $\mathbf{B} = [\mathbf{B}_0\|\mathbf{B}']$, we have

$$[\mathbf{B}' - \mathbf{x}' \otimes \mathbf{G}]\widehat{\mathbf{H}}'_{\mathbf{B}',F',\mathbf{x}'} = \mathbf{B}'\mathbf{H}'_{F'} - F(\mathbf{x})\mathbf{G} \mod q,$$

where F' is the same function as F except that it ignores only the first bit, $\widehat{\mathbf{H}}'_{\mathbf{B}',F',\mathbf{x}'} = \mathsf{EvalFX}(\mathbf{B}', F', \mathbf{x}')$, and $\mathsf{EvalF}(\mathbf{B}', F') \to \mathbf{H}'_{F'}$. We then define the new evaluation algorithms EvalF and EvalFX as $\mathsf{EvalFX}(\mathbf{B}, F, \mathbf{x}) = \widehat{\mathbf{H}}_{\mathbf{B},F,\mathbf{x}} = \begin{bmatrix} \mathbf{I} \\ \widehat{\mathbf{H}}'_{F',\mathbf{x}'} \end{bmatrix}$ and $\mathsf{EvalF}(\mathbf{B}, F) = \mathbf{H}_F = \begin{bmatrix} \mathbf{I} \\ \mathbf{H}'_{F'} \end{bmatrix}$. It is easy to see that the new evaluation algorithms satisfy all the desired properties. In our paper, we implicitly assume that $\mathbf{x}$ input to the circuit F always has $0\|1$ as its prefix so that the above lemma holds and will not explicitly write the leading bits for the sake of notational simplicity. In our context, this means that the first two bits of an attribute $\mathbf{x}$ associated with a ciphertext should be $0\|1$.

Low Norm Variant. We also consider the low norm variant of the lattice evaluation algorithm defined in [Wee22], where $\mathbf{B}$ has low-norm and $\mathbf{G}$ is replaced with $\mathbf{I}$.

Lemma 2.8. *Fix parameters* m, ℓ*. Given a matrix* $\mathbf{B} \in \mathbb{Z}^{m \times m\ell}$ *and a circuit* $F : \{0,1\}^\ell \to \{0,1\}$ *of depth* d*, we can efficiently compute a matrix* $\mathbf{H}_F \in \mathbb{Z}^{m\ell \times m}$ *such that* $\|\mathbf{H}_F\|_\infty = (\|\mathbf{B}\|_\infty m)^{O(2^d)}$ *and for all* $\mathbf{x} \in \{0,1\}^\ell$*, there exists a matrix* $\widehat{\mathbf{H}}_{\mathbf{B},F,\mathbf{x}} \in \mathbb{Z}^{\ell m \times m}$ *with* $\|\widehat{\mathbf{H}}_{\mathbf{B},F,\mathbf{x}}\|_\infty = (\|\mathbf{B}\|_\infty m)^{O(2^d)}$ *such that*

$$(\mathbf{B} - \mathbf{x} \otimes \mathbf{I}_m) \cdot \widehat{\mathbf{H}}_{\mathbf{B},F,\mathbf{x}} = \mathbf{B}\mathbf{H}_F - F(\mathbf{x})\mathbf{I}_m$$

Moreover, $\widehat{\mathbf{H}}_{\mathbf{B},F,\mathbf{x}}$ *is efficiently computable given* $\mathbf{B}, F, \mathbf{x}$*. We use* $\mathsf{EvalF}(\mathbf{B}, F)$*,* $\mathsf{EvalFX}(\mathbf{B}, F, \mathbf{x})$ *to denote the algorithms computing* $\mathbf{H}_F, \widehat{\mathbf{H}}_{\mathbf{B},F,\mathbf{x}}$ *respectively. Finally, the topmost* m *rows of* $\widehat{\mathbf{H}}_{\mathbf{B},F,\mathbf{x}}$ *constitutes an identity matrix.*

Remark 2.9. The condition that the top most m rows of $\widehat{\mathbf{H}}_{\mathbf{B},F,\mathbf{x}}$ constitutes an identity matrix can be satisfied by adding suitable modifications to the evaluation algorithms without this property. See Remark 2.7 for the detail.

Smudging Lemma. We will also require the standard smudging lemma.

Lemma 2.10 (Smudging Lemma [WWW22])**.** *Let λ be a security parameter. Take any $a \in \mathbb{Z}$ where $|a| \leq B$. Suppose $\chi \geq B\lambda^{\omega(1)}$. Then the statistical distance between the distributions $\{z : z \leftarrow \mathcal{D}_{\mathbb{Z},\chi}\}$ and $\{z + a : z \leftarrow \mathcal{D}_{\mathbb{Z},\chi}\}$ is* $\mathrm{negl}(\lambda)$.

2.3 Tensors

In this work, similarly to [Wee22], we use the tensor product techniques. Let $\mathbf{A} = (a_{i,j}) \in \mathbb{Z}_q^{m\times n}$ and $\mathbf{B} \in \mathbb{Z}_q^{s\times t}$. The tensor product is defined as:

$$\mathbf{A} \otimes \mathbf{B} := \begin{pmatrix} a_{1,1}\mathbf{B} & \cdots & a_{1,n}\mathbf{B} \\ \vdots & & \vdots \\ a_{m,1}\mathbf{B} & \cdots & a_{m,n}\mathbf{B} \end{pmatrix} \in \mathbb{Z}_q^{ms\times nt}.$$

Throughout the paper, we will heavily use the mixed-product equality, stated as follows. Let $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$, $\mathbf{B} \in \mathbb{Z}_q^{s\times t}$, $\mathbf{C} \in \mathbb{Z}_q^{n\times u}$ and $\mathbf{D} \in \mathbb{Z}_q^{t\times v}$,

$$(\mathbf{A} \otimes \mathbf{B}) \cdot (\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD}) \in \mathbb{Z}_q^{ms\times uv}.$$

The mixed-product can be naturally generalized following

$$(\mathbf{A}^1 \otimes \cdots \otimes \mathbf{A}^k) \cdot (\mathbf{B}^1 \otimes \cdots \otimes \mathbf{B}^k) = (\mathbf{A}^1\mathbf{B}^1) \otimes \cdots \otimes (\mathbf{A}^k\mathbf{B}^k).$$

Note that we adopt the same convention as in [Wee22] where matrix multiplication takes precedence over tensor products, i.e. $\mathbf{A} \otimes \mathbf{BC} = \mathbf{A} \otimes (\mathbf{BC})$.

3 Assumptions and New Implications

In this section, we discuss the evasive and tensor LWE assumptions. Our variants of these assumptions differ slightly from the original formulation by [Wee22] as discussed below.

3.1 Evasive LWE

Below, we state a variant of the Evasive-LWE assumption which will be useful for our constructions.

Assumption 3.1 (Evasive LWE). Let $n, m, t, m', q \in \mathbb{N}$ be parameters and λ be a security parameter. Let χ and χ' be parameters for Gaussian distributions. Let Samp be a PPT algorithm that outputs

$$\mathbf{S} \in \mathbb{Z}_q^{m'\times n}, \mathbf{P} \in \mathbb{Z}_q^{n\times t}, \mathsf{aux} \in \{0,1\}^*$$

on input 1^λ. For a PPT adversary Adv, we define the following advantage functions:

$$\mathcal{A}_{\mathsf{Adv}}^{\mathrm{PRE}}(\lambda) := \Pr[\mathsf{Adv}_0(\mathbf{B}, \mathbf{SB} + \mathbf{E}, \mathbf{SP} + \mathbf{E}', \mathsf{aux}) = 1] - \Pr[\mathsf{Adv}_0(\mathbf{B}, \mathbf{C}_0, \mathbf{C}', \mathsf{aux}) = 1]$$

$$\mathcal{A}_{\mathsf{Adv}}^{\mathrm{POST}}(\lambda) := \Pr[\mathsf{Adv}_1(\mathbf{B}, \mathbf{SB} + \mathbf{E}, \mathbf{K}, \mathsf{aux}) = 1] - \Pr[\mathsf{Adv}_1(\mathbf{B}, \mathbf{C}_0, \mathbf{K}, \mathsf{aux}) = 1]$$

where

$$(\mathbf{S}, \mathbf{P}, \mathsf{aux}) \leftarrow \mathsf{Samp}(1^\lambda),$$
$$\mathbf{B} \leftarrow \mathbb{Z}_q^{n \times m},$$
$$\mathbf{C}_0 \leftarrow \mathbb{Z}_q^{m' \times m}, \mathbf{C}' \leftarrow \mathbb{Z}_q^{m' \times t},$$
$$\mathbf{E} \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^{m' \times m}, \mathbf{E}' \leftarrow \mathcal{D}_{\mathbb{Z},\chi'}^{m' \times t}$$
$$\mathbf{K} \leftarrow \mathbf{B}^{-1}(\mathbf{P}) \text{ with standard deviation } O(\sqrt{m \log(q)}).$$

We say that the *evasive* LWE (EvLWE) assumption holds if for every PPT Samp and Adv_1, there exists another PPT Adv_0 and a polynomial $Q(\cdot)$ such that

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

Remark 3.2. In the above definition, all the entries of $\mathbf{E}'$ are chosen from the same distribution $D_{\mathbb{Z},\chi'}$. However, in our security proof, we often consider the case where some entries of $\mathbf{E}'$ are chosen from $D_{\mathbb{Z},\chi'}$ and others from $D_{\mathbb{Z},\chi''}$ with different $\chi' \gg \chi''$. The evasive LWE assumption with such a mixed noise distribution for $\mathbf{E}'$ is implied by the evasive LWE assumption with all entries in $\mathbf{E}'$ being chosen from $D_{\mathbb{Z},\chi'}$ as above definition, since if the precondition is satisfied for the latter case, that for the former case is also satisfied. To see this, it suffices to observe that we can convert the distribution from $D_{\mathbb{Z},\chi''}$ into that from $D_{\mathbb{Z},\chi'}$ by adding extra Gaussian noise.

Remark 3.3. In the above, we chose χ' to be smaller than χ following [VWW22]. This makes the precondition stronger, which in turn makes evasive LWE weaker.

Comparison with the original evasive LWE [Wee22]. Our assumption is closely related to the evasive LWE assumption that appeared in Wee [Wee22] with minor differences. In Wee, the secret $\mathbf{S}$ is chosen uniformly whereas in our assumption, the secret can be structured and output by the sampler, subject to the pre-condition being true. On the other hand, in [VWW22], $\mathbf{S}$ is the public matrix and can be structured, while $\mathbf{B}$ is secret and is random. An additional difference is related to the auxiliary input. In Wee, aux contains all the coin tosses used by the sampler – this suffices to rule out obfuscation based counter-examples where aux may contain information of the trapdoor for $\mathbf{P}$ in a hidden way. On the other hand, in [VWW22], the coins of the sampler are private, and aux contains information including certain Gaussian preimages. They argue that their assumption nevertheless avoids the obfuscation based counter-examples, since their auxiliary input does not contain trapdoor for the matrix $\mathbf{P}$. In both their and our cases, aux is derived from the trapdoor for $\mathbf{P}$ or related information that should be kept hidden, but it does not contain the trapdoor itself. We may therefore expect that there is no space for embedding an obfuscation into our auxiliary input, similarly to [VWW22]. We also note that as observed in [VWW22], Tsabary's variant of evasive LWE is less conservative than ours and theirs, since her definition allows aux to depend on $\mathbf{B}$.

In the security proof of our constructions, we sometimes want to include information dependent on $\mathbf{S}$ into the auxiliary information. Furthermore, we may want to prove the pseudorandomness of such auxiliary information. The following lemma is useful in such a situation, which says that the auxiliary information is pseudorandom in the post condition distribution, if it is pseudorandom in the precondition distribution.

Lemma 3.4. *Let $n, m, t, m', q \in \mathbb{N}$ be parameters and λ be a security parameter. Let χ and χ' be Gaussian parameters. Let* Samp *be a PPT algorithm that outputs*

$$\mathbf{S} \in \mathbb{Z}_q^{m' \times n}, \mathsf{aux} = (\mathsf{aux}_1, \mathsf{aux}_2) \in \mathcal{S} \times \{0,1\}^* \textit{ and } \mathbf{P} \in \mathbb{Z}_q^{n \times t}$$

for some set $\mathcal{S}$. Furthermore, we assume that there exists a public deterministic poly-time algorithm $\mathsf{Reconstruct}$ *that allows to derive* $\mathbf{P}$ *from* aux_2*, i.e.* $\mathbf{P} = \mathsf{Reconstruct}(\mathsf{aux}_2)$*.*

We introduce the following advantage functions:

$$\mathcal{A}^{\mathrm{PRE}'}_{\mathsf{Adv}}(\lambda) := \Pr[\mathsf{Adv}(\mathbf{B}, \mathbf{SB}+\mathbf{E}, \mathbf{SP}+\mathbf{E}', \mathsf{aux}_1, \mathsf{aux}_2) = 1] - \Pr[\mathsf{Adv}(\mathbf{B}, \mathbf{C}_0, \mathbf{C}', \mathbf{c}, \mathsf{aux}_2) = 1]$$

$$\mathcal{A}^{\mathrm{POST}'}_{\mathsf{Adv}}(\lambda) := \Pr[\mathsf{Adv}(\mathbf{B}, \mathbf{SB}+\mathbf{E}, \mathbf{K}, \mathsf{aux}_1, \mathsf{aux}_2) = 1] - \Pr[\mathsf{Adv}(\mathbf{B}, \mathbf{C}_0, \mathbf{K}, \mathbf{c}, \mathsf{aux}_2) = 1]$$

where

$$(\mathbf{S}, \mathsf{aux} = (\mathsf{aux}_1, \mathsf{aux}_2), \mathbf{P}) \leftarrow \mathsf{Samp}(1^\lambda),$$
$$\mathbf{B} \leftarrow \mathbb{Z}_q^{n\times m}$$
$$\mathbf{C}_0 \leftarrow \mathbb{Z}_q^{m'\times m}, \mathbf{C}' \leftarrow \mathbb{Z}_q^{m'\times t}, \mathbf{c} \leftarrow \mathcal{S}$$
$$\mathbf{E} \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^{m'\times m}, \mathbf{E}' \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^{m'\times t}$$
$$\mathbf{K} \leftarrow \mathbf{B}^{-1}(\mathbf{P}) \text{ with standard deviation } O(\sqrt{m\log(q)}).$$

Then, under the Evasive-LWE *(cited above in Assumption 3.1) with respect to* Samp*, if* $\mathcal{A}^{\mathrm{PRE}'}_{\mathsf{Adv}}(\lambda)$ *is negligible for any PPT adversary* Adv*, so is* $\mathcal{A}^{\mathrm{PRE}'}_{\mathsf{Adv}}(\lambda)$ *for any PPT adversary* Adv*.*

Proof. By the assumption, we have $(\mathbf{B}, \mathbf{SB}+\mathbf{E}, \mathbf{SP}+\mathbf{E}', \mathsf{aux}_1, \mathsf{aux}_2) \approx_c (\mathbf{B}, \mathbf{C}_0, \mathbf{C}', \mathbf{c}, \mathsf{aux}_2)$. This in particular implies $(\mathsf{aux}_1, \mathsf{aux}_2) \approx_c (\mathbf{c}, \mathsf{aux}_2)$ since discarding the term making the task of distinguishing the distributions harder. This further implies

$$(\mathbf{B}, \mathbf{C}_0, \mathbf{C}', \mathsf{aux}_1, \mathsf{aux}_2) \approx_c (\mathbf{B}, \mathbf{C}_0, \mathbf{C}', \mathbf{c}, \mathsf{aux}_2)$$

since adding random terms $(\mathbf{B}, \mathbf{C}_0, \mathbf{C}')$ chosen independently from the other terms does not make the task of distinguishing the the distributions easier. We therefore establish $(\mathbf{B}, \mathbf{SB} + \mathbf{E}, \mathbf{SP}+\mathbf{E}', \mathsf{aux}_1, \mathsf{aux}_2) \approx_c (\mathbf{B}, \mathbf{C}_0, \mathbf{C}', \mathsf{aux}_1, \mathsf{aux}_2)$. Applying the evasive LWE with respect to Samp defined in the statement, we have

$$(\mathbf{B}, \mathbf{SB}+\mathbf{E}, \mathbf{K}, \mathsf{aux}_1, \mathsf{aux}_2) \approx_c (\mathbf{B}, \mathbf{C}_0, \mathbf{K}, \mathsf{aux}_1, \mathsf{aux}_2).$$

To complete the proof, it suffices to show

$$(\mathbf{B}, \mathbf{C}_0, \mathbf{K}, \mathsf{aux}_1, \mathsf{aux}_2) \approx_c (\mathbf{B}, \mathbf{C}_0, \mathbf{K}, \mathbf{c}, \mathsf{aux}_2).$$

To show this, we recall that the precondition implies $(\mathsf{aux}_1, \mathsf{aux}_2) \approx_c (\mathbf{c}, \mathsf{aux}_2)$. We then observe that $(\mathbf{B}, \mathbf{C}_0, \mathbf{K})$ can be sampled publicly given aux_2. This suffices to complete the proof, since having extra terms that can be computed efficiently from the given terms does not make the task of distinguishing the distributions easier. To sample $(\mathbf{B}, \mathbf{C}_0, \mathbf{K})$, we first sample $\mathbf{B}$ with the trapdoor as $(\mathbf{B}, \mathbf{B}^{-1}_{\tau_0}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$ where $\tau_0 = \omega(\sqrt{n\log q\log m}) \leq O(m\log q)$, compute $\mathbf{P}$ by $\mathbf{P} = \mathsf{Reconstruct}(\mathsf{aux}_2)$, and finally sample $\mathbf{K} \leftarrow \mathbf{B}^{-1}(\mathbf{P}, O(\sqrt{m\log(q)}))$. □

3.2 Tensor LWE

In this section, we define the tensor LWE assumption introduced by Wee [Wee22]. Then, we provide new arguments supporting the assumption.

Assumption 3.5 (Tensor LWE). Let $n, m, q, \ell, Q \in \mathbb{N}$ be parameters and $\gamma, \chi > 0$ be Gaussian parameters. For all $\mathbf{x}_1, \cdots, \mathbf{x}_Q \in \{0,1\}^\ell$, we have

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

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times \ell m}, \mathbf{s} \leftarrow \mathbb{Z}_q^{mn}, \mathbf{e}_i \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^{\ell m}, \mathbf{r}_i^\top \leftarrow \mathcal{D}_{\mathbb{Z},\gamma}^{m}, \mathbf{c}_i \leftarrow \mathbb{Z}_q^{\ell m}$.

To gain confidence in the tensor LWE assumption, we study conditions under which it can be reduced to standard LWE. To begin, we recall the following lemma which is implicit in [Wee22]. The lemma says that a variant of the tensor LWE assumption holds under the standard LWE assumption if $\mathbf{A}$ matrices are chosen from Gaussian distribution and $\mathbf{G}$ is replaced with $\mathbf{I}$ in certain parameter settings.

Lemma 3.6 (Implicitly proved in [Wee22]). *Let $n, m, q, \ell, Q, \beta \in \mathbb{N}$ be parameters and χ_0, χ, and γ be a Gaussian parameter satisfying $m = \Omega(n \log q)$, $\gamma = \lambda^{\omega(1)}$, $\chi = \chi_0 \gamma \lambda^{\omega(1)}$. For all $\mathbf{x}_1, \cdots, \mathbf{x}_Q \in \{0,1\}^\ell$, $\mathsf{LWE}(n, Q+m, q, \chi_0)$ hardness assumption implies*

$$\mathbf{A}, \left\{\mathbf{s}(\mathbf{I}_n \otimes \mathbf{r}_i^\top)(\mathbf{A} - \mathbf{x}_i \otimes \mathbf{I}_m) + \mathbf{e}_i, \mathbf{r}_i^\top\right\}_{i\in[Q]} \approx_c \mathbf{A}, \left\{\mathbf{c}_i, \mathbf{r}_i^\top\right\}_{i\in[Q]}$$

where $\mathbf{A} \leftarrow \mathcal{D}_{\mathbb{Z},\gamma}^{n\times \ell m}$, $\mathbf{s} \leftarrow \mathbb{Z}_q^{mn}, \mathbf{e}_i \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^{\ell m}, \mathbf{r}_i^\top \leftarrow \mathcal{D}_{\mathbb{Z},\gamma}^{m}, \mathbf{c}_i \leftarrow \mathbb{Z}_q^{\ell m}$.

3.3 New Implications for Tensor LWE

We now introduce a new lemma that also proves the same implication between LWE and Tensor LWE in another particular case. Notably, the lemma shows the hardness for the case where $\mathbf{A}$ is chosen uniformly at random rather than from a Gaussian distribution, albeit with the downside of assuming $\mathbf{x}_i = \mathbf{0}$ for all i.

Lemma 3.7 (Tensor LWE with $\{\mathbf{x}_i = \mathbf{0}\}_i$). *Let $n, m, q, \ell, Q, \beta \in \mathbb{N}$ be parameters and χ_0, χ, and γ be a Gaussian parameter satisfying $m = \Omega(n \log q)$, $\gamma = \Omega(\sqrt{n \log q})$, and $\chi = \gamma \chi_0 \lambda^{\omega(1)}$. Then, $\mathsf{LWE}(n, m, q, \chi_0)$ hardness assumption implies*

$$\mathbf{A}, \left\{\mathbf{s}(\mathbf{I}_n \otimes \mathbf{r}_i^\top)\mathbf{A} + \mathbf{e}_i, \mathbf{r}_i^\top\right\}_{i\in[Q]} \approx_c \mathbf{A}, \left\{\mathbf{c}_i, \mathbf{r}_i^\top\right\}_{i\in[Q]}$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times \ell m}$, $\mathbf{s} \leftarrow \mathbb{Z}_q^{mn}, \mathbf{e}_i \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^{\ell m}, \mathbf{r}_i^\top \leftarrow \mathcal{D}_{\mathbb{Z},\gamma}^{m}, \mathbf{c}_i \leftarrow \mathbb{Z}_q^{\ell m}$.

Proof. Let $\mathcal{A}$ be an attacker for Tensor-LWE with $\mathbf{x}_i = \mathbf{0}$ for all $i \in [Q]$. $\mathcal{A}$ is given either $\mathbf{A}, \left\{\mathbf{s}(\mathbf{I}_n \otimes \mathbf{r}_i^\top)\mathbf{A} + \mathbf{e}_i, \mathbf{r}_i^\top\right\}_{i\in[Q]}$ or $\mathbf{A}, \left\{\mathbf{c}_i, \mathbf{r}_i^\top\right\}_{i\in[Q]}$. We provide a proof to show that under the LWE assumption, $\mathcal{A}$ has a negligible advantage of distinguishing the left hand side from the right hand side.

G_0 : $\mathcal{A}$ is given $\mathbf{A}, \left\{\mathbf{s}(\mathbf{I}_n \otimes \mathbf{r}_i^\top)\mathbf{A} + \mathbf{e}_i, \mathbf{r}_i^\top\right\}_{i\in[Q]}$.

G_1 : We rewrite $\mathbf{s}(\mathbf{I}_n \otimes \mathbf{r}_i^\top)\mathbf{A} + \mathbf{e}_i$ using the tensor decomposition of $\mathbf{s} \in \mathbb{Z}_q^{mn}$. In other words,

$$\mathbf{s} = \sum_{j=1}^{m} \mathbf{s}_j \otimes \boldsymbol{\epsilon}_j,$$

where $\boldsymbol{\epsilon}_j$ are the canonical vectors of $\mathbb{Z}_q^m$ and $\mathbf{s}_j \in \mathbb{Z}_q^n$. Let us fix an index $1 \le i \le Q$ and rewrite the i-th sample. We get

$$\begin{aligned} \mathbf{s}(\mathbf{I}_n \otimes \mathbf{r}_i^\top)\mathbf{A} + \mathbf{e}_i &= \textstyle\sum_{j=1}^{m} (\mathbf{s}_j \otimes \boldsymbol{\epsilon}_j) \cdot (\mathbf{I}_n \otimes \mathbf{r}_i^\top)\mathbf{A} + \mathbf{e}_i \\ &= \textstyle\sum_{j=1}^{m} (\mathbf{s}_j \otimes \underbrace{\boldsymbol{\epsilon}_j \mathbf{r}_i^\top}_{:=\mathbf{r}_i[j] \text{ scalar}})\mathbf{A} + \mathbf{e}_i \\ &= \textstyle\sum_{j=1}^{m} \mathbf{r}_i[j] \cdot \mathbf{s}_j \cdot \mathbf{A} + \mathbf{e}_i, \end{aligned}$$

where $\mathbf{r}_i[j]$ is the j-th entry of the vector $\mathbf{r}_i$. Hence, in this game, $\mathcal{A}$ is given

$$\mathbf{A}, \left\{ \sum_{j=1}^{m} \mathbf{r}_i[j] \cdot \mathbf{s}_j \cdot \mathbf{A} + \mathbf{e}_i, \mathbf{r}_i^\top \right\}_{i \in [Q]} .$$

This is a conceptual change : $\mathsf{G}_1 \equiv \mathsf{G}_2$.

G_2 : We now add some extra noise to the distribution to introduce an LWE instance. Define $\mathbf{e}'_j \leftarrow \mathcal{D}_{\mathbb{Z},\chi_0}^{\ell m}$ for all $j \in [1, m]$. In this game, the attacker is given

$$\mathbf{A}, \left\{ \sum_{j=1}^{m} \mathbf{r}_i[j] \cdot (\mathbf{s}_j \cdot \mathbf{A} + \mathbf{e}'_j) + \mathbf{e}_i, \mathbf{r}_i^\top \right\}_{i \in [Q]} .$$

Note that this game is different from the previous game only in the noise term. In the previous game, the noise is $\mathbf{e}_i$ for the i-th sample, while it is $\mathbf{e}_i + \sum_j \mathbf{r}_i[j] \cdot \mathbf{e}'_j$ in this game. Since we have $\|\sum_j \mathbf{r}_i[j] \cdot \mathbf{e}'_j\|_\infty \leq \mathrm{poly}(\lambda)\gamma\chi_0$ and $\chi = \lambda^{\omega(1)} \cdot \gamma\chi_0$, we can apply Lemma 2.10 to conclude that this only introduces a statistical change: $\mathsf{G}_2 \approx_s \mathsf{G}_1$.

G_3 : In this game, we replace each $(\mathbf{s}_j \cdot \mathbf{A} + \mathbf{e}'_j)$ by a uniform vector $\mathbf{c}'_j \leftarrow \mathbb{Z}_q^{\ell m}$. The attacker $\mathcal{A}$ thus gets

$$\mathbf{A}, \left\{ \sum_{j=1}^{m} \mathbf{r}_i[j] \cdot \mathbf{c}'_j + \mathbf{e}_i, \mathbf{r}_i^\top \right\}_{i \in [Q]} .$$

This game is computationally indistinguishable from G_2 under the standard LWE assumption: $\mathsf{G}_3 \approx_c \mathsf{G}_2$.

G_4 : Let us define $\mathbf{C}' := \begin{pmatrix} \mathbf{c}'_1 \\ \vdots \\ \mathbf{c}'_m \end{pmatrix}$ and obtain

$$\begin{pmatrix} \sum_{j=1}^{m} \mathbf{r}_1[j] \cdot \mathbf{c}'_j + \mathbf{e}_1 \\ \vdots \\ \sum_{j=1}^{m} \mathbf{r}_Q[j] \cdot \mathbf{c}'_j + \mathbf{e}_Q \end{pmatrix} = \underbrace{\begin{pmatrix} \mathbf{r}'_1 \\ \vdots \\ \mathbf{r}'_Q \end{pmatrix}}_{\text{public}} \cdot \underbrace{\mathbf{C}'}_{\text{secret}} + \underbrace{\begin{pmatrix} \mathbf{e}_1 \\ \vdots \\ \mathbf{e}_Q \end{pmatrix}}_{\text{error}} .$$

In this game, we replace $\mathbf{r}'_i\mathbf{C}' + \mathbf{e}_i$ by a uniform random vector $\mathbf{c}_i \leftarrow \mathbb{Z}_q^{\ell m}$. Hence the adversary is given

$$\mathbf{A}, \left\{ \mathbf{c}_i, \mathbf{r}_i^\top \right\}_{i \in [Q]} .$$

This game is computationally indistinguishable from G_3: we use LWE with short public matrix and large secret [BLMR13], which is implied by the standard LWE (See Lemma 2.4). Hence, $\mathsf{G}_4 \approx_c \mathsf{G}_3$.

In the last game, the distribution corresponds to the random case, which allows to conclude the proof. □

We can introduce a corollary that follows from Lemma 3.7 where $\mathbf{A}$ is replaced by $\mathbf{A} - \mathbf{x} \otimes \mathbf{G}$.

Corollary 3.8 (Tensor LWE with the same $\mathbf{x}_i$)**.** *Let $n, m, q, \ell, Q \in \mathbb{N}$, χ_0, χ, and γ be parameters defined as Lemma 3.7. Let $\mathbf{x} \in \{0,1\}^\ell$. Then,* $\mathsf{LWE}(n, m, q, \chi_0)$ *hardness assumption implies*

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

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{m\times \ell m}$, $\mathbf{s} \leftarrow \mathbb{Z}_q^{mn}, \mathbf{e}_i \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^{\ell m}, \mathbf{r}_i^\top \leftarrow \mathcal{D}_{\mathbb{Z},\gamma}^{m}, \mathbf{c}_i \leftarrow \mathbb{Z}_q^{\ell m}$.

What prevents Lemma 3.7 to be proved in the general case? The proof of Lemma 3.7 cannot be easily adapted for arbitrary $\mathbf{x}_i$. Following the same proof strategy, we have to prove the pseudorandomness of the following terms:

$$\mathbf{s}(\mathbf{I}_n \otimes \mathbf{r}_i)(\mathbf{A} - \mathbf{x}_i \otimes \mathbf{G}) + \mathbf{e}_i \quad = \textstyle\sum_j \mathbf{r}_i[j] \cdot (\mathbf{s}_j \cdot (\mathbf{A} - \mathbf{x}_i \otimes \mathbf{G}) + \mathbf{e}'_j) - \sum_j \mathbf{r}_i[j]\mathbf{e}'_j + \mathbf{e}_i.$$

However, it is not possible to replace $\mathbf{s}_j \cdot (\mathbf{A} - \mathbf{x}_i \otimes \mathbf{G}) + \mathbf{e}'_j$ with random vectors as is done in Section 3.2, if we are given the term for multiple i with different $\mathbf{x}_i$ and for the same $\mathbf{s}_j$. Thus, the approach cannot be directly transferred.

3.4 New Implications from LWE

In this section, we provide new lemmata under the LWE assumption which will be useful for our constructions. We believe these may be of broader applicability.

Lemma 3.9. *Let $n = n(\lambda)$, $m = m(\lambda)$, $N = N(\lambda)$, $q = q(\lambda)$, $\gamma = \gamma(\lambda)$, $\chi_0 = \chi_0(\lambda) \in \lambda^{\omega(1)}$, $\chi = \chi(\lambda)$, and $k = O(1)$ be parameters satisfying $m = \Omega(n \log q)$, $\chi(\lambda) \geq (m\gamma\chi_0)^k$. If* $\mathsf{LWE}(n, Q, q, \chi_0)$ *holds, then the following distributions are computationally indistinguishable:*

$$\left\{\mathbf{c}_{j_1,\ldots,j_k} := \mathbf{s}(\mathbf{I}_N \otimes \mathbf{r}_{1,j_1}^\top \otimes \cdots \otimes \mathbf{r}_{k,j_k}^\top) + \mathbf{e}_{j_1,\ldots,j_k}\right\}_{j_1,\ldots,j_k\in[Q]} \approx_c \{\mathbf{w}_{j_1,\ldots,j_k}\}_{j_1,\ldots,j_k\in[Q]}$$

where $\mathbf{s} \leftarrow \mathbb{Z}_q^{Nm^k}$, $\mathbf{r}_{i,j_i} \leftarrow \mathcal{D}_{\mathbb{Z},\gamma}^m$, $\mathbf{e}_{j_1,\ldots,j_k} \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^N$, $\mathbf{w}_{j_1,\ldots,j_k} \leftarrow \mathbb{Z}_q^N$ *for $i \in [k]$ and $j_1, \ldots, j_k \in [Q]$.*

Proof. We prove this by induction. The case of $k = 1$ follows from LWE with short public matrices [BLMR13] (Lemma 2.4). Here, we prove the statement for $k = \tau + 1$ assuming it is is true for $k = \tau$. To show the indistinguishability, we start from the distribution on the left hand side and gradually change it to that on the right hand side. We first change the distribution of $\{\mathbf{c}_{j_1,\ldots,j_{\tau+1}}\}_{j_1,\ldots,j_{\tau+1}}$ so that they are sampled as

$$\mathbf{c}_{j_1,\ldots,j_{\tau+1}} = \underbrace{\left(\mathbf{s}(\mathbf{I}_N \otimes \mathbf{I}_m \otimes \mathbf{r}_{2,j_2}^\top \otimes \cdots \otimes \mathbf{r}_{\tau+1,j_{\tau+1}}^\top) + \mathbf{e}'_{j_2,\ldots,j_{\tau+1}}\right)}_{:=\mathbf{s}'_{j_2,\ldots,j_{\tau+1}}} \left(\mathbf{I}_N \otimes \mathbf{r}_{1,j_1}^\top\right) + \mathbf{e}_{j_1,\ldots,j_{\tau+1}}.$$

where $\mathbf{e}'_{j_2,\ldots,j_{\tau+1}} \leftarrow \mathcal{D}_{\mathbb{Z},(m\gamma\chi_0)^\tau}^{Nm}$ for $j_2, \ldots, j_{\tau+1} \in [Q]$. We claim that this is statistically indistinguishable from the original distribution. To see this, we observe that

$$\begin{aligned} &\left(\mathbf{s}(\mathbf{I}_N \otimes \mathbf{I}_m \otimes \mathbf{r}_{2,j_2}^\top \otimes \cdots \otimes \mathbf{r}_{\tau+1,j_{\tau+1}}^\top) + \mathbf{e}'_{j_2,\ldots,j_{\tau+1}}\right) \left(\mathbf{I}_N \otimes \mathbf{r}_{1,j_1}^\top\right) + \mathbf{e}_{j_1,\ldots,j_{\tau+1}} \\ = \quad &\mathbf{s}(\mathbf{I}_N \otimes \mathbf{r}_{1,j_1}^\top \otimes \mathbf{r}_{2,j_2}^\top \otimes \cdots \otimes \mathbf{r}_{\tau+1,j_{\tau+1}}^\top) + \underbrace{\mathbf{e}'_{j_2,\ldots,j_{\tau+1}} \left(\mathbf{I}_N \otimes \mathbf{r}_{1,j_1}^\top\right) + \mathbf{e}_{j_1,\ldots,j_{\tau+1}}}_{=\text{error}} \end{aligned}$$

and these distributions only differ in the error terms. We have

$$\mathbf{e}_{j_1,\ldots,j_{\tau+1}} \approx_s \mathbf{e}'_{j_2,\ldots,j_{\tau+1}} \left(\mathbf{I}_N \otimes \mathbf{r}_{1,j_1}^\top\right) + \mathbf{e}_{j_1,\ldots,j_{\tau+1}}$$

by the smudging lemma, since we have $\chi \geq (m\gamma\chi_0)^{\tau+1}$ and $\|\mathbf{e}'_{j_2,\ldots,j_{\tau+1}}\left(\mathbf{I}_N \otimes \mathbf{r}_{1,j_1}^\top\right)\|_\infty \leq m\gamma \cdot \mathrm{poly}(\lambda) \cdot \|\mathbf{e}'_{j_2,\ldots,j_{\tau+1}}\|_\infty \leq (m\gamma)^{\tau+1} \cdot \chi_0^\tau \cdot \mathrm{poly}(\lambda) \cdot$.

In the next step, we replace each $\mathbf{s}'_{j_2,\ldots,j_{\tau+1}}$ with random vectors. Namely, $\{\mathbf{c}_{j_1,\ldots,j_{\tau+1}}\}_{j_1,\ldots,j_{\tau+1}}$ are sampled as

$$\mathbf{c}_{j_1,\ldots,j_{\tau+1}} = \mathbf{s}'_{j_2,\ldots,j_{\tau+1}}\left(\mathbf{I}_N \otimes \mathbf{r}_{1,j_1}^\top\right) + \mathbf{e}_{j_1,\ldots,j_{\tau+1}},$$

where $\mathbf{s}'_{j_2,\ldots,j_{\tau+1}} \leftarrow \mathbb{Z}_q^{Nm}$. We can see that this change is computationally indistinguishable, by applying the induction hypothesis for each combination of indices $(j_2, \ldots, j_{\tau+1})$. We then use the induction hypothesis for the case of $k = 1$ to replace $\{\mathbf{c}_{j_1,\ldots,j_{\tau+1}}\}_{j_1}$ with random vectors for each combination of $j_2, \ldots, j_{\tau+1}$ one by one. This brings us to the distribution where all $\mathbf{c}_{j_1,\ldots,j_{\tau+1}}$ are random vectors. This completes the proof of the lemma. □

Lemma 3.10. *Let $n = n(\lambda)$, $m = m(\lambda)$, $N = N(\lambda)$, $q = q(\lambda)$, $\chi = \chi(\lambda)$, and $k = O(1)$ be parameters. If $\mathsf{LWE}(n, (m+1)^k N, q, \chi)$ holds, then, the following distributions are computationally indistinguishable:*

$$\begin{aligned}&\left(\{\mathbf{B}_i\}_{i\in[0,k]}, \mathbf{s}(\mathbf{B}_0 \otimes \mathbf{I}_m^{\otimes k}) + \mathbf{e}_0, \ldots, \mathbf{s}(\mathbf{B}_i \otimes \mathbf{I}_m^{\otimes(k-i)}) + \mathbf{e}_i, \ldots, \mathbf{s}\mathbf{B}_k + \mathbf{e}_k\right)\\ &\approx_c \left(\{\mathbf{B}_i\}_{i\in[0,k]}, \mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_k\right)\end{aligned}$$

where $\mathbf{B}_i \leftarrow \mathbb{Z}_q^{nm^i \times Nm^i}$, $\mathbf{e}_i \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^{m^k N}$, and $\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_k \leftarrow \mathbb{Z}_q^{m^k N}$ for $i \in [0,k]$, $\mathbf{s} \leftarrow \mathbb{Z}_q^{nm^k}$.

Proof. We prove the lemma by induction. First, the statement is trivially true when $k = 0$. We then prove that the statement is true for $k = \tau + 1$ assuming it is true for $k = \tau$. To show this, we first observe that any $\mathbf{x} \in \mathbb{Z}_q^{nm^{\tau+1}}$ can be written as $\mathbf{x} = \sum_{j\in[m]} \mathbf{x}_j \otimes \boldsymbol{\epsilon}_j$ using $\mathbf{x}_j \in \mathbb{Z}_q^{nm^\tau}$ where $\boldsymbol{\epsilon}_j$ is the j-th canonical unit vector of dimension m. We then have

$$\begin{aligned}\mathbf{s}(\mathbf{B}_i \otimes \mathbf{I}_m^{\otimes(\tau+1-i)}) + \mathbf{e}_i &= \sum_{j\in[m]} (\mathbf{s}_j \otimes \boldsymbol{\epsilon}_j)((\mathbf{B}_i \otimes \mathbf{I}_m^{\otimes(\tau-i)}) \otimes \mathbf{I}_m) + \sum_{j\in[m]} \mathbf{e}_{i,j} \otimes \boldsymbol{\epsilon}_j\\ &= \sum_{j\in[m]} \left(\mathbf{s}_j(\mathbf{B}_i \otimes \mathbf{I}_m^{\otimes(\tau-i)}) + \mathbf{e}_{i,j}\right) \otimes \boldsymbol{\epsilon}_j\end{aligned}$$

for $i \in [0,\tau]$ where we decompose $\mathbf{s}$ and $\mathbf{e}_i$ as $\mathbf{s} = \sum_{j\in[m]} \mathbf{s}_j \otimes \boldsymbol{\epsilon}_j$ and $\mathbf{e}_i = \sum_{j\in[m]} \mathbf{e}_{i,j} \otimes \boldsymbol{\epsilon}_j$. We also have

$$\mathbf{s}\mathbf{B}_{\tau+1} + \mathbf{e}_{\tau+1} = \sum_{j\in[m]} \mathbf{s}_j(\mathbf{I}_{nm^\tau} \otimes \boldsymbol{\epsilon}_j)\mathbf{B}_{\tau+1} + \mathbf{e}_{\tau+1}.$$

Therefore, omitting $\{\mathbf{B}_i\}_{i\in[0,\tau+1]}$, the input to the adversary is

$$
\begin{aligned}
& \left(\left\{\mathbf{s}(\mathbf{B}_i \otimes \mathbf{I}_m^{\otimes(\tau+1-i)}) + \mathbf{e}_i\right\}_{i\in[0,\tau]}, \mathbf{s}\mathbf{B}_{\tau+1} + \mathbf{e}_{\tau+1}\right) \\
= & \left(\left\{\sum_{j\in[m]} \left(\mathbf{s}_j(\mathbf{B}_i \otimes \mathbf{I}_m^{\otimes(\tau-i)}) + \mathbf{e}_{i,j}\right) \otimes \boldsymbol{\epsilon}_j\right\}_{i\in[0,\tau]}, \sum_{j\in[m]} \mathbf{s}_j(\mathbf{I}_{nm^\tau} \otimes \boldsymbol{\epsilon}_j)\mathbf{B}_{\tau+1} + \mathbf{e}_{\tau+1}\right) \\
\approx_c & \left(\left\{\mathbf{c}_{i,1} \otimes \boldsymbol{\epsilon}_1 + \sum_{j\in[2,m]} \left(\mathbf{s}_j(\mathbf{B}_i \otimes \mathbf{I}_m^{\otimes(\tau-i)}) + \mathbf{e}_{i,j}\right) \otimes \boldsymbol{\epsilon}_j\right\}_{i\in[0,\tau]}, \mathbf{c}_{\tau+1} + \sum_{j\in[2,m]} \mathbf{s}_j(\mathbf{I}_{nm^\tau} \otimes \boldsymbol{\epsilon}_j)\mathbf{B}_{\tau+1}\right) \\
\equiv & \left(\left\{\mathbf{c}_{i,1} \otimes \boldsymbol{\epsilon}_1 + \sum_{j\in[2,m]} \left(\mathbf{s}_j(\mathbf{B}_i \otimes \mathbf{I}_m^{\otimes(\tau-i)}) + \mathbf{e}_{i,j}\right) \otimes \boldsymbol{\epsilon}_j\right\}_{i\in[0,\tau]}, \mathbf{c}_{\tau+1}\right) \\
\approx_c & \left(\left\{\mathbf{c}_{i,1} \otimes \boldsymbol{\epsilon}_1 + \sum_{j\in[2,m]} \mathbf{c}_{i,j} \otimes \boldsymbol{\epsilon}_j\right\}_{i\in[0,\tau]}, \mathbf{c}_{\tau+1}\right) \\
\equiv & \left(\{\mathbf{c}_i\}_{i\in[0,\tau]}, \mathbf{c}_{\tau+1}\right)
\end{aligned}
$$

where $\mathbf{c}_i \leftarrow \mathbb{Z}_q^{m^{\tau+1}N}$ and $\mathbf{c}_{i,j} \leftarrow \mathbb{Z}_q^{m^\tau N}$. In the third line, we used the induction hypothesis for secret $\mathbf{s}' := \mathbf{s}_1$ and matrices

$$
\mathbf{B}'_i := \begin{cases} \mathbf{B}_i \in \mathbb{Z}_q^{n\times Nm^i} & \text{if } i \in [0, \tau-1] \\ (\mathbf{B}_\tau \| (\mathbf{I}_{nm^\tau} \otimes \boldsymbol{\epsilon}_j)\mathbf{B}_{\tau+1}) \in \mathbb{Z}_q^{n\times N(m+1)m^\tau} & \text{if } i = \tau \end{cases}
$$

and the parameter $N' = (m+1)N$. Note that the number of columns in each $\mathbf{B}'_i$ is at most $N'm^i$ and thus the indistinguishability follows from the induction hypothesis and the assumption $\mathsf{LWE}(n, (m+1)^{\tau+1}N, q, \chi)$. The indistinguishability of the fifth line also holds from the induction hypothesis similarly to the third line. Here, we apply the induction hypothesis for each $j \in [2, m]$ one by one, by setting secret $\mathbf{s}' := \mathbf{s}_j$ and matrices $\mathbf{B}'_i = \mathbf{B}_i$ for all $i \in [0, \tau]$. This completes the proof of the lemma. □

4 Two-input ABE from Evasive and Tensor LWE

4.1 Construction

In this section, we define our construction of 2ABE for P using evasive LWE (Assumption 3.1) and tensor LWE (Assumption 3.5). As discussed in Section 1, when restricted to NC_1, our construction can be modified to rely only on evasive LWE. We defer the details of this modification to Section 5 and focus on circuit class P for this section.

Let ℓ be the length of the attribute in each slot. The construction supports general circuits with bounded depth d and the decryption is possible when $f(\mathbf{x}_0\|\mathbf{x}_1) = 0$, where $\mathbf{x}_0$ is the attribute associated with a ciphertext, $\mathbf{x}_1$ is the attribute associated with the first slot key, and f is the function associated with the second slot key. Below $\mathbf{I}$ refers to $\mathbf{I}_m$.

$\mathsf{Setup}(1^\lambda)$**:** The setup algorithm takes as input the security parameter λ and does the following:

- Sample $\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_2 \leftarrow \mathbb{Z}_q^{n\times m\ell}$; $(\mathbf{B}, \mathbf{B}^{-1}_{\tau_B}) \leftarrow \mathsf{TrapGen}(1^\lambda, 2nm + nm^2, (2nm + nm^2)w)$; $(\mathbf{C}, \mathbf{C}^{-1}_{\tau_C}) \leftarrow \mathsf{TrapGen}(1^\lambda, nm, nmw)$, where $w \in O(\log q)$; $\mathbf{D} \leftarrow \mathbb{Z}_q^{n\times m}$.

- Output $\mathsf{mpk} = (\mathbf{A}_0, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$, where $\mathbf{A} = (\mathbf{A}_1 \| \mathbf{A}_2)$, $\mathsf{msk} = (\mathbf{B}^{-1}_{\tau_B}, \mathbf{C}^{-1}_{\tau_C})$.

$\mathsf{Enc}(\mathsf{mpk}, \mathbf{x}_0, \mu)$**:** The encryption algorithm takes as input the master public key mpk, an attribute $\mathbf{x}_0$ and message bit $\mu \in \{0,1\}$ and does the following:

- If $\mu = 1$, sample $\mathbf{c}_1 \leftarrow \mathbb{Z}_q^{m^2\ell}$, $\mathbf{c}_2 \leftarrow \mathbb{Z}_q^{(2nm+nm^2)w}$.
- Else,
  - Sample $\mathbf{s}, \mathbf{s}_0 \leftarrow \mathbb{Z}_q^{nm}$ and $\mathbf{s}_1 \leftarrow \mathbb{Z}_q^{nm^2}$.
  - Sample error vectors $\mathbf{e}_1 \leftarrow \mathcal{D}_{\mathbb{Z},\chi_1}^{m^2\ell}$, $\mathbf{e}_2 \leftarrow \mathcal{D}_{\mathbb{Z},\chi_2}^{(2nm+nm^2)w}$.
  - Compute $\mathbf{c}_1 = (\mathbf{s}, \mathbf{s}_0) \begin{pmatrix} (\mathbf{A}_1 - \mathbf{x}_0 \otimes \mathbf{G}) \otimes \mathbf{I} \\ \mathbf{A}_0 \otimes \mathbf{I} \end{pmatrix} + \mathbf{e}_1$.
  - Compute $\mathbf{c}_2 = (\mathbf{s}, \mathbf{s}_0, \mathbf{s}_1)\mathbf{B} + \mathbf{e}_2$.
- Output $\mathsf{ct}_{\mathbf{x}_0} = (\mathbf{c}_1, \mathbf{c}_2)$.

$\mathsf{KeyGen}_1(\mathsf{msk}, \mathbf{x}_1)$**:** The keygen algorithm for slot 1 takes as input the master secret key msk and the slot attribute $\mathbf{x}_1 \in \{0,1\}^\ell$ and does the following:

- Sample $\mathbf{r} \leftarrow \mathcal{D}^m_{\mathbb{Z},\gamma}$.
- Sample $\mathbf{L}_{\mathbf{x}_1} \leftarrow \mathbf{B}^{-1}\left(\begin{pmatrix} (\mathbf{A}_2 - \mathbf{x}_1 \otimes \mathbf{G}) \otimes \mathbf{r}^\top & & \\ & \mathbf{A}_0 \otimes \mathbf{r}^\top & \\ & & \mathbf{C} \otimes \mathbf{r}^\top \end{pmatrix}, \tau_B\right)$.
- Output $\mathsf{sk}_{1,\mathbf{x}_1} = (\mathbf{r}, \mathbf{L}_{\mathbf{x}_1})$.

$\mathsf{KeyGen}_2(\mathsf{msk}, f)$ The keygen algorithm for slot 2 takes as input the master secret key msk and slot function f, which is a function represented as a binary circuit $f : \{0,1\}^{2\ell} \to \{0,1\}$ and does the following:

- Sample $\mathbf{t} \leftarrow \mathcal{D}^m_{\mathbb{Z},\gamma}$, $\mathbf{U} \leftarrow \mathcal{D}^{m\times m}_{\mathbb{Z},\gamma}$.
- Compute $\mathbf{H}_f = \mathsf{EvalF}(\mathbf{A}, f)$ and $\mathbf{A}_f = \mathbf{A}\mathbf{H}_f$.
- Sample $\mathbf{M}_f \leftarrow \mathbf{B}^{-1}\left(\begin{pmatrix} \mathbf{A}_f\mathbf{U} \otimes \mathbf{I} \\ \mathbf{0}_{nm\times m^2} \\ \mathbf{D} \otimes \mathbf{t}^\top \otimes \mathbf{I} \end{pmatrix}, \tau_B\right)$ and $\mathbf{N}_f \leftarrow \mathbf{C}^{-1}\left((\mathbf{D} \otimes \mathbf{t}^\top), \tau_C\right)$.
- Output $\mathsf{sk}_{2,f} = (\mathbf{t}, \mathbf{U}, \mathbf{M}_f, \mathbf{N}_f)$.

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{ct}_{\mathbf{x}_0}, \mathsf{sk}_{1,\mathbf{x}_1}, \mathsf{sk}_{2,f})$ The decryption algorithm takes as input the ciphertext $\mathsf{ct}_{\mathbf{x}_0}$, key $\mathsf{sk}_{1,\mathbf{x}_1}$ for slot 1, and key $\mathsf{sk}_{2,f}$ for slot 2 and does the following:

- Parse $\mathsf{ct}_{\mathbf{x}_0}$ as $(\mathbf{c}_1, \mathbf{c}_2)$, $\mathsf{sk}_{1,\mathbf{x}_1}$ as $(\mathbf{r}, \mathbf{L}_{\mathbf{x}_1})$ and $\mathsf{sk}_{2,f}$ as $(\mathbf{t}, \mathbf{U}, \mathbf{M}_f, \mathbf{N}_f)$.
- Compute $\widehat{\mathbf{H}}_{\mathbf{A},f,(\mathbf{x}_0\|\mathbf{x}_1)} = \mathsf{EvalFX}(\mathbf{A}, f, (\mathbf{x}_0\|\mathbf{x}_1))$.
- Compute the following:
  $$\begin{aligned} &\mathbf{d}_1 = \mathbf{c}_1, && (\mathbf{d}_2,\ \mathbf{d}_3,\ \mathbf{d}_4) = \mathbf{c}_2\mathbf{L}_{\mathbf{x}_1}, && \mathbf{d}_5 = \mathbf{c}_2\mathbf{M}_f, \\ &\mathbf{d}_6 = \mathbf{N}_f, && \mathbf{d}'_1 = \mathbf{d}_1(\mathbf{I}_{m\ell} \otimes \mathbf{r}^\top) - \mathbf{d}_3, && \mathbf{d}'_5 = \mathbf{d}_5(\mathbf{I} \otimes \mathbf{r}^\top) - \mathbf{d}_4\mathbf{d}_6, \\ &\mathbf{d}_7 = (\mathbf{d}'_1\|\mathbf{d}_2)\widehat{\mathbf{H}}_{\mathbf{A},f,(\mathbf{x}_0\|\mathbf{x}_1)}\mathbf{U}, && && \mathbf{d}_8 = \mathbf{d}_7 - \mathbf{d}'_5. \end{aligned}$$
  Note that $\mathbf{d}_6$ is a matrix of size $nmw \times m$ and $\mathbf{d}_i$ for all $i \neq 6$ are vectors.
- If $\|\mathbf{d}_8\|_\infty \leq \beta_0$ (where β_0 is as defined in the Sec. 4.2) then output $\mu' = 0$, else output 1.

4.2 Correctness, Parameters and Security

Correctness: Here, we show correctness of the scheme.
When $\mu = 1$: We first show the correctness for the case of $\mu = 1$. For an honest run of the protocol, $\mathbf{d}_1$ is distributed uniformly at random over its domain. Then, since $\mathbf{r} \neq \mathbf{0}$ with overwhelming probability and thus $\mathbf{I}_{m\ell} \otimes \mathbf{r}^\top$ is a full-rank matrix, $\mathbf{d}_1'$ is distributed uniformly at random over its domain. Then, since the topmost m rows of $\widehat{\mathbf{H}}_{\mathbf{A},f,(\mathbf{x}_0\|\mathbf{x}_1)}$ constitutes an identity matrix by Lemma 2.6, $(\mathbf{d}_1'\|\mathbf{d}_2)\widehat{\mathbf{H}}_{\mathbf{A},f,(\mathbf{x}_0\|\mathbf{x}_1)}$ is distributed uniformly at random over its domain. Finally, since each column of $\mathbf{U}$ is chosen from $\mathcal{D}^m_{\mathbb{Z},\gamma}$, with overwhelming probability, there exists $i \in [m]$ such that the i-th column of $\mathbf{U}$ is not a zero vector. This in turn implies that that the i-th entry of $\mathbf{d}_7$ is distributed uniformly at random over $\mathbb{Z}_q$. Since we set $\beta_0/q = \lambda^{-\omega(1)}$, the probability that the decryption algorithm falsely outputs 0 is negligible as desired.

When $\mu = 0$: Next, we show the correctness for the case of $\mu = 0$. For an honest run of the protocol, we have

- $\mathbf{d}_1 = \mathbf{c}_1 = \mathbf{s}((\mathbf{A}_1 - \mathbf{x}_0 \otimes \mathbf{G}) \otimes \mathbf{I}) + \mathbf{s}_0(\mathbf{A}_0 \otimes \mathbf{I}) + \mathbf{e}_1$.

  Let $(\mathbf{e}_2',\ \mathbf{e}_3',\ \mathbf{e}_4') = \mathbf{e}_2 \cdot \mathbf{L}_{\mathbf{x}_1}$
- $\mathbf{d}_2 = \mathbf{s}((\mathbf{A}_2 - \mathbf{x}_1 \otimes \mathbf{G}) \otimes \mathbf{r}^\top) + \mathbf{e}_2'$,
- $\mathbf{d}_3 = \mathbf{s}_0(\mathbf{A}_0 \otimes \mathbf{r}^\top) + \mathbf{e}_3'$,
- $\mathbf{d}_4 = \mathbf{s}_1(\mathbf{C} \otimes \mathbf{r}^\top) + \mathbf{e}_4'$,
- $\mathbf{d}_5 = \mathbf{s}(\mathbf{A}_f\mathbf{U} \otimes \mathbf{I}) + \mathbf{s}_1(\mathbf{D} \otimes \mathbf{t}^\top \otimes \mathbf{I}) + \mathbf{e}_5'$, where $\mathbf{e}_5' = \mathbf{e}_2 \cdot \mathbf{M}_f$
- $\mathbf{d}_1'$ is computed as

$$\begin{aligned}
\mathbf{d}_1' &= \mathbf{d}_1(\mathbf{I}_{m\ell} \otimes \mathbf{r}^\top) - \mathbf{d}_3 \\
&= (\mathbf{s}((\mathbf{A}_1 - \mathbf{x}_0 \otimes \mathbf{G}) \otimes \mathbf{I}) + \mathbf{s}_0(\mathbf{A}_0 \otimes \mathbf{I}) + \mathbf{e}_1)(\mathbf{I}_{m\ell} \otimes \mathbf{r}^\top) - \mathbf{s}_0(\mathbf{A}_0 \otimes \mathbf{r}^\top) - \mathbf{e}_3' \\
&= \mathbf{s}((\mathbf{A}_1 - \mathbf{x}_0 \otimes \mathbf{G}) \otimes \mathbf{r}^\top) + \mathbf{s}_0(\mathbf{A}_0 \otimes \mathbf{r}^\top) - \mathbf{s}_0(\mathbf{A}_0 \otimes \mathbf{r}^\top) + \mathbf{e}_1'' \\
&= \mathbf{s}((\mathbf{A}_1 - \mathbf{x}_0 \otimes \mathbf{G}) \otimes \mathbf{r}^\top) + \mathbf{e}_1''
\end{aligned}$$

  Here $\mathbf{e}_1'' = \mathbf{e}_1(\mathbf{I}_{m\ell} \otimes \mathbf{r}^\top) - \mathbf{e}_3'$
- $\mathbf{d}_5'$ is computed as

$$\begin{aligned}
\mathbf{d}_5' &= \mathbf{d}_5(\mathbf{I} \otimes \mathbf{r}^\top) - \mathbf{d}_4\mathbf{d}_6 \\
&= (\mathbf{s}(\mathbf{A}_f\mathbf{U} \otimes \mathbf{I}) + \mathbf{s}_1(\mathbf{D} \otimes \mathbf{t}^\top \otimes \mathbf{I}) + \mathbf{e}_5')(\mathbf{I} \otimes \mathbf{r}^\top) - (\mathbf{s}_1(\mathbf{C} \otimes \mathbf{r}^\top) + \mathbf{e}_4')\mathbf{N}_f \\
&= \mathbf{s}(\mathbf{A}_f\mathbf{U} \otimes \mathbf{r}^\top) + \mathbf{s}_1(\mathbf{D} \otimes \mathbf{t}^\top \otimes \mathbf{r}^\top) - \mathbf{s}_1(\mathbf{D} \otimes \mathbf{t}^\top \otimes \mathbf{r}^\top) + \mathbf{e}_5'', \\
&= \mathbf{s}(\mathbf{A}_f\mathbf{U} \otimes \mathbf{r}^\top) + \mathbf{e}_5''
\end{aligned}$$

  where we use $(\mathbf{C} \otimes \mathbf{r}^\top)\mathbf{N}_f = \mathbf{C}\mathbf{N}_f \otimes \mathbf{r}^\top = \mathbf{D} \otimes \mathbf{t}^\top \otimes \mathbf{r}^\top$ and define $\mathbf{e}_5'' := \mathbf{e}_5'(\mathbf{I} \otimes \mathbf{r}^\top) - \mathbf{e}_4'\mathbf{N}_f$ on the third line.

- $\mathbf{d}_7$ is computed as

$$\begin{aligned}
\mathbf{d}_7 &= (\mathbf{d}'_1 \| \mathbf{d}_2) \cdot (\widehat{\mathbf{H}}_{\mathbf{A},f,(\mathbf{x}_0\|\mathbf{x}_1)} \mathbf{U}) \\
&= ((\mathbf{s}((\mathbf{A}_1 - \mathbf{x}_0 \otimes \mathbf{G}) \otimes \mathbf{r}^\top) + \mathbf{e}''_1) \| (\mathbf{s}((\mathbf{A}_2 - \mathbf{x}_1 \otimes \mathbf{G}) \otimes \mathbf{r}^\top) + \mathbf{e}'_2)) \cdot (\widehat{\mathbf{H}}_{\mathbf{A},f,(\mathbf{x}_0\|\mathbf{x}_1)} \mathbf{U}) \\
&= (\mathbf{s}((\mathbf{A}_1 \| \mathbf{A}_2 - (\mathbf{x}_0 \| \mathbf{x}_1) \otimes \mathbf{G}) \otimes \mathbf{r}^\top) + (\mathbf{e}''_1 \| \mathbf{e}'_2)) \cdot (\widehat{\mathbf{H}}_{\mathbf{A},f,(\mathbf{x}_0\|\mathbf{x}_1)} \mathbf{U}) \\
&= \mathbf{s}((\mathbf{A}_f - f(\mathbf{x}_0\|\mathbf{x}_1)\mathbf{G}) \otimes \mathbf{r}^\top)\mathbf{U} + \mathbf{e}'_7 \\
&= \mathbf{s}((\mathbf{A}_f - f(\mathbf{x}_0\|\mathbf{x}_1)\mathbf{G})\mathbf{U} \otimes \mathbf{r}^\top) + \mathbf{e}'_7 \\
&= \mathbf{s}(\mathbf{A}_f \mathbf{U} \otimes \mathbf{r}^\top) + \mathbf{e}'_7 \quad \text{if } f(\mathbf{x}_0\|\mathbf{x}_1) = 0
\end{aligned}$$

  where we define $\mathbf{e}'_7 := (\mathbf{e}''_1 \| \mathbf{e}'_2)\widehat{\mathbf{H}}_{\mathbf{A},f,(\mathbf{x}_0\|\mathbf{x}_1)}\mathbf{U}$ on the fourth line.

- $\mathbf{d}_8 = \mathbf{d}_7 - \mathbf{d}'_5 = \mathbf{s}(\mathbf{A}_f\mathbf{U} \otimes \mathbf{r}^\top) + \mathbf{e}'_7 - \mathbf{s}(\mathbf{A}_f\mathbf{U} \otimes \mathbf{r}^\top) - \mathbf{e}''_5 = \mathbf{e}'_7 - \mathbf{e}''_5$ which is small ($\leq \beta_0$).

Therefore, the decryption algorithm outputs 0 as desired.

Error Bound: The error term is bounded as follows. Let β_0 denote the error bound.

$$\begin{aligned}
\|\mathbf{e}'_7\|_\infty + \|\mathbf{e}''_5\|_\infty &= \|(\mathbf{e}''_1 \| \mathbf{e}'_2)\widehat{\mathbf{H}}_{\mathbf{A},f,(\mathbf{x}_0\|\mathbf{x}_1)}\mathbf{U}\|_\infty + \|\mathbf{e}'_5(\mathbf{I} \otimes \mathbf{r}^\top) - \mathbf{e}'_4\mathbf{N}_f\|_\infty \\
&= \|(\mathbf{e}_1(\mathbf{I}_{m\ell} \otimes \mathbf{r}^\top) - \mathbf{e}'_3 \| \mathbf{e}'_2)\widehat{\mathbf{H}}_{\mathbf{A},f,(\mathbf{x}_0\|\mathbf{x}_1)}\mathbf{U}\|_\infty + \|\mathbf{e}'_5(\mathbf{I} \otimes \mathbf{r}^\top) - \mathbf{e}'_4\mathbf{N}_f\|_\infty \\
&\leq ((\chi_1\gamma + \chi_2\tau_B)\beta\gamma + \chi_2\tau_B\gamma + \chi_2\tau_B\tau_C)\,\mathrm{poly}(m) \\
&\quad \text{since } (\mathbf{e}'_2,\ \mathbf{e}'_3,\ \mathbf{e}'_4) = \mathbf{e}_2\mathbf{L}_{\mathbf{x}_1} \text{ and } \mathbf{e}'_5 = \mathbf{e}_2\mathbf{M}_f \\
&\leq \beta_0.
\end{aligned}$$

Parameters: We set the parameters as follows.

$$\begin{array}{ccc}
n = \mathrm{poly}(\lambda, d), & m = O(n \log q), & \tau_B = O(\sqrt{(2nm + nm^2)\log q}), \\
\tau_C = O(\sqrt{nm \log q}), & \beta = (2m)^d, & \gamma = \chi_1 = \lambda^{\omega(1)}, \\
\chi_3 = \chi_4 = \chi_6 = \gamma\chi_1\lambda^{\omega(1)}, & \chi_7 = \chi_3\chi_4\beta\gamma\lambda^{\omega(1)}, & \chi_5 = \lambda^{\omega(1)}\chi_7, \\
\chi_2 = \chi_5\lambda^{\omega(1)}, & \beta_0 = \beta\gamma^2\tau_B\tau_C\chi_1\chi_2\lambda^{\omega(1)}, & q = \beta_0\lambda^{\omega(1)}.
\end{array}$$

In the above, χ_3, χ_4, χ_5, χ_6, and χ_7 are the parameters that only appear in the security proof.

Security: Here, we prove the following theorem, which asserts the security of our scheme.

Theorem 4.1. *Assuming evasive* LWE *(Assumption 3.1), tensor* LWE *(Assumption 3.5), and* LWE*, our construction for 2-input* ABE *for* P *satisfies very selective security (Definition 2.2). Moreover, for the restricted class* NC_1*, our construction for 2-input* ABE *relies only on evasive* LWE.

Proof. To prove the security, we need to prove the indistinguishability of the following two distributions. Let Q_c and Q_s be the number of slot 1 and slot 2 key queries, respectively. In the following, for simplicity, we let $Q_c = Q_s = Q$, which can be assumed without loss of generality.

Distribution D_0:

$$\left(\begin{array}{cc}
\mathsf{mpk}, & \mathbf{c}_1 = (\mathbf{s}, \mathbf{s}_0)\begin{pmatrix} (\mathbf{A}_1 - \mathbf{x}_0 \otimes \mathbf{G}) \otimes \mathbf{I} \\ \mathbf{A}_0 \otimes \mathbf{I} \end{pmatrix} + \mathbf{e}_1, \quad \mathbf{c}_2 = (\mathbf{s}, \mathbf{s}_0, \mathbf{s}_1)\mathbf{B} + \mathbf{e}_2 \\
\left\{\mathsf{sk}_{1,\mathbf{x}_{1,i}} = \left(\mathbf{r}_i, \mathbf{L}_{\mathbf{x}_{1,i}}\right)\right\}_{i\in[Q]}, & \left\{\mathsf{sk}_{2,f_j} = \left(\mathbf{t}_j, \mathbf{U}_j, \mathbf{M}_{f_j}, \mathbf{N}_{f_j}\right)\right\}_{j\in[Q]}
\end{array}\right)$$

Distribution D_1:

$$\left(\begin{array}{ccc} \mathsf{mpk}, & \mathbf{c}_1 \leftarrow \mathbb{Z}_q^{m^2\ell}, & \mathbf{c}_2 \leftarrow \mathbb{Z}_q^{(2nm+nm^2)w} \\ \left\{\mathsf{sk}_{1,\mathbf{x}_{1,i}} = \left(\mathbf{r}_i, \mathbf{L}_{\mathbf{x}_{1,i}}\right)\right\}_{i\in[Q]}, & \left\{\mathsf{sk}_{2,f_j} = \left(\mathbf{t}_j, \mathbf{U}_j, \mathbf{M}_{f_j}, \mathbf{N}_{f_j}\right)\right\}_{j\in[Q]} & \end{array}\right)$$

where $\mathbf{x}_0$ is the attribute for the encryption, $\mathbf{x}_{1,1}, \dots, \mathbf{x}_{1,Q}$ are the key queries for slot 1, $f_1, \dots, f_Q$ are key queries for slot 2, and $\mathsf{sk}_{1,\mathbf{x}_{1,i}}$ (resp., sk_{2,f_j}) is secret key for $\mathbf{x}_{1,i}$ (resp., f_j) for slot 1 (resp., slot 2). In particular, we have

$$\mathbf{L}_{\mathbf{x}_{1,i}} \leftarrow \mathbf{B}^{-1}\left(\begin{pmatrix} (\mathbf{A}_2 - \mathbf{x}_{1,i} \otimes \mathbf{G}) \otimes \mathbf{r}_i^\top & & \\ & \mathbf{A}_0 \otimes \mathbf{r}_i^\top & \\ & & \mathbf{C} \otimes \mathbf{r}_i^\top \end{pmatrix}, \tau_B\right)$$

and

$$\mathbf{M}_{f_j} \leftarrow \mathbf{B}^{-1}\left(\begin{pmatrix} \mathbf{A}_{f_j}\mathbf{U}_j \otimes \mathbf{I} \\ \mathbf{0}_{nm\times m^2} \\ \mathbf{D} \otimes \mathbf{t}_j^\top \otimes \mathbf{I} \end{pmatrix}, \tau_B\right), \quad \mathbf{N}_{f_j} \leftarrow \mathbf{C}^{-1}\left((\mathbf{D} \otimes \mathbf{t}_j^\top), \tau_C\right).$$

We note that we have $f_j(\mathbf{x}_0 \| \mathbf{x}_{1,i}) = 1$ for all $i, j \in [Q]$ by the definition of the security game. We can see that D_0 and D_1 are the views of the adversary when $\mu = 0$ and $\mu = 1$ are encrypted, respectively. We then apply our variant of evasive LWE (Lemma 3.4) assumption for matrix $\mathbf{B}$ with the sampler Samp that outputs $(\mathbf{S}, \mathbf{P}, \mathsf{aux} = (\mathsf{aux}_1, \mathsf{aux}_2))$ defined as follows:[6]

$$\begin{aligned}
\mathbf{S} &= (\mathbf{s}, \mathbf{s}_0, \mathbf{s}_1) \\
\mathsf{aux}_1 &= \mathbf{c}_1 = (\mathbf{s}, \mathbf{s}_0)\begin{pmatrix} (\mathbf{A}_1 - \mathbf{x}_0 \otimes \mathbf{G}) \otimes \mathbf{I} \\ \mathbf{A}_0 \otimes \mathbf{I} \end{pmatrix} + \mathbf{e}_1, \\
\mathsf{aux}_2 &= (\mathbf{x}_0, \mathbf{x}_{1,1}, \dots, \mathbf{x}_{1,Q}, f_1, \dots, f_Q, \mathsf{coins}_{\mathcal{A}}, \mathbf{r}_1, \dots, \mathbf{r}_Q, \mathbf{t}_1, \dots, \mathbf{t}_Q, \mathbf{U}_1, \dots, \mathbf{U}_Q, \\
&\qquad \mathbf{N}_{f_1}, \dots, \mathbf{N}_{f_Q}, \mathbf{A}_0, \mathbf{A}, \mathbf{C}, \mathbf{D}) \\
\mathbf{P}_0 &= \left((\mathbf{A}_2 - \mathbf{x}_{1,1} \otimes \mathbf{G}) \otimes \mathbf{r}_1^\top \| \cdots \| (\mathbf{A}_2 - \mathbf{x}_{1,Q} \otimes \mathbf{G}) \otimes \mathbf{r}_Q^\top\right) \\
\mathbf{P}_1 &= (\mathbf{A}_0 \otimes \mathbf{r}_1^\top \| \cdots \| \mathbf{A}_0 \otimes \mathbf{r}_Q^\top) \\
\mathbf{P}_2 &= \left(\mathbf{C} \otimes \mathbf{r}_1^\top \| \cdots \| \mathbf{C} \otimes \mathbf{r}_Q^\top\right) \\
\mathbf{P}_3 &= \left(\begin{pmatrix} \mathbf{A}_{f_1}\mathbf{U}_1 \otimes \mathbf{I} \\ \mathbf{0}_{nm\times m^2} \\ \mathbf{D} \otimes \mathbf{t}_1^\top \otimes \mathbf{I} \end{pmatrix} \Bigg\| \cdots \Bigg\| \begin{pmatrix} \mathbf{A}_{f_Q}\mathbf{U}_Q \otimes \mathbf{I} \\ \mathbf{0}_{nm\times m^2} \\ \mathbf{D} \otimes \mathbf{t}_Q^\top \otimes \mathbf{I} \end{pmatrix}\right) \\
\mathbf{P} &= \left(\begin{pmatrix} \mathbf{P}_0 & & \\ & \mathbf{P}_1 & \\ & & \mathbf{P}_2 \end{pmatrix} \Bigg\| \mathbf{P}_3\right)
\end{aligned}$$

where $\mathsf{coins}_{\mathcal{A}}$ is adversary's coin. By Lemma 3.4, to prove that D_0 and D_1 are computationally indistinguishable, it suffices to show the computational indistinguishability of the following distributions:

[6] By Lemma 3.4, it suffices to invoke the evasive LWE for a modified sampler that outputs random aux_1, instead of aux_1 that is dependent on $(\mathbf{s}, \mathbf{s}_0)$. The same comments apply to other invocations of the assumption.

Distribution D_0':

$$\begin{pmatrix} \mathbf{c}_1 = (\mathbf{s}, \mathbf{s}_0) \begin{pmatrix} (\mathbf{A}_1 - \mathbf{x}_0 \otimes \mathbf{G}) \otimes \mathbf{I} \\ \mathbf{A}_0 \otimes \mathbf{I} \end{pmatrix} + \mathbf{e}_1, & \mathbf{c}_2 = (\mathbf{s}, \mathbf{s}_0, \mathbf{s}_1)\mathbf{B} + \mathbf{e}_2, & \mathbf{B} \\ \{\mathbf{c}_{3,i}, \mathbf{c}_{4,i}, \mathbf{c}_{5,i}\}_{i \in [Q]}, & \{\mathbf{c}_{6,j}\}_{j \in [Q]}, & \mathsf{aux}' \end{pmatrix}$$

Distribution D_1':

$$\begin{pmatrix} \mathbf{w}_1, & \mathbf{w}_2, & \mathbf{B} \\ \{\mathbf{w}_{3,i}, \mathbf{w}_{4,i}, \mathbf{w}_{5,i}\}_{i \in [Q]}, & \{\mathbf{w}_{6,j}\}_{j \in [Q]}, & \mathsf{aux}' \end{pmatrix}$$

In the above distributions,

$$\begin{aligned} &(\mathbf{c}_{3,i},\ \mathbf{c}_{4,i},\ \mathbf{c}_{5,i},\ \mathbf{c}_{6,j}) = \\ &\qquad (\mathbf{s}\ , \mathbf{s}_0,\ \mathbf{s}_1) \left(\begin{pmatrix} (\mathbf{A}_2 - \mathbf{x}_{1,i} \otimes \mathbf{G}) \otimes \mathbf{r}_i^\top & & \\ & \mathbf{A}_0 \otimes \mathbf{r}_i^\top & \\ & & \mathbf{C} \otimes \mathbf{r}_i^\top \end{pmatrix} \middle\| \begin{pmatrix} \mathbf{A}_{f_j} \mathbf{U}_j \otimes \mathbf{I} \\ \mathbf{0}_{nm \times m^2} \\ \mathbf{D} \otimes \mathbf{t}_j^\top \otimes \mathbf{I} \end{pmatrix} \right) \\ &\qquad\qquad + (\mathbf{e}_{3,i},\ \mathbf{e}_{4,i},\ \mathbf{e}_{5,i},\ \mathbf{e}_{6,j}) \end{aligned}$$

where aux' above is defined as aux_2 in distribution D_0, $\mathbf{e}_{3,i} \leftarrow \mathcal{D}_{\mathbb{Z},\chi_3}^{m\ell}$, $\mathbf{e}_{4,i} \leftarrow \mathcal{D}_{\mathbb{Z},\chi_4}^{m\ell}$, $\mathbf{e}_{5,i} \leftarrow \mathcal{D}_{\mathbb{Z},\chi_5}^{nmw}$, and $\mathbf{e}_{6,j} \leftarrow \mathcal{D}_{\mathbb{Z},\chi_6}^{m^2}$ and all the $\mathbf{w}$ vectors are of the same dimension as the corresponding $\mathbf{c}$ vector and chosen randomly from their respective domains. Note that we set $\chi_2 > \chi_3, \chi_4, \chi_5, \chi_6$ so that we can rely on quantitatively weaker evasive LWE assumption (See Remark 3.3). We also note that here, we have $\chi_5 \neq \chi_3 = \chi_4 = \chi_6$, where Gaussian distributions with different standard deviations are mixed. We refer to Remark 3.2 for details. We have

$$\begin{aligned} &\mathbf{c}_{3,i} = \mathbf{s}((\mathbf{A}_2 - \mathbf{x}_{1,i} \otimes \mathbf{G}) \otimes \mathbf{r}_i^\top) + \mathbf{e}_{3,i} \\ &\mathbf{c}_{4,i} = \mathbf{s}_0(\mathbf{A}_0 \otimes \mathbf{r}_i^\top) + \mathbf{e}_{4,i} \\ &\mathbf{c}_{5,i} = \mathbf{s}_1(\mathbf{C} \otimes \mathbf{r}_i^\top) + \mathbf{e}_{5,i} \text{ which can be rewritten as } \mathbf{s}_1(\mathbf{I}_{nm} \otimes \mathbf{r}_i^\top)\mathbf{C} + \mathbf{e}_{5,i} \\ &\mathbf{c}_{6,j} = \mathbf{s}(\mathbf{A}_{f_j}\mathbf{U}_j \otimes \mathbf{I}) + \mathbf{s}_1(\mathbf{D} \otimes \mathbf{t}_j^\top \otimes \mathbf{I}) + \mathbf{e}_{6,j}. \end{aligned}$$

We then apply the evasive LWE assumption once again, now for matrix $\mathbf{C}$ with sampler Samp' that outputs $(\mathbf{S}', \mathbf{P}', \mathsf{aux}' = (\mathsf{aux}_1', \mathsf{aux}_2'))$ defined as follows:

$$\begin{aligned} \mathbf{S}' &= \begin{pmatrix} \mathbf{s}_1(\mathbf{I}_{nm} \otimes \mathbf{r}_1^\top) \\ \vdots \\ \mathbf{s}_1(\mathbf{I}_{nm} \otimes \mathbf{r}_Q^\top) \end{pmatrix} \\ \mathsf{aux}_1' &= (\mathbf{c}_1, \mathbf{c}_2, \{\mathbf{c}_{3,i}, \mathbf{c}_{4,i}\}_{i \in [Q]}, \{\mathbf{c}_{6,j}\}_{j \in [Q]}) \\ \mathsf{aux}_2' &= (\mathbf{x}_0, \mathbf{x}_{1,1}, \ldots, \mathbf{x}_{1,Q}, f_1, \ldots, f_Q, \mathsf{coins}_{\mathcal{A}}, \mathbf{r}_1, \ldots, \mathbf{r}_Q, \mathbf{t}_1, \ldots, \mathbf{t}_Q, \mathbf{U}_1, \ldots, \mathbf{U}_Q, \mathbf{A}_0, \mathbf{A}, \mathbf{B}, \mathbf{D}) \\ \mathbf{P}' &= (\mathbf{D} \otimes \mathbf{t}_1^\top \| \cdots \| \mathbf{D} \otimes \mathbf{t}_Q^\top) \end{aligned}$$

where $\mathbf{c}_1$, $\mathbf{c}_2$, $\mathbf{c}_{3,i}$, $\mathbf{c}_{4,i}$, and $\mathbf{c}_{6,j}$ are chosen as in distribution D_0'. By Lemma 3.4, it suffices to prove the computational indistinguishability of the following distributions:

Distribution D_0'':

$$\begin{pmatrix} \mathbf{c}_1, & \mathbf{c}_2, & \mathbf{C}, & \\ \{\mathbf{c}_{3,i}, \mathbf{c}_{4,i}, \mathbf{c}_{5,i}\}_{i \in [Q]}, & \{\mathbf{c}_{6,j}\}_{j \in [Q]}, & \{\mathbf{c}_{7,i,j}\}_{i,j \in [Q]}, & \mathsf{aux}'' \end{pmatrix}$$

Distribution D''_1:

$$\begin{pmatrix} \mathbf{w}_1, & \mathbf{w}_2, & \mathbf{C}, & \\ \{\mathbf{w}_{3,i}, \mathbf{w}_{4,i}, \mathbf{w}_{5,i}\}_{i\in[Q]}, & \{\mathbf{w}_{6,j}\}_{j\in[Q]}, & \{\mathbf{w}_{7,i,j}\}_{i,j\in[Q]}, & \mathsf{aux}'' \end{pmatrix}$$

where aux'' is defined as aux'_2 in distribution D'_0,

$$\mathbf{c}_{7,i,j} = \mathbf{s}_1(\mathbf{D} \otimes \mathbf{t}_j^\top \otimes \mathbf{r}_i^\top) + \mathbf{e}_{7,i,j}, \qquad \text{where } \mathbf{e}_{7,i,j} \leftarrow D_{\mathbb{Z},\chi_7}^m$$

and $\mathbf{w}_{7,i,j}$ is a random vector with the same dimension as $\mathbf{c}_{7,i,j}$. Note that we set $\chi_5 > \chi_7$ so that we can rely on quantitatively weaker evasive LWE assumption (See Remark 3.3). The rest of the vectors are defined as in distribution D'_0 and D'_1. From the above discussion, it suffices to prove Lemma 4.2 in the following to complete the proof of Theorem 4.1. □

Lemma 4.2. *Distributions D''_0 and D''_1 are computationally indistinguishable under the hardness assumption of* LWE *and tensor* LWE.

Proof. We prove the lemma via the following hybrids.

G_0 : This is same as D''_0.

G_1 : In this hybrid, the challenger computes $\mathbf{c}_{4,i}$ as

$$\mathbf{c}_{4,i} = \mathbf{c}_1(\mathbf{I}_{m\ell} \otimes \mathbf{r}_i^\top) - \underbrace{\left(\mathbf{s}((\mathbf{A}_1 - \mathbf{x}_0 \otimes \mathbf{G}) \otimes \mathbf{r}_i^\top) + \mathbf{e}_{4,i}\right)}_{:=\mathbf{c}'_{4,i}}.$$

G_2 : In this hybrid, the challenger samples $\mathbf{c}_1$ and $\mathbf{c}_2$ randomly as $\mathbf{c}_1 \leftarrow \mathbb{Z}_q^{m^2\ell}$, $\mathbf{c}_2 \leftarrow \mathbb{Z}_q^{(2nm+nm^2)w}$.

G_3: In this hybrid, $\mathbf{c}_{7,i,j}$ is computed as $\mathbf{c}_{7,i,j} = \mathbf{c}_{6,j}(\mathbf{I} \otimes \mathbf{r}_i^\top) - \underbrace{\left(\mathbf{s}(\mathbf{A}_{f_j}\mathbf{U}_j \otimes \mathbf{r}_i^\top) + \mathbf{e}_{7,i,j}\right)}_{:=\mathbf{c}'_{7,i,j}}$.

G_4 : In this hybrid, $\mathbf{c}_{5,i}$ and $\mathbf{c}_{6,j}$ are chosen randomly as $\mathbf{c}_{5,i} \leftarrow \mathbb{Z}_q^{nmw}$ and $\mathbf{c}_{6,j} \leftarrow \mathbb{Z}_q^{m^2}$.

G_5: In this hybrid, $\mathbf{c}'_{7,i,j}$ is computed differently as

$$\mathbf{c}'_{7,i,j} = [\mathbf{c}'_{4,i} \| \mathbf{c}_{3,i}]\widehat{\mathbf{H}}_{\mathbf{A},f,(\mathbf{x}_0\|\mathbf{x}_{1,i})}\mathbf{U}_j + \underbrace{\mathbf{s}(\mathbf{G}\mathbf{U}_j \otimes \mathbf{r}_i^\top) + \mathbf{e}_{7,i,j}}_{\mathbf{c}''_{7,i,j}}.$$

G_6 : In this hybrid, $\mathbf{c}_{3,i} \leftarrow \mathbb{Z}_q^{m\ell}$, $\mathbf{c}'_{4,i} \leftarrow \mathbb{Z}_q^{m\ell}$ and $\mathbf{c}''_{7,i,j} \leftarrow \mathbb{Z}_q^m$.

G_7 : In this hybrid, $\mathbf{c}_{4,i} \leftarrow \mathbb{Z}_q^{m\ell}$, $\mathbf{c}_{7,i,j} \leftarrow \mathbb{Z}_q^m$.

It is easy to see that the distribution in G_7 is the same as that of D''_1.

Indistinguishability of hybrids:

We prove the indistinguishability between the hybrid distributions via the following claims.

Claim 4.3. $\mathsf{G}_0 \approx_s \mathsf{G}_1$.

Proof. The two hybrids differ only in the error term in $\mathbf{c}_{4,i}$ and are indistinguishable due to the smudging lemma.

In G_0:

$$\mathbf{c}_{4,i} = \mathbf{s}_0(\mathbf{A}_0 \otimes \mathbf{r}_i^\top) + \mathbf{e}_{4,i}$$

In G_1:

$$\begin{aligned}\mathbf{c}_{4,i} &= \mathbf{c}_1(\mathbf{I}_{m\ell} \otimes \mathbf{r}_i^\top) - \Big(\mathbf{s}((\mathbf{A}_1 - \mathbf{x}_0 \otimes \mathbf{G}) \otimes \mathbf{r}_i^\top) + \mathbf{e}_{4,i}\Big)\\ &= (\mathbf{s}((\mathbf{A}_1 - \mathbf{x}_0 \otimes \mathbf{G}) \otimes \mathbf{I}) + \mathbf{s}_0(\mathbf{A}_0 \otimes \mathbf{I}) + \mathbf{e}_1)(\mathbf{I}_{m\ell} \otimes \mathbf{r}_i^\top) - \mathbf{s}((\mathbf{A}_1 - \mathbf{x}_0 \otimes \mathbf{G}) \otimes \mathbf{r}_i^\top) - \mathbf{e}_{4,i}\\ &= \mathbf{s}((\mathbf{A}_1 - \mathbf{x}_0 \otimes \mathbf{G}) \otimes \mathbf{r}_i^\top) + \mathbf{s}_0(\mathbf{A}_0 \otimes \mathbf{r}_i^\top) - \mathbf{s}((\mathbf{A}_1 - \mathbf{x}_0 \otimes \mathbf{G}) \otimes \mathbf{r}_i^\top) + \mathbf{e}_1(\mathbf{I}_{m\ell} \otimes \mathbf{r}_i^\top) - \mathbf{e}_{4,i}\\ &= \mathbf{s}_0(\mathbf{A}_0 \otimes \mathbf{r}_i^\top) + \mathbf{e}_1(\mathbf{I}_{m\ell} \otimes \mathbf{r}_i^\top) - \mathbf{e}_{4,i}\end{aligned}$$

Clearly, the two hybrids differ only in the error terms in $\mathbf{c}_{4,i}$. Thus, the indistinguishability follows due to the following:

$$\mathbf{e}_{4,i} \approx -\mathbf{e}_{4,i} + \mathbf{e}_1(\mathbf{I}_{m\ell} \otimes \mathbf{r}_i^\top)$$

which is true since the distribution of $-\mathbf{e}_{4,i}$ is the same as that of $\mathbf{e}_{4,i}$ by the symmetry of the discrete Gaussian distribution and $\chi_4 \geq \gamma\chi_1\lambda^{\omega(1)}$. □

Claim 4.4. $\mathsf{G}_1 \approx_c \mathsf{G}_2$ *due to* LWE*.*

Proof. Let us write $\mathbf{B}$ as $(\mathbf{B}_U^\top \ \ \mathbf{B}_M^\top \ \ \mathbf{B}_L^\top)^\top$. Then we can see that the indistinguishability follows from LWE by applying Lemma 3.10 for $k = 1$, which implies $(\mathbf{A}_0, \mathbf{B}_M, \mathbf{s}_0(\mathbf{A}_0 \otimes \mathbf{I}) + \mathbf{e}_1, \mathbf{s}_0\mathbf{B}_M + \mathbf{e}_2) \approx_c (\mathbf{A}_0, \mathbf{B}_M, \mathbf{w}_1', \mathbf{w}_2')$, where $\mathbf{w}_1' \leftarrow \mathbb{Z}_q^{m^2\ell}, \mathbf{w}_2' \leftarrow \mathbb{Z}_q^{(2nm+nm^2)w}$.

In particular, let $\mathbf{B} = (\mathbf{B}_U^\top \ \ \mathbf{B}_M^\top \ \ \mathbf{B}_L^\top)^\top$. Then
In G_1,

$$\begin{aligned}(\mathbf{c}_1, \mathbf{c}_2) &= (\mathbf{s}((\mathbf{A}_1 - \mathbf{x}_0 \otimes \mathbf{G}) \otimes \mathbf{I}) + \mathbf{s}_0(\mathbf{A}_0 \otimes \mathbf{I}) + \mathbf{e}_1, \mathbf{s}\mathbf{B}_U + \mathbf{s}_0\mathbf{B}_M + \mathbf{s}_1\mathbf{B}_L + \mathbf{e}_2)\\ &\approx_c (\mathbf{s}((\mathbf{A}_1 - \mathbf{x}_0 \otimes \mathbf{G}) \otimes \mathbf{I}) + \mathbf{w}_1', \mathbf{s}\mathbf{B}_U + \mathbf{s}_1\mathbf{B}_L + \mathbf{w}_2') \quad \text{(from LWE)}\\ &\approx_s (\mathbf{w}_1, \mathbf{w}_2) \quad \text{where } \mathbf{w}_1 \leftarrow \mathbb{Z}_q^{m^2\ell}, \mathbf{w}_2 \leftarrow \mathbb{Z}_q^{(2nm+nm^2)w}\end{aligned}$$

□

Claim 4.5. $\mathsf{G}_2 \approx_s \mathsf{G}_3$

Proof. The two hybrids differ only in the error terms in $\mathbf{c}_{7,i,j}$ and are indistinguishable due to the smudging lemma.
In G_2:

$$\mathbf{c}_{7,i,j} = \mathbf{s}_1(\mathbf{I}_{nm} \otimes \mathbf{r}_i^\top)(\mathbf{D} \otimes \mathbf{t}_j^\top) + \mathbf{e}_{7,i,j}$$

In G_3:

$$\begin{aligned}\mathbf{c}_{7,i,j} &= \mathbf{c}_{6,j}(\mathbf{I} \otimes \mathbf{r}_i^\top) - \mathbf{s}(\mathbf{A}_{f_j}\mathbf{U}_j \otimes \mathbf{r}_i^\top) - \mathbf{e}_{7,i,j}\\ &= (\mathbf{s}(\mathbf{A}_{f_j}\mathbf{U}_j \otimes \mathbf{I}) + \mathbf{s}_1(\mathbf{D} \otimes \mathbf{t}_j^\top \otimes \mathbf{I}) + \mathbf{e}_{6,j})(\mathbf{I} \otimes \mathbf{r}_i^\top) - \mathbf{s}(\mathbf{A}_{f_j}\mathbf{U}_j \otimes \mathbf{r}_i^\top) - \mathbf{e}_{7,i,j}\\ &= \mathbf{s}(\mathbf{A}_{f_j}\mathbf{U}_j \otimes \mathbf{r}_i^\top) + \mathbf{s}_1(\mathbf{D} \otimes \mathbf{t}_j^\top \otimes \mathbf{r}_i^\top) + \mathbf{e}_{6,j}(\mathbf{I} \otimes \mathbf{r}_i^\top) - \mathbf{s}(\mathbf{A}_{f_j}\mathbf{U}_j \otimes \mathbf{r}_i^\top) - \mathbf{e}_{7,i,j}\\ &= \mathbf{s}_1(\mathbf{I}_{nm} \otimes \mathbf{r}_i^\top)(\mathbf{D} \otimes \mathbf{t}_j^\top) + \mathbf{e}_{6,j}(\mathbf{I} \otimes \mathbf{r}_i^\top) - \mathbf{e}_{7,i,j}\end{aligned}$$

Clearly, the two hybrids differ only in the error terms in $\mathbf{c}_{7,i,j}$. Thus, the indistinguishability follows due to the following:

$$\mathbf{e}_{7,i,j} \quad \approx_s \quad -\mathbf{e}_{7,i,j} + \mathbf{e}_{6,j}(\mathbf{I} \otimes \mathbf{r}_i^\top)$$

which is true since $\chi_7 \geq \gamma\chi_6\lambda^{\omega(1)}$ and by the symmetry of the discrete Gaussian distribution. □

Claim 4.6. $\mathsf{G}_3 \approx \mathsf{G}_4$

Proof. The indistinguishability follows from Lemma 4.10. □

Claim 4.7. $\mathsf{G}_4 \approx_s \mathsf{G}_5$.

Proof. The two hybrids differ only in the error terms in $\mathbf{c}_{7,i,j}$. The indistinguishability follows from the smudging lemma.
In G_4,

$$\mathbf{c}'_{7,i,j} = \mathbf{s}(\mathbf{A}_{f_j}\mathbf{U}_j \otimes \mathbf{r}_i^\top) + \mathbf{e}_{7,i,j}.$$

In G_5,

$$\begin{aligned}
\mathbf{c}'_{7,i,j} &= (\mathbf{c}'_{4,i} \| \mathbf{c}_{3,i})\widehat{\mathbf{H}}_{\mathbf{A},f_j,(\mathbf{x}_0\|\mathbf{x}_{1,i})}\mathbf{U}_j + \mathbf{s}(\mathbf{G}\mathbf{U}_j \otimes \mathbf{r}_i^\top) + \mathbf{e}_{7,i,j} \\
&= \Big(\mathbf{s}((\mathbf{A}_1 - \mathbf{x}_0 \otimes \mathbf{G}) \otimes \mathbf{r}_i^\top) + \mathbf{e}_{4,i} | \mathbf{s}((\mathbf{A}_2 - \mathbf{x}_{1,i} \otimes \mathbf{G}) \otimes \mathbf{r}_i^\top) + \mathbf{e}_{3,i}\Big)\, \widehat{\mathbf{H}}_{\mathbf{A},f_j,(\mathbf{x}_0\|\mathbf{x}_{1,i})}\mathbf{U}_j \\
&\quad + \mathbf{s}(\mathbf{G}\mathbf{U}_j \otimes \mathbf{r}_i^\top) + \mathbf{e}_{7,i,j} \\
&= \mathbf{s}((\mathbf{A}_1 \| \mathbf{A}_2 - (\mathbf{x}_0 \| \mathbf{x}_{1,i}) \otimes \mathbf{G}) \otimes \mathbf{r}_i^\top)\widehat{\mathbf{H}}_{\mathbf{A},f_j,(\mathbf{x}_0\|\mathbf{x}_{1,i})}\mathbf{U}_j + (\mathbf{e}_{4,i} \| \mathbf{e}_{3,i})\widehat{\mathbf{H}}_{\mathbf{A},f_j,(\mathbf{x}_0\|\mathbf{x}_{1,i})}\mathbf{U}_j \\
&\quad + \mathbf{s}(\mathbf{G}\mathbf{U}_j \otimes \mathbf{r}_i^\top) + \mathbf{e}_{7,i,j} \\
&= \mathbf{s}((\mathbf{A}_f - f_j(\mathbf{x}_0 \| \mathbf{x}_{1,i})\mathbf{G})\mathbf{U}_j \otimes \mathbf{r}_i^\top) + (\mathbf{e}_{4,i} \| \mathbf{e}_{3,i})\widehat{\mathbf{H}}_{\mathbf{A},f_j,(\mathbf{x}_0\|\mathbf{x}_{1,i})}\mathbf{U}_j + \mathbf{s}(\mathbf{G}\mathbf{U}_j \otimes \mathbf{r}_i^\top) + \mathbf{e}_{7,i,j} \\
&= \mathbf{s}(\mathbf{A}_f\mathbf{U}_j \otimes \mathbf{r}_i^\top) + (\mathbf{e}_{4,i} \| \mathbf{e}_{3,i})\widehat{\mathbf{H}}_{\mathbf{A},f_j,(\mathbf{x}_0\|\mathbf{x}_{1,i})}\mathbf{U}_j + \mathbf{e}_{7,i,j} \quad (\text{since } f_j(\mathbf{x}_0 \| \mathbf{x}_{1,i}) = 1)
\end{aligned}$$

Clearly, the two hybrids differ only in the error terms in $\mathbf{c}'_{7,i,j}$. Thus, the indistinguishability follows due to the following:

$$\mathbf{e}_{7,i,j} \approx_s \mathbf{e}_{7,i,j} + (\mathbf{e}_{4,i} \| \mathbf{e}_{3,i})\widehat{\mathbf{H}}_{\mathbf{A},f_j,(\mathbf{x}_0\|\mathbf{x}_{1,i})}\mathbf{U}_j$$

which is true when $\chi_7 \geq \chi_3\chi_4\beta\gamma\lambda^{\omega(1)}$, where $\|\widehat{\mathbf{H}}_{\mathbf{A},f_j,(\mathbf{x}_0\|\mathbf{x}_{1,i})}\|_\infty \leq \beta$ □

Claim 4.8. $\mathsf{G}_5 \approx_c \mathsf{G}_6$ *under the tensor-*LWE *assumption.*

Proof. The indistinguishability between the two hybrids follows from tensor-LWE which implies

$$\begin{aligned}
\mathbf{A}_1, \mathbf{A}_2, \{\mathbf{U}_j, \mathbf{r}_i^\top, \mathbf{s}(\mathbf{I}_n \otimes \mathbf{r}_i^\top)(\mathbf{A}_1 - \mathbf{x}_0 \otimes \mathbf{G}) + \mathbf{e}_{4,i}, \mathbf{s}(\mathbf{I}_n \otimes \mathbf{r}_i^\top)(\mathbf{A}_2 - \mathbf{x}_{1,i} \otimes \mathbf{G}) + \mathbf{e}_{3,i}, \mathbf{s}(\mathbf{I}_n \otimes \mathbf{r}_i^\top)\mathbf{G}\mathbf{U}_j + \mathbf{e}_{7,i,j}\}_{i,j} \\
\approx_c \mathbf{A}_1, \mathbf{A}_2, \{\mathbf{U}_j, \mathbf{r}_i^\top, \mathsf{random}, \mathsf{random}\}_{i,j}.
\end{aligned}$$

□

Claim 4.9. $\mathsf{G}_6 \equiv \mathsf{G}_7$

Proof. This follows since in G_6, $\mathbf{c}_{4,i}$ and $\mathbf{c}_{7,i,j}$ are masked by random vectors $\mathbf{c}'_{4,i}$ and $\mathbf{c}'_{7,i,j}$, respectively. □

To complete the proof of Lemma 4.2, it remains to prove the following.

Lemma 4.10. *Given* $\{\mathbf{t}_j\}_{j\in[Q]}$, $\{\mathbf{r}_i\}_{i\in[Q]}$, $\mathbf{C}, \mathbf{D}$,

$$(\{\mathbf{z}_{C,i} := \mathbf{s}_1(\mathbf{I}_{nm} \otimes \mathbf{r}_i^\top)\mathbf{C} + \mathbf{e}_{5,i}\}_i, \{\mathbf{z}_{D,j} := \mathbf{s}_1(\mathbf{D} \otimes \mathbf{t}_j^\top \otimes \mathbf{I}) + \mathbf{e}_{6,j}\}_j) \approx_c \ (\{\mathbf{w}'_{5,i}\}_i, \{\mathbf{w}'_j\}_{6,j}),$$

where $\mathbf{w}'_{5,i} \leftarrow \mathbb{Z}_q^{nmw}$ *and* $\mathbf{w}'_{6,j} \leftarrow \mathbb{Z}_q^{m^2}$ *assuming* LWE.

Proof. We prove the lemma by considering a sequence of games where we start from the LHS and gradually change it to that of RHS in a way that is not noticed by the adversary.

$\tilde{\mathsf{G}}_0$: This is the same distribution as in LHS.

$\tilde{\mathsf{G}}_1$: In this hybrid, we change the distribution to be

$$\left\{\mathbf{z}_{C,i} = \underbrace{\left(\mathbf{s}_1(\mathbf{C}\otimes\mathbf{I}) + \mathbf{e}_C\right)}_{:=\mathbf{s}_C}(\mathbf{I}_{nmw}\otimes\mathbf{r}_i^\top) + \mathbf{e}_{5,i}\right\}_i, \quad \left\{\mathbf{z}_{D,j} = \underbrace{\left(\mathbf{s}_1(\mathbf{D}\otimes\mathbf{I}^{\otimes 2}) + \mathbf{e}_D\right)}_{:=\mathbf{s}_D}(\mathbf{I}\otimes\mathbf{t}_j^\top\otimes\mathbf{I}) + \mathbf{e}_{6,j}\right\}_j,$$

where $\mathbf{e}_C \leftarrow \mathcal{D}_{\mathbb{Z},\chi_1}^{nm^2w}$, $\mathbf{e}_D \leftarrow \mathcal{D}_{\mathbb{Z},\chi_1}^{m^3}$.

This hybrid differs from the previous one only in the error terms in $\mathbf{z}_{C,i}$ and $\mathbf{z}_{D,j}$. The indistinguishability follows from the smudging lemma.

To see this, observe that we have

$$\left(\mathbf{s}_1(\mathbf{C}\otimes\mathbf{I}) + \mathbf{e}_C\right)\left(\mathbf{I}_{nmw}\otimes\mathbf{r}_i^\top\right) + \mathbf{e}_{5,i} = \mathbf{s}_1(\mathbf{C}\otimes\mathbf{r}_i^\top) + \underbrace{\mathbf{e}_C(\mathbf{I}_{nmw}\otimes\mathbf{r}_i^\top) + \mathbf{e}_{5,i}}_{=\text{error}}$$

and

$$\left(\mathbf{s}_1(\mathbf{D}\otimes\mathbf{I}^{\otimes 2}) + \mathbf{e}_D\right)\left(\mathbf{I}\otimes\mathbf{t}_j^\top\otimes\mathbf{I}\right) + \mathbf{e}_{6,j} = \mathbf{s}_1(\mathbf{D}\otimes\mathbf{t}_j^\top\otimes\mathbf{I}) + \underbrace{\mathbf{e}_D(\mathbf{I}\otimes\mathbf{t}_j^\top\otimes\mathbf{I}) + \mathbf{e}_{6,j}}_{=\text{error}}.$$

Thus, the indistinguishability follows due to the following:

$$\mathbf{e}_{5,i} \approx_s \mathbf{e}_C(\mathbf{I}_{nmw}\otimes\mathbf{r}_i^\top) + \mathbf{e}_{5,i}, \quad \mathbf{e}_{6,j} \approx_s \mathbf{e}_D(\mathbf{I}\otimes\mathbf{t}_j^\top\otimes\mathbf{I}) + \mathbf{e}_{6,j},$$

which is true when $\chi_5, \chi_6 > \gamma\lambda^{\omega(1)}\chi_1$.

$\tilde{\mathsf{G}}_2$: In this hybrid, we replace $\mathbf{s}_C$ and $\mathbf{s}_D$ with random vectors sampled as $\mathbf{s}_C \leftarrow \mathbb{Z}_q^{nm^2w}$, $\mathbf{s}_D \leftarrow \mathbb{Z}_q^{m^3}$. This hybrid is indistinguishable from the previous one by Lemma 3.10 with $k = 2$ assuming LWE.

$\tilde{\mathsf{G}}_3$: In this game, $\{\mathbf{z}_{C,i}\}_i$ and $\{\mathbf{z}_{D,j}\}_j$ are replaced with random vectors sampled as $\mathbf{z}_{C,i} \leftarrow \mathbb{Z}_q^{nmw}$ and $\mathbf{z}_{D,j} \leftarrow \mathbb{Z}_q^{m^2}$ for all $i, j \in [Q]$. We can see that this hybrid is indistinguishable from the previous one by LWE with low norm samples (Lemma 2.4) once with respect to secret $\mathbf{s}_C$, and then with respect to $\mathbf{s}_D$.

It is clear that the distribution in $\tilde{\mathsf{G}}_3$ is the same as that of RHS in the statement of the lemma. This completes the proof of Lemma 4.2. □

5 Multi-Input ABE for Any Constant Arity

In this section, we extend the construction in Sec. 4 to construct k-ABE for any constant k using evasive LWE. Our main construction supports functions in NC_1 and proven secure assuming evasive LWE. We also discuss a variant that supports any polynomial size circuit of bounded depth, which can be proven secure assuming a strengthening of tensor LWE in addition.

Table 2: Summary of hybrids in proof of security for 2ABE construction. All the $\mathbf{w}$ and $\mathbf{w}'$ vectors are sampled randomly.

To prove $D_0'' \approx D_1''$: Given $\mathbf{C}$, $\mathsf{aux}'' = (\mathbf{x}, \{\mathbf{y}_i, \mathbf{r}_i\}_i, \mathsf{coins}_{\mathcal{A}}, \{f_j, \mathbf{t}_j, \mathbf{U}_j\}_j \mathbf{A}_0, \mathbf{A}, \mathbf{B}, \mathbf{D})$, to prove pseudorandomness of $\mathbf{c}_1, \mathbf{c}_2$, $\mathbf{c}_{3,i}$, $\mathbf{c}_{4,i}$, $\mathbf{c}_{5,i}$, $\mathbf{c}_{6,j}$, $\mathbf{c}_{7,i,j}$

	$\mathbf{c}_1$	$\mathbf{c}_2$	$\mathbf{c}_{3,i}$	$\mathbf{c}_{4,i}$	$\mathbf{c}_{5,i}$	$\mathbf{c}_{6,j}$	$\mathbf{c}_{7,i,j}$	Remark
G_0	$\mathbf{s}((\mathbf{A}_1 - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{I}) + \mathbf{s}_0(\mathbf{A}_0 \otimes \mathbf{I}) + \mathbf{e}_1$	$(\mathbf{s}, \mathbf{s}_0, \mathbf{s}_1)\mathbf{B} + \mathbf{e}_2$	$\mathbf{s}((\mathbf{A}_2 - \mathbf{y}_i \otimes \mathbf{G}) \otimes \mathbf{r}_i^\top) + \mathbf{e}_{3,i}$	$\mathbf{s}_0(\mathbf{A}_0 \otimes \mathbf{r}_i^\top) + \mathbf{e}_{4,i}$	$\mathbf{s}_1(\mathbf{I}_{nm} \otimes \mathbf{r}_i^\top)\mathbf{C} + \mathbf{e}_{5,i}$	$\mathbf{s}(\mathbf{A}_{f_j}\mathbf{U}_j \otimes \mathbf{I}) + \mathbf{s}_1(\mathbf{D} \otimes \mathbf{t}_j^\top \otimes \mathbf{I}) + \mathbf{e}_{6,j}$	$\mathbf{s}_1(\mathbf{D} \otimes \mathbf{t}_j^\top \otimes \mathbf{r}_i^\top) + \mathbf{e}_{7,i,j}$	
G_1	↓	↓	↓	$\mathbf{c}_1(\mathbf{I}_{m\ell} \otimes \mathbf{r}_i^\top) - (\mathbf{s}((\mathbf{A}_1 - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{r}_i^\top) + \mathbf{e}_{4,i})$	↓	↓	↓	smudging lemma
$\mathsf{G}_{1.5}$	$\mathbf{s}((\mathbf{A}_1 - \mathbf{x} \otimes \mathbf{G}) \otimes \mathbf{I}) + \mathbf{w}_1'$,	$(\mathbf{s},\ \mathbf{0},\ \mathbf{s}_1)\mathbf{B} + \mathbf{w}_2'$,	↓	↓	↓	↓	↓	LWE
G_2	$\mathbf{w}_1$	$\mathbf{w}_2$	↓	↓	↓	↓	↓	random mask
G_3	↓	↓	↓	↓	↓	↓	$\mathbf{c}_{6,j}(\mathbf{I} \otimes \mathbf{r}_i^\top) - \underbrace{\Big(\mathbf{s}(\mathbf{A}_{f_j}\mathbf{U}_j \otimes \mathbf{r}_i^\top) + \mathbf{e}_{7,i,j}\Big)}_{:=\mathbf{c}'_{7,i,j}}$	smudging
$\mathsf{G}_{3.5}$	↓	↓	↓	↓	$\mathbf{w}_{5,i}$	$\mathbf{s}(\mathbf{A}_{f_j}\mathbf{U}_j \otimes \mathbf{I}) + \mathbf{w}'_{6,j}$	↓	LWE
G_4	↓	↓	↓	↓	↓	$\mathbf{w}_{6,j}$		random mask
G_5	↓	↓	↓	↓	↓	↓	$\mathbf{c}_{6,j}(\mathbf{I} \otimes \mathbf{r}_i^\top) - [\mathbf{c}'_{4,i} \Vert \mathbf{c}_{3,i}]\widehat{\mathbf{H}}_{\mathbf{A},f_j,(\mathbf{x}\Vert\mathbf{y}_i)}\mathbf{U}_j - \underbrace{\big(\mathbf{s}(\mathbf{G}\mathbf{U}_j \otimes \mathbf{r}_i^\top) + \mathbf{e}_{7,i,j}\big)}_{:=\mathbf{c}''_{7,i,j}}$	smudging
G_6	↓	↓	$\mathbf{w}_{3,i}$	$\mathbf{c}_1(\mathbf{I}_{m\ell} \otimes \mathbf{r}_i^\top) - \mathbf{w}'_{4,i}$	↓	↓	$\mathbf{c}_{6,j}(\mathbf{I} \otimes \mathbf{r}_i^\top) - [\mathbf{c}'_{4,i} \Vert \mathbf{c}_{3,i}]\widehat{\mathbf{H}}_{\mathbf{A},f_j,(\mathbf{x}\Vert\mathbf{y}_i)}\mathbf{U}_j - \mathbf{w}''_{7,i,j}$	tensor LWE
G_7	↓	↓	↓	$\mathbf{w}_{4,i}$	↓	↓	$\mathbf{w}_{7,i,j}$	random mask

5.1 Construction for NC_1 Circuits

Here, we show our construction. Let ℓ be the length of each of the k attributes. Decryption is possible when $f(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{k-1}) = 0$, where $\mathbf{x}_0 \in \{0,1\}^\ell$ is the attribute associated with the public encryption, $\mathbf{x}_i \in \{0,1\}^\ell$ is the attribute associated with the slot i, and f is a binary circuit associated with the slot k key. Below $\mathbf{I}$ refers to $\mathbf{I}_m$. We require an upper bound on the depth of the circuit and denote it by d. We require $d = O(\log \lambda)$.

In the construction, we will use the low-norm variant of the lattice evaluation algorithms $(\mathsf{EvalF}, \mathsf{EvalFX})$ from Lemma 2.8.

$\mathsf{Setup}(1^\lambda)$**:** The setup algorithm takes as input the security parameter and does the following:

- Sample $\mathbf{A}_0, \ldots, \mathbf{A}_{k-1} \leftarrow \mathcal{D}_{\mathbb{Z},\gamma}^{m \times m\ell}$; $\mathbf{D}_0, \ldots, \mathbf{D}_{k-1} \leftarrow \mathbb{Z}_q^{n \times m\ell}$, $\mathbf{D}_k \leftarrow \mathbb{Z}_q^{n \times m}$, $\mathbf{U} \leftarrow \mathcal{D}_{\mathbb{Z},\gamma}^{m \times m}$. Let $\mathbf{A} = (\mathbf{A}_0, \ldots, \mathbf{A}_{k-1})$.
- Sample $(\mathbf{B}, \mathbf{B}_{\tau_B}^{-1}) \leftarrow \mathsf{TrapGen}(1^\lambda, m^{k+1} + (k+1)nm^k, (m^{k+1} + (k+1)nm^k)w)$, where $w \in O(\log q)$;
$\{(\mathbf{C}_i, \mathbf{C}_{i,\tau_C}^{-1}) \leftarrow \mathsf{TrapGen}(1^\lambda, (k+1)nm^{i-1}, (k+1)nm^{i-1}w)\}_{i \in [2,k]}$
- Set $\mathbf{C}_1 = \begin{pmatrix} \mathbf{D}_0 & & & \\ & \mathbf{D}_1 & & \\ & & \ddots & \\ & & & \mathbf{D}_k \end{pmatrix}$
- Output $\mathsf{mpk} = (\mathbf{A}, \mathbf{B}, \mathbf{C}_1, \ldots, \mathbf{C}_k, \mathbf{D}_0, \ldots, \mathbf{D}_k, \mathbf{U})$,
$\mathsf{msk} = (\mathbf{B}, \mathbf{B}_{\tau_B}^{-1}, \mathbf{C}_{2,\tau_C}^{-1}, \ldots, \mathbf{C}_{k,\tau_C}^{-1})$.

$\mathsf{Enc}(\mathsf{mpk}, \mathbf{x}_0, \mu)$**:** The Enc algorithm is a public encryption algorithm. It takes as input the master public key mpk, attribute $\mathbf{x}_0$ and message bit $\mu \in \{0,1\}$ and does the following:

- Sample $\mathbf{s} \leftarrow \mathbb{Z}_q^{m^{k+1}}$, $\mathbf{s}_0, \ldots, \mathbf{s}_k \leftarrow \mathbb{Z}_q^{nm^k}$.
- If $\mu = 1$, sample $\mathbf{c}_1 \leftarrow \mathbb{Z}_q^{\ell m^{k+1}}$, $\mathbf{c}_2 \leftarrow \mathbb{Z}_q^{(m^{k+1}+(k+1)nm^k)w}$.
Else, compute
  - $\mathbf{c}_1 = \mathbf{s}((\mathbf{A}_0 - \mathbf{x}_0 \otimes \mathbf{I}) \otimes \mathbf{I}^{\otimes k}) + \mathbf{s}_0(\mathbf{D}_0 \otimes \mathbf{I}^{\otimes k}) + \mathbf{e}_1$, where $\mathbf{e}_1 \leftarrow \mathcal{D}_{\mathbb{Z},\chi_1}^{\ell m^{k+1}}$.
  - $\mathbf{c}_2 = (\mathbf{s},\ \mathbf{s}_0,\ \cdots,\ \mathbf{s}_k)\mathbf{B} + \mathbf{e}_2$, where $\mathbf{e}_2 \leftarrow \mathcal{D}_{\mathbb{Z},\chi_2}^{(m^{k+1}+(k+1)nm^k)w}$.
- Output $\mathsf{ct}_{\mathbf{x}_0} = (\mathbf{c}_1, \mathbf{c}_2)$.

$\mathsf{KeyGen}_i(\mathsf{msk}, \mathbf{x}_i)$ **for** $1 \leq i \leq k-1$**:** The keygen algorithm for slot $1 \leq i \leq k-1$, takes as input the master secret key msk and attribute $\mathbf{x}_i$ and does the following:

- Samples $\mathbf{r}_i \leftarrow \mathcal{D}_{\mathbb{Z},\gamma}^m$
- Samples $\mathbf{X}_i \leftarrow \mathbf{B}^{-1}\left(\begin{pmatrix} (\mathbf{A}_i - \mathbf{x}_i \otimes \mathbf{I}) \otimes \mathbf{I}^{\otimes(i-1)} \otimes \mathbf{r}_i^\top \otimes \mathbf{I}^{\otimes(k-i)} \\ \mathbf{0}_{inm^k \times \ell m^k} \\ \mathbf{D}_i \otimes \mathbf{I}^{\otimes(i-1)} \otimes \mathbf{r}_i^\top \otimes \mathbf{I}^{\otimes(k-i)} \\ \mathbf{0}_{(k-i)nm^k \times \ell m^k} \end{pmatrix}, \tau_B\right)$, and $\mathbf{Y}_i \leftarrow \mathbf{C}_{i+1}^{-1}\left((\mathbf{C}_i \otimes \mathbf{r}_i^\top), \tau_C\right)$
- Returns $\mathsf{sk}_{i,\mathbf{x}_i} = (\mathbf{r}_i, \mathbf{X}_i, \mathbf{Y}_i)$

$\mathsf{KeyGen}_k(\mathsf{msk}, f)$**:** The keygen algorithm for slot k takes as input the master secret key, msk, and k-arity function f and does the following:

- Samples $\mathbf{r}_k \leftarrow \mathcal{D}^m_{\mathbb{Z},\gamma}$
- Computes $\mathbf{H}_f = \mathsf{EvalF}(\mathbf{A}, f)$ and $\mathbf{A}_f = \mathbf{A}\mathbf{H}_f$
- Computes $\mathbf{M}_f \leftarrow \mathbf{B}^{-1}\left(\begin{pmatrix} \mathbf{A}_f\mathbf{U} \otimes \mathbf{I}^{\otimes(k-1)} \otimes \mathbf{r}_k^\top \\ \mathbf{0}_{knm^k \times m^k} \\ \mathbf{D}_k \otimes \mathbf{I}^{\otimes(k-1)} \otimes \mathbf{r}_k^\top \end{pmatrix}, \tau_B\right)$ and

  $\mathbf{N}_f \leftarrow \mathbf{B}^{-1}\left(\begin{pmatrix} \mathbf{0}_{m^{k+1} \times (k+1)nm^{k-1}w} \\ \mathbf{C}_k \otimes \mathbf{r}_k^\top \end{pmatrix}, \tau_B\right)$
- Returns $\mathsf{sk}_{k,f} = (\mathbf{r}_k, \mathbf{M}_f, \mathbf{N}_f)$

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{ct}_{\mathbf{x}_0}, \mathsf{sk}_{1,\mathbf{x}_1}, \ldots, \mathsf{sk}_{k-1,\mathbf{x}_{k-1}}, \mathsf{sk}_{k,f})$ The decryption algorithm takes a ciphertext $\mathsf{ct}_{\mathbf{x}_0}$, k keys $\mathsf{sk}_{1,\mathbf{x}_1}, \ldots, \mathsf{sk}_{k-1,\mathbf{x}_{k-1}}$ and $\mathsf{sk}_{k,f}$ and does the following:

- Parse $\mathsf{ct}_{\mathbf{x}_0}$ as $(\mathbf{c}_1, \mathbf{c}_2)$, $\mathsf{sk}_{i,\mathbf{x}_i}$ as $(\mathbf{r}_i, \mathbf{X}_i, \mathbf{Y}_i)$ for $1 \leq i \leq k-1$ and $\mathsf{sk}_{k,f}$ as $(\mathbf{r}_k, \mathbf{M}_f, \mathbf{N}_f)$. Let $\mathbf{x} = (\mathbf{x}_0, \ldots, \mathbf{x}_{k-1})$.
- Compute $\widehat{\mathbf{H}}_{\mathbf{A},f,\mathbf{x}} = \mathsf{EvalFX}(\mathbf{A}, f, \mathbf{x})$.
- Compute the following
  * $\mathbf{d}'_0 = \mathbf{c}_1(\mathbf{I}_{m\ell} \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top)$
  * $\mathbf{d}'_i = \mathbf{c}_2\mathbf{X}_i(\mathbf{I}_{m\ell} \otimes \mathbf{r}_1^\top \cdots \otimes \mathbf{r}_{i-1}^\top \otimes \mathbf{r}_{i+1}^\top \otimes \cdots \otimes \mathbf{r}_k^\top)$, for $1 \leq i \leq k-1$,
  * $\mathbf{d}'_f = \mathbf{c}_2\mathbf{M}_f(\mathbf{I}_m \otimes \mathbf{r}_1^\top \cdots \otimes \mathbf{r}_{k-1}^\top)$
  * $(\mathbf{d}''_0,\ \cdots,\ \mathbf{d}''_{k-1},\ \mathbf{d}''_f) = \mathbf{c}_2\mathbf{N}_f\mathbf{Y}_{k-1}\cdots\mathbf{Y}_1$
  * $\mathbf{d}_i = \mathbf{d}'_i - \mathbf{d}''_i$, for $i = 0$ to $k-1$.
  * $\mathbf{d}_f = \mathbf{d}'_f - \mathbf{d}''_f$
  * $\mathbf{d} = (\mathbf{d}_0|\cdots|\mathbf{d}_{k-1})\widehat{\mathbf{H}}_{\mathbf{A},f,\mathbf{x}}\mathbf{U} - \mathbf{d}_f$
- If $\|\mathbf{d}\|_\infty \leq \beta_0$, where β_0 is as defined in section 5.2 then return $\mu = 0$, else return $\mu = 1$.

5.2 Correctness, Parameters and Security

Correctness. Here, we show the correctness of the scheme.

When $\mu = 1$: We first show the correctness for the case of $\mu = 1$. For an honest run of the protocol, $\mathbf{c}_1$ is distributed uniformly at random over its domain. Then, since $\mathbf{r}_i \neq \mathbf{0}$ for all $i \in [k]$ with overwhelming probability and thus $\mathbf{I}_{m\ell} \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top$ is a full-rank matrix, $\mathbf{d}'_0$ and thus $\mathbf{d}_0$ are distributed uniformly at random over their domains. Then, since the topmost m rows of $\widehat{\mathbf{H}}_{\mathbf{A},f,\mathbf{x}}$ constitutes an identity matrix by Lemma 2.8, $(\mathbf{d}_0\|\cdots\|\mathbf{d}_{k-1})\widehat{\mathbf{H}}_{\mathbf{A},f,\mathbf{x}}$ is distributed uniformly at random over its domain. Finally, since each column of $\mathbf{U}$ is chosen from $\mathcal{D}^m_{\mathbb{Z},\gamma}$, with overwhelming probability, there exists $i \in [m]$ such that the i-th column of $\mathbf{U}$ is not a zero vector. This in turn implies that that the i-th entry of $\mathbf{d}$ is distributed uniformly at random over $\mathbb{Z}_q$. Since we set $\beta_0/q = \lambda^{-\omega(1)}$, the probability that the decryption algorithm falsely outputs 0 is negligible as desired.

When $\mu = 0$: We now show the correctness for the case of $\mu = 0$.

* Let us first compute $\mathbf{d}'_0$.

$$\begin{aligned}
\mathbf{d}'_0 &= \mathbf{c}_1(\mathbf{I}_{m\ell} \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top) \\
&= \big(\mathbf{s} \cdot \big((\mathbf{A}_0 - \mathbf{x}_0 \otimes \mathbf{I}_m) \otimes \mathbf{I}^{\otimes k}\big) + \mathbf{s}_0 \cdot (\mathbf{D}_0 \otimes \mathbf{I}^{\otimes k})\big) \cdot \big(\mathbf{I}_{m\ell} \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top\big) + \mathbf{e}_{\mathbf{d}'_0} \\
&= \mathbf{s} \cdot \big((\mathbf{A}_0 - \mathbf{x}_0 \otimes \mathbf{I}_m) \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top\big) + \mathbf{s}_0 \cdot \big(\mathbf{D}_0 \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top\big) + \mathbf{e}_{\mathbf{d}'_0}
\end{aligned}$$

where $\mathbf{e}_{\mathbf{d}'_0} := \mathbf{e}_1 \cdot (\mathbf{I}_{m\ell} \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top)$.

* Let $1 \le i \le k-1$,

$$\begin{aligned}
\mathbf{d}'_i &= \mathbf{c}_2 \cdot \mathbf{X}_i \cdot \big(\mathbf{I}_{m\ell} \otimes \mathbf{r}_1^\top \cdots \otimes \mathbf{r}_{i-1}^\top \otimes \mathbf{r}_{i+1}^\top \otimes \cdots \otimes \mathbf{r}_k^\top\big) \\
&= ((\mathbf{s},\ \mathbf{s}_0,\ \cdots,\ \mathbf{s}_k)\,\mathbf{B} + \mathbf{e}_2) \cdot \mathbf{B}^{-1} \left(\begin{pmatrix} (\mathbf{A}_i - \mathbf{x}_i \otimes \mathbf{I}_m) \otimes \mathbf{I}^{\otimes(i-1)} \otimes \mathbf{r}_i^\top \otimes \mathbf{I}^{\otimes(k-i)} \\ \mathbf{0}_{inm^k \times \ell m^k} \\ \mathbf{D}_i \otimes \mathbf{I}^{\otimes(i-1)} \otimes \mathbf{r}_i^\top \otimes \mathbf{I}^{\otimes(k-i)} \\ \mathbf{0}_{(k-i)nm^k \times \ell m^k} \end{pmatrix}, \tau_B \right) \\
&\quad \cdot (\mathbf{I}_{m\ell} \otimes \mathbf{r}_1^\top \cdots \otimes \mathbf{r}_{i-1}^\top \otimes 1 \otimes \mathbf{r}_{i+1}^\top \otimes \cdots \otimes \mathbf{r}_k^\top) \\
&= \big(\mathbf{s}((\mathbf{A}_i - \mathbf{x}_i \otimes \mathbf{I}_m) \otimes \mathbf{I}^{\otimes(i-1)} \otimes \mathbf{r}_i^\top \otimes \mathbf{I}^{\otimes(k-i)}) + \mathbf{s}_i \cdot (\mathbf{D}_i \otimes \mathbf{I}^{\otimes(i-1)} \otimes \mathbf{r}_i^\top \otimes \mathbf{I}^{\otimes(k-i)})\big) \\
&\quad \cdot (\mathbf{I}_{m\ell} \otimes \mathbf{r}_1^\top \cdots \otimes \mathbf{r}_{i-1}^\top \otimes 1 \otimes \mathbf{r}_{i+1}^\top \otimes \cdots \otimes \mathbf{r}_k^\top) + \mathbf{e}_{\mathbf{d}'_i}. \\
&= \mathbf{s} \cdot ((\mathbf{A}_i - \mathbf{x}_i \otimes \mathbf{I}_m) \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top) + \mathbf{s}_i \cdot (\mathbf{D}_i \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top) + \mathbf{e}_{\mathbf{d}'_i}
\end{aligned}$$

where $\mathbf{e}_{\mathbf{d}'_i} := \mathbf{e}_2 \cdot \mathbf{X}_i(\mathbf{I}_{m\ell} \otimes \mathbf{r}_1^\top \cdots \otimes \mathbf{r}_{i-1}^\top \otimes \mathbf{r}_{i+1}^\top \otimes \cdots \otimes \mathbf{r}_k^\top)$.

* Now we compute $\mathbf{d}'_f$.

$$\begin{aligned}
\mathbf{d}'_f &= \mathbf{c}_2\mathbf{M}_f(\mathbf{I}_m \otimes \mathbf{r}_1^\top \cdots \otimes \mathbf{r}_{k-1}^\top) \\
&= ((\mathbf{s},\ \mathbf{s}_0,\ \cdots,\ \mathbf{s}_k)\mathbf{B} + \mathbf{e}_2)\mathbf{B}^{-1} \left(\begin{pmatrix} \mathbf{A}_f\mathbf{U} \otimes \mathbf{I}^{\otimes(k-1)} \otimes \mathbf{r}_k^\top \\ \mathbf{0}_{knm^k \times m^k} \\ \mathbf{D}_k \otimes \mathbf{I}^{\otimes(k-1)} \otimes \mathbf{r}_k^\top \end{pmatrix}, \tau_B \right) (\mathbf{I}_m \otimes \mathbf{r}_1^\top \cdots \otimes \mathbf{r}_{k-1}^\top \otimes 1) \\
&= \big(\mathbf{s} \cdot (\mathbf{A}_f\mathbf{U} \otimes \mathbf{I}^{\otimes(k-1)} \otimes \mathbf{r}_k^\top) + \mathbf{s}_k \cdot (\mathbf{D}_k \otimes \mathbf{I}^{\otimes(k-1)} \otimes \mathbf{r}_k^\top)\big) \cdot (\mathbf{I}_m \otimes \mathbf{r}_1^\top \cdots \otimes \mathbf{r}_{k-1}^\top \otimes 1) + \mathbf{e}_{\mathbf{d}'_f} \\
&= \mathbf{s} \cdot (\mathbf{A}_f\mathbf{U} \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top) + \mathbf{s}_k \cdot (\mathbf{D}_k \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top) + \mathbf{e}_{\mathbf{d}'_f}
\end{aligned}$$

where $\mathbf{e}_{\mathbf{d}'_f} := \mathbf{e}_2\mathbf{M}_f(\mathbf{I} \otimes \mathbf{r}_1^\top \cdots \otimes \mathbf{r}_{k-1}^\top)$.

* Next, we compute:

$$\begin{aligned}
(\mathbf{d}''_0,\ \cdots,\ \mathbf{d}''_{k-1},\ \mathbf{d}''_f) &= \mathbf{c}_2\mathbf{N}_f\mathbf{Y}_{k-1}\cdots\mathbf{Y}_1 \\
&= ((\mathbf{s},\ \mathbf{s}_0,\ \cdots,\ \mathbf{s}_k)\mathbf{B} + \mathbf{e}_2)\mathbf{B}^{-1} \left(\begin{pmatrix} \mathbf{0}_{m^{k+1} \times (k+1)nm^{k-1}w} \\ \mathbf{C}_k \otimes \mathbf{r}_k^\top \end{pmatrix}, \tau_B \right) \cdot \mathbf{Y}_{k-1}\cdots\mathbf{Y}_1 \\
&= (\mathbf{s}_0, \cdots, \mathbf{s}_k) \cdot (\mathbf{C}_k \otimes \mathbf{r}_k^\top) \cdot \mathbf{Y}_{k-1}\cdots\mathbf{Y}_1 + (\mathbf{e}_{\mathbf{d}''_0}, \cdots, \mathbf{e}_{\mathbf{d}''_k}) \\
&= (\mathbf{s}_0, \cdots, \mathbf{s}_k) \cdot (\mathbf{C}_k \otimes \mathbf{r}_k^\top) \cdot (\mathbf{C}_k^{-1}\big((\mathbf{C}_{k-1} \otimes \mathbf{r}_{k-1}^\top), \tau_C\big) \otimes 1) \cdot \mathbf{Y}_{k-2}\cdots\mathbf{Y}_1 + (\mathbf{e}_{\mathbf{d}''_0}, \cdots, \mathbf{e}_{\mathbf{d}''_k}) \\
&= (\mathbf{s}_0, \cdots, \mathbf{s}_k) \cdot (\mathbf{C}_k \cdot \mathbf{C}_k^{-1}\big((\mathbf{C}_{k-1} \otimes \mathbf{r}_{k-1}^\top), \tau_C\big) \otimes \mathbf{r}_k^\top) \cdot \mathbf{Y}_{k-2}\cdots\mathbf{Y}_1 + (\mathbf{e}_{\mathbf{d}''_0}, \cdots, \mathbf{e}_{\mathbf{d}''_k}) \\
&= (\mathbf{s}_0, \cdots, \mathbf{s}_k) \cdot (\mathbf{C}_{k-1} \otimes \mathbf{r}_{k-1}^\top \otimes \mathbf{r}_k^\top) \cdot \mathbf{Y}_{k-2}\cdots\mathbf{Y}_1 + (\mathbf{e}_{\mathbf{d}''_0}, \cdots, \mathbf{e}_{\mathbf{d}''_k}) \\
&= \vdots \\
&= (\mathbf{s}_0, \cdots, \mathbf{s}_k) \cdot (\mathbf{C}_1 \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top) + (\mathbf{e}_{\mathbf{d}''_0}, \cdots, \mathbf{e}_{\mathbf{d}''_k}) \\
&= (\mathbf{s}_0 \cdot (\mathbf{D}_0 \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top) + \mathbf{e}_{\mathbf{d}''_0}, \cdots, \mathbf{s}_k \cdot (\mathbf{D}_k \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top) + \mathbf{e}_{\mathbf{d}''_k})
\end{aligned}$$

with $(\mathbf{e}_{\mathbf{d}''_0}, \cdots, \mathbf{e}_{\mathbf{d}''_k}) := \mathbf{e}_2\mathbf{N}_f\mathbf{Y}_{k-1}\cdots\mathbf{Y}_1$.

- * Let $0 \leq i \leq k-1$,

  $$\begin{aligned}\mathbf{d}_i &= \mathbf{d}'_i - \mathbf{d}''_i \\ &= \mathbf{s}((\mathbf{A}_i - \mathbf{x}_i \otimes \mathbf{I}_m) \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top) + \mathbf{s}_i(\mathbf{D}_i \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top) + \mathbf{e}_{\mathbf{d}'_i} - \mathbf{s}_i(\mathbf{D}_i \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top) - \mathbf{e}_{\mathbf{d}''_i} \\ &= \mathbf{s}((\mathbf{A}_i - \mathbf{x}_i \otimes \mathbf{I}_m) \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top) + \mathbf{e}_{\mathbf{d}_i}\end{aligned}$$

  with $\mathbf{e}_{\mathbf{d}_i} := \mathbf{e}_{\mathbf{d}'_i} - \mathbf{e}_{\mathbf{d}''_i}$.

- * Next, $\mathbf{d}_f = \mathbf{d}'_f - \mathbf{d}''_f$. So,

  $$\begin{aligned}\mathbf{d}_f &= \mathbf{s}(\mathbf{A}_f\mathbf{U} \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top) + \mathbf{s}_k(\mathbf{D}_k \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top) + \mathbf{e}_{\mathbf{d}'_f} - \mathbf{s}_k(\mathbf{D}_k \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top) - \mathbf{e}_{\mathbf{d}''_f} \\ &= \mathbf{s}(\mathbf{A}_f\mathbf{U} \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top) + \mathbf{e}_{\mathbf{d}_f}\end{aligned}$$

  with $\mathbf{e}_{\mathbf{d}_f} := \mathbf{e}_{\mathbf{d}'_f} - \mathbf{e}_{\mathbf{d}''_f}$ where $\mathbf{e}_{\mathbf{d}''_f} := \mathbf{e}_{\mathbf{d}''_k}$.

- * And finally, $\mathbf{d} = (\mathbf{d}_0\|\cdots\|\mathbf{d}_{k-1}) \cdot \widehat{\mathbf{H}}_{\mathbf{A},f,\mathbf{x}}\mathbf{U} - \mathbf{d}_f$. First,

  $$\begin{aligned}&(\mathbf{d}_0\|\cdots\|\mathbf{d}_{k-1}) \\ &= (\mathbf{s}((\mathbf{A}_0 - \mathbf{x}_0 \otimes \mathbf{I}) \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top)\|\cdots\|\mathbf{s}((\mathbf{A}_{k-1} - \mathbf{x}_{k-1} \otimes \mathbf{I}) \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top)) + (\mathbf{e}_{\mathbf{d}_0}\|\cdots\|\mathbf{e}_{\mathbf{d}_{k-1}}) \\ &= (\mathbf{s}(((\mathbf{A}_0\|\mathbf{A}_1\|\cdots\|\mathbf{A}_{k-1}) - (\mathbf{x}_0\|\mathbf{x}_1\|\cdots\|\mathbf{x}_{k-1}) \otimes \mathbf{I}) \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top)) + (\mathbf{e}_{\mathbf{d}_0}\|\cdots\|\mathbf{e}_{\mathbf{d}_{k-1}}) \\ &= (\mathbf{s}((\mathbf{A} - \underbrace{(\mathbf{x}_0\|\mathbf{x}_1\|\cdots\|\mathbf{x}_{k-1})}_{=\mathbf{x}} \otimes \mathbf{I}) \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top)) + (\mathbf{e}_{\mathbf{d}_0}\|\cdots\|\mathbf{e}_{\mathbf{d}_{k-1}}).\end{aligned}$$

  From Lemma 2.8, we deduce

  $$(\mathbf{A} - \mathbf{x} \otimes \mathbf{I})\widehat{\mathbf{H}}_{\mathbf{A},f,\mathbf{x}} = \mathbf{A}_f - f(\mathbf{x})\mathbf{I} \mod q.$$

  Hence,

  $$\begin{aligned}&(\mathbf{d}_0\|\cdots\|\mathbf{d}_{k-1})\widehat{\mathbf{H}}_{\mathbf{A},f,\mathbf{x}}\mathbf{U} - \mathbf{d}_f \\ &= \mathbf{s}((\mathbf{A} - \mathbf{x} \otimes \mathbf{I}) \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top)(\widehat{\mathbf{H}}_{\mathbf{A},f,\mathbf{x}}\mathbf{U} \otimes 1 \otimes 1 \cdots \otimes 1) + \mathbf{e}_{\mathbf{d}} - \mathbf{s}(\mathbf{A}_f\mathbf{U} \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top) - \mathbf{e}_{\mathbf{d}_f} \\ &= \mathbf{s}((\mathbf{A} - \mathbf{x} \otimes \mathbf{I})\widehat{\mathbf{H}}_{\mathbf{A},f,\mathbf{x}}\mathbf{U} \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top) - \mathbf{s}(\mathbf{A}_f\mathbf{U} \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top) + \mathbf{e}_{\mathbf{d}} - \mathbf{e}_{\mathbf{d}_f} \\ &= \mathbf{s}((\mathbf{A}_f\mathbf{U} - f(\mathbf{x})\mathbf{U}) \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top) - \mathbf{s}(\mathbf{A}_f\mathbf{U} \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top) + \mathbf{e}_{\mathbf{d}} - \mathbf{e}_{\mathbf{d}_f} \\ &= -\mathbf{s}(f(\mathbf{x})\mathbf{U} \otimes \mathbf{r}_1^\top \otimes \cdots \otimes \mathbf{r}_k^\top) + \mathbf{e}_{\mathbf{d}} - \mathbf{e}_{\mathbf{d}_f} \\ &= \mathbf{e}_{\mathbf{d}} - \mathbf{e}_{\mathbf{d}_f} \text{ if } f(\mathbf{x}) = 0.\end{aligned}$$

  where $\mathbf{e}_{\mathbf{d}} := (\mathbf{e}_{\mathbf{d}_0}\|\cdots\|\mathbf{e}_{\mathbf{d}_{k-1}})\widehat{\mathbf{H}}_{\mathbf{A},f,\mathbf{x}}\mathbf{U}$. Thus, when $\mu = 0$, $\|\mathbf{d}\|_\infty$ is small ($\leq \beta_0$), and hence, the decryption correctly outputs 0.

Error Bound: The error term is bounded as follows. Let β_0 denote the error bound.

$$\begin{aligned}&\|\mathbf{e}_{\mathbf{d}}\|_\infty + \|\mathbf{e}_{\mathbf{d}_f}\|_\infty \\ &= \|(\mathbf{e}_{\mathbf{d}_0}\|\cdots\|\mathbf{e}_{\mathbf{d}_{k-1}})\widehat{\mathbf{H}}_{\mathbf{A},f,\mathbf{x}}\mathbf{U}\|_\infty + \|\mathbf{e}_{\mathbf{d}'_f} - \mathbf{e}_{\mathbf{d}''_f}\|_\infty \\ &= \|\Big((\mathbf{e}_{\mathbf{d}'_0}\|\cdots\|\mathbf{e}_{\mathbf{d}'_{k-1}}) - (\mathbf{e}_{\mathbf{d}''_0}\|\cdots\|\mathbf{e}_{\mathbf{d}''_{k-1}})\Big)\widehat{\mathbf{H}}_{\mathbf{A},f,\mathbf{x}}\mathbf{U}\|_\infty + \|\mathbf{e}_{\mathbf{d}'_f} - \mathbf{e}_{\mathbf{d}''_f}\|_\infty \\ &\leq \Big(\Big(\chi_1(m\gamma)^{O(k)} + w\chi_2\tau_B(m\gamma)^{O(k)} + \chi_2\tau_B(w\tau_C)^{O(k)}m^{O(k^2)}\Big) \cdot m\beta\gamma \\ &+w\chi_2\tau_B(m\gamma)^{O(k)} + \chi_2\tau_B(w\tau_C)^{O(k)}m^{O(k^2)}\Big)\operatorname{poly}(m) \\ &\leq (m\chi_1\chi_2w\gamma\tau_B\tau_C)^{O(k^2)} \\ &\leq \beta_0,\end{aligned}$$

where we used $\|\mathbf{e}_{\mathbf{d}'_0}\|_\infty \leq \chi_1(m\gamma)^{O(k)}$, $\|\mathbf{e}_{\mathbf{d}'_i}\|_\infty, \|\mathbf{e}_{\mathbf{d}'_f}\|_\infty \leq w\chi_2\tau_B(m\gamma)^{O(k)}$, and $\|\mathbf{e}_{\mathbf{d}''_i}\|_\infty, \|\mathbf{e}_{\mathbf{d}''_f}\|_\infty \leq \chi_2\tau_B(w\tau_C)^{O(k)}m^{O(k^2)}$.

Parameters: We set the parameters as follows.

$$\begin{array}{llll}
n = \mathrm{poly}(\lambda, 2^d), & m = O(n \log q), & \tau_B = O(\sqrt{2km^{k+1}\log q}), & \tau_C = O(\sqrt{2km^k \log q}), \\
\beta = (m\gamma)^{O(2^d)}, & \gamma = \lambda^{\omega(1)}, & \chi_1 = (m\gamma)^{2k}, & \chi_3 = \chi_4 = (m\gamma)^{4k}, \\
\chi_6 = (m\gamma)^{6k}, & \chi_7 = m\beta\ell\chi_6\lambda^{\omega(1)}, & \chi_{5,i} = \gamma^{k-i}\cdot\chi_7 \text{ for } i \in [0,k], & \chi_2 = \gamma\chi_5 \\
\beta_0 = (m\chi_1\chi_2 w\gamma\tau_B\tau_C)^{O(k^2)}, & q = \beta_0\lambda^{\omega(1)} & &
\end{array}$$

where we define $\chi_5 := \chi_{5,0}$. We note that in the above, χ_3, χ_4, $\chi_{5,i}$, χ_6, and χ_7 are the parameters that only appear in the security proof.

Security: Here, we prove the following theorem, which asserts the security of our scheme.

Theorem 5.1. *Assuming evasive* LWE *(Assumption 3.1) and* LWE*, our construction for k-input* ABE *for* NC_1 *satisfies very selective security (Definition 2.2).*

Proof. To prove the security, we need to prove the indistinguishability of the two distributions given below. Let Q_i be the number of key queries to $\mathsf{KeyGen}_i(\mathsf{msk}, \cdot)$ oracle for $i \in [k]$. In the following, for simplicity, we let $Q_1 = \cdots = Q_k = Q$. Note that this can be assumed without loss of generality.

Note that compared to Section 4.2 where i and j are the indexes for the keys, in this proof, $i \in [k]$ is the index of the key generator, and we denote $j_1, \cdots, j_k \in [Q]$ the indexes of the keys. In the sequel, for the ease of the reading, we often suppress the subscript and simply write j when differentiating the indexes is not necessary.

Distribution D_0:

$$\left(\begin{array}{ll}\mathsf{mpk}, \quad \mathbf{c}_1 = (\mathbf{s},\ \mathbf{s}_0)\begin{pmatrix}(\mathbf{A}_0 - \mathbf{x}_0 \otimes \mathbf{I}_m) \otimes \mathbf{I}^{\otimes k} \\ \mathbf{D}_0 \otimes \mathbf{I}^{\otimes k}\end{pmatrix} + \mathbf{e}_1, & \mathbf{c}_2 = (\mathbf{s},\ \mathbf{s}_0,\ \cdots,\ \mathbf{s}_k)\mathbf{B} + \mathbf{e}_2, \\ \qquad \{\mathsf{sk}_{i,\mathbf{x}_{i,j}} = (\mathbf{r}_{i,j}, \mathbf{X}_{i,j}, \mathbf{Y}_{i,j})\}_{i\in[k-1], j\in[Q]}, & \{\mathsf{sk}_{k,f_j} = (\mathbf{r}_{k,j}, \mathbf{M}_{f_j}, \mathbf{N}_{f_j})\}_{j\in[Q]}\end{array}\right)$$

Distribution D_1:

$$\left(\begin{array}{lll}\mathsf{mpk}, & \mathbf{c}_1 \leftarrow \mathbb{Z}_q^{\ell m^{k+1}}, & \mathbf{c}_2 \leftarrow \mathbb{Z}_q^{(m^{k+1}+(k+1)nm^k)w}, \\ & \{\mathsf{sk}_{i,\mathbf{x}_{i,j}} = (\mathbf{r}_{i,j}, \mathbf{X}_{i,j}, \mathbf{Y}_{i,j})\}_{i\in[k-1], j\in[Q]}, & \{\mathsf{sk}_{k,f_j} = (\mathbf{r}_{k,j}, \mathbf{M}_{f_j}, \mathbf{N}_{f_j})\}_{j\in[Q]}\end{array}\right),$$

where $\mathbf{x}_0$ is the attribute for public encryption, $\mathbf{x}_{i,j}$ for $i \in [k-1]$ is the j-th key query for slot i, and f_j is the j-th key query to $\mathsf{KeyGen}_k(\mathsf{msk}, \cdot)$, $\mathsf{sk}_{i,\mathbf{x}_{i,j}}$ is the j-th key for slot i and sk_{k,f_j} is the key for function f_j. In particular, we have

$$\begin{aligned}
\mathbf{X}_{i,j} &\leftarrow \mathbf{B}^{-1}\left(\begin{pmatrix} (\mathbf{A}_i - \mathbf{x}_{i,j} \otimes \mathbf{I}_m) \otimes \mathbf{I}^{\otimes(i-1)} \otimes \mathbf{r}_{i,j}^\top \otimes \mathbf{I}^{\otimes(k-i)} \\ \mathbf{0}_{inm^k \times \ell m^k} \\ \mathbf{D}_i \otimes \mathbf{I}^{\otimes(i-1)} \otimes \mathbf{r}_{i,j}^\top \otimes \mathbf{I}^{\otimes(k-i)} \\ \mathbf{0}_{(k-i)nm^k \times \ell m^k} \end{pmatrix}, \tau_B\right) \\
\mathbf{Y}_{i,j} &\leftarrow \mathbf{C}_{i+1}^{-1}\left((\mathbf{C}_i \otimes \mathbf{r}_{i,j}^\top), \tau_C\right) \\
\mathbf{M}_{f_j} &\leftarrow \mathbf{B}^{-1}\left(\begin{pmatrix} \mathbf{A}_{f_j}\mathbf{U}_j \otimes \mathbf{I}^{\otimes(k-1)} \otimes \mathbf{r}_{k,j}^\top \\ \mathbf{0}_{knm^k \times m^k} \\ \mathbf{D}_k \otimes \mathbf{I}^{\otimes(k-1)} \otimes \mathbf{r}_{k,j}^\top \end{pmatrix}, \tau_B\right) \\
\mathbf{N}_{f_j} &\leftarrow \mathbf{B}^{-1}\left(\begin{pmatrix} \mathbf{0}_{m^{k+1} \times (k+1)nm^{k-1}w} \\ \mathbf{C}_k \otimes \mathbf{r}_{k,j}^\top \end{pmatrix}, \tau_B\right)
\end{aligned}$$

$\mathbf{e}_1 \leftarrow \mathcal{D}_{\mathbb{Z},\chi_1}^{\ell m^{k+1}}$, $\mathbf{e}_2 \leftarrow \mathcal{D}_{\mathbb{Z},\chi_2}^{(m^{k+1}+(k+1)nm^k)w}$.

We can see that D_0 and D_1 are the views of the adversary when $\mu = 0$ and $\mu = 1$ are encrypted, respectively. We then apply Evasive LWE (EvLWE) with respect to matrix $\mathbf{B}$ with sampler Samp^1 that outputs $\mathsf{aux}^1 = (\mathsf{aux}_1^1, \mathsf{aux}_2^1), \mathbf{P}^1, \mathbf{S}^1$ as follows:[7]

$$\begin{aligned}
\mathbf{S}^1 &= (\mathbf{s}, \mathbf{s}_0, \ldots, \mathbf{s}_k) \\
\mathsf{aux}_1^1 &= \mathbf{s}((\mathbf{A}_0 - \mathbf{x}_0 \otimes \mathbf{I}_m) \otimes \mathbf{I}^{\otimes k}) + \mathbf{s}_0(\mathbf{D}_0 \otimes \mathbf{I}^{\otimes k}) + \mathbf{e}_1 \\
\mathsf{aux}_2^1 &= (\mathbf{x}_0, \{\mathbf{x}_{i,j}, \mathbf{r}_{i,j}\}_{i\in[k-1],j\in[Q]}, \{\mathbf{Y}_{i,j}\}_{i\in[k-1],j\in[Q]}, \{f_j, \mathbf{r}_{k,j}\}_{j\in[Q]}, \mathbf{A}, \mathbf{C}_1, \ldots, \mathbf{C}_k, \mathbf{U}) \\
\mathbf{P}_{i,j} &= \begin{pmatrix} (\mathbf{A}_i - \mathbf{x}_{i,j} \otimes \mathbf{I}) \otimes \mathbf{I}^{\otimes(i-1)} \otimes \mathbf{r}_{i,j}^\top \otimes \mathbf{I}^{\otimes(k-i)} \\ \mathbf{0}_{inm^k \times \ell m^k} \\ \mathbf{D}_i \otimes \mathbf{I}^{\otimes(i-1)} \otimes \mathbf{r}_{i,j}^\top \otimes \mathbf{I}^{\otimes(k-i)} \\ \mathbf{0}_{(k-i)nm^k \times \ell m^k} \end{pmatrix}, \text{ for } i \in [k-1], j \in [Q] \\
\mathbf{P}_{k,j} &= \begin{pmatrix} \mathbf{A}_{f_j}\mathbf{U} \otimes \mathbf{I}^{\otimes(k-1)} \otimes \mathbf{r}_{k,j}^\top \\ \mathbf{0}_{knm^k \times m^k} \\ \mathbf{D}_k \otimes \mathbf{I}^{\otimes(k-1)} \otimes \mathbf{r}_{k,j}^\top \end{pmatrix}, \text{ for } j \in [Q] \\
\mathbf{P}_{k+1,j} &= \begin{pmatrix} \mathbf{0}_{m^{k+1} \times (k+1)nm^{k-1}w} \\ \mathbf{C}_k \otimes \mathbf{r}_{k,j}^\top \end{pmatrix}, \text{ for } j \in [Q] \\
\mathbf{P}^1 &= (\mathbf{P}_{1,1} \| \cdots \| \mathbf{P}_{1,Q} \| \cdots \| \mathbf{P}_{k-1,1} \| \cdots \| \mathbf{P}_{k-1,Q} \| \mathbf{P}_{k,1} \| \cdots \| \mathbf{P}_{k,Q} \| \mathbf{P}_{k+1,1} \| \cdots \| \mathbf{P}_{k+1,Q})
\end{aligned}$$

Then from Lemma 3.4, to prove that D_0 and D_1 are computationally indistinguishable, it suffices to prove the computational indistinguishability between the following distributions:

Distribution D_0^1:

$$\begin{pmatrix} \mathsf{aux}_2^1, & \mathbf{B}, & \mathbf{c}_1, & \mathbf{c}_2, \\ & & \{\mathbf{c}_{i,j}\}_{i\in[k-1],j\in[Q]}, & \{\mathbf{c}_{k,j}, \mathbf{d}_j\}_{j\in[Q]} \end{pmatrix}$$

Distribution D_1^1:

$$\begin{pmatrix} \mathsf{aux}_2^1, & \mathbf{B}, & \mathbf{v}_1, & \mathbf{v}_2, \\ & & \{\mathbf{v}_{i,j}\}_{i\in[k-1],j\in[Q]}, & \{\mathbf{v}_{k,j}, \mathbf{w}_j\}_{j\in[Q]} \end{pmatrix},$$

[7] By Lemma 3.4, it suffices to invoke the evasive LWE for a modified sampler that outputs random aux_1. The same comments apply to other invocations of the assumption.

where $\mathbf{v}$ (resp., $\mathbf{w}$) vectors above are sampled uniformly at random from the same domain as the corresponding $\mathbf{c}$ (resp., $\mathbf{d}$) vectors and

$$\begin{aligned}
\mathbf{c}_{i,j} &= \mathbf{S}^1\mathbf{P}_{i,j} + \mathbf{e}_{i,j} \\
&= \mathbf{s}((\mathbf{A}_i - \mathbf{x}_{i,j} \otimes \mathbf{I}) \otimes \mathbf{I}^{\otimes(i-1)} \otimes \mathbf{r}_{i,j}^\top \otimes \mathbf{I}^{\otimes(k-i)}) + \mathbf{s}_i(\mathbf{D}_i \otimes \mathbf{I}^{\otimes(i-1)} \otimes \mathbf{r}_{i,j}^\top \otimes \mathbf{I}^{\otimes(k-i)}) + \mathbf{e}_{i,j} \\
\mathbf{c}_{k,j} &= \mathbf{S}^1\mathbf{P}_{k,j} + \mathbf{e}_{k,j} \\
&= \mathbf{s}(\mathbf{A}_{f_j}\mathbf{U}_j \otimes \mathbf{I}^{\otimes(k-1)} \otimes \mathbf{r}_{k,j}^\top) + \mathbf{s}_k(\mathbf{D}_k \otimes \mathbf{I}^{\otimes(k-1)} \otimes \mathbf{r}_{k,j}^\top) + \mathbf{e}_{k,j} \\
\mathbf{d}_j &= \mathbf{S}^1\mathbf{P}_{k+1,j} + \mathbf{e}'_j \\
&= (\mathbf{s}_0, \ldots, \mathbf{s}_k)(\mathbf{C}_k \otimes \mathbf{r}_{k,j}^\top) + \mathbf{e}'_j \\
&= (\mathbf{s}_0, \ldots, \mathbf{s}_k)(\mathbf{I}_{(k+1)nm^{k-1}} \otimes \mathbf{r}_{k,j}^\top)\mathbf{C}_k + \mathbf{e}'_j
\end{aligned}$$

where $\mathbf{e}_{i,j} \leftarrow \mathcal{D}_{\mathbb{Z},\chi_3}^{\ell m^k}$, $\mathbf{e}_{k,j} \leftarrow \mathcal{D}_{\mathbb{Z},\chi_4}^{m^k}$, $\mathbf{e}'_j \leftarrow \mathcal{D}_{\mathbb{Z},\chi_5}^{(k+1)nm^{k-1}w}$.

Note that we set $\chi_2 > \chi_3, \chi_4, \chi_5$ so that we can rely on quantitatively weaker evasive LWE assumption (See Remark 3.3). We also note that here, we have $\chi_3 = \chi_4 \neq \chi_5$, where Gaussian distributions with different standard deviations are mixed in the precondition distribution. We refer to Remark 3.2 for the detail.

To show the indistinguishability between the two distributions D_0^1 and D_1^1, we again apply Evasive LWE, this time with respect to matrix $\mathbf{C}_k$ and a sampler Samp^2 as described below:

$$\begin{aligned}
\mathbf{S}^2 &= \begin{pmatrix} (\mathbf{s}_0, \ldots, \mathbf{s}_k)(\mathbf{I}_{(k+1)nm^{k-1}} \otimes \mathbf{r}_{k,1}^\top) \\ \vdots \\ (\mathbf{s}_0, \ldots, \mathbf{s}_k)(\mathbf{I}_{(k+1)nm^{k-1}} \otimes \mathbf{r}_{k,Q}^\top) \end{pmatrix} \\
\mathsf{aux}_1^2 &= \mathbf{c}_1, \mathbf{c}_2, \{\mathbf{c}_{i,j}\}_{i\in[k-1],j\in[Q]}, \{\mathbf{c}_{k,j}\}_{j\in[Q]} \\
\mathsf{aux}_2^2 &= (\mathbf{x}_0, \{\mathbf{x}_{i,j}, \mathbf{r}_{i,j}\}_{i\in[k-1],j\in[Q]}, \{\mathbf{Y}_{i,j}\}_{i\in[k-2],j\in[Q]}, \{f_j, \mathbf{r}_{k,j}\}_{j\in[Q]}, \mathbf{A}, \mathbf{B}, \mathbf{C}_1, \ldots, \mathbf{C}_{k-1}, \mathbf{U}) \\
\mathbf{P}^2 &= (\mathbf{C}_{k-1} \otimes \mathbf{r}_{k-1,1}^\top \| \cdots \| \mathbf{C}_{k-1} \otimes \mathbf{r}_{k-1,Q}^\top),
\end{aligned}$$

where $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_{i,j}, \mathbf{c}_{k,j}$for $i \in [k-1], j \in [Q]$ are as defined in distribution D_0^1. Then again using Lemma 3.4, to prove that the two distributions are computationally indistinguishable, it suffices to prove the computational indistinguishability between the following two distributions:

Distribution D_0^2:

$$\begin{pmatrix} \mathsf{aux}_2^2, \quad \mathbf{C}_k, & \mathbf{c}_1, & \mathbf{c}_2, \\ \{\mathbf{c}_{i,j}\}_{i\in[k-1],j\in[Q]}, & \{\mathbf{d}_{j_1}, \mathbf{d}_{j_1,j_2}\}_{j_1,j_2\in[Q]}, & \{\mathbf{c}_{k,j}\}_{j\in[Q]} \end{pmatrix}$$

Distribution D_1^2:

$$\begin{pmatrix} \mathsf{aux}_2^2, \quad \mathbf{C}_k, & \mathbf{v}_1, & \mathbf{v}_2, \\ \{\mathbf{v}_{i,j}\}_{i\in[k-1],j\in[Q]}, & \{\mathbf{w}_{j_1}, \mathbf{w}_{j_1,j_2}\}_{j_1,j_2\in[Q]}, & \{\mathbf{v}_{k,j}\}_{j\in[Q]} \end{pmatrix},$$

where

$$(\mathbf{d}_{j_1,j_2})_{j_1,j_2\in[Q]} = \mathbf{S}^2\mathbf{P}^2 + (\mathbf{e}'_{j_1,j_2})_{j_1,j_2\in[Q]}, \qquad \mathbf{e}'_{j_1,j_2} \leftarrow \mathcal{D}_{\mathbb{Z},\chi_{5,2}}^{(k+1)nm^{k-2}w},$$

all the $\mathbf{c}$ vectors and $\{\mathbf{d}_{j_1}\}_{j_1}$ are defined same as previously, and $\mathbf{v}$ (resp., $\mathbf{w}$) vectors are sampled uniformly at random from the same domain as their corresponding $\mathbf{c}$ (resp., $\mathbf{d}$) vectors. In the above, $(\mathbf{a}_{j_1,j_2})_{j_1,j_2\in[Q]}$ denotes a matrix obtained by vertically concatenating vectors $\{\mathbf{a}_{j_1,j_2}\}_{j_1,j_2}$ of the same dimensions for all possible combinations of $j_1, j_2 \in [Q]$. In particular, we have

$$\mathbf{d}_{j_1,j_2} = (\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_k)(\mathbf{I}_{(k+1)nm^{k-1}} \otimes \mathbf{r}_{k,j_1}^\top)(\mathbf{I}_{(k+1)nm^{k-2}} \otimes \mathbf{r}_{k-1,j_2}^\top)\mathbf{C}_{k-1} + \mathbf{e}'_{j_1,j_2}.$$

To show that the two distributions - D_0^2 and D_1^2 - are computationally indistinguishable, we again apply Evasive LWE, now with respect to matrix $\mathbf{C}_{k-1}$. In general, we apply evasive LWE k times, where the sampler Samp^l for $l \in [k]$ for the l-th application of the evasive LWE assumption is defined as follows: Samp^1 is as defined before.
For $l \in [2, k]$, evasive LWE is applied with respect to the matrix $\mathbf{C}_{k-(l-2)}$ and Samp^l outputs the following:

$$
\begin{aligned}
\mathbf{S}^l &= \\
&\begin{pmatrix} \vdots \\ (\mathbf{s}_0, \dots, \mathbf{s}_k)(\mathbf{I}_{(k+1)nm^{k-1}} \otimes \mathbf{r}_{k,j_1}^\top)(\mathbf{I}_{(k+1)nm^{k-2}} \otimes \mathbf{r}_{k-1,j_2}^\top) \cdots (\mathbf{I}_{(k+1)nm^{k-l+1}} \otimes \mathbf{r}_{k-l+2,j_{l-1}}^\top) \\ \vdots \end{pmatrix}_{j_1,\dots,j_{l-1} \in [Q]} \\
\mathsf{aux}_1^l &= \mathbf{c}_1, \mathbf{c}_2, \{\mathbf{c}_{i,j}\}_{i \in [k-1], j \in [Q]}, \{\mathbf{c}_{k,j}\}_{j \in [Q]}, \{\mathbf{d}_{j_1}, \mathbf{d}_{j_1,j_2}, \mathbf{d}_{j_1,j_2,j_3}, \dots, \mathbf{d}_{j_1,\dots,j_{l-2}}\}_{j_1,\dots,j_{l-2} \in [Q]} \\
\mathsf{aux}_2^l &= (\mathbf{x}_0, \{\mathbf{x}_{i,j}, \mathbf{r}_{i,j}\}_{i \in [k-1], j \in [Q]}, \{\mathbf{Y}_{i,j}\}_{i \in [k-l], j \in [Q]}, \{f_j, \mathbf{r}_{k,j}\}_{j \in [Q]}, \mathbf{A}, \mathbf{B}, \{\mathbf{C}_i\}_{i \in [k] \setminus \{k-l+2\}}, \mathbf{U}) \\
\mathbf{P}^l &= (\mathbf{C}_{k-l+1} \otimes \mathbf{r}_{k-l+1,1}^\top \| \cdots \| \mathbf{C}_{k-l+1} \otimes \mathbf{r}_{k-l+1,Q}^\top),
\end{aligned}
$$

where

$$
\begin{aligned}
&\mathbf{d}_{j_1,j_2,\dots,j_t} \\
= \; &(\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_k)(\mathbf{I}_{(k+1)nm^{k-1}} \otimes \mathbf{r}_{k,j_1}^\top) \cdots (\mathbf{I}_{(k+1)nm^{k-t}} \otimes \mathbf{r}_{k-t+1,j_t}^\top)\mathbf{C}_{k-t+1} + \mathbf{e}'_{j_1,\dots,j_t} \\
= \; &(\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_k)(\mathbf{I}_{(k+1)nm^{k-t}} \otimes \mathbf{r}_{k-t+1,j_t}^\top \otimes \mathbf{r}_{k-t+2,j_{t-1}}^\top \otimes \cdots \otimes \mathbf{r}_{k,j_1}^\top)\mathbf{C}_{k-t+1} + \mathbf{e}'_{j_1,\dots,j_t}, \text{for } t \in [k]
\end{aligned}
$$

where $\mathbf{e}'_{j_1,\dots,j_t} \leftarrow \mathcal{D}_{\mathbb{Z},\chi_{5,t}}^{(k+1)nm^{k-t}}$ when $t \le k-1$. When $t = k$, $\mathbf{e}'_{j_1,\dots,j_k}$ is chosen as $\mathbf{e}'_{j_1,\dots,j_k} = (\mathbf{e}'_{0,j_1,\dots,j_k}, \dots, \mathbf{e}'_{k,j_1,\dots,j_k})$, where $\mathbf{e}'_{i,j_1,\dots,j_k} \leftarrow \mathcal{D}_{\mathbb{Z},\chi_6}^{m\ell}$ for $i \in [0, k-1]$ and $\mathbf{e}'_{k,j_1,\dots,j_k} \leftarrow \mathcal{D}_{\mathbb{Z},\chi_7}^{m}$. Similarly to the first application of evasive LWE, we set $\chi_5 > \chi_{5,2} > \cdots > \chi_{5,k-1} > \chi_6, \chi_7$ so that we can rely on quantitatively weaker evasive LWE assumption (See Remark 3.3). We also note that here, we have $\chi_6 \neq \chi_7$ for the final usage of evasive LWE, which means that Gaussian distributions with different standard deviations are mixed in the precondition distribution. We refer to Remark 3.2 for the detail. Thus, after applying EvLWE l times and using Lemma 3.4, it suffices to prove the indistinguishability between the following two distributions.

Distribution D_0^l:

$$
\begin{pmatrix} \mathsf{aux}_2^l, & \mathbf{C}_{k-l+2}, & \mathbf{c}_1, & \mathbf{c}_2, \\ \{\mathbf{c}_{i,j}\}_{i \in [k-1], j \in [Q]}, & \{\mathbf{d}_{j_1}, \mathbf{d}_{j_1,j_2}, \mathbf{d}_{j_1,j_2,j_3}, \dots, \mathbf{d}_{j_1,j_2,\dots,j_l}\}_{j_1,\dots,j_l \in [Q]}, & \{\mathbf{c}_{k,j}\}_{j \in [Q]} \end{pmatrix}
$$

Distribution D_1^l:

$$
\begin{pmatrix} \mathsf{aux}_2^l, & \mathbf{C}_{k-l+2}, & \mathbf{v}_1, & \mathbf{v}_2, \\ \{\mathbf{v}_{i,j}\}_{i \in [k-1], j \in [Q]}, & \{\mathbf{w}_{j_1}, \mathbf{w}_{j_1,j_2}, \mathbf{w}_{j_1,j_2,j_3}, \dots, \mathbf{w}_{j_1,j_2,\dots,j_l}\}_{j_1,\dots,j_l \in [Q]}, & \{\mathbf{v}_{k,j}\}_{j \in [Q]} \end{pmatrix},
$$

where all the $\mathbf{c}$ and the $\mathbf{d}$ vectors are same as defined previously and $\mathbf{v}$ (resp., $\mathbf{w}$) vectors are sampled uniformly at random from the same domain as their corresponding $\mathbf{c}$ (resp., $\mathbf{d}$) vectors.

In particular, we get that after applying EvLWE k times, it suffices to prove the indistinguishability between the following two distributions:

Distribution $D'_0 = D_0^k$:

$$
\begin{pmatrix} \mathsf{aux}'_2, & \mathbf{C}_2, & \mathbf{c}_1, & \mathbf{c}_2, \\ \{\mathbf{c}_{i,j}\}_{i \in [k-1], j \in [Q]}, & \{\mathbf{d}_{j_1}, \mathbf{d}_{j_1,j_2}, \mathbf{d}_{j_1,j_2,j_3}, \dots, \mathbf{d}_{j_1,j_2,\dots,j_k}\}_{j_1,\dots,j_k \in [Q]}, & \{\mathbf{c}_{k,j}\}_{j \in [Q]} \end{pmatrix}
$$

Distribution $D_1' = D_1^k$:

$$\begin{pmatrix} \mathsf{aux}_2', & \mathbf{C}_2, & \mathbf{v}_1, & \mathbf{v}_2, \\ \{\mathbf{v}_{i,j}\}_{i\in[k-1],j\in[Q]}, & \{\mathbf{w}_{j_1}, \mathbf{w}_{j_1,j_2}, \mathbf{w}_{j_1,j_2,j_3}, \dots, \mathbf{w}_{j_1,j_2,\dots,j_l}\}_{j_1,\dots,j_l\in[Q]}, & & \{\mathbf{v}_{k,j}\}_{j\in[Q]} \end{pmatrix},$$

where $\mathsf{aux}_2' = (\mathbf{x}_0, \{\mathbf{x}_{i,j}, \mathbf{r}_{i,j}\}_{i\in[k-1],j\in[Q]}, \{f_j, \mathbf{r}_{k,j}\}_{j\in[Q]}, \mathbf{A}, \mathbf{B}, \{\mathbf{C}_i\}_{i\in[k]\setminus\{2\}}, \mathbf{U})$. All the $\mathbf{c}$, $\mathbf{d}$, $\mathbf{v}$, and $\mathbf{w}$ vectors are same as defined before.

From the discussion above, to complete the proof of Theorem 5.1, it suffices to prove Lemma 5.2 in the following. □

Lemma 5.2. *Distributions D_0' and D_1' are computationally indistinguishable under the hardness assumption of* LWE*.*

Proof. We prove the computational indistinguishability between the two hybrids - D_0' and D_1' via the following hybrids:

G_0 : This is same as the distribution D_0'. For ease of reading and setting up notations, let us list what the adversary can see here. The adversary can see

$$\begin{aligned}
\mathsf{aux} : &= \big(\mathbf{x}_0, \{\mathbf{x}_{i,j}, \mathbf{r}_{i,j}\}_{i\in[k-1],j\in[Q]}, \{f_j, \mathbf{r}_{k,j}\}_{j\in[Q]}, \mathbf{A}, \mathbf{B}, \{\mathbf{C}_i\}_{i\in[k]}, \mathbf{U}\big) \\
\mathbf{c}_1 &= \mathbf{s}((\mathbf{A}_0 - \mathbf{x}_0 \otimes \mathbf{I}) \otimes \mathbf{I}^{\otimes k}) + \mathbf{s}_0(\mathbf{D}_0 \otimes \mathbf{I}^{\otimes k}) + \mathbf{e}_1 \\
\mathbf{c}_2 &= (\mathbf{s},\ \mathbf{s}_0,\ \cdots,\ \mathbf{s}_k)\mathbf{B} + \mathbf{e}_2 \\
\mathbf{c}_{i,j} &= \mathbf{s}((\mathbf{A}_i - \mathbf{x}_{i,j} \otimes \mathbf{I}) \otimes \mathbf{I}^{\otimes(i-1)} \otimes \mathbf{r}_{i,j}^\top \otimes \mathbf{I}^{\otimes(k-i)}) + \mathbf{s}_i(\mathbf{D}_i \otimes \mathbf{I}^{\otimes(i-1)} \otimes \mathbf{r}_{i,j}^\top \otimes \mathbf{I}^{\otimes(k-i)}) + \mathbf{e}_{i,j} \\
&\quad \text{for } i \in [k-1],\ j \in [Q] \\
\mathbf{c}_{k,j} &= \mathbf{s}(\mathbf{A}_{f_j}\mathbf{U} \otimes \mathbf{I}^{\otimes(k-1)} \otimes \mathbf{r}_{k,j}^\top) + \mathbf{s}_k(\mathbf{D}_k \otimes \mathbf{I}^{\otimes(k-1)} \otimes \mathbf{r}_{k,j}^\top) + \mathbf{e}_{k,j} \\
&\quad \text{for } j \in [Q] \\
\mathbf{d}_{j_t,j_{t+1},\dots,j_k} &= (\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_k)(\mathbf{I}_{(k+1)nm^{t-1}} \otimes \mathbf{r}_{t,j_t}^\top \otimes \mathbf{r}_{t+1,j_{t+1}}^\top \otimes \cdots \otimes \mathbf{r}_{k,j_k}^\top)\mathbf{C}_t + \mathbf{e}_{j_t,\dots,j_k}', \\
&\quad \text{for } t \in [k],\ j_t, \dots j_k \in [Q]
\end{aligned}$$

where we have relabeled the subscripts $j_1, j_2, \dots$, for making the notation simpler. Note that this can be done without loss of generality. We then observe that

$$\begin{aligned}
&\mathbf{d}_{j_1,j_2,\dots,j_k} \\
= &(\mathbf{s}_0,\ \mathbf{s}_1,\ \dots,\ \mathbf{s}_k)(\mathbf{I}_{(k+1)n} \otimes \mathbf{r}_{1,j_1}^\top \otimes \cdots \otimes \mathbf{r}_{k,j_k}^\top)\mathbf{C}_1 + \mathbf{e}_{j_1,\dots,j_k}' \\
= &(\mathbf{s}_0,\ \mathbf{s}_1,\ \dots,\ \mathbf{s}_k) \begin{pmatrix} \mathbf{I}_n \otimes \mathbf{r}_{j_1,\dots,j_k}^\top & & \\ & \ddots & \\ & & \mathbf{I}_n \otimes \mathbf{r}_{j_1,\dots,j_k}^\top \end{pmatrix} \begin{pmatrix} \mathbf{D}_0 & & \\ & \ddots & \\ & & \mathbf{D}_k \end{pmatrix} + \mathbf{e}_{j_1,\dots,j_k}' \\
= &\Bigg(\underbrace{\mathbf{s}_0(\mathbf{I}_n \otimes \mathbf{r}_{j_1,\dots,j_k}^\top)\mathbf{D}_0 + \mathbf{e}_{0,j_1,\dots,j_k}'}_{:=\mathbf{p}_{0,j_1,\dots,j_k}}, \dots \underbrace{\mathbf{s}_k(\mathbf{I}_n \otimes \mathbf{r}_{j_1,\dots,j_k}^\top)\mathbf{D}_k + \mathbf{e}_{k,j_1,\dots,j_k}'}_{:=\mathbf{p}_{k,j_1,\dots,j_k}} \Bigg)
\end{aligned}$$

where $\mathbf{r}_{j_1,\dots,j_k}^\top = \mathbf{r}_{1,j_1}^\top \otimes \cdots \otimes \mathbf{r}_{k,j_k}^\top$ and $\mathbf{e}_{j_1,\dots,j_k}' = (\mathbf{e}_{0,j_1,\dots,j_k}', \dots, \mathbf{e}_{k,j_1,\dots,j_k}')$.

G_1 : In this hybrid, $\mathbf{d}_{j_1,j_2,\dots,j_k} = \{\mathbf{p}_{i,j_1,\dots,j_k}\}_{i\in[0,k],j_1,\dots,j_k\in[Q]}$ is computed differently. Namely, for

$j_1, \ldots, j_k \in [Q]$, they are computed as

$$\begin{aligned}
\mathbf{p}_{0,j_1,\ldots,j_k} &= \mathbf{c}_1(\mathbf{I}_{m\ell} \otimes \mathbf{r}^\top_{j_1,\ldots,j_k}) - \underbrace{\Big(\mathbf{s}((\mathbf{A}_0 - \mathbf{x}_0 \otimes \mathbf{I}_m) \otimes \mathbf{r}^\top_{j_1,\ldots,j_k}) + \mathbf{e}'_{0,j_1,\ldots,j_k}\Big)}_{:=\mathbf{c}'_{0,j_1,,\ldots,j_k}} \\
\mathbf{p}_{i,j_1,\ldots,j_k} &= \mathbf{c}_{i,j_i}(\mathbf{I}_{m\ell} \otimes \mathbf{r}^\top_{j_1,\ldots,j_{i-1},j_{i+1},\ldots,j_k}) - \underbrace{\Big(\mathbf{s}((\mathbf{A}_i - \mathbf{x}_{i,j_i} \otimes \mathbf{I}_m) \otimes \mathbf{r}^\top_{j_1,\ldots,j_k}) + \mathbf{e}'_{i,j_1,\ldots,j_k}\Big)}_{:=\mathbf{c}'_{i,j_1,,\ldots,j_k}} \\
&\quad \text{for } i \in [k-1], \\
\mathbf{p}_{k,j_1,\ldots,j_k} &= \mathbf{c}_{k,j_k}(\mathbf{I}_m \otimes \mathbf{r}^\top_{j_1,\ldots,j_{k-1}}) - \underbrace{\Big(\mathbf{s}(\mathbf{A}_{f_{j_k}} \otimes \mathbf{r}^\top_{j_1,\ldots,j_k}) + \mathbf{e}'_{k,j_1,\ldots,j_k}\Big)}_{:=\mathbf{c}'_{k,j_1,,\ldots,j_k}}
\end{aligned}$$

G_2 : In this hybrid, the challenger samples $\mathbf{d}_{j_t,j_{t+1},\ldots,j_k}$ for $t \geq 2$ differently. Namely, for $t \in [2,k]$ and $j_t, \ldots, j_k \in [Q]$, they are computed as

$$\begin{aligned}
&\mathbf{d}_{j_t,j_{t+1},\ldots,j_k} \\
= \;& \underbrace{\Big((\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_k)(\mathbf{C}_t \otimes \mathbf{I}^{\otimes(k-t+1)}) + \mathbf{e}''_t\Big)}_{:=\mathbf{s}'_t}(\mathbf{I}_{(k+1)nm^{t-1}w} \otimes \mathbf{r}^\top_{t,j_t} \otimes \cdots \otimes \mathbf{r}^\top_{k,j_k}) + \mathbf{e}'_{j_t,\ldots,j_k}
\end{aligned}$$

where $\mathbf{e}''_t$ for $t \in [2,k]$ are sampled as $\mathbf{e}''_t \leftarrow D^{(k+1)nm^k w}_{\mathbb{Z},\chi_1}$.

G_3 : In this hybrid, $\mathbf{c}_1$, $\mathbf{c}_2$, and $\mathbf{s}'_t$ for $t \in [2,k]$ are replaced with random vectors sampled as $\mathbf{c}_1 \leftarrow \mathbb{Z}_q^{\ell m^{k+1}}$, $\mathbf{c}_2 \leftarrow \mathbb{Z}_q^{(m^{k+1}+(k+1)nm^k)w}$, and $\mathbf{s}'_t \leftarrow \mathbb{Z}_q^{(k+1)nm^k w}$ for $t \in [2,k]$.

G_4 : In this hybrid, the challenger samples $\mathbf{d}_{j_t,j_{t+1},\ldots,j_k}$ for $t \geq 2$ randomly as $\mathbf{d}_{j_t,j_{t+1},\ldots,j_k} \leftarrow \mathbb{Z}_q^{(k+1)nm^{t-1}w}$.

G_5 : In this hybrid, the challenger samples $\mathbf{c}_{i,j}$ for $i \in [k]$, $j \in [Q]$ randomly. Namely, they are sampled as $\mathbf{c}_{i,j} \leftarrow \mathbb{Z}_q^{\ell m^k}$ for $i \in [k-1]$, $j \in [Q]$ and $\mathbf{c}_{k,j} \leftarrow \mathbb{Z}_q^{m^k}$ for $j \in [Q]$. Note that in this hybrid, all the vectors except for $\{\mathbf{d}_{j_1,j_2,\ldots,j_k}\}_{j_1,\ldots,j_k \in [Q]} = \{\mathbf{p}_{i,j_1,\ldots,j_k}\}_{i \in [0,k], j_1,\ldots,j_k \in [Q]}$ are random.

G_6 : In this hybrid, $\mathbf{c}'_{i,j_1,\ldots,j_k}$ for $i \in [0,k], j_1, \ldots, j_k \in [Q]$ are sampled differently. Namely, they are sampled as

$$\begin{aligned}
\mathbf{c}'_{0,j_1,\ldots,j_k} &= \underbrace{\Big(\mathbf{s}(\mathbf{I} \otimes \mathbf{r}^\top_{j_1,\ldots,j_k}) + \mathbf{e}''_{j_1,\ldots,j_k}\Big)}_{:=\mathbf{s}'_{j_1,\ldots,j_k}}(\mathbf{A}_0 - \mathbf{x}_0 \otimes \mathbf{I}) + \mathbf{e}'_{0,j_1,\ldots,j_k} \\
\mathbf{c}'_{i,j_1,\ldots,j_k} &= \underbrace{\Big(\mathbf{s}(\mathbf{I} \otimes \mathbf{r}^\top_{j_1,\ldots,j_k}) + \mathbf{e}''_{j_1,\ldots,j_k}\Big)}_{=\mathbf{s}'_{j_1,\ldots,j_k}}(\mathbf{A}_i - \mathbf{x}_{i,j_i} \otimes \mathbf{I}) + \mathbf{e}'_{i,j_1,\ldots,j_k} \\
\mathbf{c}'_{k,j_1,\ldots,j_k} &= \underbrace{\Big(\mathbf{s}(\mathbf{I} \otimes \mathbf{r}^\top_{j_1,\ldots,j_k}) + \mathbf{e}''_{j_1,\ldots,j_k}\Big)}_{=\mathbf{s}'_{j_1,\ldots,j_k}}\mathbf{A}_{f_{j_k}} + \mathbf{e}'_{k,j_1,\ldots,j_k}
\end{aligned}$$

for $i \in [k-1]$, $j_1, \ldots, j_k \in [Q]$, where $\mathbf{e}''_{j_1,\ldots,j_k} \leftarrow D^m_{\mathbb{Z},\chi_1}$.

G_7 : In this hybrid, $\mathbf{s}'_{j_1,\ldots,j_k}$ for $j_1, \ldots, j_k \in [Q]$ are replaced with random vectors sampled as $\mathbf{s}'_{j_1,\ldots,j_k} \leftarrow \mathbb{Z}_q^m$.

G_8 : In this hybrid, $\mathbf{c}'_{k,j_1,\ldots,j_k}$ for $j_1,\ldots,j_k \in [Q]$ are computed differently as

$$\mathbf{c}'_{k,j_1,\ldots,j_k} = \mathbf{c}'_{[0,k-1],j_1,\ldots j_k} \widehat{\mathbf{H}}_{\mathbf{A},f_{j_k},\mathbf{x}_{j_1,\ldots j_{k-1}}} \mathbf{U} + \left(\mathbf{s}'_{j_1,\ldots,j_k}\mathbf{U} + \mathbf{e}'_{k,j_1,\ldots,j_k}\right).$$

where $\mathbf{c}'_{[0,k-1],j_1,\ldots j_k} := (\mathbf{c}'_{0,j_1,\ldots j_k}|\cdots|\mathbf{c}'_{k-1,j_1,\ldots j_k})$ and $\mathbf{x}_{j_1,\ldots j_{k-1}} = (\mathbf{x}_0|\mathbf{x}_{1,j_1}|\cdots|\mathbf{x}_{k-1,j_{k-1}})$

G_9 : In this hybrid, $\mathbf{c}'_{i,j_1,\ldots,j_k}$ for $i \in [0,k], j_1,\ldots,j_k \in [Q]$ are sampled randomly. Namely, for $j_1,\ldots,j_k \in [Q]$, we have $\mathbf{c}'_{i,j_1,\ldots,j_k} \leftarrow \mathbb{Z}_q^{m\ell}$ for $i \in [0,k-1]$ and $\mathbf{c}'_{k,j_1,\ldots,j_k} \leftarrow \mathbb{Z}_q^m$.

It is easy to see that the distribution in G_9 is the same as that of D'_1.

Indistinguishability of hybrids:

We prove the indistinguishability between the hybrid distributions via the following claims.

Claim 5.3. $\mathsf{G}_0 \approx_s \mathsf{G}_1$

Proof. The two hybrids differ only in the error terms in $\{\mathbf{p}_{i,j_1,\ldots,j_k}\}_{i\in[0,k]}$ and are indistinguishable due to the smudging lemma 2.10. We show this for the case of $i \in [k-1]$ here. The case of $i = 0$ and $i = k$ can be shown similarly.

In G_0:

$$\mathbf{p}_{i,j_1,\ldots,j_k} = \mathbf{s}_i(\mathbf{I}_n \otimes \mathbf{r}^\top_{j_1,\ldots,j_k})\mathbf{D}_i + \mathbf{e}'_{i,j_1,\ldots,j_k}$$

In G_1:

$$\begin{aligned}
\mathbf{p}_{i,j_1,\ldots,j_k} &= \mathbf{c}_{i,j_i}(\mathbf{I}_{m\ell} \otimes \mathbf{r}^\top_{j_1,\ldots,j_{i-1},j_{i+1},\ldots,j_k}) - \left(\mathbf{s}((\mathbf{A}_i - \mathbf{x}_{i,j_i} \otimes \mathbf{I}_m) \otimes \mathbf{r}^\top_{j_1,\ldots,j_k}) + \mathbf{e}'_{i,j_1,\ldots,j_k}\right) \\
&= \mathbf{s}((\mathbf{A}_i - \mathbf{x}_{i,j_i} \otimes \mathbf{I}_m) \otimes \mathbf{r}^\top_{j_1,\ldots,j_k}) + \mathbf{s}_i(\mathbf{D}_i \otimes \mathbf{r}^\top_{j_1,\ldots,j_k}) + \mathbf{e}_{i,j}(\mathbf{I}_{m\ell} \otimes \mathbf{r}^\top_{j_1,\ldots,j_{i-1},j_{i+1},\ldots,j_k}) \\
&\quad - \left(\mathbf{s}((\mathbf{A}_i - \mathbf{x}_{i,j_i} \otimes \mathbf{I}_m) \otimes \mathbf{r}^\top_{j_1,\ldots,j_k}) + \mathbf{e}'_{i,j_1,\ldots,j_k}\right) \\
&= \mathbf{s}_i(\mathbf{D}_i \otimes \mathbf{r}^\top_{j_1,\ldots,j_k}) + \underbrace{\mathbf{e}_{i,j}(\mathbf{I}_{m\ell} \otimes \mathbf{r}^\top_{j_1,\ldots,j_{i-1},j_{i+1},\ldots,j_k}) - \mathbf{e}'_{i,j_1,\ldots,j_k}}_{:=\text{error}}
\end{aligned}$$

Clearly, the two hybrids differ only in the error terms. Thus, the indistinguishability follows due to the following:

$$\mathbf{e}'_{i,j_1,\ldots,j_k} \approx_s -\mathbf{e}'_{i,j_1,\ldots,j_k} + \mathbf{e}_{i,j}(\mathbf{I}_{m\ell} \otimes \mathbf{r}^\top_{j_1,\ldots,j_{i-1},j_{i+1},\ldots,j_k}).$$

The above is true since the distribution of $-\mathbf{e}_{i,j_1,\ldots,j_k}$ is the same as that of $\mathbf{e}_{i,j_1,\ldots,j_k}$ by the symmetry of the discrete Gaussian distribution and by the sumdging lemma, which is applicable since $\chi_6 \geq (m\gamma)^k\lambda^{\omega(1)}\chi_3$ and we have $\|\mathbf{e}_{i,j}(\mathbf{I}_{m\ell} \otimes \mathbf{r}^\top_{j_1,\ldots,j_{i-1},j_{i+1},\ldots,j_k})\|_\infty \leq (m\gamma\,\mathrm{poly}(\lambda))^k\chi_3$. The case of $i = k$ is handled similarly, by using $\chi_7 \geq (m\gamma)^k\lambda^{\omega(1)}\chi_4$. □

Claim 5.4. $\mathsf{G}_1 \approx_s \mathsf{G}_2$

Proof. The two hybrids differ only in the error term in $\{\mathbf{d}_{j_t,j_{t+1},\ldots,j_k}\}_{t\geq 2,j_t,\ldots,j_k\in[Q]}$ and are indistinguishable due to the smudging lemma. In G_1:

$$\mathbf{d}_{j_t,j_{t+1},\ldots,j_k} = (\mathbf{s}_0,\mathbf{s}_1,\ldots,\mathbf{s}_k)(\mathbf{I}_{(k+1)nm^{t-1}} \otimes \mathbf{r}^\top_{t,j_t} \otimes \cdots \otimes \mathbf{r}^\top_{k,j_k})\mathbf{C}_t + \mathbf{e}'_{j_t,\ldots,j_k}$$

In G_2:

$$\begin{aligned}
&\mathbf{d}_{j_t,j_{t+1},\ldots,j_k} \\
= \;& \left((\mathbf{s}_0,\mathbf{s}_1,\ldots,\mathbf{s}_k)(\mathbf{C}_t \otimes \mathbf{I}^{\otimes(k-t+1)}) + \mathbf{e}''_t\right)(\mathbf{I}_{(k+1)nm^{t-1}w} \otimes \mathbf{r}^\top_{t,j_t} \otimes \cdots \otimes \mathbf{r}^\top_{k,j_k}) + \mathbf{e}'_{j_t,\ldots,j_k} \\
= \;& (\mathbf{s}_0,\mathbf{s}_1,\ldots,\mathbf{s}_k)(\mathbf{I}_{(k+1)nm^{t-1}} \otimes \mathbf{r}^\top_{t,j_t} \otimes \cdots \otimes \mathbf{r}^\top_{k,j_k})\mathbf{C}_t \\
& \qquad + \underbrace{\mathbf{e}'_{j_t,\ldots,j_k} + \mathbf{e}''_t(\mathbf{I}_{(k+1)nm^{t-1}w} \otimes \mathbf{r}^\top_{t,j_t} \otimes \cdots \otimes \mathbf{r}^\top_{k,j_k})}_{=\text{error}}
\end{aligned}$$

Clearly, the two hybrids differ only in the error terms. Thus, the indistinguishability follows due to the following:

$$\mathbf{e}'_{j_t,\ldots,j_k} \approx_s \mathbf{e}'_{j_t,\ldots,j_k} + \mathbf{e}''_t(\mathbf{I}_{(k+1)nm^{t-1}w} \otimes \mathbf{r}^\top_{t,j_t} \otimes \cdots \otimes \mathbf{r}^\top_{k,j_k}).$$

The above is true by the smudging lemma, since we have $\chi_{5,t} \geq (m\gamma)^k \chi_1 \cdot \lambda^{\omega(1)}$ for $t \geq 2$ and $\|\mathbf{e}''_t(\mathbf{I}_{(k+1)nm^{t-1}w} \otimes \mathbf{r}^\top_{t,j_t} \otimes \cdots \otimes \mathbf{r}^\top_{k,j_k}\|_\infty \leq (m\gamma\, \mathrm{poly}(\lambda))^k \chi_1$. □

Claim 5.5. $\mathsf{G}_2 \approx_c \mathsf{G}_3$ *due to* LWE.

Proof. Let us write $\mathbf{B}$ as $(\mathbf{B}_U^\top | \mathbf{B}_M^\top | \mathbf{B}_L^\top)^\top$ so that

$$\mathbf{c}_2 = (\mathbf{s}, \mathbf{s}_0, \cdots, \mathbf{s}_k)\mathbf{B} + \mathbf{e}_2 = \mathbf{s}\mathbf{B}_U + (\mathbf{s}_1, \ldots, \mathbf{s}_k)\mathbf{B}_L + (\mathbf{s}_0\mathbf{B}_M + \mathbf{e}_2).$$

We also write $\mathbf{C}_t$ as $\mathbf{C}_t = (\mathbf{C}_{t,U}^\top | \mathbf{C}_{t,L}^\top)^\top$ so that

$$\begin{aligned}
\mathbf{s}'_t &= (\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_k)(\mathbf{C}_t \otimes \mathbf{I}^{\otimes k-t+1}) + \mathbf{e}''_t \\
&= (\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_k) \begin{pmatrix} \mathbf{C}_{t,U} \otimes \mathbf{I}^{\otimes k-t+1} \\ \mathbf{C}_{t,L} \otimes \mathbf{I}^{\otimes k-t+1} \end{pmatrix} + \mathbf{e}''_t \\
&= (\mathbf{s}_1, \ldots, \mathbf{s}_k)(\mathbf{C}_{t,L} \otimes \mathbf{I}^{\otimes k-t+1}) + \left(\mathbf{s}_0(\mathbf{C}_{t,U} \otimes \mathbf{I}^{\otimes k-t+1}) + \mathbf{e}''_t\right)
\end{aligned}$$

By Lemma 3.10, we have that $\mathbf{s}_0(\mathbf{D} \otimes \mathbf{I}^{\otimes k}) + \mathbf{e}_1$, $\mathbf{s}_0\mathbf{B}_M + \mathbf{e}_2$, and $\{\mathbf{s}_0(\mathbf{C}_{t,U} \otimes \mathbf{I}^{\otimes k-t+1}) + \mathbf{e}''_t\}_{t\in[2,k]}$ are indistinguishable from random vectors. The claim follows since these terms mask $\mathbf{c}_1$, $\mathbf{c}_2$, and $\{\mathbf{s}'_t\}_{t\in[2,k]}$, respectively. □

Claim 5.6. $\mathsf{G}_3 \approx_c \mathsf{G}_4$ *due to* LWE.

Proof. In G_3, $\mathbf{d}_{j_t,j_{t+1},\ldots,j_k}$ is chosen as $\mathbf{s}'_t(\mathbf{I}_{(k+1)nm^{t-1}w} \otimes \mathbf{r}^\top_{t,j_t} \otimes \cdots \otimes \mathbf{r}^\top_{k,j_k}) + \mathbf{e}'_{j_t,\ldots,j_k}$ where $\mathbf{s}'_t$ is chosen uniformly at random for all t. The indistinguishability follows by applying Lemma 3.9 for each $t \in [2,k]$, which is possible since we set $\chi_{5,t} \geq (m\gamma \cdot \lambda^{\omega(1)})^k$. □

Claim 5.7. $\mathsf{G}_4 \approx_c \mathsf{G}_5$ *due to* LWE.

Proof. We observe that $\mathbf{c}_{i,j}$ is masked by $\mathbf{v}_{i,j} := \mathbf{s}_i(\mathbf{D}_i \otimes \mathbf{I}^{\otimes(i-1)} \otimes \mathbf{r}^\top_{i,j} \otimes \mathbf{I}^{\otimes(k-i)}) + \mathbf{e}_{i,j}$ for $i \in [k]$, $j \in [Q]$. We show that $\{\mathbf{v}_{i,j}\}_{j\in[Q]}$ is pseudorandom for the case of $i = k$. Other cases can be shown similarly. To show the indistinguishability, we first change the distribution of $\{\mathbf{v}_{k,j}\}_j$ so that they are sampled as

$$\mathbf{v}_{k,j} = \underbrace{\left(\mathbf{s}_k(\mathbf{D}_k \otimes \mathbf{I}^{\otimes k}) + \mathbf{e}''_k\right)}_{:=\mathbf{s}'_k} \left(\mathbf{I}^{\otimes k} \otimes \mathbf{r}_{k,j}\right) + \mathbf{e}_{k,j}.$$

where $\mathbf{e}''_k \leftarrow \mathcal{D}^{m+1}_{\mathbb{Z},\chi_1}$. We claim that this is statistically indistinguishable from the original distribution. To see this, we observe that

$$\mathbf{v}_{k,j} = \mathbf{s}_k(\mathbf{D}_k \otimes \mathbf{r}_{k,j}) + \underbrace{\mathbf{e}''_k\left(\mathbf{I}^{\otimes k} \otimes \mathbf{r}_{k,j}\right) + \mathbf{e}_{k,j}}_{=\text{error}}$$

and these distributions only differ in the error terms. We have

$$\mathbf{e}_{k,j} \approx_s \mathbf{e}''_k\left(\mathbf{I}^{\otimes k} \otimes \mathbf{r}_{k,j}\right) + \mathbf{e}_{k,j}$$

by the smudging lemma, since we have $\chi_3 \geq (m\gamma)^k \lambda^{\omega(1)}\chi_1$ and $\|\mathbf{e}''_k\left(\mathbf{I}^{\otimes k} \otimes \mathbf{r}_{k,j}\right)\|_\infty \leq (m\gamma\,\mathrm{poly}(\lambda))^k\chi_1$. The case of $i \neq k$ is shown similarly, using $\chi_4 \geq (m\gamma)^k\lambda^{\omega(1)}\chi_1$. We then observe that we can replace $\mathbf{s}'_k$ with a random vector by applying LWE with secret $\mathbf{s}_k$. We then apply LWE once again, now the variant with short public matrix and with the secret $\mathbf{s}'_k$, we can conclude that $\{\mathbf{v}_{k,j}\}_{k,j}$ are indistinguishable from random vectors. □

Claim 5.8. $\mathsf{G}_5 \approx_s \mathsf{G}_6$

Proof. The two hybrids differ only in the error terms in $\{\mathbf{c}'_{i,j_1,\dots,j_k}\}_{i\in[0,k],j_1,\dots,j_k\in[Q]}$ and are indistinguishable due to the smudging lemma. We first show this for the case of $i \in [k-1]$. In G_5:

$$\mathbf{c}'_{i,j_1,,\dots,j_k} = \mathbf{s}((\mathbf{A}_i - \mathbf{x}_{i,j_i} \otimes \mathbf{I}) \otimes \mathbf{r}^\top_{j_1,\dots,j_k}) + \mathbf{e}'_{i,j_1,\dots,j_k}$$

In G_6:

$$\begin{aligned}\mathbf{c}'_{i,j_1,,\dots,j_k} &= \left(\mathbf{s}(\mathbf{I}_m \otimes \mathbf{r}^\top_{j_1,\dots,j_k}) + \mathbf{e}''_{j_1,\dots,j_k}\right)(\mathbf{A}_i - \mathbf{x}_{i,j_i} \otimes \mathbf{I}) + \mathbf{e}'_{i,j_1,\dots,j_k}\\ &= \mathbf{s}((\mathbf{A}_i - \mathbf{x}_{i,j_i} \otimes \mathbf{I}) \otimes \mathbf{r}^\top_{j_1,\dots,j_k}) + \underbrace{\mathbf{e}''_{j_1,\dots,j_k}(\mathbf{A}_i - \mathbf{x}_{i,j_i} \otimes \mathbf{I}) + \mathbf{e}'_{i,j_1,\dots,j_k}}_{=\text{error}}\end{aligned}$$

Clearly, the two hybrids differ only in the error terms. Thus, the indistinguishability follows due to the following:

$$\mathbf{e}'_{i,j_1,\dots,j_k} \approx_s \mathbf{e}'_{i,j_1,\dots,j_k} + \mathbf{e}''_{j_1,\dots,j_k}(\mathbf{A}_i - \mathbf{x}_{i,j_i} \otimes \mathbf{I})$$

The above is true by the smudging lemma, since we have $\chi_6 \geq m\gamma\chi_1\lambda^{\omega(1)}$ and $\|\mathbf{e}''_{j_1,\dots,j_k}(\mathbf{A}_i - \mathbf{x}_{i,j_i} \otimes \mathbf{I})\|_\infty \leq m\gamma\,\mathrm{poly}(\lambda)$. The case of $i = 0$ is shown in the same manner. The case of $i = k$ is shown similarly, noting that

$$\mathbf{e}'_{k,j_1,\dots,j_k} \approx_s \mathbf{e}'_{k,j_1,\dots,j_k} + \mathbf{e}''_{j_1,\dots,j_k}\mathbf{A}_{f_{j_k}}$$

holds since we have $\chi_7 \geq m\beta\chi_1 \cdot \lambda^{\omega(1)}$ and $\|\mathbf{e}''_{j_1,\dots,j_k}\mathbf{A}_{f_{j_k}}\|_\infty \leq m\chi_1\|\mathbf{A}_{f_{j_k}}\|_\infty \cdot \mathrm{poly}(\lambda) \leq m\beta\chi_1 \cdot \mathrm{poly}(\lambda)$. □

Claim 5.9. $\mathsf{G}_6 \approx_c \mathsf{G}_7$

Proof. The indistinguishability follows from LWE by Lemma 3.9, which is applicable since $\chi_1 \geq (m\gamma)^k\lambda^{\omega(1)}$. □

Claim 5.10. $\mathsf{G}_7 \approx_s \mathsf{G}_8$

Proof. The two hybrids differ only in the error terms in $\mathbf{c}'_{k,j_1,\dots,j_k}$. The indistinguishability follows from the smudging lemma. In G_7,

$$\mathbf{c}'_{k,j_1,\dots,j_k} = \mathbf{s}'_{j_1,\dots,j_k}\mathbf{A}_{f_{j_k}}\mathbf{U} + \mathbf{e}'_{k,j_1,\dots,j_k}$$

In G_8,

$$\begin{aligned}\mathbf{c}'_{k,j_1,\dots,j_k} &= \mathbf{c}'_{[0,k-1],j_1,\dots j_k}\widehat{\mathbf{H}}_{\mathbf{A},f_{j_k},\mathbf{x}_{j_1,\dots j_{k-1}}}\mathbf{U} + \left(\mathbf{s}'_{j_1,\dots,j_k}\mathbf{U} + \mathbf{e}'_{k,j_1,\dots,j_k}\right)\\ &= \left(\mathbf{s}'_{j_1,\dots,j_k}(\mathbf{A} - \mathbf{x}_{j_1,\dots,j_{k-1}} \otimes \mathbf{I}) + \mathbf{e}'_{[0,k-1],j_1,\dots,j_k}\right)\widehat{\mathbf{H}}_{\mathbf{A},f_{j_k},\mathbf{x}_{j_1,\dots j_{k-1}}}\mathbf{U} + \left(\mathbf{s}'_{j_1,\dots,j_k}\mathbf{U} + \mathbf{e}'_{k,j_1,\dots,j_k}\right)\\ &= \mathbf{s}'_{j_1,\dots,j_k}(\mathbf{A}_{f_{j_k}} - f_{j_k}(\mathbf{x}_{j_1,\dots,j_{k-1}}) \cdot \mathbf{I})\mathbf{U} + \mathbf{s}'_{j_1,\dots,j_k}\mathbf{U} + \mathbf{e}'_{k,j_1,\dots,j_k} + \mathbf{e}'_{[0,k-1],j_1,\dots,j_k}\widehat{\mathbf{H}}_{\mathbf{A},f_{j_k},\mathbf{x}_{j_1,\dots j_{k-1}}}\mathbf{U}\\ &= \mathbf{s}'_{j_1,\dots,j_k}\mathbf{A}_{f_{j_k}}\mathbf{U} + \underbrace{\mathbf{e}'_{k,j_1,\dots,j_k} + \mathbf{e}'_{[0,k-1],j_1,\dots,j_k}\widehat{\mathbf{H}}_{\mathbf{A},f_{j_k},\mathbf{x}_{j_1,\dots j_{k-1}}}\mathbf{U}}_{=\text{error}},\end{aligned}$$

where we define $\mathbf{e}'_{[0,k-1],j_1,\ldots,j_k} = (\mathbf{e}'_{0,j_1,\ldots,j_k},\ldots,\mathbf{e}'_{k-1,j_1,\ldots,j_k})$ in the second line and we use $f_{j_k}(\mathbf{x}_{j_1,\ldots,j_{k-1}}) = 1$ in the last line. Clearly, the two hybrids differ only in the error terms. Thus, the indistinguishability follows due to the following:

$$\mathbf{e}'_{k,j_1,\ldots,j_k} + \mathbf{e}'_{[0,k-1],j_1,\ldots,j_k}\widehat{\mathbf{H}}_{\mathbf{A},f_{j_k},\mathbf{x}_{j_1,\ldots j_{k-1}}} \approx_s \mathbf{e}'_{k,j_1,\ldots,j_k}$$

which is true when $\chi_7 \geq m\beta\ell\chi_6 \cdot \lambda^{\omega(1)}$, since we have $\|\mathbf{e}'_{[0,k-1],j_1,\ldots,j_k}\widehat{\mathbf{H}}_{\mathbf{A},f_{j_k},\mathbf{x}_{j_1,\ldots j_{k-1}}}\|_\infty \leq m\beta\ell\chi_6\,\mathrm{poly}(\lambda)$, where $\|\widehat{\mathbf{H}}_{\mathbf{A},f_{j_k},\mathbf{x}_{j_1,\ldots j_{k-1}}}\|_\infty \leq \beta$. □

Claim 5.11. $\mathsf{G}_8 \approx_c \mathsf{G}_9$

Proof. The indistinguishability between the two hybrids follows from the fact that the following distribution is indistinguishable from random:

$$\mathbf{A},\mathbf{U},\left\{\mathbf{r}_{i,j_i},\mathbf{s}'_{j_1,\ldots,j_k}(\mathbf{A}-\mathbf{x}_{j_1,\ldots,j_k}\otimes\mathbf{I})+\mathbf{e}'_{[0,k-1],j_1,\ldots,j_k},\mathbf{s}'_{j_1,\ldots,j_k}\mathbf{U}+\mathbf{e}'_{k,j_1,\ldots,j_k}\right\}_{i\in[k],j_1,\ldots,j_k\in[Q]}$$

This can be shown by LWE with short public matrix as follows. Here, we change the LWE sample with respect to matrix $(\mathbf{A}-\mathbf{x}_{j_1,\ldots,j_k}\otimes\mathbf{I}|\mathbf{U})$ into random vectors for each combination of $(j_1,\ldots,j_k)$ one by one. To do so, we first use the smudging lemma to see that the distribution of $\mathbf{A}$ and $\mathbf{A}-\mathbf{x}_{j_1,\ldots,j_k}\otimes\mathbf{I}$ are statistically indistinguishable, since each entry of $\mathbf{x}_{j_1,\ldots,j_k}\otimes\mathbf{I}$ is either 0 or 1, while that of $\mathbf{A}$ is chosen from $\mathcal{D}_{\mathbb{Z},\gamma}$ with $\gamma = \lambda^{\omega(1)}$. We then apply the LWE with short public matrix to see that the LWE samples with respect to the secret $\mathbf{s}_{j_1,\ldots,j_k}$ are indistinguishable from the random vectors. □

This completes the proof of Lemma 5.2. □

5.3 A Construction for P

Here, we discuss the variant of our scheme that can deal with circuits with arbitrary (bounded) polynomial depth. Because the construction is very similar to our construction for NC_1 circuits, we only highlight the difference here.

- We sample the matrices $\mathbf{A}_0,\ldots,\mathbf{A}_{k-1}$ uniformly at random from $\mathbb{Z}_q^{n\times m\ell}$ rather than a Gaussian distribution over $\mathbb{Z}^{m\times m\ell}$.
- We replace $\mathbf{A}_0-\mathbf{x}_0\otimes\mathbf{I}$ in the encryption algorithm with $\mathbf{A}_0-\mathbf{x}_0\otimes\mathbf{G}$. Similarly, we also replace $\mathbf{A}_i-\mathbf{x}_i\otimes\mathbf{I}$ in $\mathbf{X}_i$ with $\mathbf{A}_i-\mathbf{x}_i\otimes\mathbf{G}$. Accordingly, $\mathbf{s}$ is chosen randomly from $\mathbb{Z}_q^{nm^{k-1}}$ rather than $\mathbb{Z}_q^{m^k}$.
- The low-norm variant of the lattice evaluation algorithms $(\mathsf{EvalF},\mathsf{EvalFX})$ from Lemma 2.8 used in KeyGen_k and Dec are replaced with those of Lemma 2.6 (i.e., the regular one).
- We use the same parameters as Sec. 5.2 except that β is set to be $(2m)^{O(d)}$ reflecting the fact that we replace the homomorphic lattice evaluation algorithm.

The correctness of the scheme can be shown similarly to Sec. 5.2. The scheme can be proven secure assuming the strengthening of the tensor LWE assumption defined below.

Assumption 5.12 (Extended Tensor LWE). Let $n,m,q,\ell,Q\in\mathbb{N}$ be parameters, χ and γ be Gaussian distributions, and k be a constant. For all $\mathbf{x}_{j_1,\ldots,j_k}\in\{0,1\}^\ell$ indexed by $j_1,\ldots,j_k\in[Q]$, we have

$$\begin{aligned}&\mathbf{A},\left\{\mathbf{s}(\mathbf{I}_n\otimes\mathbf{r}^\top_{1,j_1}\otimes\cdots\otimes\mathbf{r}^\top_{k,j_k})(\mathbf{A}-\mathbf{x}_{j_1,\ldots,j_k}\otimes\mathbf{G})+\mathbf{e}_{j_1,\ldots,j_k},\mathbf{r}_{i,j_i}\right\}_{i\in[k],j_1,\ldots,j_k\in[Q]}\\ \approx_c\quad&\mathbf{A},\{\mathbf{c}_{j_1,\ldots,j_k},\mathbf{r}_{i,j_i}\}_{i\in[k],j_1,\ldots,j_k\in[Q]}\end{aligned}$$

where $\mathbf{A} \leftarrow \mathbb{Z}_q^{m\times \ell m}, \mathbf{s} \leftarrow \mathbb{Z}_q^{m^k n}, \mathbf{e}_{j_1,\ldots,j_k} \leftarrow \mathcal{D}_{\mathbb{Z},\chi}^{\ell m}, \mathbf{r}_{i,j} \leftarrow \mathcal{D}_{\mathbb{Z},\gamma}^{m}, \mathbf{c}_{i,j_i} \leftarrow \mathbb{Z}_q^{\ell m}$ for $i \in [k], j_1, \ldots, j_k \in [Q]$.

Theorem 5.13. *Assuming evasive* LWE *(Assumption 3.1) and extended tensor* LWE *(Assumption 5.12) our k input* miABE *for* P *satisfies very selective security (Definition 2.2).*

Since the proof is very similar to that of Theorem 5.1 in Sec. 5.2, we only provide an overview while highlighting the difference. The first step of the proof is the same as that of Theorem 5.1, where we invoke the evasive LWE assumption k times to conclude that in order to prove the security of the scheme, it suffices to show the indistinguishability of the two distributions D_0' and D_1'. These distributions are defined similarly, except that $\mathbf{A}_0 - \mathbf{x}_0 \otimes \mathbf{I}$ and $\mathbf{A}_i - \mathbf{x}_{i,j_i} \otimes \mathbf{I}$ are replaced with $\mathbf{A}_0 - \mathbf{x}_0 \otimes \mathbf{G}$ and $\mathbf{A}_i - \mathbf{x}_{i,j_i} \otimes \mathbf{G}$. Then, the indistinguishability is shown by similar sequence of hybrids with the following difference.

- We skip G_6 and G_7 and directly argue that $\mathsf{G}_5 \approx_c \mathsf{G}_8$, where $\mathbf{c}'_{k,j_1,\ldots,j_k}$ is replaced with
$$\mathbf{c}'_{k,j_1,\ldots,j_k} = \mathbf{c}'_{[0,k-1],j_1,\ldots j_k} \widehat{\mathbf{H}}_{\mathbf{A},f_{j_k},\mathbf{x}_{j_1,\ldots j_{k-1}}} \mathbf{U} + \left(\mathbf{s}(\mathbf{I}_n \otimes \mathbf{r}_{j_1,\ldots,j_k})\mathbf{G}\mathbf{U} + \mathbf{e}'_{k,j_1,\ldots,j_k}\right).$$
in G_8.
- $\mathsf{G}_8 \approx_c \mathsf{G}_9$ is shown directly from the extension of the evasive LWE assumption above.

□

Acknowledgements This collaboration was initiated when Mélissa Rossi was visiting IIT Madras. Such a visit has been made possible thanks to the funding of the cybersecurity center of excellence of IIT Madras and of a L'Oreal-UNESCO for Women in Science Young Talent France 2019 grant. This work was also supported in part by the DST "Swarnajayanti" fellowship, Cybersecurity Center of Excellence, IIT Madras, National Blockchain Project and the Algorand Centres of Excellence programme managed by Algorand Foundation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of sponsors. Shota Yamada was partially supported by JST AIP Acceleration Research JPMJCR22U5, JST CREST Grant Number JPMJCR22M1, and JSPS KAKENHI Grant Number 19H01109.

References

[ABB10a] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT*, 2010.

[ABB10b] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In *CRYPTO*, 2010.

[ABG19] Michel Abdalla, Fabrice Benhamouda, and Romain Gay. From single-input to multi-client inner-product functional encryption. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 552–582. Springer, Heidelberg, December 2019.

[ABKW19] Michel Abdalla, Fabrice Benhamouda, Markulf Kohlweiss, and Hendrik Waldner. Decentralizing inner-product functional encryption. In Dongdai Lin and Kazue Sako, editors, *PKC 2019, Part II*, volume 11443 of *LNCS*, pages 128–157. Springer, Heidelberg, April 2019.

[ACF$^+$18] Michel Abdalla, Dario Catalano, Dario Fiore, Romain Gay, and Bogdan Ursu. Multi-input functional encryption for inner products: Function-hiding realizations and constructions without pairings. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part I*, volume 10991 of *LNCS*, pages 597–627. Springer, Heidelberg, August 2018.

[Agr19] Shweta Agrawal. Indistinguishability obfuscation without multilinear maps: New techniques for bootstrapping and instantiation. In *Eurocrypt*, 2019.

[AGRW17] Michel Abdalla, Romain Gay, Mariana Raykova, and Hoeteck Wee. Multi-input inner-product functional encryption from pairings. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 601–626. Springer, Heidelberg, April / May 2017.

[AGT21a] Shweta Agrawal, Rishab Goyal, and Junichi Tomida. Multi-input quadratic functional encryption from pairings. In *CRYPTO*, 2021.

[AGT21b] Shweta Agrawal, Rishab Goyal, and Junichi Tomida. Multi-party functional encryption. In *TCC*, 2021.

[AGT22] Shweta Agrawal, Rishab Goyal, and Junichi Tomida. Multi-input quadratic functional encryption: Stronger security, broader functionality. In *TCC*, 2022.

[AIK04] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in NC^0. In *45th FOCS*, pages 166–175. IEEE Computer Society Press, October 2004.

[AJ15] Prabhanjan Ananth and Abhishek Jain. Indistinguishability obfuscation from compact functional encryption. In *CRYPTO*, 2015.

[AJS23] Paul Lou Aayush Jain, Huijia Lin and Amit Sahai. Polynomial-time cryptanalysis of the subspace flooding assumption for post-quantum io. In *Eurocrypt*, 2023.

[AKYY23] Shweta Agrawal, Simran Kumari, Anshu Yadav, and Shota Yamada. Trace and revoke with optimal parameters from polynomial hardness. In *Eurocrypt*, 2023.

[APM20] Shweta Agrawal and Alice Pellet-Mary. Indistinguishability obfuscation without maps: Attacks and fixes for noisy linear fe. In *Advances in Cryptology–EUROCRYPT 2020: 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10–14, 2020, Proceedings, Part I*, pages 110–140. Springer, 2020.

[Att14] Nuttapong Attrapadung. Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 557–577. Springer, Heidelberg, May 2014.

[AWY20] Shweta Agrawal, Daniel Wichs, and Shota Yamada. Optimal broadcast encryption from lwe and pairings in the standard model. In *TCC*, 2020.

[AY20] Shweta Agrawal and Shota Yamada. Optimal broadcast encryption from pairings and lwe. In *EUROCRYPT*, 2020.

[AYY22] Shweta Agrawal, Anshu Yadav, and Shota Yamada. Multi-input attribute based encryption and predicate encryption. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 590–621. Springer, Heidelberg, August 2022.

[BGG$^+$14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In *EUROCRYPT*, 2014.

[BGI$^+$01] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. In *CRYPTO*, 2001.

[BJK$^+$18] Zvika Brakerski, Aayush Jain, Ilan Komargodski, Alain Passelègue, and Daniel Wichs. Non-trivial witness encryption and null-io from standard assumptions. In *SCN*, 2018.

[BLMR13] Dan Boneh, Kevin Lewi, Hart William Montgomery, and Ananth Raghunathan. Key homomorphic PRFs and their applications. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 410–428. Springer, Heidelberg, August 2013.

[BLP$^+$13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *STOC*, 2013.

[BPR12] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 719–737. Springer, Heidelberg, April 2012.

[BV15] Zvika Brakerski and Vinod Vaikuntanathan. Constrained key-homomorphic PRFs from standard lattice assumptions - or: How to secretly embed a circuit in your PRF. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 1–30. Springer, Heidelberg, March 2015.

[BV22] Zvika Brakerski and Vinod Vaikuntanathan. Lattice-inspired broadcast encryption and succinct ciphertext policy abe. In *ITCS*, 2022.

[CDG$^+$18] Jérémy Chotard, Edouard Dufour Sans, Romain Gay, Duong Hieu Phan, and David Pointcheval. Decentralized multi-client functional encryption for inner product. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part II*, volume 11273 of *LNCS*, pages 703–732. Springer, Heidelberg, December 2018.

[CHKP10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In *EUROCRYPT*, 2010.

[DOT18] Pratish Datta, Tatsuaki Okamoto, and Junichi Tomida. Full-hiding (unbounded) multi-input inner product functional encryption from the k-Linear assumption. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018, Part II*, volume 10770 of *LNCS*, pages 245–277. Springer, Heidelberg, March 2018.

[DQV$^+$21] Lalita Devadas, Willy Quach, Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Succinct lwe sampling, random polynomials, and obfuscation. In *Theory of Cryptography: 19th International Conference, TCC 2021, Raleigh, NC, USA, November 8–11, 2021, Proceedings, Part II 19*, pages 256–287. Springer, 2021.

[FFMV23] Danilo Francati, Daniele Friolo, Giulio Malavolta, and Daniele Venturi. Multi-key and multi-input predicate encryption from learning with errors. In *Eurocrypt*, 2023.

[GGG$^+$14] Shafi Goldwasser, S. Dov Gordon, Vipul Goyal, Abhishek Jain, Jonathan Katz, Feng-Hao Liu, Amit Sahai, Elaine Shi, and Hong-Sheng Zhou. Multi-input functional encryption. In *EUROCRYPT*, 2014.

[GGH$^+$13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *FOCS*, 2013. http://eprint.iacr.org/.

[GP21] Romain Gay and Rafael Pass. Indistinguishability obfuscation from circular security. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 736–749, 2021.

[GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM CCS*, 2006.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, 2008.

[KNYY20] Shuichi Katsumata, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. Adaptively secure inner product encryption from LWE. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part III*, volume 12493 of *LNCS*, pages 375–404. Springer, Heidelberg, December 2020.

[LT19] Benoît Libert and Radu Titiu. Multi-client functional encryption for linear functions in the standard model from LWE. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 520–551. Springer, Heidelberg, December 2019.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Eurocrypt*, 2012.

[NR97] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *38th FOCS*, pages 458–467. IEEE Computer Society Press, October 1997.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J.ACM*, 56(6), 2009.

[SW05] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, 2005.

[Tom19] Junichi Tomida. Tightly secure inner product functional encryption: Multi-input and function-hiding constructions. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part III*, volume 11923 of *LNCS*, pages 459–488. Springer, Heidelberg, December 2019.

[Tsa19] Rotem Tsabary. Fully secure attribute-based encryption for t-CNF from LWE. In *CRYPTO*, 2019.

[Tsa22] Rotem Tsabary. Candidate witness encryption from lattice techniques. In *Advances in Cryptology–CRYPTO 2022: 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15–18, 2022, Proceedings, Part I*, pages 535–559. Springer, 2022.

[VWW22] Vinod Vaikuntanathan, Hoeteck Wee, and Daniel Wichs. Witness encryption and null-io from evasive lwe. In *ASIACRYPT*, pages 195–221. Springer, 2022.

[Wee14] Hoeteck Wee. Dual system encryption via predicate encodings. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 616–637. Springer, Heidelberg, February 2014.

[Wee22] Hoeteck Wee. Optimal broadcast encryption and CP-ABE from evasive lattice assumptions. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 217–241. Springer, Heidelberg, May / June 2022.

[WW21] Hoeteck Wee and Daniel Wichs. Candidate obfuscation via oblivious lwe sampling. In *Advances in Cryptology–EUROCRYPT 2021: 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17–21, 2021, Proceedings, Part III*, pages 127–156. Springer, 2021.

[WWW22] Brent Waters, Hoeteck Wee, and David J Wu. Multi-authority abe from lattices without random oracles. In *TCC*, 2022.