# Unbounded ABE for Circuits from LWE, Revisited

Valerio Cini[1] and Hoeteck Wee[1]

NTT Research, Sunnyvale, CA, USA

**Abstract.** We introduce new lattice-based techniques for building ABE for circuits with unbounded attribute length based on the LWE assumption, improving upon the previous constructions of Brakerski and Vaikuntanathan (CRYPTO 16) and Goyal, Koppula, and Waters (TCC 16). Our main result is a simple and more efficient unbounded ABE scheme for circuits where only the circuit depth is fixed at set-up; this is the first unbounded ABE scheme for circuits that rely only on black-box access to cryptographic and lattice algorithms. The scheme achieves semi-adaptive security against unbounded collusions under the LWE assumption. The encryption time and ciphertext size are roughly 3× larger than the prior bounded ABE of Boneh et al. (EUROCRYPT 2014), substantially improving upon the encryption times in prior works. As a secondary contribution, we present an analogous result for unbounded inner product predicate encryption that satisfies weak attribute-hiding.

## 1 Introduction

Attribute-based encryption (ABE) [SW05,GPSW06] is a generalization of public-key encryption to support fine-grained access control for encrypted data. Here, ciphertexts are associated with attributes like '(author:Waters), (inst:UT), (topic:PK)' and keys with access policies like ((topic:Thy) OR (topic:Qu)) AND (NOT(inst:CWI)), and decryption is possible only when the attributes satisfy the access policy. Over past decade, substantial progress has been made in the design and analysis of ABE schemes, leading to a large families of schemes that achieve various trade-offs between efficiency, security and underlying assumptions. Meanwhile, ABE has found use in a variety of settings such as electronic medical records, messaging systems and online social networks; companies like Cloudflare already use ABE to distribute private key storage across data centers [Ver23].

As institutions grow and with new emerging and more complex applications for ABE, we need ABE schemes that can readily accommodate the addition of new roles, entities, attributes and policies. This means that the ABE set-up algorithm should put no restriction on the length of the attributes or the size of the policies that will be used in the ciphertexts and keys. This requirement was introduced and first realized in the work of Lewko and Waters [LW11] under the term *unbounded ABE*; we would henceforth also refer to standard ABE as *bounded ABE*. The Lewko-Waters schemes rely on pairings without random oracles, and have since been improved and extended in several subsequent works [Lew12,OT12,RW13,Att14,KL15,Att16,CGKW18]. All of these schemes are limited to policies described by $\mathrm{NC}^1$ circuits or branching programs, as is the case with all pairing-based ABE schemes.

In 2016, Brakerski and Vaikuntanathan (BV16) gave the first construction of unbounded ABE for circuits [BV16] based on the Learning with Errors (LWE) assumption, building upon bounded ABE schemes in [BGG$^+$14,GVW13]. This was followed shortly by a generalization in Goyal-Koppula-Waters (GKW16) [GKW16] showing a generic compiler of bounded ABE schemes to unbounded ones assuming additionally adaptively secure identity-based encryption (IBE). Both BV16 and GKW16 schemes also achieve *semi-adaptive security* [CW14], a slight strengthening of selective security where an adversary can choose its encryption challenge after seeing the public key. We note that both schemes do inherit the limitation from prior bounded ABE for circuits, in that the depth of the circuits needs to be fixed at set-up; nonetheless, this already capture $\mathrm{NC}^1$ circuits, whose depth can be bounded by security parameter $\lambda$.

One theoretical and practical draw-back of the BV16 and GKW16 schemes is that they require non-black-box access to the underlying cryptographic building blocks and algorithms, which not only incur substantial efficiency overheads during encryption, but also make these schemes harder to implement and deploy in practice. In particular, the BV16 scheme uses homomorphic computation of a pseudorandom function, whereas the GKW16 applies circuit garbling techniques to the underlying ABE schemes. This is in contrast to the afore-mentioned pairing-based unbounded ABE schemes as well as prior LWE-based ABE schemes for circuits, which avoid non-black-box techniques.

### 1.1 Our Results

In this work, we present new LWE techniques for building simple and more efficient unbounded ABE from bounded ones that avoid non-black-box techniques, leading to substantial savings in encryption times. Our constructions are inspired in part by prior pairing-based schemes in [Lew12,OT12,CGKW18], as well as ideas from [Agr17] on how to combine inner-product functionality and BGGHNSVV14 structure.

**Unbounded ABE for circuits.** Our main result is a more efficient unbounded ABE for circuits of a-priori bounded depth $d$ based on the LWE assumption. From a feasibility stand-point, this is the first unbounded ABE scheme for circuits that rely only on black-box access to cryptographic and lattice algorithms. As with BV16 and GKW16, we achieve semi-adaptive security against unbounded collusions. For depth $d$ circuits over $\ell$-bit inputs where only $d$ is fixed at set-up, we have

$$|\mathsf{mpk}| = \mathsf{poly}(d, \lambda), \quad |\mathsf{ct}| = \ell \cdot \mathsf{poly}(d, \lambda), \quad |\mathsf{sk}| = \ell \cdot \mathsf{poly}(d, \lambda)$$

Compared to the BGGHNSVV14 ABE (which only achieves selective security),

- the encryption time and the ciphertext size are roughly $3\times$ larger;
- the decryption time incurs an additive $\ell \cdot \mathsf{poly}(d, \lambda)$ overhead; the overhead is sublinear in the BGGHNSVV14 ABE decryption time $s \cdot \mathsf{poly}(d, \lambda)$.

The efficiency savings over prior works are as follows:

- compared to the BV16 unbounded ABE, the savings in running times and ciphertext/key sizes are two-fold: cutting down $\mathsf{poly}(d + d_{\mathrm{PRF}})$ dependencies to $\mathsf{poly}(d)$ where $d_{\mathrm{PRF}}$ is the depth of a PRF and removing additive overheads corresponding to PRF evaluation; in particular, (i) encryption time in BV16 is mostly dominated by homomorphic evaluation of a PRF with $\ell$-bit output, and our encryption time should be a $\mathsf{poly}(\lambda)$ factor smaller, (ii) for constant-depth circuits and shallow circuits where $d \ll d_{\mathrm{PRF}}$, our scheme is substantially more efficient for all running times and sizes.
- compared to the GKW16 unbounded ABE, our encryption time and ciphertext size are a multiplicative $O(\lambda)$ factor smaller, which corresponds to the overhead from garbling the BGGHNSVV14 ABE encryption circuit.

Decryption times in our scheme and GKW16 are comparable to that in BGGHNSVV14 ABE, and faster than that in BV16. In all three unbounded ABE schemes, the secret key has two components: a private component corresponding to a BGGHNSVV14 ABE secret key of size $\mathsf{poly}(d, \lambda)$ as well as a public component of size $\ell \cdot \mathsf{poly}(d, \lambda)$ that can be reused across all keys for circuits of input length $\ell$. In BV16, the private component is slightly larger $\mathsf{poly}(d + d_{\mathrm{PRF}}, \lambda)$, but the public component is just $\ell + \mathsf{poly}(\lambda)$ bits.

**Unbounded inner product predicate encryption.** Next, we turn our attention to inner product predicate encryption (IPPE) [KSW08], where ciphertexts are associated with (row) vectors $\mathbf{x} \in \mathbb{Z}_q^\ell$ and keys with vectors $\mathbf{y} \in \mathbb{Z}_q^\ell$ and decryption is possible only if their inner product $\mathbf{x}\mathbf{y}^\top$ equals $0$. In addition to hiding the message as in ABE, we require attribute-hiding, namely that ciphertexts hide the attribute $\mathbf{x}$. Unbounded IPPE schemes can be realized from pairings [OT12] with black-box techniques, or from LWE by applying the GKW16 transformation to the bounded IPPE scheme of Agrawal, Freeman, and Vaikuntanathan (AFV11) [AFV11] with non-black-box techniques.

Our second result is a more efficient unbounded inner product predicate encryption scheme based on the LWE assumption. We achieve semi-adaptive, weak attribute-hiding security against unbounded collusions. For vectors over $\mathbb{Z}_q^\ell$ where only $q$ is fixed at set-up, we have

$$|\mathsf{mpk}| = \mathsf{poly}(\log q, \lambda), \quad |\mathsf{ct}| = \ell \cdot \mathsf{poly}(\log q, \lambda), \quad |\mathsf{sk}| = \ell \cdot \mathsf{poly}(\log q, \lambda)$$

Compared to the scheme derived from combining GKW16 with the AFV11 scheme, our encryption time and ciphertext size are a multiplicative $O(\lambda \log q)$ factor smaller, where the $O(\lambda)$ factor comes from garbling as before, and the $O(\log q)$ comes from the fact that we can directly support attributes over $\mathbb{Z}_q$ in our scheme. In contrast, the techniques in BV16 and GKW16 are inherently limited to attributes over a binary alphabet.

| Scheme | Time(Enc) | Time(Dec) |
|---|---|---|
| [BGG+14] | $T_{\mathsf{Enc}}(\ell, d) = \ell \cdot \mathsf{poly}(\lambda, d)$ | $T_{\mathsf{Dec}}(s, d) = s \cdot \mathsf{poly}(\lambda, d)$ |
| [BV16] | $O(T_{\mathsf{Enc}}(\ell, d + d_{\mathrm{PRF}})) + s_{\mathrm{PRF}} \cdot \mathsf{poly}(\lambda, d + d_{\mathrm{PRF}})$ | $T_{\mathsf{Dec}}(s + s_{\mathrm{PRF}}, d + d_{\mathrm{PRF}}) + \ell \cdot \mathsf{poly}(\lambda, d)$ |
| [GKW16] | $\mathsf{poly}(\lambda) \cdot T_{\mathsf{Enc}}(\ell, d)$ | $T_{\mathsf{Dec}}(s, d) + \ell \cdot \mathsf{poly}(\lambda, d)$ |
| this work | $(3 + o(1)) \cdot T_{\mathsf{Enc}}(\ell, d)$ | $T_{\mathsf{Dec}}(s, d) + \ell \cdot \mathsf{poly}(\lambda, d)$ |

Fig. 1: Comparison of running times with prior KP-ABE for circuits of size $s$ and depth $d$ over $\{0,1\}^\ell$. [BGG+14] is used as a benchmark. Here, $d_{\mathrm{PRF}} = O(\log \lambda + \log \ell)$ and $s_{\mathrm{PRF}} = O(\ell \cdot \lambda)$ denotes the depth and size of a PRF for $\ell$-bit inputs. The ciphertext sizes satisfy an analogous relationship, where we replace $T_{\mathsf{Enc}}$ by $S_{\mathsf{Enc}}$, namely: $S_{\mathsf{Enc}}(\ell, d) = \ell \cdot \mathsf{poly}(d, \lambda), O(S_{\mathsf{Enc}}(\ell, d + d_{\mathrm{PRF}})), \mathsf{poly}(\lambda) \cdot S_{\mathsf{Enc}}(\ell, d), (3 + o(1)) \cdot S_{\mathsf{Enc}}(\ell, d)$ respectively. The total key sizes are $S_{\mathsf{Dec}}(d) = \mathsf{poly}(\lambda, d), S_{\mathsf{Dec}}(d + d_{\mathrm{PRF}}) + \ell + \mathsf{poly}(\lambda), S_{\mathsf{Dec}}(d) + \ell \cdot \mathsf{poly}(\lambda), \ell \cdot S_{\mathsf{Dec}}(d)$ respectively.

**Our construction, in a nutshell.** The starting point, following BV16 and GKW16, is to compute/sample a BGGHNSVV14 mpk during key generation, which would be reused across all key queries; this (deceptively) simple idea buys us both short mpk and semi-adaptive security. Decryption would then first reconstruct a BGGHNSVV14 ciphertext w.r.t. mpk and then proceed as in BGGHNSVV14 decryption. The key technical novelties in this work lie in how we enable reconstruction of BGGHNSVV14 ciphertext using simple LWE algebra and techniques (instead of non-black-box techniques), along with a new simple idea for handling circuit with different input lengths in the key queries.

### 1.2 Technical Overview

We proceed to provide a technical overview of our constructions, focusing on the unbounded ABE.

**BGGHNSVV14 ABE.** We begin with an overview of the BGGHNSVV14 bounded ABE scheme for depth $d$ circuits over $\{0,1\}^\ell$ [BGG+14]. Let $\mathbf{A} \in \mathbb{Z}_q^{n \times \ell \cdot m}$ be a matrix where $q \in \mathbb{N}$ is prime and $m = O(n \log q)$. Given $\mathbf{A}$ and a circuit $f : \{0,1\}^\ell \to \{0,1\}$ of depth $d$, we can derive [BGG+14,GSW13] a matrix $\mathbf{A}_f \in \mathbb{Z}_q^{n \times m}$ such that for any $\mathbf{x} \in \{0,1\}^\ell$, we can compute a low-norm matrix $\mathbf{H}_{\mathbf{A},f,\mathbf{x}}$ satisfying

$$(\mathbf{A} - \mathbf{x} \otimes \mathbf{G}) \cdot \mathbf{H}_{\mathbf{A},f,\mathbf{x}} = \mathbf{A}_f - f(\mathbf{x}) \cdot \mathbf{G}, \tag{1}$$

where $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$ is the gadget matrix [MP12] and $\|\mathbf{H}_{\mathbf{A},f,\mathbf{x}}\| \le m^{O(d)}$. The ABE scheme is as follows, omitting error terms in the ciphertext:

$$\begin{aligned}
\mathsf{mpk} &= \mathbf{A}_0 \leftarrow \mathbb{Z}_q^{n \times m}, \mathbf{b} \leftarrow \mathbb{Z}_p^n, \mathbf{A} \leftarrow \mathbb{Z}_q^{n \times \ell \cdot m}. \\
\mathsf{ct} &= (\overbrace{\mathbf{s} \cdot \mathbf{A}_0}^{\mathbf{c}_0}, \overbrace{\mathbf{s} \cdot \mathbf{b}^\top + \mu \cdot \lfloor q/2 \rfloor}^{c_2}, \overbrace{\mathbf{s} \cdot (\mathbf{A} - \mathbf{x} \otimes \mathbf{G})}^{\mathbf{c}_3}),\ \mathbf{s} \leftarrow \mathbb{Z}_q^n. \\
\mathsf{sk} &= \mathbf{k}_f^\top \leftarrow \mathcal{D}_{\mathbb{Z}^{2m},\tau} \text{ s.t. } [\mathbf{A}_0 \mid \mathbf{A}_f] \cdot \mathbf{k}_f^\top = \mathbf{b}^\top.
\end{aligned}$$

Decryption computes an approximation to $\mu \cdot \lfloor q/2 \rfloor$ for $f(\mathbf{x}) = 0$ as follows:

$$\mathbf{c}_2 - \overbrace{[\mathbf{c}_0 \mid \mathbf{c}_3 \cdot \mathbf{H}_{\mathbf{A},f,\mathbf{x}}]}^{\approx \mathbf{s} \cdot [\mathbf{A}_0 | \mathbf{A}_f]} \cdot \mathbf{k}_f^\top.$$

**Compressing mpk.** As a warm-up, we describe an ABE for circuits over $\{0,1\}^\ell$ where $|\mathsf{mpk}| = \mathsf{poly}(d, \lambda)$. It is convenient to then write $\mathbf{A}$ in the BGGHNSVV14 ABE as $\mathbf{A}_i \in \mathbb{Z}_q^{n \times m}, i \in [\ell]$ and $\mathbf{c}_3$ in the ciphertext as $\mathbf{s} \cdot (\mathbf{A}_i - x_i \mathbf{G}), i \in [\ell]$. We want to sample $\mathbf{A}_i$ during key generation (and not set-up) and then compute $\mathbf{s} \cdot (\mathbf{A}_i - x_i \mathbf{G})$ during decryption. In particular,

- $\mathsf{mpk}$ now contains random matrices $\mathbf{B}_0, \mathbf{W}, \mathbf{V} \leftarrow \mathbb{Z}_q^{n \times m}$ in addition to $\mathbf{A}_0, \mathbf{b}$, and $\mathsf{msk}$ contains the trapdoors for $\mathbf{A}_0$ and $\mathbf{B}_0$;
- the ciphertext contains

$$\mathbf{s}_i \cdot \mathbf{B}_0, \quad \{\mathbf{s}_i \cdot \mathbf{W} + \mathbf{s} \cdot \mathbf{G}, \quad \mathbf{s}_i \cdot \mathbf{V} + x_i \cdot \mathbf{s} \cdot \mathbf{G}\}_{i \in [\ell]},$$

  where $\mathbf{s}, \mathbf{s}_i \leftarrow \mathbb{Z}_q^n$ are sampled during encryption;
- during decryption, we compute

$$\mathbf{s} \cdot (\mathbf{A}_i - x_i \cdot \mathbf{G}) \approx (\mathbf{s}_i \cdot \mathbf{W} + \mathbf{s} \cdot \mathbf{G}) \cdot \mathbf{G}^{-1}(\mathbf{A}_i) - (\mathbf{s}_i \cdot \mathbf{V} + x_i \cdot \mathbf{s} \cdot \mathbf{G}) + \mathbf{s}_i \cdot \mathbf{B}_0 \cdot \mathbf{Z}_i$$

  where $\mathbf{Z}_i \leftarrow \mathbf{B}_0^{-1}(\mathbf{V} - \mathbf{W} \cdot \mathbf{G}^{-1}(\mathbf{A}_i))$ is provided in the secret key.
- key generation for $f$ returns the same $(\mathbf{A}_i, \mathbf{Z}_i)$ across all secret keys — generated using a PRF key in $\mathsf{msk}$ so that we don't need to maintain state across key queries — as well as a BGGHNSVV14 secret key for $f$.

This is sufficient for functionality. However, an adversary can also compute $\mathbf{s}_i \cdot \mathbf{B}_0 \cdot \mathbf{Z}_j$ and thus $\mathbf{s}(\mathbf{A}_j - x_i \cdot \mathbf{G})$ for any $i \neq j$. To prevent this attack, we replace $\mathbf{W}$ with $\mathbf{W} + i \cdot \mathbf{G}$ in both the ciphertext and the secret key. This yields the following ABE scheme:

$$\begin{aligned}
\mathsf{mpk} &= \mathbf{A}_0, \mathbf{B}_0, \mathbf{W}, \mathbf{V} \leftarrow \mathbb{Z}_q^{n \times m}, \mathbf{b} \leftarrow \mathbb{Z}_q^n. \\
\mathsf{ct} &= \Big(\overbrace{\mathbf{s} \cdot \mathbf{A}_0}^{\mathbf{c}_0},\ \{\overbrace{\mathbf{s}_i \cdot \mathbf{B}_0}^{\mathbf{c}_{1,i}},\ \overbrace{\mathbf{s}_i \cdot (\mathbf{W} + i \cdot \mathbf{G}) + \mathbf{s} \cdot \mathbf{G}}^{\mathbf{c}_{2,i}},\ \overbrace{\mathbf{s}_i \cdot \mathbf{V} + x_i \cdot \mathbf{s} \cdot \mathbf{G}}^{\mathbf{c}_{3,i}}\}_{i \in [\ell]},\ \overbrace{\mathbf{s} \cdot \mathbf{b}^\top + \mu \cdot \lfloor q/2 \rceil}^{c_4}\Big),\ \mathbf{s}, \mathbf{s}_i \leftarrow \mathbb{Z}_q^n. \\
\mathsf{sk} &= \big(\{\mathbf{Z}_j, \mathbf{R}_j\}_{j \in [\ell]}, \mathbf{k}_f\big), \text{where } \mathbf{Z}_j, \mathbf{R}_j \text{ are fixed across all keys} \\
&\quad \begin{bmatrix} \mathbf{Z}_j \\ \mathbf{R}_j \end{bmatrix} \leftarrow \mathcal{D}_{\mathbb{Z}^{2m \times m}, \tau} \text{ s.t. } \big[\mathbf{B}_0 \mid \mathbf{W} + j \cdot \mathbf{G}\big] \cdot \begin{bmatrix} \mathbf{Z}_j \\ \mathbf{R}_j \end{bmatrix} = \mathbf{V}, \\
&\quad \mathbf{A}_j = \mathbf{G} \cdot \mathbf{R}_j \\
&\quad \mathbf{k}_f^\top \leftarrow \mathcal{D}_{\mathbb{Z}^{2m}, \tau} \text{ s.t. } [\mathbf{A}_0 \mid \mathbf{A}_f] \cdot \mathbf{k}_f^\top = \mathbf{b}^\top
\end{aligned}$$

Decryption first uses

$$\mathbf{c}_{1,i} \cdot \mathbf{Z}_i + \mathbf{c}_{2,i} \cdot \mathbf{R}_i - \mathbf{c}_{3,i} \approx \mathbf{s} \cdot (\mathbf{A}_i - x_i \cdot \mathbf{G}) \tag{2}$$

to recover a BGGHNSVV14 ciphertext, and then proceed as in BGGHNSVV14 decryption.

**Proof overview.** The proof proceeds in two steps:

*Step 1.* For $i = 1, 2, \ldots, \ell$, we rely on pseudorandomness of $\mathbf{s}_i \cdot [\mathbf{B}_0 \mid \mathbf{W} + i \cdot \mathbf{G}]$ to replace $\mathbf{c}_{1,i}, \mathbf{c}_{2,i}$ with random and rewrite $\mathbf{c}_{3,i}$ in terms of $\mathbf{s} \cdot (\mathbf{A}_i - x_i \cdot \mathbf{G})$ using (2). In more detail,

- we program $\mathbf{W} + i \cdot \mathbf{G} = \mathbf{B}_0 \cdot \tilde{\mathbf{W}}$ for a random low-norm $\tilde{\mathbf{W}}$;
- we sample random Gaussian $\mathbf{Z}_i, \mathbf{R}_i$ and program $\mathbf{V}$ accordingly;
- for all $j \neq i$, we sample $\mathbf{Z}_j, \mathbf{R}_j$ using the trapdoor for $[\mathbf{B}_0 \mid \mathbf{W} + j \cdot \mathbf{G}] = [\mathbf{B}_0 \mid \mathbf{B}_0 \cdot \tilde{\mathbf{W}} + (j - i) \cdot \mathbf{G}]$;
- use the LWE assumption to replace $\mathbf{s}_i \cdot \mathbf{B}_0$ with random.

*Step 2.* Run the BGGHNSVV14 security proof. This step knows the trapdoor for $\mathbf{B}_0$, which is used to solve for $\mathbf{Z}_j$.

Our construction and proof strategy achieve semi-adaptive security for the same reason as in BV16, GKW16: the matrices $\mathbf{A}_i$ from the BGGHNSVV14 $\mathsf{mpk}$ are sampled after the adversary chooses its encryption challenge attribute.

**Getting to an unbounded ABE scheme.** In an unbounded ABE scheme, we need to allow both an honest party and an adversary to ask for keys corresponding to functions with different input lengths. The previous scheme already satisfies the syntax of an unbounded ABE scheme, since we can sample the $\mathbf{A}_i$ matrices "on the fly", while using a PRF to ensure that we use the same $\mathbf{A}_i$ across all secret keys. However, it is insecure as an unbounded ABE: consider an attack that fixes $\mathbf{x}^*$ for the challenge ciphertext and then query a $f$ such that $f$ evaluates to true on a prefix of $\mathbf{x}^*$. To defeat this attack, we add $\mathbf{s} \cdot (\mathbf{B}_1 - |\mathbf{x}| \cdot \mathbf{G})$ to the challenge ciphertext and modify $\mathsf{sk}$ for $f : \{0,1\}^\ell \to \{0,1\}$ to satisfy

$$[\mathbf{A}_0 \mid \mathbf{A}_f \mid \mathbf{B}_1 - \ell \cdot \mathbf{G}] \cdot \mathbf{k}_f^\top = \mathbf{b}^\top$$

To handle semi-adaptive security, we would simply guess $|\mathbf{x}^*|$ when simulating $\mathbf{B}_1$ in the reduction, which incurs an additional polynomial loss. This is where GKW16 uses an adaptively secure IBE (for which the known instantiations from LWE in e.g. [Yam16] are more complex than their selectively secure counter-parts), since they embed $|\mathbf{x}^*|$ into the identity of the IBE ciphertext. The BV16 scheme similarly embeds $|\mathbf{x}^*|$ as part of the attribute in an "outer ABE" that plays an analogous role to the IBE ciphertext in GKW16.

**Inner product predicate encryption scheme.** Here, we start with the AFV11 inner product predicate encryption, where an encryption for an attribute $\mathbf{x} = (x_1, \ldots, x_\ell) \in \mathbb{Z}_q^\ell$ is exactly the same as that in the BGGHNSVV14 ABE. Here, we exploit the fact that our construction directly support attributes over $\mathbb{Z}_q$. We can then proceed essentially as before.

### 1.3 Discussion

**Additional comparison with prior approaches.** As mentioned at the beginning of Section 1.1, our constructions are inspired in part by prior pairing-based schemes. To better convey this, compare our ciphertext with that in the KP-ABE for arithmetic span programs in [CGKW18, Section 9.3], where an encryption of $\mathbf{x} = (x_1, \ldots, x_\ell) \in \mathbb{Z}_p^\ell$ has the form:

$$\Big(\overbrace{[\mathbf{s}\mathbf{B}_0]_1}^{\mathbf{c}_0},\ \overbrace{[\mathbf{s}_i\mathbf{B}_0]_1}^{\mathbf{c}_{2,i}},\ \overbrace{[\mathbf{s}_i'\mathbf{B}_0]_1}^{\mathbf{c}_{2,i}'},\ \overbrace{[\mathbf{s}_i(\mathbf{W} + i\mathbf{W}_1) + \mathbf{s}_i'(\mathbf{W}' + i\mathbf{W}_1') + \mathbf{s}(\mathbf{V} + x_i\mathbf{V}')]_1}^{\mathbf{c}_{1,i}},\ [\mathbf{s}\mathbf{b}^\top]_T\mu\Big),\ \mathbf{s}, \mathbf{s}_i, \mathbf{s}_i' \leftarrow \mathbb{Z}_p^k.$$

where $[\cdot]_1$ denotes exponentiation in the group $G_1$. Our $\mathbf{s}_i(\mathbf{W} + i \cdot \mathbf{G})$ is inspired by $\mathbf{s}_i(\mathbf{W} + i \cdot \mathbf{W}_1)$ above. However, note that $\mathbf{s}, \mathbf{s}x_i$ appear together as $\mathbf{s}(\mathbf{V} + x_i\mathbf{V}')$ in $\mathbf{c}_{1,i}$. In our scheme, $\mathbf{sG}, x_i\mathbf{sG}$ appear separately in $\mathbf{s}_{2,i}$ and $\mathbf{s}_{3,i}$ respectively.

In our analysis, we implicitly treat $\{\mathbf{s}_i(\mathbf{W} + i \cdot \mathbf{G})\}_{i \in [\ell]}$ as independent IBE ciphertexts (for the LWE-based IBE in [ABB10]) corresponding to the identities $1, 2, \ldots, \ell$ with randomness $\mathbf{s}_i$. This is again inspired by the pairing-based scheme [CGKW18] which uses IBE techniques in a similar way. IBE schemes are also used in the BV16, GKW16 constructions in a generic manner, whereas our schemes exploit specific algebraic structure in the underlying IBE.

Our construction can be viewed as using a one-key secure inner product functional encryption (IPFE) scheme to compute $\mathbf{s}(\mathbf{A}_i - x_i\mathbf{G})$, where the IPFE ciphertext encrypts $(\mathbf{s}, x_i\mathbf{s})$. The IPFE approach was used in [Agr17, Section 5] to construct a "bounded" ABE for circuits with semi-adaptive security. Our construction is simpler in that we do not need to encrypt a LWE error term, but also more delicate since we want an unbounded ABE scheme.

**Perspective.** Apart from the landmark results of ABE for circuits from LWE about a decade ago now, research on LWE-based ABE has largely lagged behind their pairing-based counter-parts. One reason is that we have a much larger arsenal of techniques in the pairings world, which exploit the rich algebraic structure in pairing groups. We see this work as taking another step towards discovering analogues of these algebraic techniques in the LWE setting, in the specific context of realizing short $\mathsf{mpk}$. We stress that realizing short $\mathsf{mpk}$ (where $|\mathsf{mpk}|$ is much shorter than the ciphertext attributes) is not only relevant for constructing unbounded ABE and IPPE schemes, but also a necessity for several outstanding open problems in the LWE-based ABE literature, notably (i) CP-ABE for unbounded size circuits (even just $\mathsf{NC}^1$), (ii) ABE for DFA and Turing machines, (iii) broadcast encryption where the total parameter size $|\mathsf{mpk}| + |\mathsf{ct}| + |\mathsf{sk}|$ is sublinear in the total number of users, all of which we have made much more substantial progress in the pairings setting. We hope that developing new algebraic techniques for short $\mathsf{mpk}$ as well as LWE analogues of existing pairing-based techniques in this work could help facilitate progress on these open problems.

# 2 Preliminaries

**Notations.** We use boldface lower case for row vectors (e.g. $\mathbf{r}$) and boldface upper case for matrices (e.g. $\mathbf{R}$). For integral vectors and matrices (i.e., those over $\mathbb{Z}$), we use the notation $\|\mathbf{r}\|$, $\|\mathbf{R}\|$ to denote the maximum absolute value over all the entries. We use $v \leftarrow D$ to denote a random sample from a distribution $D$, as well as $v \leftarrow S$ to denote a uniformly random sample from a set $S$. We use $\approx_s$ and $\approx_c$ as the abbreviation for statistically close and computationally indistinguishable. We denoted by $\mathcal{D}_{\mathbb{Z}^m,\chi}$ the (centered) discrete Gaussian distribution over $\mathbb{Z}^m$ with parameter $\chi$, i.e., the distribution over $\mathbb{Z}^m$ where for all $\mathbf{x}$, $\Pr[\mathbf{x}] \propto e^{-\pi\cdot(x_1^2+\cdots+x_m^2)/\chi^2}$.

## 2.1 Pseudoradom Functions

A pseudorandom function (PRF) is a family of functions $\{\mathsf{F}(\mathsf{k},\cdot) : \{0,1\}^{m(\lambda)} \to \{0,1\}^{\ell(\lambda)}\}_{\lambda\in\mathbb{N},\mathsf{k}\in\{0,1\}^\lambda}$ such that:

- *efficiency*: one can compute $\mathsf{F}(\mathsf{k},x)$ in $\mathsf{poly}(\lambda)$-time given $x$ and $\mathsf{k}$,
- *security*: for any PPT adversary $\mathcal{A}$ let

$$\mathsf{Adv}^{\mathrm{PRF}}_{\mathcal{A},\mathsf{F}}(\lambda) := \left|\Pr\left[\mathcal{A}^{\mathsf{F}(\mathsf{k},\cdot)}(1^\lambda) = 1\right] - \Pr\left[\mathcal{A}^{R(\cdot)}(1^\lambda) = 1\right]\right|,$$

where $\mathsf{k} \leftarrow \{0,1\}^\lambda$ and $R \leftarrow \mathcal{F}(\{0,1\}^{m(\lambda)} \to \{0,1\}^{\ell(\lambda)})$, with $\mathcal{F}(\{0,1\}^{m(\lambda)} \to \{0,1\}^{\ell(\lambda)})$ denoting the set of all functions mapping $m(\lambda)$ bits to $\ell(\lambda)$ bits. A PRF $\mathsf{F}$ is secure if for all PPT adversary $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathrm{PRF}}_{\mathcal{A},\mathsf{F}}(\lambda)$ is a negligible function in $\lambda$.

## 2.2 Attribute-based encryption

*Syntax.* A key policy attribute-based encryption (KP-ABE) scheme $\mathit{\Pi}$ for some class $\mathcal{F}$ consists of four algorithms:

- $\mathsf{Setup}(1^\lambda, \mathcal{F}) \to (\mathsf{mpk}, \mathsf{msk})$. The setup algorithm gets as input the security parameter $1^\lambda$ and class description $\mathcal{F}$. It outputs the master public key $\mathsf{mpk}$ and the master secret key $\mathsf{msk}$.
- $\mathsf{Enc}(\mathsf{mpk}, \mathbf{x}, \boldsymbol{\mu}) \to \mathsf{ct}_\mathbf{x}$. The encryption algorithm gets as input $\mathsf{mpk}$, an input $\mathbf{x}$ and a message $\boldsymbol{\mu} \in \{0,1\}^\lambda$. It outputs a ciphertext $\mathsf{ct}_\mathbf{x}$. Note that $\mathbf{x}$ is public given $\mathsf{ct}_\mathbf{x}$.
- $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, f) \to \mathsf{sk}_f$. The key generation algorithm gets as input $\mathsf{mpk}, \mathsf{msk}$ and $f \in \mathcal{F}$. It outputs a secret key $\mathsf{sk}_f$. Note that $f$ is public given $\mathsf{sk}_f$.
- $\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_f, f, \mathsf{ct}_\mathbf{x}, \mathbf{x}) \to \boldsymbol{\mu}$. The decryption algorithm gets as input $\mathsf{sk}_f$ and $\mathsf{ct}_\mathbf{x}$ along with $\mathsf{mpk}$. It outputs a message $\boldsymbol{\mu}$.

*Correctness.* For all $\ell \in \mathbb{N}$, inputs $\mathbf{x} \in \{0,1\}^\ell$, functions $f : \{0,1\}^\ell \to \{0,1\}$ with $f(\mathbf{x}) = 0$, and all $\boldsymbol{\mu} \in \{0,1\}^\lambda$, we require

$$\Pr\left[\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_f, \mathsf{ct}_\mathbf{x}) = \boldsymbol{\mu} : \begin{array}{l} (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{F}) \\ \mathsf{sk}_f \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, f) \\ \mathsf{ct}_\mathbf{x} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathbf{x}, \boldsymbol{\mu}) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$

*Security Definition.* For a stateful adversary $\mathcal{A}$, we define the advantage function

$$\mathsf{Adv}^{\mathrm{ABE}}_{\mathcal{A},\mathit{\Pi}}(\lambda) := \Pr\left[b = b' : \begin{array}{l} (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{F}) \\ \mathbf{x}^* \leftarrow \mathcal{A}(1^\lambda, \mathsf{mpk}) \\ (\boldsymbol{\mu}_0, \boldsymbol{\mu}_1) \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{mpk},\mathsf{msk},\cdot)}(\mathsf{mpk}) \\ b \leftarrow \{0,1\}; \mathsf{ct}_{\mathbf{x}^*} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathbf{x}^*, \boldsymbol{\mu}_b) \\ b' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{mpk},\mathsf{msk},\cdot)}(\mathsf{ct}_{\mathbf{x}^*}) \end{array}\right] - \frac{1}{2},$$

with the restriction that all queries $f : \{0,1\}^\ell \to \{0,1\}$ that $\mathcal{A}$ sent to $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \cdot)$ satisfy either $\ell \neq |\mathbf{x}^*|$ or $f(\mathbf{x}^*) = 1$. An ABE scheme $\mathit{\Pi}$ is *semi-adaptively secure* if for all PPT adversaries $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathrm{ABE}}_{\mathcal{A},\mathit{\Pi}}(\lambda)$ is a negligible function in $\lambda$.

### 2.3 Lattices background

**Learning with Errors.** Given $n, m, q, \chi_e \in \mathbb{N}$, the $\mathsf{LWE}_{n,m,q,\chi_e}$ assumption states that

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

where

$$\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}, \mathbf{s} \leftarrow \mathbb{Z}_q^n, \mathbf{e} \leftarrow \mathcal{D}_{\mathbb{Z}^m, \chi_e}, \mathbf{c} \leftarrow \mathbb{Z}_q^m.$$

**Leftover Hash Lemma and Generalizations** A result that we will use is the so-called leftover hash lemma (LHL) [HILL99], which states that for $m \geq (n+1) \cdot \log q + 2 \cdot \lambda$ the distribution of $(\mathbf{A}, \mathbf{u} = \mathbf{A} \cdot \mathbf{x})$ for uniform and independent $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and $\mathbf{x} \leftarrow \{1, -1\}^m$ is statistically indistinguishable from uniformly random.

**Lemma 1 (Generalized Leftover Hash Lemma [DRS04,ABB10]).** *Suppose that $m > (n+1)\log q + \omega(\log n)$ and that $q > 2$ is prime. Let $\mathbf{R}$ be an $m \times k$ matrix chosen uniformly in $\{1, -1\}^{m \times k} \mod q$ where $k = k(n)$ is polynomial in $n$. Let $\mathbf{A}$ and $\mathbf{B}$ be matrices chosen uniformly in $\mathbb{Z}_q^{n \times m}$ and $\mathbb{Z}_q^{n \times k}$ respectively. Then, for all vectors $\mathbf{w}$ in $\mathbb{Z}_q^m$, the distribution $(\mathbf{A}, \mathbf{A} \cdot \mathbf{R}, \mathbf{w}^\top \cdot \mathbf{R})$ is statistically close to the distribution $(\mathbf{A}, \mathbf{B}, \mathbf{w}^\top \cdot \mathbf{R})$.*

**Trapdoor and preimage sampling.** Let $n, q \in \mathbb{Z}$,

$$\mathbf{g}_q = (1, 2, 4, \ldots, 2^{\lceil \log q \rceil - 1}) \in \mathbb{Z}^{\lceil \log q \rceil}.$$

The gadget matrix $\mathbf{G}_{n,q}$ is defined as the diagonal concatenation of $\mathbf{g}_q$ $n$ times. Formally, $\mathbf{G}_{n,q} = \mathbf{g}_q \otimes \mathbf{I}_n \in \mathbb{Z}^{n \times n \cdot \lceil \log q \rceil}$. For any $t \in \mathbb{Z}$, the function $\mathbf{G}_{n,q}^{-1} : \mathbb{Z}_q^{n \times t} \to \{0,1\}^{n \cdot \lceil \log q \rceil \times t}$ expands each entry $a \in \mathbb{Z}_q$ of the input matrix into a column of size $\lceil \log q \rceil$ consisting of the bit-representation of $a$. For any matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times t}$ it holds that $\mathbf{G}_{n,q} \cdot \mathbf{G}_{n,q}^{-1}(\mathbf{A}) = \mathbf{A} \mod q$. We refer to the gadget matrix simply as $\mathbf{G}$ when parameters $n$ and $q$ are clear from the context.

Let $n, m, q \in \mathbb{N}$ and 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 $\mathbf{A}^{-1}(\mathbf{V}, \tau)$ denote the random variable whose distribution is the discrete Gaussian $\mathcal{D}_{\mathbb{Z}^{m \times m'}, \tau}$ conditioned on $\mathbf{A} \cdot \mathbf{A}^{-1}(\mathbf{V}, \tau) = \mathbf{V} \mod q$. If $\mathbf{Y} \leftarrow \mathbf{A}^{-1}(\mathbf{V}, \tau)$ then $\|\mathbf{Y}\| \leq k \cdot \tau \cdot \sqrt{m \cdot m'}$ with probability at least $1 - e^{-\Omega(k^2)}$. A matrix $\mathbf{T} \in \mathbb{Z}^{m \times m}$ such that $\mathbf{A} \cdot \mathbf{T} = \mathbf{H} \cdot \mathbf{G}$, for some invertible matrix $\mathbf{H} \in \mathbb{Z}_q^{n \times n}$ is called a $\tau$-trapdoor for $\mathbf{A}$, for $\tau \geq 2 \cdot m \cdot \sqrt{n \cdot \log q} \cdot \|\mathbf{T}\|$. The following properties have been established in a long sequence of works.

**Lemma 2 (Trapdoor Generation and Sampling [Ajt96,GPV08,MP12]).** *There exists a pair of probabilistic polynomial-time algorithms:*

- $\mathsf{TrapGen}(1^n, 1^m, q)$ *that for all $m \geq m_0 = m_0(n,q) = O(n \log q)$, outputs $(\mathbf{A}, \mathbf{T_A})$ s.t. $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ is within statistical distance $2^{-n}$ from uniform and $\mathbf{T_A}$ is a $\tau$-trapdoor for $\mathbf{A}$ where $\tau = O(\sqrt{n} \cdot \log q \cdot \log n)$.*
- $\mathsf{SamplePre}(\mathbf{A}, \mathbf{T}, \mathbf{V}, \tau)$ *that given $\mathbf{A}$ and any $\tau$-trapdoor $\mathbf{T}$ of $\mathbf{A}$, outputs a sample from $\mathbf{A}^{-1}(\mathbf{V}, \tau)$.*

*Moreover*

1. *for $\mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z}^m, \tau}$, the marginal distribution of $\mathbf{y} = \mathbf{A} \cdot \mathbf{x} \in \mathbb{Z}_q^n$ is uniform (up to $\mathsf{negl}(n)$ statistical distance), and the conditional distribution of $\mathbf{x}$ given $\mathbf{y}$ is $\mathbf{A}^{-1}(\mathbf{y}, \tau)$.*

**Lemma 3 (Trapdoor Extension [ABB10,CHKP10]).** *Given $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, with a $\tau$-trapdoor $\mathbf{T}$, it is efficient to sample from $[\mathbf{A}|\mathbf{B}]^{-1}(\cdot, \tau)$ for all $\mathbf{B} \in \mathbb{Z}_q^{n \times k}$. Moreover, for any $\mathbf{V} \in \mathbb{Z}_q^{n \times m'}$, the following two distributions are statistically close*

- $\mathbf{U} \in \mathbb{Z}^{m+k \times m'}$, *where $\mathbf{U} \leftarrow [\mathbf{A}|\mathbf{B}]^{-1}(\mathbf{V}, \tau)$,*
- $\begin{bmatrix} \mathbf{U}_0 \\ \mathbf{U}_1 \end{bmatrix} \in \mathbb{Z}^{m+k \times m'}$, *where $\mathbf{U}_1 \leftarrow \mathcal{D}_{\mathbb{Z}^{k \times m'}, \tau}$ and $\mathbf{U}_0 \leftarrow \mathbf{A}^{-1}(\mathbf{V} - \mathbf{B}\mathbf{U}_1, \tau)$.*

Another related result that we will use is the so-called leftover hash lemma (LHL) [HILL99], which states that for $m \geq (n+1) \cdot \log q + 2 \cdot \lambda$ the distribution of $(\mathbf{A}, \mathbf{u} = \mathbf{A} \cdot \mathbf{x})$ for uniform and independent $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and $\mathbf{x} \leftarrow \{0, 1\}^m$ is statistically indistinguishable from uniformly random.

**Homomorphic Computation on Matrices.** We recall basic homomorphic computation on matrices used in BGGHNSVV14:

**Theorem 1 ([BGG+14,GSW13]).** *There exist efficient deterministic algorithms* $\mathsf{EvalF}$ *and* $\mathsf{EvalFX}$ *such that for all* $n, q, \ell \in \mathbb{N}$*, and for any sequence of matrices* $\mathbf{A} = (\mathbf{A}_1, \ldots, \mathbf{A}_\ell) \in (\mathbb{Z}^{n \times n \cdot \lceil \log q \rceil})^\ell$*, for any depth-*$d$ *Boolean circuit* $f : \{0,1\}^\ell \to \{0,1\}$ *and for every* $\mathbf{x} = (x_1, \ldots, x_\ell) \in \{0,1\}^\ell$*, the following properties hold.*

- *The outputs* $\mathbf{A}_f = \mathsf{EvalF}(\mathbf{A}, f)$ *and* $\mathbf{H}_{\mathbf{A},f,\mathbf{x}} = \mathsf{EvalFX}(\mathbf{A}, f, \mathbf{x})$ *are matrices in* $\mathbb{Z}_q^{n \times (n \cdot \lceil \log q \rceil)}$ *and* $\mathbb{Z}^{(\ell \cdot n \cdot \lceil \log q \rceil) \times (n \cdot \lceil \log q \rceil)}$,
- *It holds that* $\|\mathbf{H}_{\mathbf{A},f,\mathbf{x}}\| \leq (n \cdot \log q)^{O(d)}$,
- *It holds that*

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

For a proof of this theorem, we refer the reader to [BCTW16, Fact 3.4].

## 3 Unbounded ABE for Circuits

We refer to Section 1.2 for an overview of the scheme and the security proof.

**Construction.** Let the ABE $\Pi = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ for the family $\mathcal{F}_d$ of circuits of depth $d$, over $\ell$-bit inputs for any $\ell \in \mathbb{N}$, be defined as follows:

- $\mathsf{Setup}(1^\lambda, 1^d)$: Sample

$$(\mathbf{A}_0, \mathbf{T}_{\mathbf{A}_0}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q), (\mathbf{B}_0, \mathbf{T}_{\mathbf{B}_0}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q),$$
$$\mathbf{B}_1, \mathbf{W}, \mathbf{V} \leftarrow \mathbb{Z}_p^{n \times m}, \mathbf{D} \leftarrow \mathbb{Z}_q^{n \times \lambda},$$
$$\mathsf{k} \leftarrow \{0,1\}^\lambda.$$

  where $q$ is prime[1]. Set $\mathsf{mpk} = (\mathbf{A}_0, \mathbf{B}_0, \mathbf{B}_1, \mathbf{W}, \mathbf{V}, \mathbf{D})$, and $\mathsf{msk} = (\mathbf{T}_{\mathbf{A}_0}, \mathbf{T}_{\mathbf{B}_0}, \mathsf{k})$. Return $(\mathsf{mpk}, \mathsf{msk})$.
- $\mathsf{Enc}(\mathsf{mpk}, \mathbf{x} \in \{0,1\}^\ell, \boldsymbol{\mu} \in \{0,1\}^\lambda)$: Let $\ell = |\mathbf{x}|$. Sample

$$\mathbf{s} \leftarrow \mathbb{Z}_q^n, \mathbf{e}_0 \leftarrow \mathcal{D}_{\mathbb{Z}^m,\chi}, \mathbf{e}_4 \leftarrow \mathcal{D}_{\mathbb{Z}^\lambda,\chi}, \mathbf{e}_5 \leftarrow \mathcal{D}_{\mathbb{Z}^m,\chi'},$$
$$\mathbf{s}_j \leftarrow \mathbb{Z}_q^n, \mathbf{e}_{1,j} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\chi}, \mathbf{e}_{2,j} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\chi'}, \mathbf{e}_{3,j} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\chi''} \quad \text{for all } j \in [\ell].$$

  Compute

$$\begin{aligned}
\mathbf{c}_0 &:= \mathbf{s} \cdot \mathbf{A}_0 + \mathbf{e}_0 \bmod q,\\
\mathbf{c}_{1,j} &:= \mathbf{s}_j \cdot \mathbf{B}_0 + \mathbf{e}_{1,j} \bmod q \quad \text{for all } j \in [\ell],\\
\mathbf{c}_{2,j} &:= \mathbf{s}_j \cdot (\mathbf{W} + j \cdot \mathbf{G}) + \mathbf{s} \cdot \mathbf{G} + \mathbf{e}_{2,j} \bmod q \quad \text{for all } j \in [\ell],\\
\mathbf{c}_{3,j} &:= \mathbf{s}_j \cdot \mathbf{V} + x_j \cdot \mathbf{s} \cdot \mathbf{G} + \mathbf{e}_{3,j} \bmod q \quad \text{for all } j \in [\ell],\\
\mathbf{c}_4 &:= \mathbf{s} \cdot \mathbf{D} + \boldsymbol{\mu} \cdot \lfloor q/2 \rfloor + \mathbf{e}_4 \bmod q.\\
\mathbf{c}_5 &:= \mathbf{s} \cdot (\mathbf{B}_1 - \ell \cdot \mathbf{G}) + \mathbf{e}_5 \bmod q.
\end{aligned}$$

  Output $\mathsf{ct}_\mathbf{x} := (\mathbf{c}_0, \{\mathbf{c}_{1,j}\}_{j\in[\ell]}, \{\mathbf{c}_{2,j}\}_{j\in[\ell]}, \{\mathbf{c}_{3,j}\}_{j\in[\ell]}, \mathbf{c}_4, \mathbf{c}_5)$.
- $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, f)$: Let $\ell$ equal the size of $f$'s inputs. For all $j \in [\ell]$, sample

$$\mathbf{K}_j \leftarrow \mathsf{SamplePre}\left([\mathbf{B}_0 | \mathbf{W} + j \cdot \mathbf{G}], \begin{bmatrix} \mathbf{T}_{\mathbf{B}_0} \\ \mathbf{0}_{m \times m} \end{bmatrix}, \mathbf{V}, \tau_1; \mathsf{F}(\mathsf{k}, j)\right).$$

  Parse

$$\mathbf{K}_j = \begin{bmatrix} \mathbf{Z}_j \\ \mathbf{R}_j \end{bmatrix},$$

[1] We can also adapt the construction to support non-prime moduli using techniques from [MP12].

and let $\mathbf{A}_j := \mathbf{G} \cdot \mathbf{R}_j \bmod q$. Let $\mathbf{A} := [\mathbf{A}_1 | \ldots | \mathbf{A}_\ell]$ and $\mathbf{A}_f = \mathsf{EvalF}(\mathbf{A}, f)$. Sample

$$\mathbf{K}_f \leftarrow \mathsf{SamplePre}\left( [\mathbf{A}_0 | \mathbf{A}_f | \mathbf{B}_1 - \ell \cdot \mathbf{G}], \begin{bmatrix} \mathbf{T}_{\mathbf{A}_0} \\ \mathbf{0}_{m \times m} \\ \mathbf{0}_{m \times m} \end{bmatrix}, \mathbf{D}, \tau_2 \right).$$

Output $\mathsf{sk}_f := (\{\mathbf{K}_j\}_{j \in [\ell]}, \mathbf{K}_f)$. Here, $\mathbf{K}_f$ is the private component, and $\{\mathbf{K}_j\}$ is the public component and can be reused over all functions of input length at most $\ell$.

- $\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_f, f, \mathsf{ct}_{\mathbf{x}}, \mathbf{x})$: Let $\ell = |\mathbf{x}|$. Parse $\mathsf{ct}_{\mathbf{x}} = (\mathbf{c}_0, \{\mathbf{c}_{1,j}\}_{j \in [\ell]}, \{\mathbf{c}_{2,j}\}_{j \in [\ell]}, \{\mathbf{c}_{3,j}\}_{j \in [\ell]}, \mathbf{c}_4, \mathbf{c}_5)$, $\mathsf{sk}_f = (\{\mathbf{K}_j\}_{j \in [\ell]}, \mathbf{K}_f)$, and $\mathbf{K}_j = \begin{bmatrix} \mathbf{Z}_j \\ \mathbf{R}_j \end{bmatrix}$ for all $j \in [\ell]$. Let $\mathbf{A}_j = \mathbf{G} \cdot \mathbf{R}_j$ and $\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \ldots & \mathbf{A}_\ell \end{bmatrix}$. Compute $\mathbf{H}_{\mathbf{A}, f, \mathbf{x}} = \mathsf{EvalFX}(\mathbf{A}, f, \mathbf{x})$. For each $j \in [\lambda]$, check if the $j$-th entry of

$$\mathbf{c}_4 - \left[ \mathbf{c}_0 \,\middle|\, \left( \left[ \begin{bmatrix} \mathbf{c}_{1,1} & \mathbf{c}_{2,1} \end{bmatrix} \cdot \mathbf{K}_1 \mid \ldots \mid \begin{bmatrix} \mathbf{c}_{1,\ell} & \mathbf{c}_{2,\ell} \end{bmatrix} \cdot \mathbf{K}_\ell \right] - \begin{bmatrix} \mathbf{c}_{3,1} & \ldots & \mathbf{c}_{3,\ell} \end{bmatrix} \right) \cdot \mathbf{H}_{\mathbf{A}, f, \mathbf{x}} \,\middle|\, \mathbf{c}_5 \right] \cdot \mathbf{K}_f$$

is $q/4$-close to $q/2$. If so, set $\mu_j := 1$. Else, $\mu_j := 0$. Return $\boldsymbol{\mu}$.

**Parameters.** We have 3 gaussian parameters:

$$\overbrace{\chi}^{\approx \|\mathbf{e}_0\|, \|\mathbf{e}_{1,j}\|, \|\mathbf{e}_4\|} \quad \leq \quad \overbrace{\chi'}^{\approx \|\mathbf{e}_{2,j}\|, \|\mathbf{e}_5\|} \quad \leq \quad \overbrace{\chi''}^{\approx \|\mathbf{e}_{3,j}\|}.$$

The parameters requirements can be compactly specified as:

$$\begin{aligned}
& m \geq O(n \log q) && \text{trapdoor generation (Lemma 2)} \\
& 2^{n^\delta} \geq q/\chi_0, \qquad \chi \geq O(n + \lambda) && \mathsf{LWE}_{n, \chi_{s_0}, q} \text{ hardness } (\mathsf{H}_{3,i,6} \approx_c \mathsf{H}_{3,i,7}, \mathsf{H}_2 \approx_c \mathsf{H}_3, \mathsf{H}_7 \approx_c \mathsf{H}_8) \\
& \chi' \geq \chi \cdot \mathsf{poly}(\lambda, m) \cdot \lambda^{\omega(1)} && \text{noise flooding } (\mathsf{H}_{3,i,5} \approx_s \mathsf{H}_{3,i,6}, \mathsf{H}_6 \approx_s \mathsf{H}_7) \\
& \chi'' \geq \chi' \cdot \tau_1 \cdot \mathsf{poly}(\lambda, m) \cdot \lambda^{\omega(1)} && \text{noise flooding } (\mathsf{H}_{3,i,4} \approx_s \mathsf{H}_{3,i,5}, \mathsf{H}_6 \approx_s \mathsf{H}_7) \\
& m \geq (n+1) \cdot \log q + \omega(\log n) + 2\lambda && \text{(G)LHL } (\mathsf{H}_{3,i,0} \approx_s \mathsf{H}_{3,i,1}, \mathsf{H}_{3,i,7} \approx_s \mathsf{H}_{3,i,8}) \\
& \tau_1 \geq O(m^2) && \text{trapdoor generation } (\mathsf{H}_{3,i,2} \approx_s \mathsf{H}_{3,i,3}) \\
& \tau_2 \geq \lambda^{\omega(1)} \cdot m^3 \cdot B && \text{trapdoor generation } (\mathsf{H}_4 \approx_s \mathsf{H}_5) \\
& q \geq \mathsf{poly}(\lambda, m, \ell) \cdot \tau_2 \cdot \tau_1 \cdot B \cdot (\chi + \chi' + \chi'') && \text{correctness (Theorem 2)}
\end{aligned}$$

We bound the adversarially chosen parameter $d, \ell$ by $\lambda^{\omega(1)}$. Taking $\lambda_1 = \lambda^{\omega(1)}$, and additionally bounding the $\mathsf{poly}(\lambda, m, \ell)$ terms by $\lambda_1$, we can set

$$\begin{aligned}
& m = O(n \log q), && && \\
& \chi = \lambda_1, && \chi' = \lambda_1^3, && \chi'' = \lambda_1^6, \\
& \tau_1 = \lambda_1, && \tau_2 = B \cdot \lambda_1^2 = \lambda_1^{O(d)}, && \\
& q = B \cdot \tau_2 \cdot \lambda_1^8 = \lambda_1^{O(d)}, && n = O(\log B + \log \lambda_1)^{1/\delta} = O(d \cdot \log \lambda_1)^{1/\delta}, &&
\end{aligned} \tag{3}$$

where in the last two lines, we use $B \leq m^{O(d)} \leq \lambda_1^{O(d)}$.

**Efficiency.** Our ABE scheme achieves

$$|\mathsf{mpk}| = O((n \cdot \log q)^2), \quad |\mathsf{ct}| = O(\ell \cdot n \cdot (\log q)^2), \quad |\mathsf{sk}| = O(\ell \cdot (n \cdot \log q)^2 \cdot \log \tau_1 + \lambda \cdot n \cdot \log q \cdot \log \tau_2).$$

This yields the following parameter sizes (in bits) for our ABE scheme:

$$|\mathsf{mpk}| = O_\lambda(d^{2+2/\delta}), \quad |\mathsf{ct}| = O_\lambda(\ell \cdot d^{2+1/\delta}), \quad |\mathsf{sk}| = O_\lambda(\ell \cdot d^{2+2/\delta}).$$

where $O_\lambda(\cdot)$ hides factors polynomial in $\lambda$ (bounded by $\lambda^4$). Here, we use $n = O(d^{1/\delta} \cdot \lambda), \log q = O(d \cdot \lambda)$, where we do optimize on the dependency on $d$ but not on $\lambda$.

**Comparison with BGGHNSVV14 ABE.** To compare concrete efficiency of our construction against the BGGHNSVV14 ABE, let $n, m, q$ denote the parameters in our scheme and $n_0, m_0, q_0$ those in BGGHNSVV14. Since $q_0 \geq B$, we can set
$$q = q_0 \cdot \lambda^{\omega(1)}.$$

This implies that we have
$$\log q = (1 + o(1)) \cdot \log q_0, \qquad n = (1 + o(1)) \cdot n_0, \quad \text{and} \quad m = (1 + o(1)) \cdot m_0.$$

In particular, $n, m$, and $\log q$ factors are essentially the same in both schemes. Therefore, to compare concrete efficiency with BGGHNSVV14 ABE, we can compare the number of field (i.e., $\mathbb{Z}_q$) elements and operations.

- Our ciphertext size is $(3\ell + 2) \cdot m + \lambda$ elements in $\mathbb{Z}_q$, whereas that in BGGHNSVV14 is $(\ell + 1) \cdot m + \lambda$ elements.
- Encryption requires $(3\ell + 2) \cdot m + \lambda$ vector-vector products over $\mathbb{Z}_q^n$ and sampling $(3\ell + 2) \cdot m + \lambda$ gaussians over $\mathbb{Z}_q$, whereas that in BGGHNSVV14 requires $(\ell + 1) \cdot m + \lambda$ vector-vector products and $(\ell + 1) \cdot m + \lambda$ gaussians.
- Our secret key contains a private component with $m\lambda$ $\mathbb{Z}_q$-elements, and a public component with $m\ell n$ $\mathbb{Z}_q$-elements, whereas that in BGGHNSVV14 is $m\lambda$ elements.
- Decryption in both schemes are dominated by $s \cdot \mathsf{poly}(\lambda)$ time to compute $\mathbf{H}_{\mathbf{A},f,\mathbf{x}}$, with an additive $\ell \cdot \mathsf{poly}(\lambda)$ overhead in our scheme.

Notice that this also applies to GKW16 since it uses the BGGHNSVV14 scheme as the underlying building block.

**Theorem 2 (Correctness).** *Let $\Pi$ be the* ABE *construction described in Section 3, with parameters as in Equation (3). Then $\Pi$ is correct.*

*Proof.* Fix $\mathbf{x}, f$ such that $f(\mathbf{x}) = 0$. The bulk of the proof lies in showing that
$$\Big[\mathbf{c}_0 \,\Big|\, \big(\big[[\mathbf{c}_{1,1}\ \mathbf{c}_{2,1}] \cdot \mathbf{K}_1 \mid \ldots \mid [\mathbf{c}_{1,\ell}\ \mathbf{c}_{2,\ell}] \cdot \mathbf{K}_\ell\big] - [\mathbf{c}_{3,1} \ldots \mathbf{c}_{3,\ell}]\big) \cdot \mathbf{H}_{\mathbf{A},f,\mathbf{x}} \,\Big|\, \mathbf{c}_5\Big]$$
$$= \mathbf{s} \cdot [\mathbf{A}_0 \mid \mathbf{A}_f \mid \mathbf{B}_1 - \ell \cdot \mathbf{G}] + \mathbf{e}'_{f,\mathbf{x}} \bmod q \tag{4}$$

where $\|\mathbf{e}'_{f,\mathbf{x}}\|$ is small. Correctness then follows readily from the fact that
$$\mathbf{c}_4 - (\mathbf{s} \cdot [\mathbf{A}_0 \mid \mathbf{A}_f \mid \mathbf{B}_1 - \ell \cdot \mathbf{G}] + \mathbf{e}'_{f,\mathbf{x}}) \cdot \mathbf{K}_f = \boldsymbol{\mu} \cdot \lfloor q/2 \rfloor + \mathbf{e}_4 - \mathbf{e}'_{f,\mathbf{x}} \cdot \mathbf{K}_f \bmod q.$$

To prove Eq. (4):

- First, for any $j \in [\ell]$, we have
$$\begin{aligned}[\mathbf{c}_{1,j}\ \mathbf{c}_{2,j}] \cdot \mathbf{K}_i &= (\mathbf{s}_j \cdot [\mathbf{B}_0 \mid \mathbf{W} + j \cdot \mathbf{G}] + \mathbf{s} \cdot [\mathbf{0}_{n \times m} \mid \mathbf{G}] + [\mathbf{e}_{1,j} \mid \mathbf{e}_{2,j}]) \cdot \mathbf{K}_j \\ &= \mathbf{s}_j \cdot [\mathbf{B}_0 \mid \mathbf{W} + j \cdot \mathbf{G}] \cdot \mathbf{K}_j + \mathbf{s} \cdot [\mathbf{0}_{n \times m} \mid \mathbf{G}] \cdot \mathbf{K}_j + \boxed{[\mathbf{e}_{1,j} \mid \mathbf{e}_{2,j}] \cdot \mathbf{K}_j} \\ &\approx \mathbf{s}_i \cdot \mathbf{V} + \mathbf{s} \cdot \mathbf{A}_i \bmod q.\end{aligned}$$
- Further, we have
$$\begin{aligned}\mathbf{s}_j \cdot \mathbf{V} + \mathbf{s} \cdot \mathbf{A}_j - \mathbf{c}_{3,j} &= \mathbf{s}_j \cdot \mathbf{V} + \mathbf{s} \cdot \mathbf{A}_j - (\mathbf{s}_j \cdot \mathbf{V} + x_j \cdot \mathbf{s} \cdot \mathbf{G} + \boxed{\mathbf{e}_{3,j}}) \\ &\approx \mathbf{s} \cdot (\mathbf{A}_j - x_j \cdot \mathbf{G}) \bmod q.\end{aligned}$$
- We deduce that
$$\big[[\mathbf{c}_{1,1}\ \mathbf{c}_{2,1}] \cdot \mathbf{K}_1 \mid \ldots \mid [\mathbf{c}_{1,\ell}\ \mathbf{c}_{2,\ell}] \cdot \mathbf{K}_\ell\big] - [\mathbf{c}_{3,1} \ldots \mathbf{c}_{3,\ell}] = \mathbf{s} \cdot (\mathbf{A} - \mathbf{x} \otimes \mathbf{G}) + \mathbf{e}_{\mathbf{x}} \bmod q,$$
where $\mathbf{A} := [\mathbf{A}_1 \mid \ldots \mid \mathbf{A}_\ell]$ and $\mathbf{e}_{\mathbf{x}} := \big[[\mathbf{e}_{1,1}\ \mathbf{e}_{2,1}] \cdot \mathbf{K}_1 \mid \ldots \mid [\mathbf{e}_{1,\ell}\ \mathbf{e}_{2,\ell}] \cdot \mathbf{K}_\ell\big] - [\mathbf{e}_{3,1} \mid \ldots \mid \mathbf{e}_{3,\ell}]$.

- Using the key equation
$$(\mathbf{A} - \mathbf{x} \otimes \mathbf{G}) \cdot \mathbf{H}_{\mathbf{A},f,\mathbf{x}} = \mathbf{A}_f \bmod q,$$
as $f(\mathbf{x}) = 0$, we have
$$\begin{aligned}\left[\left[\mathbf{c}_{1,1}\ \mathbf{c}_{2,1}\right] \cdot \mathbf{K}_1 \mid \ldots \mid \left[\mathbf{c}_{1,\ell}\ \mathbf{c}_{2,\ell}\right] \cdot \mathbf{K}_\ell\right] \cdot \mathbf{H}_{\mathbf{A},f,\mathbf{x}} &= \mathbf{s} \cdot (\mathbf{A} - \mathbf{x} \otimes \mathbf{G}) + \mathbf{e}_\mathbf{x}) \cdot \mathbf{H}_{\mathbf{A},f,\mathbf{x}} \\ &= \mathbf{s} \cdot \mathbf{A}_f + \boxed{\mathbf{e}_\mathbf{x} \cdot \mathbf{H}_{\mathbf{A},f,\mathbf{x}}} \\ &\approx \mathbf{s} \cdot \mathbf{A}_f \bmod q.\end{aligned}$$
- Putting everything together, we obtain Eq. (4) with
$$\begin{aligned}\mathbf{e}'_{f,\mathbf{x}} &= \left[\mathbf{e}_0 \mid \mathbf{e}_\mathbf{x} \cdot \mathbf{H}_{\mathbf{A},f,\mathbf{x}} \mid \mathbf{e}_5\right] \\ &= \left[\mathbf{e}_0 \mid \left(\left[\left[\mathbf{e}_{1,1}\ \mathbf{e}_{2,1}\right] \cdot \mathbf{K}_1 \mid \ldots \mid \left[\mathbf{e}_{1,\ell}\ \mathbf{e}_{2,\ell}\right] \cdot \mathbf{K}_\ell\right] - \left[\mathbf{e}_{3,1} \mid \ldots \mid \mathbf{e}_{3,\ell}\right]\right) \cdot \mathbf{H}_{\mathbf{A},f,\mathbf{x}} \mid \mathbf{e}_5\right],\end{aligned}$$
where
$$\begin{aligned}\|\mathbf{e}'_{f,\mathbf{x}}\| \leq \lambda \cdot \chi& \\ &+ \lambda^2 \cdot 2 \cdot \ell \cdot m^2 \cdot (\chi + \chi' + \chi'') \cdot \tau_1 \cdot B \\ &+ \lambda \cdot \chi'.\end{aligned}$$

In particular, the norm of the final error term is, with overwhelming probability in $\lambda$, bounded by
$$\begin{aligned}\|\mathbf{e}_4\| + \|\mathbf{e}'_{f,\mathbf{x}} \cdot \mathbf{K}_f\| \leq \lambda \cdot \chi& \\ &+ \lambda \cdot 3 \cdot m \cdot \tau_2 \cdot \Big(\lambda \cdot \chi \\ &\qquad + \lambda^2 \cdot 2 \cdot \ell \cdot m^2 \cdot (\chi + \chi' + \chi'') \cdot \tau_1 \cdot B \\ &\qquad + \lambda \cdot \chi'\Big),\end{aligned}$$
where we have used that $\|\mathbf{K}_f\| \leq \lambda \cdot \tau_2$ and that $\mathbf{e}'_{f,\mathbf{x}}$ is a vector of length $3 \cdot m$. Since
$$q \geq \mathsf{poly}(\lambda, m, \ell) \cdot \tau_2 \cdot \tau_1 \cdot B \cdot (\chi + \chi' + \chi'')$$
the theorem follows. □

**Theorem 3 (Security).** *Let $\Pi$ be the* KP-ABE *construction described in Section 3, with parameters set as in Eq. (3), and* F *a PRF. Then, for any semi-adaptive adversary $\mathcal{A}$ that runs is time $T = T(\lambda)$, there exists adversaries $\mathcal{B}_0$, $\mathcal{B}_1$ and $\mathcal{B}_2$ against PRF-security,* $\mathsf{LWE}_{n,m,\chi,q}$*, and* $\mathsf{LWE}_{n,m+\lambda,\chi,q}$ *respectively, such that*
$$\mathsf{Adv}^{\textit{sa-ABE}}_{\mathcal{A},\Pi}(\lambda) \leq T \cdot \left(\mathsf{Adv}^{\textit{PRF}}_{\mathcal{B}_0,\mathsf{F}}(\lambda) + \mathsf{Adv}^{\mathsf{LWE}_{n,m,\chi,q}}_{\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{LWE}_{n,m+\lambda,\chi,q}}_{\mathcal{B}_2}(\lambda) + \mathsf{negl}(\lambda)\right).$$

*Proof.* Consider the following sequence of hybrids, summarized in Fig. 2 and Fig. 3. Let $\mathsf{Adv}_i(\mathcal{A})$ denote the advantage of $\mathcal{A}$ in hybrid $\mathsf{H}_i$. Notice that we can bound the length $\ell$ of the input domain of any function $f$ queried to the KeyGen oracle by $T$, i.e., an adverary $\mathcal{A}$ running in time $T$ will never obtain $\mathbf{K}_j$ for $j > T$.

- $\mathsf{H}_0$: This is identical to the real security game. Therefore
$$\mathsf{Adv}^{\text{sa-ABE}}_{\mathcal{A},\Pi}(\lambda) = \mathsf{Adv}_0(\mathcal{A}).$$
- $\mathsf{H}_1$: This is identical to $\mathsf{H}_0$, except for the fact that the reduction guesses $|\mathbf{x}^*| = \ell^*$ before generating the public parameters. If the guess is not correct, the reduction aborts. Since $\mathcal{A}$ runs in time $T$, one has that $\ell^* \leq T$, so the reduction can guess $\ell^*$ and incur a security loss of $T$. In other words, we have that
$$\mathsf{Adv}_0(\mathcal{A}) \leq T \cdot \mathsf{Adv}_1(\mathcal{A}).$$

| Hybrid | mpk | ct | $\mathsf{sk}_f$ | justification |
|---|---|---|---|---|
| $\mathsf{H}_0$ | $(\mathbf{A}_0, \mathbf{T}_{\mathbf{A}_0}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$<br>$(\mathbf{B}_0, \mathbf{T}_{\mathbf{B}_0}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$<br>$\mathbf{W} \leftarrow \mathbb{Z}_q^{n \times m}$<br>$\mathbf{V} \leftarrow \mathbb{Z}_q^{n \times m}$<br>$\mathbf{B}_1 \leftarrow \mathbb{Z}_q^{n \times m}$<br>$\mathbf{D} \leftarrow \mathbb{Z}_q^{n \times \lambda}$<br>$\mathsf{k} \leftarrow \mathcal{K}$ | $\mathbf{c}_0 \approx \mathbf{s} \cdot \mathbf{A}_0$<br>$\mathbf{c}_{1,j} \approx \mathbf{s}_j \cdot \mathbf{B}_0$<br>$\mathbf{c}_{2,j} \approx \mathbf{s}_j \cdot (\mathbf{W} + j \cdot \mathbf{G}) + \mathbf{s} \cdot \mathbf{G}$<br>$\mathbf{c}_{3,j} \approx \mathbf{s}_j \cdot \mathbf{V} + x_j^* \cdot \mathbf{s} \cdot \mathbf{G}$<br>$\mathbf{c}_4 \approx \mathbf{s} \cdot \mathbf{D} + \boldsymbol{\mu} \cdot \lfloor q/2 \rfloor$<br>$\mathbf{c}_5 \approx \mathbf{s} \cdot (\mathbf{B}_1 - \vert\mathbf{x}^*\vert \cdot \mathbf{G})$ | $\mathbf{T}_{\mathbf{A}_0}, \mathbf{T}_{\mathbf{B}_0}, \mathsf{k}$<br>$\mathbf{K}_j = \begin{bmatrix} \mathbf{Z}_j \\ \mathbf{R}_j \end{bmatrix} \leftarrow \mathsf{SamplePre}\left( \begin{matrix} [\mathbf{B}_0 \vert \mathbf{W} + j \cdot \mathbf{G}], \\ \begin{bmatrix} \mathbf{T}_{\mathbf{B}_0} \\ \mathbf{0}_{m \times m} \end{bmatrix}, \mathbf{V}, \tau_1; \mathsf{F}(\mathsf{k}, j) \end{matrix} \right)$<br>$\mathbf{A}_j = \mathbf{G} \cdot \mathbf{R}_j$<br>$\mathbf{K}_f \leftarrow \mathsf{SamplePre}\left( [\mathbf{A}_0 \vert \mathbf{A}_f \vert \mathbf{B}_1 - \ell \cdot \mathbf{G}], \begin{bmatrix} \mathbf{T}_{\mathbf{A}_0} \\ \mathbf{0}_{m \times m} \\ \mathbf{0}_{m \times m} \end{bmatrix}, \mathbf{D}, \tau_2 \right)$ | |
| $\mathsf{H}_1$ | $\downarrow$ | $\downarrow$ | $\downarrow$ | guess $\vert\mathbf{x}^*\vert = \ell^*$ |
| $\mathsf{H}_2$ | $\downarrow$ | $\downarrow$ | $\mathbf{K}_j = \begin{bmatrix} \mathbf{Z}_j \\ \mathbf{R}_j \end{bmatrix} \leftarrow \mathsf{SamplePre}\left( \begin{matrix} [\mathbf{B}_0 \vert \mathbf{W} + j \cdot \mathbf{G}], \\ \begin{bmatrix} \mathbf{T}_{\mathbf{B}_0} \\ \mathbf{0}_{m \times m} \end{bmatrix}, \mathbf{V}, \tau_1; \mathbf{r}_j \end{matrix} \right)$ | PRF security |
| $\mathsf{H}_3$ | $\downarrow$ | $\mathbf{c}_{1,j}, \mathbf{c}_{2,j} \leftarrow \mathbb{Z}_q^m$<br>$\mathbf{c}_{3,j} \approx [\mathbf{c}_{1,j} \mid \mathbf{c}_{2,j}] \cdot \mathbf{K}_j$<br>$-\mathbf{s} \cdot (\mathbf{A}_j - x_j^* \cdot \mathbf{G})$ | $\downarrow$ | $\mathsf{LWE}_{n,m,\chi,q}$<br>(Fig. 3) |
| $\mathsf{H}_4$ | $\downarrow$ | $\downarrow$ | $\mathbf{R}_j \leftarrow \mathcal{D}_{\mathbb{Z}^{m \times m}, \tau_1}$<br>$\mathbf{Z}_j = \mathsf{SamplePre}\left( \begin{matrix} \mathbf{B}_0, \mathbf{T}_{\mathbf{B}_0}, \\ \mathbf{V} - [\mathbf{W} + j \cdot \mathbf{G}] \cdot \mathbf{R}_j, \tau_1; \mathbf{r}_{j,1} \end{matrix} \right)$<br>$\mathbf{K}_j = \begin{bmatrix} \mathbf{Z}_j \\ \mathbf{R}_j \end{bmatrix}$ | Lemma 3 |
| $\mathsf{H}_5$ | $\downarrow$ | $\downarrow$ | $\mathbf{A}_j \leftarrow \mathbb{Z}_q^{n \times m}$<br>$\mathbf{R}_j = \mathsf{SamplePre}(\mathbf{G}, \mathbf{I}, \mathbf{A}_j, \tau_1; \mathbf{r}_{j,2})$<br>$\mathbf{Z}_j = \mathsf{SamplePre}\left( \begin{matrix} \mathbf{B}_0, \mathbf{T}_{\mathbf{B}_0}, \\ \mathbf{V} - [\mathbf{W} + j \cdot \mathbf{G}] \cdot \mathbf{R}_j, \tau_1; \mathbf{r}_{j,1} \end{matrix} \right)$<br>$\mathbf{K}_j = \begin{bmatrix} \mathbf{Z}_j \\ \mathbf{R}_j \end{bmatrix}$ | Lemma 2<br>(Item 1) |
| $\mathsf{H}_6$ | $\mathbf{B}_1 = \mathbf{A}_0 \cdot \mathbf{U} + \ell^* \cdot \mathbf{G}$ | $\downarrow$ | $\mathbf{A}_j = \mathbf{A}_0 \cdot \mathbf{U}_j + x_j^* \cdot \mathbf{G}$ | LHL |
| $\mathsf{H}_7$ | $\downarrow$ | $\downarrow$ | $\mathbf{K}_f$ using $\begin{cases} \begin{bmatrix} [\mathbf{U}_1 \vert \dots \vert \mathbf{U}_{\ell^*}] \cdot \mathbf{H}_{\mathbf{A}, f, \mathbf{x}^*} \\ \mathbf{I}_m \\ \mathbf{0}_{m \times m} \end{bmatrix} & \text{if } \ell = \ell^* \\ \begin{bmatrix} -\mathbf{U} \\ \mathbf{0}_{m \times m} \\ \mathbf{I}_m \end{bmatrix} & \text{if } \ell \neq \ell^* \end{cases}$ | Lemma 2<br>SamplePre |
| $\mathsf{H}_8$ | $\mathbf{A}_0 \leftarrow \mathbb{Z}_q^{n \times m}$ | $\downarrow$ | $\downarrow$ | Lemma 2<br>TrapGen |
| $\mathsf{H}_9$ | $\downarrow$ | $\mathbf{c}_{3,j} \approx [\mathbf{c}_{1,j} \mid \mathbf{c}_{2,j}] \cdot \mathbf{K}_j$<br>$-\mathbf{c}_0 \cdot \mathbf{U}_j$<br>$\mathbf{c}_5 \approx \mathbf{c}_0 \cdot \mathbf{U}$ | $\downarrow$ | noise flooding |
| $\mathsf{H}_{10}$ | $\downarrow$ | $\mathbf{c}_0 \leftarrow \mathbb{Z}_q^m, \mathbf{c}_4 \leftarrow \mathbb{Z}_q^\lambda$ | $\downarrow$ | $\mathsf{LWE}_{n,m+\lambda,\chi,q}$ |

Fig. 2: Summary of our security proof. $\downarrow$ denotes the same as the previous hybrid. We omit the noise terms in $\mathsf{H}_0$. Starting from $\mathsf{H}_6$, the proof is essentially the same as the BGGHNSVV14 ABE.

- $\mathsf{H}_2$: This is identical to $\mathsf{H}_1$, except for the following modification to KeyGen:
  - for all $j \in [T]$, sample once and for all a random string $\mathbf{r}_j \leftarrow \{0,1\}^{\mathsf{poly}(\lambda)}$,
  - use $\mathbf{r}_i$ as randomness to sample $\mathbf{K}_j$, i.e.

  $$\mathbf{K}_j \leftarrow \mathsf{SamplePre}\left( [\mathbf{B}_0 \vert \mathbf{W} + j \cdot \mathbf{G}], \begin{bmatrix} \mathbf{T}_{\mathbf{B}_0} \\ \mathbf{0}_{m \times m} \end{bmatrix}, \mathbf{V}, \tau_1; \mathbf{r}_j \right).$$

  To show that $\mathsf{H}_1 \approx_c \mathsf{H}_2$, we rely on the PRF security of $\mathsf{F}$. The reduction works as follows:
  - it samples $\mathbf{A}, \mathbf{B}_0, \mathbf{B}_1, \mathbf{W}, \mathbf{V}$ and $\mathbf{D}$ as in $\mathsf{H}_1$,
  - it obtains $\mathbf{x}^*$ from the adversary $\mathcal{A}$,
  - it answers KeyGen queries as in $\mathsf{H}_1$ but using the output $\mathcal{O}(j)$ of its PRF oracle as randomness to sample $\mathbf{K}_j$,
  - whenever the adversary $\mathcal{A}$ produces $(\boldsymbol{\mu}_0, \boldsymbol{\mu}_1)$, it produces the challenge ciphertext $\mathsf{ct}_{\mathbf{x}^*}$ as in $\mathsf{H}_1$.

  Observe that
  - if $\mathcal{O}(\cdot) = \mathsf{F}(\mathsf{k}, \cdot)$ is pseudorandom function instance, the view of the adversary $\mathcal{A}$ is identical to $\mathsf{H}_1$;

| Hybrid | mpk | ct | $\mathsf{sk}_f$ | justification |
|---|---|---|---|---|
| $\mathsf{H}_{3,i,0}$ | $\downarrow$ | $\downarrow$ | $\downarrow$ | $\mathsf{H}_{3,1,0} = \mathsf{H}_2$,<br>$\mathsf{H}_{3,i,0} = \mathsf{H}_{3,i-1,9}, i > 1$ |
| $\mathsf{H}_{3,i,1}$ | $\mathbf{W} = \mathbf{B}_0 \cdot \tilde{\mathbf{W}}_i - i \cdot \mathbf{G}$ | $\downarrow$ | $\downarrow$ | LHL |
| $\mathsf{H}_{3,i,2}$ | $\mathbf{V} = [\mathbf{B}_0 \mid \mathbf{W} + i \cdot \mathbf{G}] \cdot \mathbf{K}_i$ | $\downarrow$ | $\mathbf{Z}_i, \mathbf{R}_i \leftarrow \mathcal{D}_{\mathbb{Z}^{m\times m}, \tau_1}$<br>$\mathbf{K}_i = \begin{bmatrix} \mathbf{Z}_i \\ \mathbf{R}_i \end{bmatrix}$ | Lemma 2<br>(Item 1) |
| $\mathsf{H}_{3,i,3}$ | $\downarrow$ | $\downarrow$ | $\mathbf{K}_j \leftarrow \mathsf{SamplePre}\left( \begin{matrix} [\mathbf{B}_0 \vert \mathbf{W} + j \cdot \mathbf{G}], \\ \begin{bmatrix} \tilde{\mathbf{W}}_j \\ -\mathbf{I}_m \end{bmatrix}, \mathbf{V}, \tau_1; \mathbf{r}_j \end{matrix} \right),\ j \neq i$ | Lemma 2<br>SamplePre |
| $\mathsf{H}_{3,i,4}$ | $\mathbf{B}_0 \leftarrow \mathbb{Z}_q^{n\times m}$ | $\downarrow$ | $\downarrow$ | Lemma 2<br>TrapGen |
| $\mathsf{H}_{3,i,5}$ | $\downarrow$ | $\mathbf{c}_{3,i} = [\mathbf{c}_{1,i} \mid \mathbf{c}_{2,i}] \cdot \mathbf{K}_i$<br>$-\mathbf{s} \cdot (\mathbf{A}_i - x_i^* \cdot \mathbf{G}) + \mathbf{e}'_{3,i}$ | $\downarrow$ | noise flooding |
| $\mathsf{H}_{3,i,6}$ | $\downarrow$ | $\mathbf{c}_{2,i} = \mathbf{c}_{1,i} \cdot \tilde{\mathbf{W}}_i + i \cdot \mathbf{s} \cdot \mathbf{G} + \mathbf{e}'_{2,i}$ | $\downarrow$ | noise flooding |
| $\mathsf{H}_{3,i,7}$ | $\downarrow$ | $\mathbf{c}_{1,i} \leftarrow \mathbb{Z}_q^m$ | $\downarrow$ | $\mathsf{LWE}_{n,m,\chi,q}$ |
| $\mathsf{H}_{3,i,8}$ | $\downarrow$ | $\mathbf{c}_{2,i} \leftarrow \mathbb{Z}_q^m$ | $\downarrow$ | **??**<br>GLHL |
| $\mathsf{H}_{3,i,9}$ | same as $\mathsf{H}_2$ | $\downarrow$ | same as $\mathsf{H}_2$ | $\mathsf{H}_{3,i,4} \approx_s \mathsf{H}_{3,i,0}$ |

Fig. 3: Summary for $\mathsf{H}_2 \approx_c \mathsf{H}_3$. The sequence of hybrid is repeated for all $i \in [\ell^*]$. That is, $\mathsf{H}_2 = \mathsf{H}_{3,1,0} \approx \cdots \approx \mathsf{H}_{3,1,9} = \mathsf{H}_{3,2,0} \approx \cdots \approx \mathsf{H}_{3,\ell^*,9} = \mathsf{H}_3$.

- if $\mathcal{O}(\cdot) = F(\cdot)$ is a truly random function instance, the view of $\mathcal{A}$ is identical to $\mathsf{H}_2$.

We conclude that

$$\mathsf{Adv}_1(\mathcal{A}) \leq \mathsf{Adv}_2(\mathcal{A}) + \mathsf{Adv}^{\mathrm{PRF}}_{\mathcal{B}_0,\mathsf{F}}(\lambda).$$

and in particular, that $\mathsf{H}_1 \approx_c \mathsf{H}_2$.

For $i \in [\ell^*]$:

- $\mathsf{H}_{3,i,1}$: This is the same as previous hybrid, except for the following modification to $\mathbf{W}$ in $\mathsf{mpk}$:
  - sample $\tilde{\mathbf{W}}_i \leftarrow \{1, -1\}^{m\times m}$,
  - set $\mathbf{W} := \mathbf{B}_0 \cdot \tilde{\mathbf{W}}_i - i \cdot \mathbf{G}$.

  Since $\tilde{\mathbf{W}}_i$ is sampled uniformly and $m \geq (n+1) \cdot \log q + 2 \cdot \lambda$, statistical indistinguishability of $\mathsf{H}_{3,i,1}$ from previous hybrid follows from the leftover hash lemma. Notice that, for all $j \in [\ell^*]$, we can now rewrite

  $$\mathbf{W} + j \cdot \mathbf{G} = \mathbf{B}_0 \cdot \tilde{\mathbf{W}}_i + (j - i) \cdot \mathbf{G}$$

- $\mathsf{H}_{3,i,2}$ This is the same as $\mathsf{H}_{3,i,1}$, except for the following modification to $\mathbf{V}$ in $\mathsf{mpk}$ and to $\mathbf{K}_i$ in $\mathsf{KeyGen}$ queries:
  - Parse $\mathbf{r}_i = [\mathbf{r}_{i,1} \vert \mathbf{r}_{i,2}]$ and sample $\mathbf{Z}_i, \mathbf{R}_i \leftarrow \mathcal{D}_{\mathbb{Z}^{m\times m}, \tau_1}$ using as random coins $\mathbf{r}_{i,1}$ and $\mathbf{r}_{i,2}$ respectively,
  - set $\mathbf{K}_i := \begin{bmatrix} \mathbf{Z}_i \\ \mathbf{R}_i \end{bmatrix}$,
  - set $\mathbf{V} := [\mathbf{B}_0 \mid \mathbf{W} + i \cdot \mathbf{G}] \cdot \mathbf{K}_i$.

  By the properties of the $\mathsf{SamplePre}$ algorithm (Lemma 2, Item 1), the distribution of $\mathbf{V}$ and $\mathbf{K}_i$ is statistically indistinguishable between $\mathsf{H}_{3,i,1}$ and $\mathsf{H}_{3,i,2}$. Therefore

  $$\mathsf{Adv}_{3,i,1}(\mathcal{A}) \leq \mathsf{Adv}_{3,i,2}(\mathcal{A}) + \mathsf{negl}(\lambda).$$

- $\mathsf{H}_{3,i,3}$: This is the same as $\mathsf{H}_{3,i,2}$, except for the following modification to $\mathsf{KeyGen}$ queries when $j \neq i$:
  - compute $\mathbf{T} := \begin{bmatrix} \tilde{\mathbf{W}}_i \\ -\mathbf{I}_m \end{bmatrix}$ and observe that

  $$[\mathbf{B}_0 \mid \mathbf{W} + j \cdot \mathbf{G}] \cdot \mathbf{T} = [\mathbf{B}_0 \mid \mathbf{B}_0 \cdot \tilde{\mathbf{W}}_i + (j - i) \cdot \mathbf{G}] \cdot \begin{bmatrix} \tilde{\mathbf{W}}_i \\ -\mathbf{I}_m \end{bmatrix} = (i - j) \cdot \mathbf{G}.$$

- compute

  $$\mathbf{K}_j \leftarrow \mathsf{SamplePre}([\mathbf{B}_0|\mathbf{W} + j \cdot \mathbf{G}], \mathbf{T}, \mathbf{V}, \tau_1; \mathbf{r}_j)$$

  to answer $\mathsf{KeyGen}$ queries. This works as long as

  $$\tau_1 \geq O(m^2) \geq O(m^2 \cdot \|\tilde{\mathbf{W}}\|). \tag{5}$$

Therefore, since $\tau_1$ satisfies such constraint by our choice of parameters, we have that

$$\mathsf{Adv}_{3,i,2}(\mathcal{A}) \leq \mathsf{Adv}_{3,i,3}(\mathcal{A}) + \mathsf{negl}(\lambda).$$

Notice that the reduction does not use $\mathbf{T}_{\mathbf{B}_0}$ anymore.

– $\mathsf{H}_{3,i,4}$: This is the same as $\mathsf{H}_{3,i,3}$, except for the following modification to $\mathbf{B}_0$ in $\mathsf{mpk}$:
  - sample $\mathbf{B}_0 \leftarrow \mathbb{Z}_q^{n \times m}$ instead of $(\mathbf{B}_0, \mathbf{T}_{\mathbf{B}_0}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$.

  By the properties of the $\mathsf{TrapGen}$ algorithm (Lemma 2), the distribution of $\mathbf{B}_0$ is statistically indistinguishable between $\mathsf{H}_{3,i,3}$ and $\mathsf{H}_{3,i,4}$. Therefore,

  $$\mathsf{Adv}_{3,i,3}(\mathcal{A}) \leq \mathsf{Adv}_{3,i,4}(\mathcal{A}) + \mathsf{negl}(\lambda).$$

– $\mathsf{H}_{3,i,5}$: This is the same as $\mathsf{H}_{3,i,4}$ except for the following modification to $\mathbf{c}_{3,i}$ in the challenge ciphertext:
  - set

    $$\mathbf{c}_{3,i} \coloneqq [\mathbf{c}_{1,i} \mid \mathbf{c}_{2,i}] \cdot \mathbf{K}_i - \mathbf{s} \cdot (\mathbf{A}_i - x_i^* \cdot \mathbf{G}) + \mathbf{e}'_{3,i},$$

    for $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $\mathbf{e}'_{3,1} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\chi''}$.

  First, recall that in $\mathsf{H}_{3,i,4}$, we have

  $$\begin{aligned} \mathbf{c}_{3,i} &= \mathbf{s}_i \cdot \mathbf{V} + x_i^* \cdot \mathbf{s} \cdot \mathbf{G} + \mathbf{e}_{3,i} \\ &= \underbrace{\mathbf{s}_i \cdot [\mathbf{B}_0 \mid \mathbf{W} + i \cdot \mathbf{G}]}_{[\mathbf{c}_{1,i}|\mathbf{c}_{2,i}]-[\mathbf{0}|\mathbf{s}\cdot\mathbf{G}]-[\mathbf{e}_{1,i}|\mathbf{e}_{2,i}]} \cdot \underbrace{\begin{bmatrix} \mathbf{Z}_i \\ \mathbf{R}_i \end{bmatrix}}_{\mathbf{K}_i} + x_i^* \cdot \mathbf{s} \cdot \mathbf{G} + \mathbf{e}_{3,i} \\ &= [\mathbf{c}_{1,i} \mid \mathbf{c}_{2,i}] \cdot \mathbf{K}_i - \mathbf{s} \cdot (\mathbf{A}_i - x_i^* \cdot \mathbf{G}) + \boxed{\mathbf{e}_{3,i} - [\mathbf{e}_{1,i} \mid \mathbf{e}_{2,i}] \cdot \mathbf{K}_i} \bmod q \end{aligned}$$

  where in the last equality we have used that $\mathbf{A}_i = \mathbf{G} \cdot \mathbf{R}_i$ and the boxed term is the term in $\mathsf{H}_{3,i,4}$ that will be modified in $\mathsf{H}_{3,i,5}$. By noise flooding, we have

  $$([\mathbf{e}_{1,i} \mid \mathbf{e}_{2,i}], \mathbf{K}_i, \boxed{\mathbf{e}_{3,i} - [\mathbf{e}_{1,i} \mid \mathbf{e}_{2,i}] \cdot \mathbf{K}_i}) \approx_s ([\mathbf{e}_{1,i} \mid \mathbf{e}_{2,i}], \mathbf{K}_i, \mathbf{e}'_{3,i}),$$

  as long as

  $$\begin{aligned} \chi'' &\geq 2 \cdot m \cdot \chi' \cdot \tau_1 \cdot \lambda^{\omega(1)}, \\ &\geq \|[\mathbf{e}_{1,i} \mid \mathbf{e}_{2,i}] \cdot \mathbf{K}_i\| \cdot \lambda^{\omega(1)}. \end{aligned}$$

  We conclude that

  $$\mathsf{Adv}_{3,i,4}(\mathcal{A}) \leq \mathsf{Adv}_{3,i,5}(\mathcal{A}) + \mathsf{negl}(\lambda).$$

– $\mathsf{H}_{3,i,6}$: This is the same as $\mathsf{H}_{3,i,5}$, except for the following modification to $\mathbf{c}_{2,i}$ in the challenge ciphertext:
  - set

    $$\mathbf{c}_{2,i} \coloneqq \mathbf{c}_{1,i} \cdot \tilde{\mathbf{W}}_i + \mathbf{s} \cdot \mathbf{G} + \mathbf{e}'_{2,i},$$

    for $\mathbf{e}'_{2,i} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\chi'}$ and $\mathbf{s} \leftarrow \mathbb{Z}_q^n$.

First, recall that in $\mathsf{H}_{3,i,5}$, we have

$$\begin{aligned}\mathbf{c}_{2,i} &= \mathbf{s}_i \cdot (\mathbf{W} + i \cdot \mathbf{G}) + \mathbf{s} \cdot \mathbf{G} + \mathbf{e}_{2,i} \\ &= \mathbf{s}_i \cdot (\mathbf{B}_0 \cdot \tilde{\mathbf{W}}_i) + \mathbf{s} \cdot \mathbf{G} + \mathbf{e}_{2,i} \\ &= \underbrace{(\mathbf{s}_i \cdot \mathbf{B}_0 + \mathbf{e}_{1,i})}_{\mathbf{c}_{1,i}} \cdot \tilde{\mathbf{W}}_i + \mathbf{s} \cdot \mathbf{G} + \boxed{\mathbf{e}_{2,i} - \mathbf{e}_{1,i} \cdot \tilde{\mathbf{W}}_i} \bmod q\end{aligned}$$

where the boxed term is the term in $\mathsf{H}_{3,i,5}$ that will be modified in $\mathsf{H}_{3,i,6}$. By noise flooding, we have

$$\left(\mathbf{e}_{1,i}, \tilde{\mathbf{W}}_i, \boxed{\mathbf{e}_{2,i} - \mathbf{e}_{1,i} \cdot \tilde{\mathbf{W}}_i}\right) \approx_s \left(\mathbf{e}_{1,i}, \tilde{\mathbf{W}}_i, \mathbf{e}'_{2,i}\right),$$

as long as

$$\begin{aligned}\chi' &\geq m \cdot \chi \cdot \lambda^{\omega(1)} \\ &\geq \|\mathbf{e}_{1,i} \cdot \tilde{\mathbf{W}}_i\| \cdot \lambda^{\omega(1)}.\end{aligned}$$

We conclude that

$$\mathsf{Adv}_{3,i,5}(\mathcal{A}) \leq \mathsf{Adv}_{3,i,6}(\mathcal{A}) + \mathsf{negl}(\lambda).$$

– $\mathsf{H}_{3,i,7}$: This is the same as $\mathsf{H}_{3,i,6}$, except for the following modification to $\mathbf{c}_{1,i}$ in the challenge ciphertext:
  - sample
$$\mathbf{c}_{1,i} \leftarrow \mathbb{Z}_q^m.$$

Recall that in $\mathsf{H}_{3,i,6}$, we have

$$\mathbf{c}_{1,i} = \mathbf{s}_i \cdot \mathbf{B}_0 + \mathbf{e}_{1,i},$$

where $\mathbf{s}_i \leftarrow \mathbb{Z}_q^n$, $\mathbf{e}_{1,i} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\chi}$. To show that $\mathsf{H}_{3,i,6} \approx_c \mathsf{H}_{3,i,7}$, we rely on $\mathsf{LWE}_{n,m,\chi,q}$. The reduction works as follows:

- it parses $\mathbf{B} = \mathbf{B}_0 \in \mathbb{Z}_q^{n \times m}$ and $\tilde{\mathbf{c}} = \mathbf{c}_{1,i} \in \mathbb{Z}_q^m$ from the $\mathsf{LWE}_{n,m,\chi,q}$ instance,
- it samples $\tilde{\mathbf{W}}_i \leftarrow \{0,1\}^{m \times m}$ and computes $\mathbf{A}_0, \mathbf{W}, \mathbf{V}, \mathbf{D}$ as in $\mathsf{H}_{3,i,6}$, while using $\mathbf{B}_0$ obtained from the LWE instance,
- it receives $\mathbf{x}^*$ from the adversary $\mathcal{A}$,
- it answers $\mathsf{KeyGen}$ queries using $\mathbf{T}$ (which can be computed from $\tilde{\mathbf{W}}_i$) as in $\mathsf{H}_{3,i,6}$,
- whenever the adversary $\mathcal{A}$ outputs $(\boldsymbol{\mu}_0, \boldsymbol{\mu}_1)$, it samples

$$\begin{gathered} b \leftarrow \{0,1\}, \mathbf{s} \leftarrow \mathbb{Z}_q^n, \mathbf{e}_0 \leftarrow \mathcal{D}_{\mathbb{Z}^m,\chi}, \mathbf{e}_4 \leftarrow \mathcal{D}_{\mathbb{Z}^\lambda,\chi}, \mathbf{e}_5 \leftarrow \mathcal{D}_{\mathbb{Z}^\lambda,\chi'} \\ \mathbf{c}_{1,j} \leftarrow \mathbb{Z}_q^m, \mathbf{c}_{2,j} \leftarrow \mathbb{Z}_q^m \quad \text{for } j \in [i-1], \\ \mathbf{e}'_{2,i} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\chi'}, \mathbf{e}_{3,i} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\chi''}, \text{ and} \\ \mathbf{s}_j \leftarrow \mathbb{Z}_q^n, \mathbf{e}_{1,j} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\chi}, \mathbf{e}_{2,j} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\chi'}, \mathbf{e}_{3,j} \leftarrow \mathcal{D}_{\mathbb{Z}^m,\chi''} \text{ for } j \in [i+1:\ell^*] \end{gathered}$$

  and outputs

$$\mathsf{ct} = \begin{pmatrix} \mathbf{s} \cdot \mathbf{A}_0 + \mathbf{e}_0 \\ \{\mathbf{c}_{1,j}\}_{j\in[i]}, \quad \{\mathbf{s}_j \cdot \mathbf{B}_0 + \mathbf{e}_{1,j}\}_{j\in[i+1:\ell^*]} \\ \{\mathbf{c}_{2,j}\}_{j\in[i-1]}, \quad \mathbf{c}_{1,i} \cdot \tilde{\mathbf{W}}_i + \mathbf{s} \cdot \mathbf{G} + \mathbf{e}'_{2,i}, \quad \{\mathbf{s}_j \cdot (\mathbf{W} + j \cdot \mathbf{G}) + \mathbf{s} \cdot \mathbf{G} + \mathbf{e}_{2,j}\}_{j\in[i+1:\ell^*]} \\ \{[\mathbf{c}_{1,j} \mid \mathbf{c}_{2,j}] \cdot \mathbf{K}_j - \mathbf{s} \cdot (\mathbf{A}_i - x_i^* \cdot \mathbf{G}) + \mathbf{e}'_{3,i}\}_{j\in[i]}, \quad \{\mathbf{s}_j \cdot \mathbf{V} + x_i^* \cdot \mathbf{s} \cdot \mathbf{G} + \mathbf{e}_{3,j}\}_{j\in[i+1:\ell^*]} \\ \mathbf{s} \cdot \mathbf{D} + \boldsymbol{\mu} \cdot \lfloor q/2 \rfloor + \mathbf{e}_4 \\ \mathbf{s} \cdot (\mathbf{B}_1 - |\mathbf{x}^*| \cdot \mathbf{G}) + \mathbf{e}_5 \end{pmatrix}$$

$$\mathsf{ct} = \begin{pmatrix} \mathbf{s} \cdot \mathbf{A}_0 + \mathbf{e}_0 \\ \{\mathbf{c}_{1,j}\}_{j\in[i]}, \quad \{\mathbf{s}_j \cdot \mathbf{B}_0 + \mathbf{e}_{1,j}\}_{j\in[i+1:\ell^*]} \\ \{\mathbf{c}_{2,j}\}_{j\in[i-1]}, \quad \mathbf{c}_{1,i} \cdot \tilde{\mathbf{W}}_i + \mathbf{s} \cdot \mathbf{G} + \mathbf{e}'_{2,i}, \quad \{\mathbf{s}_j \cdot (\mathbf{W} + j \cdot \mathbf{G}) + \mathbf{s} \cdot \mathbf{G} + \mathbf{e}_{2,j}\}_{j\in[i+1:\ell^*]} \\ \{[\mathbf{c}_{1,j} \mid \mathbf{c}_{2,j}] \cdot \mathbf{K}_j - \mathbf{s} \cdot (\mathbf{A}_i - x_i^* \cdot \mathbf{G}) + \mathbf{e}'_{3,i}\}_{j\in[i]}, \quad \{\mathbf{s}_j \cdot \mathbf{V} + x_i^* \cdot \mathbf{s} \cdot \mathbf{G} + \mathbf{e}_{3,j}\}_{j\in[i+1:\ell^*]} \\ \mathbf{s} \cdot \mathbf{D} + \boldsymbol{\mu} \cdot \lfloor q/2 \rfloor + \mathbf{e}_4 \\ \mathbf{s} \cdot (\mathbf{B}_1 - |\mathbf{x}^*| \cdot \mathbf{G}) + \mathbf{e}_5 \end{pmatrix}$$

Observe that

- if $(\mathbf{B}, \tilde{\mathbf{c}})$ is a structured $\mathsf{LWE}_{n,m,\chi,q}$ instance, the view of the adversary $\mathcal{A}$ is identical to $\mathsf{H}_{3,i,6}$;
- if $(\mathbf{B}, \tilde{\mathbf{c}})$ is a uniform random instance, the view of $\mathcal{A}$ is identical to $\mathsf{H}_{3,i,7}$.

We conclude that

$$\mathsf{Adv}_{3,i,6}(\mathcal{A}) \leq \mathsf{Adv}_{3,i,7}(\mathcal{A}) + \mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{LWE}_{n,m,\chi,q}}(\lambda).$$

– $\mathsf{H}_{3,i,8}$: This is the same as $\mathsf{H}_{3,i,7}$, except for the following modification to $\mathbf{c}_{2,i}$ in the challenge ciphertext:
- sample
$$\mathbf{c}_{2,i} \leftarrow \mathbb{Z}_q^m.$$

Recall that in $\mathsf{H}_{3,i,7}$, we have

$$\mathbf{c}_{2,i} = \mathbf{c}_{1,i} \cdot \tilde{\mathbf{W}}_i + \mathbf{s} \cdot \mathbf{G} + \mathbf{e}'_{2,i}.$$

Since $\mathbf{c}_{1,i}$ is uniform random in $\mathbb{Z}_q^m$ in $\mathsf{H}_{3,i,7}$, $\tilde{\mathbf{W}}_i$ is sampled uniformly and $m \geq (n+1) \cdot \log q + 2 \cdot \lambda + \omega(\log n)$, indistinguishability ($\mathsf{H}_{3,i,7} \approx_s \mathsf{H}_{3,i,8}$) follows from the generalized leftover hash lemma (the adversary's view includes $\mathbf{W} = \mathbf{B}_0 \cdot \tilde{W}_i - i \cdot \mathbf{G}$), that is

$$\mathsf{Adv}_{3,i,7}(\mathcal{A}) \leq \mathsf{Adv}_{3,i,8}(\mathcal{A}) + \mathsf{negl}(\lambda).$$

– $\mathsf{H}_{3,i,9}$: This is the same as $\mathsf{H}_{3,i,8}$, except for the following modification to $\mathbf{B}_0$ in $\mathsf{mpk}$ and to the $\mathsf{KeyGen}$ queries:
- sample $(\mathbf{B}_0, \mathbf{T}_{\mathbf{B}_0}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$, instead of $\mathbf{B}_0 \leftarrow \mathbb{Z}_q^{n \times m}$,
- compute
$$\mathbf{K}_j \leftarrow \mathsf{SamplePre}\left([\mathbf{B}_0 | \mathbf{W} + j \cdot \mathbf{G}], \begin{bmatrix} \mathbf{T}_{\mathbf{B}_0} \\ \mathbf{0}_{m \times m} \end{bmatrix}, \mathbf{V}, \tau_1; \mathbf{r}_j\right),$$
to answer $\mathsf{KeyGen}$ queries.

By the properties of the $\mathsf{TrapGen}$ algorithm (Lemma 2) and that of the $\mathsf{SamplePre}$ algorithm, the distribution of $\mathbf{B}_0$ and that of answers to the $\mathsf{KeyGen}$ queries are statistically indistinguishable between $\mathsf{H}_{3,i,8}$ and $\mathsf{H}_{3,i,9}$. Therefore,

$$\mathsf{Adv}_{3,i,8}(\mathcal{A}) \leq \mathsf{Adv}_{3,i,9}(\mathcal{A}) + \mathsf{negl}(\lambda).$$

– $\mathsf{H}_4$: This is the same as $\mathsf{H}_{3,\ell,9}$, except for the following modification to $\mathbf{K}_j$ for all $j \in [T]$, and to the relative $\mathsf{KeyGen}$ queries:
- sample, once and for all key queries, $\mathbf{R}_j \leftarrow \mathcal{D}_{\mathbb{Z}^{m \times m}, \tau_1}$,
- compute
$$\mathbf{Z}_j \leftarrow \mathsf{SamplePre}(\mathbf{B}_0, \mathbf{T}_{\mathbf{B}_0}, \mathbf{V} - [\mathbf{W} + j \cdot \mathbf{G}] \cdot \mathbf{R}_j, \tau_1; \mathbf{r}_{j,1}),$$
- set $\mathbf{K}_j := \begin{bmatrix} \mathbf{Z}_j \\ \mathbf{R}_j \end{bmatrix}$.

By the properties of the $\mathsf{SamplePre}$ algorithm and Lemma 3, the distribution of $\{\mathbf{K}_j\}_{j \in [T]}$ is statistically indistinguishable between $\mathsf{H}_{3,\ell,9}$ and $\mathsf{H}_4$. Therefore,

$$\mathsf{Adv}_{3,\ell,9}(\mathcal{A}) \leq \mathsf{Adv}_4(\mathcal{A}) + \mathsf{negl}(\lambda).$$

– $\mathsf{H}_5$: This is the same as $\mathsf{H}_4$, except for the following modification to $\mathbf{A}_j, \mathbf{K}_j$ for all $j \in [T]$, and to the relative $\mathsf{KeyGen}$ queries:
- sample $\mathbf{A}_j \leftarrow \mathbb{Z}_q^{n \times m}$,
- set $\mathbf{R}_j = \mathsf{SamplePre}(\mathbf{G}, \mathbf{I}, \mathbf{A}_j, \tau_1; \mathbf{r}_{j,2})$,
- compute
$$\mathbf{Z}_j \leftarrow \mathsf{SamplePre}(\mathbf{B}_0, \mathbf{T}_{\mathbf{B}_0}, \mathbf{V} - [\mathbf{W} + j \cdot \mathbf{G}] \cdot \mathbf{R}_j, \tau_1; \mathbf{r}_{j,1}),$$
- set $\mathbf{K}_j := \begin{bmatrix} \mathbf{Z}_j \\ \mathbf{R}_j \end{bmatrix}$.

By the properties of the $\mathsf{SamplePre}$ algorithm (Lemma 2, Item 1), the distribution of $\{\mathbf{A}_j, \mathbf{K}_j\}_{j \in [T]}$ is statistically indistinguishable between $\mathsf{H}_4$ and $\mathsf{H}_5$. Therefore,

$$\mathsf{Adv}_4(\mathcal{A}) \leq \mathsf{Adv}_5(\mathcal{A}) + \mathsf{negl}(\lambda).$$

- $\mathsf{H}_6$: This is the same as $\mathsf{H}_5$, except for the following modification to $\mathbf{B}_1$ and $\mathbf{A}_j$ for all $j \in [\ell^*]$:
  - sample $\mathbf{U} \leftarrow \{0,1\}^{m \times m}$ and $\mathbf{U}_j \leftarrow \{0,1\}^{m \times m}$ for $j \in [\ell^*]$,
  - set $\mathbf{B}_1 = \mathbf{A}_0 \cdot \mathbf{U} + \ell^* \cdot \mathbf{G}$ and $\mathbf{A}_j := \mathbf{A}_0 \cdot \mathbf{U}_j + x_j^* \cdot \mathbf{G}$ for $j \in [\ell^*]$.

  Since $\mathbf{U}, \mathbf{U}_j$ are sampled uniformly and $m \geq (n+1) \cdot \log q + 2 \cdot \lambda$, indistinguishability ($\mathsf{H}_5 \approx_s \mathsf{H}_6$) follows from the leftover hash lemma, that is

$$\mathsf{Adv}_5(\mathcal{A}) \leq \mathsf{Adv}_6(\mathcal{A}) + \mathsf{negl}(\lambda).$$

- $\mathsf{H}_7$: This is the same as $\mathsf{H}_6$, except for the following modification to change answers to KeyGen queries
  - recall the key equation

$$(\mathbf{A} - \mathbf{x} \otimes \mathbf{G}) \cdot \mathbf{H}_{\mathbf{A},f,\mathbf{x}} = \mathbf{A}_f - f(\mathbf{x}) \cdot \mathbf{G} \mod q,$$

    and that a valid adversary can only make KeyGen queries for functions $f$ for which $f(\mathbf{x}^*) = 1$, and that $|\mathbf{x}^*| = \ell^*$. Using these facts, for functions $f$ with input length $\ell^*$, one has that

$$\begin{aligned}
\mathbf{A}_f &= (\mathbf{A} - \mathbf{x}^* \otimes \mathbf{G}) \cdot \mathbf{H}_{\mathbf{A},f,\mathbf{x}^*} + f(\mathbf{x}^*) \cdot \mathbf{G} \\
&= ([\mathbf{A}_1 | \ldots | \mathbf{A}_{\ell^*}] - \mathbf{x}^* \otimes \mathbf{G}) \cdot \mathbf{H}_{\mathbf{A},f,\mathbf{x}^*} + f(\mathbf{x}^*) \cdot \mathbf{G} \\
&= (\mathbf{A}_0 \cdot [\mathbf{U}_1 | \ldots | \mathbf{U}_{\ell^*}] + \mathbf{x}^* \otimes \mathbf{G} - \mathbf{x}^* \otimes \mathbf{G}) \cdot \mathbf{H}_{\mathbf{A},f,\mathbf{x}^*} + f(\mathbf{x}^*) \cdot \mathbf{G} \\
&= \mathbf{A}_0 \cdot \underbrace{[\mathbf{U}_1 | \ldots | \mathbf{U}_{\ell^*}] \cdot \mathbf{H}_{\mathbf{A},f,\mathbf{x}^*}}_{\mathbf{U}_f} + f(\mathbf{x}^*) \cdot \mathbf{G} \\
&= \mathbf{A}_0 \cdot \mathbf{U}_f + \mathbf{G} \bmod q,
\end{aligned}$$

    where in the second equality we have used the definition of $\mathbf{A}_j$ for $j \in [\ell^*]$.
  - compute $\mathbf{T}_f := \begin{bmatrix} -\mathbf{U}_f \\ \mathbf{I}_m \\ \mathbf{0}_{m \times m} \end{bmatrix}$ and observe that $[\mathbf{A}_0 | \mathbf{A}_f | \mathbf{B}_1 - \ell^* \cdot \mathbf{G}] \cdot \mathbf{T}_f = \mathbf{G}$.
  - for functions $f$ whose input length is $\ell^*$, compute

$$\mathbf{K}_f \leftarrow \mathsf{SamplePre}\left([\mathbf{A}_0 | \mathbf{A}_f | \mathbf{B}_1 - \ell \cdot \mathbf{G}], \mathbf{T}_f, \mathbf{D}, \tau_2\right)$$

    to answer KeyGen queries.
  - for functions $f$ with input length $\ell \neq \ell^*$, observe that

$$\begin{aligned}
[\mathbf{A}_0 | \mathbf{A}_f | \mathbf{B}_1 - \ell \cdot \mathbf{G}] \cdot \begin{bmatrix} -\mathbf{U} \\ \mathbf{0}_{m \times m} \\ \mathbf{I}_m \end{bmatrix} &= [\mathbf{A}_0 | \mathbf{A}_f | \mathbf{A}_0 \cdot \mathbf{U} + \ell^* \cdot \mathbf{G} - \ell \cdot \mathbf{G}] \cdot \begin{bmatrix} -\mathbf{U} \\ \mathbf{0}_{m \times m} \\ \mathbf{I}_m \end{bmatrix} \\
&= [\mathbf{A}_0 | \mathbf{A}_f | \mathbf{A}_0 \cdot \mathbf{U} + (\ell^* - \ell) \cdot \mathbf{G}] \cdot \begin{bmatrix} -\mathbf{U} \\ \mathbf{0}_{m \times m} \\ \mathbf{I}_m \end{bmatrix} \\
&= (\underbrace{\ell^* - \ell}_{\neq 0}) \cdot \mathbf{G} \bmod q.
\end{aligned}$$

  - for such functions (with input length $\ell \neq \ell^*$), compute

$$\mathbf{K}_f \leftarrow \mathsf{SamplePre}\left([\mathbf{A}_0 | \mathbf{A}_f | \mathbf{B}_1 - \ell \cdot \mathbf{G}], \begin{bmatrix} -\mathbf{U} \\ \mathbf{0}_{m \times m} \\ \mathbf{I}_m \end{bmatrix}, \mathbf{D}, \tau_2\right)$$

    to answer KeyGen queries.
  - these procedures work as long as

$$\begin{aligned}
\tau_2 &\geq m^3 \cdot \ell^* \cdot B \\
&\geq m^2 \cdot m \cdot \ell^* \cdot B \\
&\geq O(m^2 \cdot (\|\mathbf{U}_f\| + \|\mathbf{U}\|))
\end{aligned} \tag{6}$$

By properties of the $\mathsf{SamplePre}$ algorithm,

$$\mathsf{Adv}_6(\mathcal{A}) \leq \mathsf{Adv}_7(\mathcal{A}) + \mathsf{negl}(\lambda).$$

Notice that the reduction does not use $\mathbf{T}_{\mathbf{A}_0}$ anymore.

- $\mathsf{H}_8$: This is the same as $\mathsf{H}_7$, except for the following modification to $\mathbf{A}_0$ in $\mathsf{mpk}$:
  - sample $\mathbf{A}_0 \leftarrow \mathbb{Z}_q^{n \times m}$ instead of $(\mathbf{A}_0, \mathbf{T}_{\mathbf{A}_0}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$.

  By the properties of the $\mathsf{TrapGen}$ algorithm (Lemma 2), the distribution of $\mathbf{A}_0$ is statistically indistinguishable between $\mathsf{H}_7$ and $\mathsf{H}_8$. Therefore

  $$\mathsf{Adv}_7(\mathcal{A}) \leq \mathsf{Adv}_8(\mathcal{A}) + \mathsf{negl}(\lambda).$$

- $\mathsf{H}_9$: This is the same as $\mathsf{H}_8$ except for the following modification to $\mathbf{c}_{3,j}$ and $\mathbf{c}_5$ in the challenge ciphertext:
  - set

  $$\begin{aligned} \mathbf{c}_{3,j} &:= [\mathbf{c}_{1,j} \mid \mathbf{c}_{2,j}] \cdot \mathbf{K}_j - \mathbf{c}_0 \cdot \mathbf{U}_j + \mathbf{e}''_{3,j}, \text{ and} \\ \mathbf{c}_5 &:= \mathbf{c}_0 \cdot \mathbf{U} + \mathbf{e}'_5, \end{aligned}$$

  for $\mathbf{e}_{3,j} \leftarrow \mathcal{D}_{\mathbb{Z}^m, \chi''}$, $\mathbf{e}'_5 \leftarrow \mathcal{D}_{\mathbb{Z}^m, \chi'}$.

  First, recall that in $\mathsf{H}_8$, we have

  $$\begin{aligned} \mathbf{c}_{3,j} &= [\mathbf{c}_{1,j} \mid \mathbf{c}_{2,j}] \cdot \mathbf{K}_j - \underbrace{\mathbf{s} \cdot \mathbf{A}_0 \cdot \mathbf{U}_j}_{\mathbf{c}_0 \cdot \mathbf{U}_j - \mathbf{e}_0 \cdot \mathbf{U}_j} + \mathbf{e}'_{3,j} \\ &= [\mathbf{c}_{1,j} \mid \mathbf{c}_{2,j}] \cdot \mathbf{K}_j - \mathbf{c}_0 \cdot \mathbf{U}_j + \boxed{\mathbf{e}_0 \cdot \mathbf{U}_j + \mathbf{e}'_{3,j}} \bmod q, \end{aligned}$$

  and

  $$\begin{aligned} \mathbf{c}_5 &= \underbrace{\mathbf{s} \cdot \mathbf{A}_0 \cdot \mathbf{U}}_{\mathbf{c}_0 \cdot \mathbf{U} - \mathbf{e}_0 \cdot \mathbf{U}} + \mathbf{e}_5 \\ &= \mathbf{c}_0 \cdot \mathbf{U} + \boxed{\mathbf{e}_5 - \mathbf{e}_0 \cdot \mathbf{U}} \bmod q. \end{aligned}$$

  where the boxed terms are the term in $\mathsf{H}_8$ that will be modified in $\mathsf{H}_9$. By noise flooding, we have

  $$\left(\mathbf{e}'_{3,j}, \mathbf{U}_j, \boxed{\mathbf{e}_0 \cdot \mathbf{U}_j + \mathbf{e}'_{3,j}}\right) \approx_s \left(\mathbf{e}'_{3,j}, \mathbf{U}_j, \mathbf{e}''_{3,j}\right),$$

  and

  $$\left(\mathbf{e}_0, \mathbf{U}, \boxed{\mathbf{e}_5 - \mathbf{e}_0 \cdot \mathbf{U}}\right) \approx_s \left(\mathbf{e}_0, \mathbf{U}, \mathbf{e}'_5\right),$$

  as long as

  $$\begin{aligned} \chi'' &\geq m \cdot \chi \cdot \lambda^{\omega(1)}, \\ &\geq \|\mathbf{e}_0 \cdot \mathbf{U}_j\| \cdot \lambda^{\omega(1)}, \end{aligned}$$

  and

  $$\begin{aligned} \chi' &\geq m \cdot \chi \cdot \lambda^{\omega(1)}, \\ &\geq \|\mathbf{e}_0 \cdot \mathbf{U}\| \cdot \lambda^{\omega(1)}, \end{aligned}$$

  respectively. We conclude that

  $$\mathsf{Adv}_8(\mathcal{A}) \leq \mathsf{Adv}_9(\mathcal{A}) + \mathsf{negl}(\lambda).$$

- $\mathsf{H}_{10}$: This is the same as $\mathsf{H}_9$, except for the following modification to $\mathbf{c}_0$ and $\mathbf{c}_4$ in the challenge ciphertext:

- sample

$$\mathbf{c}_0 \leftarrow \mathbb{Z}_q^m, \mathbf{c}_4 \leftarrow \mathbb{Z}_q^\lambda$$

Recall that in $\mathsf{H}_9$, we have

$$\mathbf{c}_0 = \mathbf{s} \cdot \mathbf{A}_0 + \mathbf{e}_0, \quad \mathbf{c}_4 = \mathbf{s} \cdot \mathbf{D} + \boldsymbol{\mu} \cdot \lfloor q/2 \rfloor + \mathbf{e}_4$$

where $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, $\mathbf{e}_0 \leftarrow \mathcal{D}_{\mathbb{Z}^m,\chi}$, and $\mathbf{e}_4 \leftarrow \mathcal{D}_{\mathbb{Z}^m,\chi}$. To show that $\mathsf{H}_9 \approx_c \mathsf{H}_{10}$, we rely on $\mathsf{LWE}_{n,m+\lambda,\chi,q}$. The reduction works as follows:

- it parses $\mathbf{B} = [\mathbf{A}_0 | \mathbf{D}] \in \mathbb{Z}_q^{n\times(m+\lambda)}$ and $\tilde{\mathbf{c}} = [\mathbf{c}_0 | \mathbf{c}_4] \in \mathbb{Z}_q^{m+\lambda}$ from the $\mathsf{LWE}_{n,m+\lambda,\chi,q}$ instance,
- it samples $\mathbf{B}_0, \mathbf{W}, \mathbf{V}, \mathbf{U}$ as in $\mathsf{H}_7$, while using $\mathbf{A}_0, \mathbf{D}$ obtained from the LWE instance,
- it receives $\mathbf{x}^*$ from the adversary $\mathcal{A}$,
- it samples $\{\mathbf{U}_j\}_{j\in[\ell^*]}$ and implicitly sets $\{\mathbf{A}_j\}_{j\in[\ell^*]}$ as in $\mathsf{H}_9$,
- it answers $\mathsf{KeyGen}$ queries using $\mathbf{U}_f$ or $\mathbf{U}$ as in $\mathsf{H}_9$,
- whenever the adversary $\mathcal{A}$ outputs $(\boldsymbol{\mu}_0, \boldsymbol{\mu}_1)$, it samples

$$\begin{gathered} b \leftarrow \{0,1\}, \mathbf{e}_5' \leftarrow \mathcal{D}_{\mathbb{Z}^\lambda,\chi'} \\ \mathbf{c}_{1,j} \leftarrow \mathbb{Z}_q^m, \mathbf{c}_{2,j} \leftarrow \mathbb{Z}_q^m \quad \text{for } j \in [\ell^*], \text{ and} \\ \mathbf{e}_{3,j}'' \leftarrow \mathcal{D}_{\mathbb{Z}^m,\chi''} \text{ for } j \in [\ell^*] \end{gathered}$$

and outputs

$$\mathsf{ct} = \begin{pmatrix} \mathbf{c}_0 \\ \{\mathbf{c}_{1,j}\}_{j\in[\ell^*]}, \\ \{\mathbf{c}_{2,j}\}_{j\in[\ell^*]}, \\ \{[\mathbf{c}_{1,j} \mid \mathbf{c}_{2,j}] \cdot \mathbf{K}_j - \mathbf{c}_0 \cdot \mathbf{U}_j + \mathbf{e}_{3,i}''\}_{j\in[\ell^*]} \\ \mathbf{c}_4 + \boldsymbol{\mu}_b \cdot \lfloor q/2 \rfloor \\ \mathbf{c}_0 \cdot \mathbf{U} + \mathbf{e}_5' \end{pmatrix}$$

Observe that

- if $(\mathbf{B}, \tilde{\mathbf{c}})$ is a structured $\mathsf{LWE}_{n,m+\lambda,\chi,q}$ instance, the view of the adversary $\mathcal{A}$ is identical to $\mathsf{H}_9$;
- if $(\mathbf{B}, \tilde{\mathbf{c}})$ is a uniform random instance, the view of $\mathcal{A}$ is identical to $\mathsf{H}_{10}$.

We conclude that

$$\mathsf{Adv}_9(\mathcal{A}) \le \mathsf{Adv}_{10}(\mathcal{A}) + \mathsf{Adv}_{\mathcal{B}_2}^{\mathsf{LWE}_{n,m+\lambda,\chi,q}}(\lambda).$$

Putting everything together, we obtain

$$\mathsf{Adv}_{\mathcal{A},\Pi}^{\text{sa-ABE}}(\lambda) \le T \cdot \left(\mathsf{Adv}_{\mathcal{B}_0,\mathsf{F}}^{\mathrm{PRF}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{LWE}_{n,m,\chi,q}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_2}^{\mathsf{LWE}_{n,m+\lambda,\chi,q}}(\lambda) + \mathsf{negl}(\lambda)\right),$$

as claimed. □

## 4 Unbounded Inner Product Predicate Encryption

We consider inner product predicate encrytion, where ciphertexts are associated with $\mathbf{x} \in \mathbb{Z}_q^\ell$, keys with $\mathbf{y} \in \mathbb{Z}_q^\ell$, and decryption is possible iff $\mathbf{x} \cdot \mathbf{y}^\top = 0$, where $q$ is prime.

**Predicate Encryption.** The syntax is exactly the same as ABE except Dec only gets $(\mathsf{mpk}, \mathsf{sk}_f, f, \mathsf{ct}_\mathbf{x})$ but not $\mathbf{x}$. Correctness is defined analogously. For security, we require weak attribute-hiding with semi-adaptive security as captured by the following advantage function:

$$\mathsf{Adv}_{\mathcal{A},\Pi}^{\mathrm{PE}}(\lambda) := \Pr\left[ b = b' : \begin{array}{c} (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{F}) \\ \mathbf{x}_0^*, \mathbf{x}_1^*, \leftarrow \mathcal{A}(1^\lambda, \mathsf{mpk}) \\ (\boldsymbol{\mu}_0, \boldsymbol{\mu}_1) \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{mpk},\mathsf{msk},\cdot)}(\mathsf{mpk}) \\ b \leftarrow \{0,1\}; \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, \mathbf{x}_b^*, \boldsymbol{\mu}_b) \\ b' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{mpk},\mathsf{msk},\cdot)}(\mathsf{ct}) \end{array} \right] - \frac{1}{2},$$

with the restriction that $|\mathbf{x}_0^*| = |\mathbf{x}_1^*|$ and all queries $f : \{0,1\}^\ell \to \{0,1\}$ that $\mathcal{A}$ sent to $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \cdot)$ satisfy either (i) $\ell \neq |\mathbf{x}_0^*|$ or (ii) $f(\mathbf{x}_0^*) \neq 0$ and $f(\mathbf{x}_1^*) \neq 0$.

**Homomorphic computation on matrices.** Following [AFV11], we have:

$$(\mathbf{A} - \mathbf{x} \otimes \mathbf{G}_{n,q}) \cdot \overbrace{(\mathbf{I}_\ell \otimes \mathbf{G}_{n,q})^{-1}(\mathbf{y}^\top)}^{\mathbf{H}_{\mathbf{A},\mathbf{y}}} = \overbrace{\mathbf{A} \cdot (\mathbf{I}_\ell \otimes \mathbf{G}_{n,q})^{-1}(\mathbf{y}^\top)}^{\mathbf{A}_\mathbf{y}} - \mathbf{x} \cdot \mathbf{y}^\top \otimes \mathbf{G} \tag{7}$$

Observe that computing $\mathbf{H}_{\mathbf{A},\mathbf{y}}$ does not require knowing $\mathbf{x}$.

### 4.1 Our Construction

Our inner product predicate encryption scheme $\Pi'$ is exactly the same as our ABE, with the following modifications:

- In $\mathsf{Enc}$, we have $\mathbf{x} \in \mathbb{Z}_q^\ell, x_j \in \mathbb{Z}_q$ instead of $\mathbf{x} \in \{0,1\}^\ell, x_j \in \{0,1\}$;
- In $\mathsf{KeyGen}$, we replace $\mathbf{A}_f$ in $\mathbf{K}_f$ with $\mathbf{A}_\mathbf{y}$ in $\mathbf{K}_y$, namely

$$\boxed{\mathbf{K}_\mathbf{y}} \leftarrow \mathsf{SamplePre}\left( [\mathbf{A}_0 | \boxed{\mathbf{A}_\mathbf{y}} | \mathbf{B}_1 - \ell \cdot \mathbf{G}], \begin{bmatrix} \mathbf{T}_{\mathbf{A}_0} \\ \mathbf{0}_{m \times m} \\ \mathbf{0}_{m \times m} \end{bmatrix}, \mathbf{D}, \tau_2 \right).$$

- In $\mathsf{Dec}$, we replace $\mathbf{H}_{\mathbf{A},f,\mathbf{x}}$ with $\mathbf{H}_{\mathbf{A},\mathbf{y}}$ and $\mathbf{K}_f$ with $\mathbf{K}_\mathbf{y}$, namely

$$\mathbf{c}_4 - \left[ \mathbf{c}_0 \,\middle|\, \left( \left[ [\mathbf{c}_{1,1}\ \mathbf{c}_{2,1}] \cdot \mathbf{K}_1 \mid \ldots \mid [\mathbf{c}_{1,\ell}\ \mathbf{c}_{2,\ell}] \cdot \mathbf{K}_\ell \right] - [\mathbf{c}_{3,1} \ldots \mathbf{c}_{3,\ell}] \right) \cdot \boxed{\mathbf{H}_{\mathbf{A},\mathbf{y}}} \,\middle|\, \mathbf{c}_5 \right] \cdot \boxed{\mathbf{K}_\mathbf{y}}$$

- We can set the parameters as before, but with $B = 1$ (since $\|\mathbf{H}_{\mathbf{A},\mathbf{y}}\| \leq 1$), which yields:

**Theorem 4 (Correctness).** *Let $\Pi'$ be the inner product predicate encryption scheme just described, with parameters as in Equation (3) and $B = 1$. Then $\Pi'$ is correct.*

Correctness is exactly the same as in the ABE, except we use (7). In particular, in the derivation of correctness we only need to replace Eq. (4) with

$$\begin{aligned} &\left[ \mathbf{c}_0 \,\middle|\, \left( \left[ [\mathbf{c}_{1,1}\ \mathbf{c}_{2,1}] \cdot \mathbf{K}_1 \mid \ldots \mid [\mathbf{c}_{1,\ell}\ \mathbf{c}_{2,\ell}] \cdot \mathbf{K}_\ell \right] - [\mathbf{c}_{3,1} \ldots \mathbf{c}_{3,\ell}] \right) \cdot \mathbf{H}_{\mathbf{A},\mathbf{y}} \,\middle|\, \mathbf{c}_5 \right] \\ &\qquad = \mathbf{s} \cdot [\mathbf{A}_0 \mid \mathbf{A}_\mathbf{y} \mid \mathbf{B}_1 - \ell \cdot \mathbf{G}] + \mathbf{e}'_{f,\mathbf{y}} \bmod q. \end{aligned} \tag{8}$$

### 4.2 Security Proof

We sketch here the main modifications required. The security proof starts out exactly as in our ABE, except

- we replace $\mathbf{A}_f$ with $\mathbf{A}_\mathbf{y}$ and $\mathbf{H}_{\mathbf{A},f,\mathbf{x}^*}$ with $\mathbf{H}_{\mathbf{A},\mathbf{y}}$ and $\mathbf{x}^*$ with $\mathbf{x}_b^*$.
- to show $\mathsf{H}_6 \approx_s \mathsf{H}_7$, we use $\mathbf{x}_b^* \cdot \mathbf{y}^\top \neq 0$, instead of $f(\mathbf{x}^*) \neq 0$.

In addition, we require the following additional games analogous to those in [AFV11]:

- $\mathsf{H}_{11}$: sample $\mathbf{A}_0$ together with a trapdoor $\mathbf{T}_{\mathbf{A}_0}$ via $(\mathbf{A}_0, \mathbf{T}_{\mathbf{A}_0}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$. We have $\mathsf{H}_{10} \approx_s \mathsf{H}_{11}$ by properties of the $\mathsf{TrapGen}$ algorithm.
- $\mathsf{H}_{12}$: sample $\mathbf{K}_\mathbf{y}$ using the trapdoor for $\mathbf{A}_0$:

$$\mathbf{K}_\mathbf{y} \leftarrow \mathsf{SamplePre}\left( [\mathbf{A}_0 | \mathbf{A}_\mathbf{y} | \mathbf{B}_1 - \ell \cdot \mathbf{G}], \begin{bmatrix} \mathbf{T}_{\mathbf{A}_0} \\ \mathbf{0}_{m \times m} \\ \mathbf{0}_{m \times m} \end{bmatrix}, \mathbf{D}, \tau_2 \right)$$

  We have $\mathsf{H}_{11} \approx_s \mathsf{H}_{12}$ by properties of the $\mathsf{SamplePre}$ algorithm.

– $\mathsf{H}_{13}$: replace $\mathbf{A}_j, \mathbf{c}_{3,j}$ with random.[2] We have $\mathsf{H}_{12} \approx_s \mathsf{H}_{13}$ via the leftover hash lemma as follows. First, in $\mathsf{H}_{12}$, $\mathbf{K}_{\mathbf{y}}$ does not leak any additional information about $\mathbf{U}_j$ beyond $\mathbf{A}_1, \ldots, \mathbf{A}_{\ell^*}$. Then, the leftover hash lemma tells us

$$(\mathbf{A}_0, \mathbf{c}_0, \mathbf{c}_0 \cdot \mathbf{U}_1, \ldots, \mathbf{c}_0 \cdot \mathbf{U}_{\ell^*}, \mathbf{A}_0 \cdot \mathbf{U}_0, \ldots, \mathbf{A}_0 \cdot \mathbf{U}_{\ell^*})$$

is statistically close to random. This means:

$$(\mathbf{A}_0, \mathbf{c}_0, \ldots, \underbrace{[\mathbf{c}_{1,j} | \mathbf{c}_{2,j}] \cdot \mathbf{K}_j - \mathbf{c}_0 \cdot \mathbf{U}_j}_{\mathbf{c}_{3,j}}, \ldots, \underbrace{\mathbf{A}_0 \cdot \mathbf{U}_j + x^*_{b,j} \cdot \mathbf{G}}_{\mathbf{A}_j}, \ldots)$$

is statistically close to random.

Finally, observe that in $\mathsf{H}_{13}$, the view of the adversary is statistically independent of the challenges $\mathbf{x}^*_b$. The various hybrids are described in Fig. 4. This result is summarized in the following theorem.

**Theorem 5 (Security).** *Let $\Pi'$ be the inner product predicate encryption scheme described in Section 4.1, with parameters set as in Eq. (3) with $B = 1$, and $\mathsf{F}$ a PRF. Then, for any semi-adaptive adversary $\mathcal{A}$ that runs is time $T = T(\lambda)$, there exists adversaries $\mathcal{B}_0$, $\mathcal{B}_1$ and $\mathcal{B}_2$ against PRF-security, $\mathsf{LWE}_{n,m,\chi,q}$, and $\mathsf{LWE}_{n,m+\lambda,\chi,q}$ respectively, such that*

$$\mathsf{Adv}^{PE}_{\mathcal{A},\Pi}(\lambda) \leq T \cdot \left( \mathsf{Adv}^{PRF}_{\mathcal{B}_0,\mathsf{F}}(\lambda) + \mathsf{Adv}^{\mathsf{LWE}_{n,m,\chi,q}}_{\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{LWE}_{n,m+\lambda,\chi,q}}_{\mathcal{B}_2}(\lambda) + \mathsf{negl}(\lambda) \right).$$

## References

ABB10. Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 553–572. Springer, Heidelberg, May / June 2010. 5, 7

AFV11. Shweta Agrawal, David Mandell Freeman, and Vinod Vaikuntanathan. Functional encryption for inner product predicates from learning with errors. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 21–40. Springer, Heidelberg, December 2011. 2, 20, 22

Agr17. Shweta Agrawal. Stronger security for reusable garbled circuits, general definitions and attacks. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 3–35. Springer, Heidelberg, August 2017. 2, 5

Ajt96. Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th ACM STOC*, pages 99–108. ACM Press, May 1996. 7

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. 1

Att16. Nuttapong Attrapadung. Dual system encryption framework in prime-order groups via computational pair encodings. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 591–623. Springer, Heidelberg, December 2016. 1

BCTW16. Zvika Brakerski, David Cash, Rotem Tsabary, and Hoeteck Wee. Targeted homomorphic attribute-based encryption. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 330–360. Springer, Heidelberg, October / November 2016. 8

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 Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 533–556. Springer, Heidelberg, May 2014. 1, 3, 8

BV16. Zvika Brakerski and Vinod Vaikuntanathan. Circuit-ABE from LWE: Unbounded attributes and semi-adaptive security. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 363–384. Springer, Heidelberg, August 2016. 1, 3

CGKW18. Jie Chen, Junqing Gong, Lucas Kowalczyk, and Hoeteck Wee. Unbounded ABE via bilinear entropy expansion, revisited. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 503–534. Springer, Heidelberg, April / May 2018. 1, 2, 5

[2] Similar argument shows that $\mathbf{c}_5$ is also pseudorandom.

| Hybrid | mpk | ct | sk$_f$ | justification |
|---|---|---|---|---|
| $\mathsf{H}_0$ | $(\mathbf{A}_0, \mathbf{T}_{\mathbf{A}_0}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$<br>$(\mathbf{B}_0, \mathbf{T}_{\mathbf{B}_0}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$<br>$\mathbf{W} \leftarrow \mathbb{Z}_q^{n \times m}$<br>$\mathbf{V} \leftarrow \mathbb{Z}_q^{n \times m}$<br>$\mathbf{B}_1 \leftarrow \mathbb{Z}_q^{n \times m}$<br>$\mathbf{D} \leftarrow \mathbb{Z}_q^{n \times \lambda}$<br>$\mathsf{k} \leftarrow \mathcal{K}$ | $\mathbf{c}_0 \approx \mathbf{s} \cdot \mathbf{A}_0$<br>$\mathbf{c}_{1,j} \approx \mathbf{s}_j \cdot \mathbf{B}_0$<br>$\mathbf{c}_{2,j} \approx \mathbf{s}_j \cdot (\mathbf{W} + j \cdot \mathbf{G}) + \mathbf{s} \cdot \mathbf{G}$<br>$\mathbf{c}_{3,j} \approx \mathbf{s}_j \cdot \mathbf{V} + x^*_{b,j} \cdot \mathbf{s} \cdot \mathbf{G}$<br>$\mathbf{c}_4 \approx \mathbf{s} \cdot \mathbf{D} + \boldsymbol{\mu} \cdot \lfloor q/2 \rfloor$<br>$\mathbf{c}_5 \approx \mathbf{s} \cdot (\mathbf{B}_1 - \vert\mathbf{x}^*_b\vert \cdot \mathbf{G})$ | $\mathbf{T}_{\mathbf{A}_0}, \mathbf{T}_{\mathbf{B}_0}, \mathsf{k}$<br>$\mathbf{K}_j = \begin{bmatrix} \mathbf{Z}_j \\ \mathbf{R}_j \end{bmatrix} \leftarrow \mathsf{SamplePre}\left( \begin{array}{c} [\mathbf{B}_0 \vert \mathbf{W} + j \cdot \mathbf{G}], \\ \begin{bmatrix} \mathbf{T}_{\mathbf{B}_0} \\ \mathbf{0}_{m \times m} \end{bmatrix}, \mathbf{V}, \tau_1; \mathsf{F}(\mathsf{k}, j) \end{array} \right)$<br>$\mathbf{A}_j = \mathbf{G} \cdot \mathbf{R}_j$<br>$\mathbf{K}_f \leftarrow \mathsf{SamplePre}\left( [\mathbf{A}_0 \vert \mathbf{A}_f \vert \mathbf{B}_1 - \ell \cdot \mathbf{G}], \begin{bmatrix} \mathbf{T}_{\mathbf{A}_0} \\ \mathbf{0}_{m \times m} \\ \mathbf{0}_{m \times m} \end{bmatrix}, \mathbf{D}, \tau_2 \right)$ | |
| $\mathsf{H}_1$ | ↓ | ↓ | ↓ | guess $\vert\mathbf{x}^*\vert = \ell^*$ |
| $\mathsf{H}_2$ | ↓ | ↓ | $\mathbf{K}_j = \begin{bmatrix} \mathbf{Z}_j \\ \mathbf{R}_j \end{bmatrix} \leftarrow \mathsf{SamplePre}\left( \begin{array}{c} [\mathbf{B}_0 \vert \mathbf{W} + j \cdot \mathbf{G}], \\ \begin{bmatrix} \mathbf{T}_{\mathbf{B}_0} \\ \mathbf{0}_{m \times m} \end{bmatrix}, \mathbf{V}, \tau_1; \mathbf{r}_j \end{array} \right)$ | PRF security |
| $\mathsf{H}_3$ | ↓ | $\mathbf{c}_{1,j}, \mathbf{c}_{2,j} \leftarrow \mathbb{Z}_q^m$<br>$\mathbf{c}_{3,j} \approx [\mathbf{c}_{1,j} \mid \mathbf{c}_{2,j}] \cdot \mathbf{K}_j$<br>$-\mathbf{s} \cdot (\mathbf{A}_j - x^*_{b,j} \cdot \mathbf{G})$ | ↓ | $\mathsf{LWE}_{n,m,\chi,q}$<br>(Fig. 3) |
| $\mathsf{H}_4$ | ↓ | ↓ | $\mathbf{R}_j \leftarrow \mathcal{D}_{\mathbb{Z}^{m \times m}, \tau_1}$<br>$\mathbf{Z}_j = \mathsf{SamplePre}\left( \begin{array}{c} \mathbf{B}_0, \mathbf{T}_{\mathbf{B}_0}, \\ \mathbf{V} - [\mathbf{W} + j \cdot \mathbf{G}] \cdot \mathbf{R}_j, \tau_1; \mathbf{r}_{j,1} \end{array} \right)$<br>$\mathbf{K}_j = \begin{bmatrix} \mathbf{Z}_j \\ \mathbf{R}_j \end{bmatrix}$ | Lemma 3 |
| $\mathsf{H}_5$ | ↓ | ↓ | $\mathbf{A}_j \leftarrow \mathbb{Z}_q^{n \times m}$<br>$\mathbf{R}_j = \mathsf{SamplePre}(\mathbf{G}, \mathbf{I}, \mathbf{A}_j, \tau_1; \mathbf{r}_{j,2})$<br>$\mathbf{Z}_j = \mathsf{SamplePre}\left( \begin{array}{c} \mathbf{B}_0, \mathbf{T}_{\mathbf{B}_0}, \\ \mathbf{V} - [\mathbf{W} + j \cdot \mathbf{G}] \cdot \mathbf{R}_j, \tau_1; \mathbf{r}_{j,1} \end{array} \right)$<br>$\mathbf{K}_j = \begin{bmatrix} \mathbf{Z}_j \\ \mathbf{R}_j \end{bmatrix}$ | Lemma 2<br>(Item 1) |
| $\mathsf{H}_6$ | $\mathbf{B}_1 = \mathbf{A}_0 \cdot \mathbf{U} + \ell^* \cdot \mathbf{G}$ | ↓ | $\mathbf{A}_j = \mathbf{A}_0 \cdot \mathbf{U}_j + x^*_{b,j} \cdot \mathbf{G}$ | LHL |
| $\mathsf{H}_7$ | ↓ | ↓ | $\mathbf{K}_\mathbf{y}$ using $\begin{array}{ll} \begin{bmatrix} [\mathbf{U}_1 \vert \dots \vert \mathbf{U}_{\ell^*}] \cdot \mathbf{H}_{\mathbf{A},\mathbf{y}} \\ \mathbf{I}_m \\ \mathbf{0}_{m \times m} \end{bmatrix} & \text{if } \ell = \ell^* \\ \begin{bmatrix} -\mathbf{U} \\ \mathbf{0}_{m \times m} \\ \mathbf{I}_m \end{bmatrix} & \text{if } \ell \neq \ell^* \end{array}$ | Lemma 2<br>SamplePre |
| $\mathsf{H}_8$ | $\mathbf{A}_0 \leftarrow \mathbb{Z}_q^{n \times m}$ | ↓ | ↓ | Lemma 2<br>TrapGen |
| $\mathsf{H}_9$ | ↓ | $\mathbf{c}_{3,j} \approx [\mathbf{c}_{1,j} \mid \mathbf{c}_{2,j}] \cdot \mathbf{K}_j$<br>$-\mathbf{c}_0 \cdot \mathbf{U}_j$<br>$\mathbf{c}_5 \approx \mathbf{c}_0 \cdot \mathbf{U}$ | ↓ | noise flooding |
| $\mathsf{H}_{10}$ | ↓ | $\mathbf{c}_0 \leftarrow \mathbb{Z}_q^m, \mathbf{c}_4 \leftarrow \mathbb{Z}_q^\lambda$ | ↓ | $\mathsf{LWE}_{n,m+\lambda,\chi,q}$ |
| $\mathsf{H}_{11}$ | $(\mathbf{A}_0, \mathbf{T}_{\mathbf{A}_0}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$ | ↓ | ↓ | Lemma 2<br>TrapGen |
| $\mathsf{H}_{12}$ | ↓ | ↓ | $\mathbf{K}_\mathbf{y} \leftarrow \mathsf{SamplePre}\left( \begin{array}{c} [\mathbf{A}_0 \vert \mathbf{A}_\mathbf{y} \vert \mathbf{B}_1 - \ell \cdot \mathbf{G}], \\ \begin{bmatrix} \mathbf{T}_{\mathbf{A}_0} \\ \mathbf{0}_{m \times m} \\ \mathbf{0}_{m \times m} \end{bmatrix}, \mathbf{D}, \tau_2 \end{array} \right)$ | Lemma 2<br>SamplePre |
| $\mathsf{H}_{13}$ | ↓ | $\mathbf{c}_{3,j} \leftarrow \mathbb{Z}_q^m$ | $\mathbf{A}_j \leftarrow \mathbb{Z}_q^{n \times m}$ | LHL |

Fig. 4: Summary of our security hybrids. ↓ denotes the same as the previous hybrid. We omit the noise terms in $\mathsf{H}_0$. Starting from $\mathsf{H}_{10}$, the proof is essentially analogous to that in [AFV11].

CHKP10. David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 523–552. Springer, Heidelberg, May / June 2010. 7

CW14. Jie Chen and Hoeteck Wee. Semi-adaptive attribute-based encryption and improved delegation for Boolean formula. In Michel Abdalla and Roberto De Prisco, editors, *SCN 14*, volume 8642 of *LNCS*, pages 277–297. Springer, Heidelberg, September 2014. 1

DRS04. Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 523–540. Springer, Heidelberg, May 2004. 7

GKW16. Rishab Goyal, Venkata Koppula, and Brent Waters. Semi-adaptive security and bundling functionalities made generic and easy. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 361–388. Springer, Heidelberg, October / November 2016. 1, 3

GPSW06. Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 89–98. ACM Press, October / November 2006. Available as Cryptology ePrint Archive Report 2006/309. 1

GPV08. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008. 7

GSW13. Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 75–92. Springer, Heidelberg, August 2013. 3, 8

GVW13. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 545–554. ACM Press, June 2013. 1

HILL99. Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999. 7

KL15. Lucas Kowalczyk and Allison Bishop Lewko. Bilinear entropy expansion from the decisional linear assumption. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 524–541. Springer, Heidelberg, August 2015. 1

KSW08. Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 146–162. Springer, Heidelberg, April 2008. 2

Lew12. Allison B. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 318–335. Springer, Heidelberg, April 2012. 1, 2

LW11. Allison B. Lewko and Brent Waters. Unbounded HIBE and attribute-based encryption. In Kenneth G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 547–567. Springer, Heidelberg, May 2011. 1

MP12. Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, Heidelberg, April 2012. 3, 7, 8

OT12. Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure unbounded inner-product and attribute-based encryption. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 349–366. Springer, Heidelberg, December 2012. 1, 2

RW13. Yannis Rouselakis and Brent Waters. Practical constructions and new proof methods for large universe attribute-based encryption. In Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung, editors, *ACM CCS 2013*, pages 463–474. ACM Press, November 2013. 1

SW05. Amit Sahai and Brent R. Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, Heidelberg, May 2005. 1

Ver23. Tanya Verma. Inside geo key manager v2: re-imagining access control for distributed systems. https://blog.cloudflare.com/inside-geo-key-manager-v2/, 2023. 1

Yam16. Shota Yamada. Adaptively secure identity-based encryption from lattices with asymptotically shorter public parameters. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 32–62. Springer, Heidelberg, May 2016. 5