Ciphertext-Policy ABE for NC^1 Circuits with Constant-Size Ciphertexts from Succinct LWE

Jiaqi Liu, Yuanyi Zhang, Fang-Wei Fu

Abstract

We construct a lattice-based ciphertext-policy attribute-based encryption (CP-ABE) scheme for NC^1 access policies with constant-size ciphertexts. Let λ be the security parameter. For an NC^1 circuit of depth d and size s on ℓ-bit inputs, our scheme has the public-key and ciphertext sizes $O(1)$ (independent of d), and secret-key size $O(\ell)$, where the $O(\cdot)$ hides $\mathrm{poly}(\lambda)$ factors. As an application, we obtain a broadcast encryption scheme for N users with ciphertext size $\mathrm{poly}(\lambda)$ independent of $\log N$ and key sizes $\mathrm{poly}(\lambda, \log N)$. Our construction is selectively secure in the standard model under the $\mathrm{poly}(\lambda)$-succinct LWE assumption introduced by Wee (CRYPTO 2024).

I. Introduction

In a traditional public-key encryption scheme, only a user who possesses the secret key corresponding to the public key used for encryption can decrypt the ciphertext. As a generalization of public-key encryption, *attribute-based encryption* (ABE) realizes fine-grained access control over encrypted data. There are two types of ABE schemes: *ciphertext-policy* ABE (CP-ABE) and *key-policy* ABE (KP-ABE). In a CP-ABE scheme, each secret key $\mathsf{sk}_{\mathbf{x}}$ is associated with an attribute $\mathbf{x} \in \{0,1\}^\ell$, and each ciphertext ct_f is associated with an access policy f that specifies which attribute vectors are authorized to decrypt it. In contrast, in a KP-ABE scheme, each secret key sk_f is associated with an access policy f, and each ciphertext $\mathsf{ct}_{\mathbf{x}}$ is associated with an attribute $\mathbf{x}$. In both cases, decryption succeeds only if the attribute satisfies the access policy, i.e., $f(\mathbf{x}) = 0$; otherwise, the decryptor learns no information about the underlying plaintext.

The concept of ABE was first proposed independently by Sahai and Waters [SW05] and by Goyal et al. [GPSW06]. Since then, ABE has become a fundamental cryptographic primitive, and many concrete schemes have been proposed. A large fraction of existing ABE constructions rely on assumptions over bilinear groups. However, due to known quantum attacks on bilinear-group-based assumptions, there has been substantial interest in constructing schemes from alternative assumptions. In particular, many subsequent ABE schemes base their security on the LWE assumption or its variants, which are believed to be resistant to quantum attacks. In [AFV11, Boy13, GGH$^+$13, BGG$^+$14, GV15, BV16, Tsa19, AWY20, CW23], a series of KP-ABE schemes were proposed, all based on the LWE assumption.

A generic transformation can convert a KP-ABE scheme into a CP-ABE one via a universal circuit. However, this transformation typically leads to significantly worse parameters. To optimize the parameters of CP-ABE scheme, Agrawal and Yamada [AY20] proposed a CP-ABE scheme for all NC^1 circuits with ciphertext size independent of the circuit size s, relying on the LWE and generic bilinear groups. Datta et al. [DKW21] proposed a CP-ABE scheme for all NC^1 circuits via linear secret sharing schemes, relying on the hardness of the LWE assumption. Subsequently, there are many CP-ABE schemes relying on the nonstandard LWE-related assumptions. Wee [Wee22] proposed a CP-ABE scheme based on the evasive LWE assumption and tensor LWE assumption. Hsieh et al. [HLL24] presented a CP-ABE scheme based on the (structured) evasive LWE assumption. Agrawal et al. [AKY24] proposed a CP-ABE scheme for all Turing machines based on the circular evasive LWE and tensor LWE. Wee [Wee24, Wee25] presents CP-ABE scheme for all circuits based on the weaker $\mathrm{poly}(\lambda, d)$-succinct LWE assumption with ciphertext size independent of the attribute length ℓ.

A. Our results

We list our results as below:

Jiaqi Liu, Yuanyi Zhang and Fang-Wei Fu are with Chern Institute of Mathematics and LPMC, Nankai University, Tianjin 300071, P. R. China, Emails: ljqi@mail.nankai.edu.cn, yuanyiz@mail.nankai.edu.cn, fwfu@nankai.edu.cn.

1. In this work, we present a CP-ABE scheme for access policies represented by NC^1 circuits. For circuits on ℓ-bit inputs with depth d and size s (hence $s = \mathrm{poly}(\ell)$ and $d = \log(\ell)$) where s, d, ℓ are fixed at setup. We construct a CP-ABE scheme with parameters:

$$|\mathsf{pk}| = O(s), \quad |\mathsf{sk}| = O(\ell), \quad |\mathsf{ct}| = O(1),$$

where the notation $O(\cdot)$ hides $\mathrm{poly}(\lambda)$ factors. It proves to be selectively secure against unbounded collusions. The security relies on the $\mathrm{poly}(\lambda)$-succinct LWE assumptions with sub-exponential modulus-to-noise ratio.

- The scheme we construct is a *succinct* CP-ABE. Namely, the ciphertext size in our scheme relies only on the security parameter λ and independent of the size s, the input length ℓ, and the depth d, which is asymptotically optimal up to $\mathrm{poly}(\lambda)$ factors.
- The size of the public parameters scales with the circuit size s. Though it is seen as not competitive compared with previous schemes (many schemes have public key size $O(\ell)$, such as [BV22, Wee22, HLL24]), we observe that in the public parameters, all elements are sampled uniformly except for the public parameters used in the succinct LWE instance (i.e., $\mathsf{pp} = (\mathbf{B}, \mathbf{W}, \mathbf{T})$), which is of size $\mathrm{poly}(\lambda)$. Therefore, we can generate the purely uniform part with a pseudorandom generator instead. We can then compress the size of the public parameters into $O(1)$ size.
- The bound on the size of sk is not tight. It depends on the number of attributes associated with the secret key rather than strictly on ℓ. In scenarios where the users possesses only $O(1)$ attributes, the secret-key size can be reduced to $O(1)$.

2. CP-ABE for NC^1 circuits gives a broadcast encryption for $N = \ell$ users as a special case. In the broadcast encryption, we can use a circuit of size $O(N \log N)$ and depth $O(\log N)$ to check the membership of the users. We obtain a broadcast encryption scheme for N users with parameters

$$|\mathsf{pk}| = N \cdot \mathrm{poly}(\lambda, \log N), \quad |\mathsf{sk}| = \mathrm{poly}(\lambda, \log N), \quad |\mathsf{ct}| = \mathrm{poly}(\lambda).$$

By using a pseudorandom generator to compress the public parameter, we can reduce $|\mathsf{pk}|$ to $\mathrm{poly}(\lambda)$. Since $|\mathsf{pk}| + |\mathsf{ct}| + |\mathsf{sk}| = \mathrm{poly}(\lambda, \log N)$, we obtain an optimal broadcast encryption (under $\mathrm{poly}(\lambda)$-succinct LWE assumption).

3. In the proof of the selective security of our construction, we use the technique of using the trapdoor $\mathbf{T}$ for $[\mathbf{I}_\ell \otimes \mathbf{B} \mid \mathbf{W}]$ in the ℓ-succinct LWE assumption to generate other trapdoors for the matrix of the form $[\mathbf{I}_{\ell'} \otimes \mathbf{B} \mid \mathbf{W}']$, where $\mathbf{W}'$ is statistically close to uniform.

B. Related Works

Comparison with [Wee25]. We use the $\{0,1\}$-LSSS technique from [DKW21] and the matrix commitment scheme from [Wee25] to construct our CP-ABE scheme for NC^1 scheme and obtain an optimal broadcast encryption scheme. Compared with [Wee25], our scheme supports a narrower class of access policies and incurs a larger secret-key size. Nevertheless, we improve the parameters in the following respects: (1) $|\mathsf{pk}|$, $|\mathsf{sk}|$, and $|\mathsf{ct}|$ are independent of the circuit depth d; (2) the resulting broadcast encryption has ciphertext size $|\mathsf{ct}|$ independent of $\log N$, where N is the number of users in the system; (3) during encryption, we execute only a single matrix commitment on $O(\ell)$-width inputs, rather than running a circuit commitment scheme on an ℓ-input circuit that composes a sequence of matrix commitments.

C. Paper Organization

The paper is organized as follows. Section II provides a technical overview of our main construction and ideas. Section III introduces necessary notations, basic concepts of lattices, the LWE assumption, and its variants. In Section IV, we formalize the linear secret sharing scheme (LSSS) and ciphertext-policy attribute-based encryption (CP-ABE) for access structures realized by LSSS. Section V presents our construction of an CP-ABE scheme and its security analysis.

II. TECHNICAL OVERVIEWS

In this section, we demonstrate a high-level overview of our construction. Throughout this section, we adopt the wavy underline to indicate that a term is perturbed by some noise. For example, we use the notation $\underset{\sim}{\mathbf{s}^\top\mathbf{A}}$ to denote the term $\mathbf{s}^\top\mathbf{A}+\mathbf{e}$ for an unspecified noise vector $\mathbf{e}$. For a discrete set S, we write $\mathbf{x} \xleftarrow{\$} S$ to denote that $\mathbf{x}$ is uniformly sampled from the set S. The notation $\mathbf{A}^{-1}(\mathbf{y})$ means sampling a short preimage $\mathbf{x}$ from discrete Gaussian distribution such that $\mathbf{A}\mathbf{x}=\mathbf{y}$. For a positive integer $n \in \mathbb{N}$, denote $[n]$ as the set $\{k \in \mathbb{N} : 1 \le k \le n\}$. We use $\overset{c}{\approx}$ to denote computational indistinguishability. We defer the formal definitions of all notations to Section III, and here use them informally for simplicity.

ℓ-succinct LWE. The security of our construction relies on the ℓ-succinct LWE assumption introduced by [Wee24]. More formally, let $n, q, m = O(n \log q)$ be lattice parameters. First, sample $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, $\mathbf{W} \xleftarrow{\$} \mathbb{Z}_q^{\ell n\times m}$, and a Gaussian $\mathbf{T} \in \mathbb{Z}^{(\ell+1)m\times \ell m}$ such that $[\mathbf{I}_\ell \otimes \mathbf{B} \mid \mathbf{W}]\cdot\mathbf{T} = \mathbf{I}_\ell \otimes \mathbf{G}$. The ℓ-succinct LWE assumption asserts that

$$(\mathbf{B}, \underset{\sim}{\mathbf{s}^\top\mathbf{B}}, \mathbf{W}, \mathbf{T}) \overset{c}{\approx} (\mathbf{B}, \mathbf{c}, \mathbf{W}, \mathbf{T}),$$

where $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \mathbf{c} \xleftarrow{\$} \mathbb{Z}_q^m$. Throughout the paper, we denote $\mathsf{pp} := (\mathbf{B}, \mathbf{W}, \mathbf{T})$. The ℓ-succinct LWE assumption is a falsifiable assumption implied by evasive LWE introduced in [Wee22]. As the parameter ℓ increases, the assumption becomes stronger. We note that the 1-succinct LWE assumption is equivalent to the standard LWE assumption.

A. Our schemes

We begin by describing the CP-ABE scheme in [DKW21], which supports access structures realized by a linear secret sharing scheme (LSSS).

DKW21 scheme. Let $\mathbb{U}$ denote the attribute universe, let $N := |\mathbb{U}|$, and identify $\mathbb{U}$ with the set $[N]$.

- The public key is given by $(\{\mathbf{A}_u\}_{u\in\mathbb{U}}, \{\mathbf{Q}_u\}_{u\in\mathbb{U}}, \mathbf{y})$, where $\mathbf{A}_u, \mathbf{Q}_u \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ and $\mathbf{y} \xleftarrow{\$} \mathbb{Z}_q^n$.
- The master secret key is given by $\{\mathsf{td}_{\mathbf{A}_u}\}_{u\in\mathbb{U}}$, where $\mathsf{td}_{\mathbf{A}_u}$ is the trapdoor for the matrix $\mathbf{A}_u$ associated with the attribute u. This trapdoor allows efficient sampling of short preimages for $\mathbf{A}_u$.
- Given an attribute set $U \subseteq \mathbb{U}$, the secret key associated with U is sampled as follows: Sample $\hat{\mathbf{t}} \leftarrow \chi^{m-1}$, where χ is some error distribution, set $\mathbf{t} = (1, \hat{\mathbf{t}}^\top)^\top$ and sample $\mathbf{k}_u \leftarrow \mathbf{A}_u^{-1}(-\mathbf{Q}_u\mathbf{t})$. Then the secret key is given by $(\{\mathbf{k}_u\}_{u\in U}, \mathbf{t})$.
- To encrypt a message $\mathsf{msg} \in \{0,1\}$ under the access structure determined by matrix $\mathbf{M} \in \{-1,0,1\}^{\ell\times s_{\max}}$, the ciphertext is given as follows: Assume that ρ is an injective function that maps the row indices of $\mathbf{M}$ to the attribute. First sample $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$ and $\mathbf{v}_2, \ldots, \mathbf{v}_{s_{\max}} \in \mathbb{Z}_q^m$. The ciphertext components are computed as

$$\begin{aligned}
\mathbf{c}_{1,i}^\top &= \underset{\sim}{\mathbf{s}^\top\mathbf{A}_{\rho(i)}},\\
\mathbf{c}_{2,i}^\top &= M_{i,1}(\underset{\sim}{\mathbf{s}^\top\mathbf{y}}, \underbrace{0,\ldots,0}_{m-1}) + \sum_{j\in\{2,\ldots,s_{\max}\}} M_{i,j}\mathbf{v}_j^\top + \underset{\sim}{\mathbf{s}^\top\mathbf{Q}_{\rho(i)}},\\
c_3 &= \underset{\sim}{\mathbf{s}^\top\mathbf{y}} + \mathsf{msg}\cdot\lceil q/2\rfloor.
\end{aligned}$$

  Then the ciphertext is given as $(\{\mathbf{c}_{1,i}\}_{i\in[\ell]}, \{\mathbf{c}_{2,i}\}_{i\in[\ell]}, c_3)$.
- Suppose that the secret key is associated with an authorized attribute set. Let I be the set of row indices of $\mathbf{M}$ corresponding to the attributes associated with the secret key. Let $\{w_i\}_{i\in I} \in \{0,1\}$

be the reconstruction coefficients in the LSSS scheme with respect to the matrix $\mathbf{M}$. The decryption algorithm proceeds as follows:

$$\begin{aligned}\sum_{i\in I} w_i(\mathbf{c}_{1,i}^\top \mathbf{k}_{\rho(i)} + \mathbf{c}_{2,i}^\top \mathbf{t}) &\approx \sum_{i\in I} w_i(\mathbf{s}^\top \mathbf{A}_{\rho(i)}\mathbf{k}_{\rho(i)}) + \sum_{i\in I} w_i M_{i,1}(\mathbf{s}^\top \mathbf{y}, 0, \ldots, 0)\mathbf{t} \\ &\quad + \sum_{i\in I, j\in\{2,\ldots,s_{\max}\}} w_i M_{i,j}\mathbf{v}_j^\top \mathbf{t} + \sum_{i\in I} w_i \mathbf{s}^\top \mathbf{Q}_{\rho(i)}\mathbf{t} \\ &= \sum_{i\in I} w_i M_{i,1}\mathbf{s}^\top \mathbf{y} + \sum_{i\in I, j\in\{2,\ldots,s_{\max}\}} w_i M_{i,j}\mathbf{v}_j^\top \mathbf{t}\end{aligned}$$

Since the reconstruction coefficients $\{w_i\}_{i\in I}$ are such that $\sum_{i\in I} w_i M_{i,1} = 1$ and $\sum_{i\in I} w_i M_{i,j} = 0$ for $j \in \{2,\ldots,s_{\max}\}$. Hence $\sum_{i\in I} w_i(\mathbf{c}_{1,i}^\top \mathbf{k}_{\rho(i)} + \mathbf{c}_{2,i}^\top \mathbf{t}) \approx \mathbf{s}^\top \mathbf{y}$, and we can recover msg by subtracting it from c_3 and decoding.

From the above description, the ciphertext size scales with ℓ, i.e., the number of rows of the matrix $\mathbf{M}$. We expect the ciphertext size to be independent of ℓ, and we use the matrix commitment technique in [Wee25] to compress the ciphertext. Precisely, we replace the matrices $\{\mathbf{A}_u\}_{u\in\mathbb{U}} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ with a matrix $\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ of the same size. Instead of generating $\mathbf{v}_2, \ldots, \mathbf{v}_{s_{\max}} \xleftarrow{\$} \mathbb{Z}_q^m$, we select uniformly random matrices $\mathbf{B}_2, \ldots, \mathbf{B}_{s_{\max}} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$. We modify the second component of the ciphertext as follows:

$$\mathbf{c}'_{2,i} = M_{i,1}(\mathbf{s}^\top \mathbf{y}, 0, \ldots, 0) + \sum_{j\in\{2,\ldots,s_{\max}\}} M_{i,j}\mathbf{s}^\top \mathbf{B}_j + \underbrace{\mathbf{s}^\top \mathbf{Q}_{\rho(i)}}$$

for our commitment purposes (with this modification, the scheme remains selectively secure in the same model). To compress the $\{\mathbf{c}'_{2,i}\}$, for $i \in [\ell]$, let

$$\mathbf{U}_{\rho(i)} \leftarrow M_{i,1}(\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}) + \sum_{j\in\{2,\ldots,s_{\max}\}} M_{i,j}\mathbf{B}_j + \mathbf{Q}_{\rho(i)}$$

and for $u \in [N] \setminus \rho([\ell])$ we sample $\mathbf{U}_u \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$. For $i \in [\ell]$, we have $\mathbf{c}'_{2,i} \approx \mathbf{s}^\top \mathbf{U}_{\rho(i)}$. Then we commit to $[\mathbf{U}_1 \mid \cdots \mathbf{U}_N] \in \mathbb{Z}_q^{n\times mN}$, and denote the commitment matrix by $\mathbf{C} \in \mathbb{Z}_q^{n\times m}$. Roughly, the matrix $\mathbf{C}$ contains partial information about all $\mathbf{U}_u$.

Modification. With the idea given above, we modify the DKW21 CP-ABE scheme as follows:

- The public key is given by $(\mathsf{pp}, \mathbf{A}, \{\mathbf{B}_i\}_{i\in[s_{\max}]}, \{\mathbf{Q}_u\}_{u\in\mathbb{U}}, \mathbf{y})$, where pp is the public parameter of the $2m^2$-succinct LWE problem, $\mathbf{A}, \mathbf{B}_i, \mathbf{Q}_u \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ and $\mathbf{y} \xleftarrow{\$} \mathbb{Z}_q^n$.
- The master secret key is given by $\mathsf{td}_{\mathbf{B}}$.
- Given an attribute set $U \subseteq \mathbb{U}$, the secret key associated with U is sampled as follows: Sample $\hat{\mathbf{t}} \leftarrow \chi^{m-1}$, where χ is some error distribution, set $\mathbf{t} = (1, \hat{\mathbf{t}}^\top)^\top$ and sample
$$\mathbf{k}_u \leftarrow \mathbf{B}^{-1}((\mathbf{A}\mathbf{V}_u + \mathbf{Q}_u)\mathbf{t}),$$
where $\mathbf{V} := [\mathbf{V}_1 \mid \cdots \mid \mathbf{V}_N]$ is the verification matrix of a commitment of a matrix of width Nm (which can be computed without the knowledge of the matrix committed). Then the secret key is given by $(\{\mathbf{k}_u\}_{u\in U}, \mathbf{t})$.
- To encrypt a message $\mathsf{msg} \in \{0,1\}$ under the access structure determined by matrix $\mathbf{M} \in \{-1,0,1\}^{\ell\times s_{\max}}$, the ciphertext is given as follows: Assume that ρ is an injective function that maps the row indices of $\mathbf{M}$ to the attribute. First sample $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$. The ciphertext components are computed as
$$\mathbf{c}_1^\top = \underbrace{\mathbf{s}^\top \mathbf{B}}, \quad \mathbf{c}_2^\top = \underbrace{\mathbf{s}^\top (\mathbf{A} + \mathbf{C})}, \quad c_3 = \underbrace{\mathbf{s}^\top \mathbf{y}} + \mathsf{msg} \cdot \lceil q/2 \rfloor.$$
Then the ciphertext is given as $(\mathbf{c}_1, \mathbf{c}_2, c_3)$.

In the second component $\mathbf{c}_2$, the commitment matrix $\mathbf{C}$ is such that

$$\mathbf{CV} = [\mathbf{U}_1 \mid \cdots \mid \mathbf{U}_N] - \mathbf{BZ},$$

where for each $i \in [\ell]$,

$$\mathbf{U}_{\rho(i)} \leftarrow M_{i,1}(\mathbf{y} \mid \underbrace{\mathbf{0} \mid \cdots \mid \mathbf{0}}_{m-1}) + \sum_{j \in \{2,\ldots,s_{\max}\}} M_{i,j}\mathbf{B}_j + \mathbf{Q}_{\rho(i)}$$

and for $u \in [N] \setminus \rho([\ell])$, $\mathbf{U}_u \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$.

- When decrypting the ciphertexts, one can first compute $\mathbf{U}_{\rho(i)}$ for all $i \in I$, where I is the set of the row indices related to the attributes associated with the secret keys. If we denote $\mathbf{V} = [\mathbf{V}_1 \mid \cdots \mid \mathbf{V}_N]$ and $\mathbf{Z} = [\mathbf{Z}_1 \mid \cdots \mid \mathbf{Z}_N]$, we obtain that

$$\mathbf{CV}_i = \mathbf{U}_i - \mathbf{BZ}_i, \qquad \forall i \in [N].$$

Therefore, we can recover $\mathbf{c}'_{2,i}$ in the DKW21 scheme by

$$\mathbf{c}_2^\top \mathbf{V}_{\rho(i)} \approx \mathbf{s}^\top(\mathbf{A}+\mathbf{C})\mathbf{V}_{\rho(i)} = \mathbf{s}^\top \mathbf{A}\mathbf{V}_{\rho(i)} + \underbrace{\mathbf{s}^\top \mathbf{U}_{\rho(i)}}_{\approx \mathbf{c}'_{2,i}} - \underbrace{\mathbf{s}^\top \mathbf{B}\mathbf{Z}_{\rho(i)}}_{\approx \mathbf{c}_1^\top \mathbf{Z}_{\rho(i)}}.$$

Combining it with the reconstruction coefficients $\{w_i\}_{i \in I}$, we obtain that

$$\begin{aligned}
&\sum_{i \in I} w_i(\mathbf{c}_2^\top \mathbf{V}_{\rho(i)} + \mathbf{c}_1^\top \mathbf{Z}_{\rho(i)}) \\
\approx &\sum_{i \in I} w_i \mathbf{s}^\top \mathbf{A}\mathbf{V}_{\rho(i)} + \sum_{i \in I} w_i \mathbf{s}^\top \mathbf{U}_{\rho(i)} \\
= &\sum_{i \in I} w_i \mathbf{s}^\top (\mathbf{A}\mathbf{V}_{\rho(i)} + \mathbf{Q}_{\rho(i)}) + \sum_{i \in I} w_i M_{i,1} \mathbf{s}^\top [\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}] + \sum_{i \in I, j \in \{2,\ldots,s_{\max}\}} w_i M_{i,j} \mathbf{s}^\top \mathbf{B}_j \\
= &\sum_{i \in I} w_i \mathbf{s}^\top (\mathbf{A}\mathbf{V}_{\rho(i)} + \mathbf{Q}_{\rho(i)}) + \mathbf{s}^\top [\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}].
\end{aligned}$$

Then using the secret key equation $\mathbf{B}\mathbf{k}_{\rho(i)} = (\mathbf{A}\mathbf{V}_{\rho(i)} + \mathbf{Q}_{\rho(i)})\mathbf{t}$, we can recover the message by computing

$$c_3 - \sum_{i \in I} w_i(\mathbf{c}_2^\top \mathbf{V}_{\rho(i)}\mathbf{t} + \mathbf{c}_1^\top \mathbf{Z}_{\rho(i)}\mathbf{t} - \mathbf{c}_1^\top \mathbf{k}_{\rho(i)}).$$

B. Static Security

We prove that our scheme is selectively secure under the $2m^2$-succinct LWE assumption in V-B. We achieve our goal by defining a sequence of hybrid games. The first hybrid game corresponds to the real game as defined in 16, while the last hybrid game is independent of the messages, where the advantage of the adversary is exactly 0. By arguing each adjacent pair of the hybrid games is indistinguishable, the adversary in the real game can win the game with negligible advantage.

More precisely, in the reduction, we attempt to prove the ciphertext components $\mathbf{c}_1, \mathbf{c}_2, c_3$ are pseudo-random, since they resemble LWE samples intuitively. However, in the key query phase, the challenger uses the trapdoors $\mathsf{td}_{\mathbf{B}}$ for $\mathbf{B}$ to generate corresponding secret keys, which may result in the additional information leakage of the LWE samples. To overcome this problem, we notice that using the matrix $\mathbf{T}$ in the public parameters in the $2m^2$-succinct LWE instance, we can construct trapdoors for the matrix of the form $[\mathbf{I}_s \otimes \mathbf{B} \mid \mathbf{W}']$, where $\mathbf{W}'$ is statistically close to uniform. Using this technique, the challenger can respond to the secret-key queries with the newly generated trapdoor when the secret-key corresponds to an unauthorized attribute set. Since the new trapdoor is efficiently constructed from the public parameters

without using the trapdoor $\mathsf{td}_{\mathbf{B}}$, this avoids giving the adversary additional information correlated with the LWE samples in the hybrids where the ciphertext components are replaced by uniform.

During the reduction, suppose that the adversary submits a secret-key query with respect to the attribute set U. The challenger needs to sample the preimage of $(\mathbf{A}\mathbf{V}_u + \mathbf{Q}_u)\mathbf{t}$ for each $u \in U$ with respect to $\mathbf{B}$. We first simulate the uniform matrix $\mathbf{A} = \mathbf{B}\mathbf{R} - \mathbf{C}, \mathbf{Q}_u = \mathbf{B}\mathbf{R}_u + M_{\rho^{-1}(u),1}(\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0})$ with $\mathbf{R}, \mathbf{R}_u \xleftarrow{\$} \{-1,1\}^{m\times m}$ for all $u \in \mathbb{U}$, which is guaranteed by the leftover hash lemma under appropriate parameters. Parsing $\mathbf{B}_i$ as $[\mathbf{b}'_i \mid \mathbf{B}'_i]$, where $\mathbf{b}'_i \in \mathbb{Z}_q^n$, then

$$\begin{aligned}(\mathbf{A}\mathbf{V}_u + \mathbf{Q}_u)\mathbf{t} &= ((\mathbf{B}\mathbf{R} - \mathbf{C})\mathbf{V}_u + \mathbf{Q}_u)\mathbf{t}\\ &= (\mathbf{B}\mathbf{R}\mathbf{V}_u - \mathbf{U}_u + \mathbf{B}\mathbf{Z}_u + \mathbf{Q}_u)\mathbf{t}\\ &= \mathbf{B}\mathbf{R}\mathbf{V}_u\mathbf{t} + \mathbf{B}\mathbf{Z}_u\mathbf{t} + \mathbf{Q}_u\mathbf{t} - M_{\rho^{-1}(u),1}(\mathbf{y} \mid \underbrace{\mathbf{0} \mid \cdots \mid \mathbf{0}}_{m-1})\mathbf{t} - \sum_{j\in\{2,\ldots,s_{\max}\}} M_{\rho^{-1}(u),j}\mathbf{B}_j\mathbf{t}\\ &= \mathbf{B}(\mathbf{R}\mathbf{V}_u\mathbf{t} + \mathbf{Z}_u\mathbf{t} + \mathbf{R}_u\mathbf{t}) - \sum_{j\in\{2,\ldots,s_{\max}\}} M_{\rho^{-1}(u),j}\mathbf{b}'_j - \sum_{j\in\{2,\ldots,s_{\max}\}} M_{\rho^{-1}(u),j}\mathbf{B}'_j\hat{\mathbf{t}}.\end{aligned}$$

Then the challenger can instead first sample short $\hat{\mathbf{t}}$ and $\tilde{\mathbf{k}}_u$ as:

$$\left[\mathbf{B}\middle|\sum_{j\in\{2,\ldots,s_{\max}\}} M_{\rho^{-1}(u),j}\mathbf{B}'_j\right]\begin{pmatrix}\tilde{\mathbf{k}}_u\\ \hat{\mathbf{t}}\end{pmatrix} = -\sum_{j\in\{2,\ldots,s_{\max}\}} M_{\rho^{-1}(u),j}\mathbf{b}'_j \tag{1}$$

for each $u \in \rho([\ell]) \cap U$. Then the challenger sets

$$\mathbf{k}_u \leftarrow \tilde{\mathbf{k}}_u + \mathbf{R}\mathbf{V}_u\mathbf{t} + \mathbf{Z}_u\mathbf{t} + \mathbf{R}_u\mathbf{t}$$

with $\mathbf{t} = (1, \hat{\mathbf{t}}^\top)^\top$, and hence $\{\mathbf{k}_u\}$ are distributed the same as in the real game up to negligible statistical distance by the noise-smudging technique. By simulating $\mathbf{B}_i := \mathbf{B}\mathbf{R}_i + \mathbf{C}_i$ with $\mathbf{R}_i \xleftarrow{\$} \{-1,1\}$ and $\mathbf{C}_i$ the commitment matrix of some matrix of specific form, we can use $\mathbf{T}$ to generate a trapdoor td' for the matrix

$$\left[\mathbf{I}_{s_{\max}-1}\otimes\mathbf{B}\middle|\begin{matrix}\mathbf{B}'_2\\ \vdots\\ \mathbf{B}'_{s_{\max}}\end{matrix}\right]. \tag{2}$$

Moreover, we can prove that as long as U is unauthorized to decrypt the ciphertext, td' can then be used to generate the trapdoor for the linear combination of the row block matrices of (2), i.e.,

$$(\mathbf{M}_U \otimes \mathbf{I}_n)\left[\mathbf{I}_{s_{\max}-1}\otimes\mathbf{B}\middle|\begin{matrix}\mathbf{B}'_2\\ \vdots\\ \mathbf{B}'_{s_{\max}}\end{matrix}\right],$$

where $\mathbf{M}_U$ are the row vectors of $\mathbf{M}$ corresponding to the attributes in U. Therefore, without using the trapdoor $\mathsf{td}_{\mathbf{B}}$, the challenger can sample $\{\tilde{\mathbf{k}}_u\}_{u\in U}$ in (1). For details, please refer to V-B.

III. Preliminaries

Notations. Let $\lambda \in \mathbb{N}$ denote the security parameter used throughout this paper. For a positive integer $n \in \mathbb{N}$, denote $[n]$ as the set $\{k \in \mathbb{N} : 1 \le k \le n\}$. For a positive integer $q \in \mathbb{N}$, let $\mathbb{Z}_q$ denote the ring of integers modulo q. Throughout this paper, vectors are assumed to be column vectors by default. We use bold lower-case letters (e.g., $\mathbf{u}, \mathbf{v}$) for vectors and bold upper-case letters (e.g., $\mathbf{A}, \mathbf{B}$) for matrices. Let $\mathbf{v}[i]$ denote the i-th entry of the vector $\mathbf{v}$, and let $\mathbf{U}[i,j]$ denote the (i,j)-entry of the matrix $\mathbf{U}$. We

write $\mathbf{0}_n$ for the all-zero vector of length n, and $\mathbf{0}_{n\times m}$ for the all-zero matrix of dimension $n \times m$. The *infinity norm* of a vector $\mathbf{v}$ and the corresponding operator norm of a matrix $\mathbf{U}$ are defined as:

$$\|\mathbf{v}\| = \max_i |\mathbf{v}[i]|, \ \|\mathbf{U}\| = \max_{i,j} |\mathbf{U}[i,j]|.$$

For two matrices $\mathbf{A}, \mathbf{B}$ of dimensions $n_1 \times m_1$ and $n_2 \times m_2$, respectively, their *Kronecker product* is an $n_1 n_2 \times m_1 m_2$ matrix given by

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} \mathbf{A}[1,1]\mathbf{B} & \cdots & \mathbf{A}[1,m_1]\mathbf{B} \\ \vdots & \ddots & \vdots \\ \mathbf{A}[n_1,1]\mathbf{B} & \cdots & \mathbf{A}[n_1,m_1]\mathbf{B} \end{bmatrix}.$$

We have $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD})$ if the multiplication is compatible.

Discrete Gaussians. Let $D_{\mathbb{Z},\sigma}$ represent the centered *discrete Gaussian distribution* over $\mathbb{Z}$ with standard deviation $\sigma \in \mathbb{R}^+$ (e.g., [Ban93]). For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a vector $\mathbf{v} \in \mathbb{Z}_q^n$, let $\mathbf{A}_\sigma^{-1}(\mathbf{v})$ denote the distribution of a random variable $\mathbf{u} \leftarrow D_{\mathbb{Z},\sigma}^m$ conditioned on $\mathbf{Au} = \mathbf{v} \mod q$. When $\mathbf{A}_\sigma^{-1}$ is applied to a matrix, it is understood as being applied independently to each column of the matrix.

The following lemma (e.g., [MR07]) shows that for a given Gaussian width parameter $\sigma = \sigma(\lambda)$, the probability of a vector drawn from the discrete Gaussian distribution having norm greater than $\sqrt{n}\sigma$ is negligible.

Lemma 1. *Let λ be a security parameter and $\sigma = \sigma(\lambda)$ be a Gaussian width parameter. Then for all polynomials $n = n(\lambda)$, there exists a negligible function $\mathrm{negl}(\lambda)$ such that for all $\lambda \in \mathbb{N}$,*

$$\Pr\left[\|\mathbf{v}\| > \sqrt{n}\sigma : \mathbf{v} \leftarrow D_{\mathbb{Z},\sigma}^n\right] = \mathrm{negl}(\lambda).$$

Smudging Lemma. In the following, we present the standard smudging lemma, which formalizes the intuition that sufficiently large standard deviation can "smudge out" small perturbations, making the resulting distributions statistically indistinguishable.

Lemma 2 ([BDE+18])**.** *Let λ be a security parameter, and let $e \in \mathbb{Z}$ satisfy $|e| \le B$. Suppose that $\sigma \ge B \cdot \lambda^{\omega(1)}$. Then, the following two distributions are statistically indistinguishable:*

$$\{z : z \leftarrow D_{\mathbb{Z},\sigma}\} \ \text{and} \ \{z + e : z \leftarrow D_{\mathbb{Z},\sigma}\}.$$

The Gadget Matrix. Here we recall the definition of the *gadget matrix* introduced in [MP12]. For positive integers $n, q \in \mathbb{N}$, let $\mathbf{g}^\top = (1, 2, \ldots, 2^{\lceil \log q \rceil - 1})$ denote the *gadget vector*. Define the *gadget matrix* $\mathbf{G}_n = \mathbf{I}_n \otimes \mathbf{g}^\top \in \mathbb{Z}_q^{n\times m}$, where $m = n\lceil \log q \rceil$.

Trapdoors for lattices. In this work, we adopt the trapdoor frameworks outlined in [Ajt96, GPV08, MP12], following the formalization in [BTVW17].

Lemma 3 ([GPV08, MP12])**.** *Let n, m, q be lattice parameters. Then there exist two efficient algorithms (*$\mathsf{TrapGen}$*,* $\mathsf{SamplePre}$*) with the following syntax:*

- $\mathsf{TrapGen}(1^n, 1^m, q) \to (\mathbf{A}, \mathbf{T})$*: On input the lattice dimension n, number of samples m, modulus q, this randomized algorithm outputs a matrix $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ together with a trapdoor $\mathbf{T} \in \mathbb{Z}_q^{m\times m}$.*
- $\mathsf{SamplePre}(\mathbf{A}, \mathbf{T}, \mathbf{y}, \sigma) \to \mathbf{x}$*: On input $(\mathbf{A}, \mathbf{T})$ from* $\mathsf{TrapGen}$*, a target vector $\mathbf{y} \in \mathbb{Z}_q^n$ and a Gaussian width parameter σ, this randomized algorithm outputs a vector $\mathbf{x} \in \mathbb{Z}^m$.*

Moreover, there exists a polynomial $m_0(n, q) = O(n \log q)$ such that for all $m \ge m_0$, the above algorithms satisfy the following properties:

- **Trapdoor distribution:** *The matrix $\mathbf{A}$ output by $\mathsf{TrapGen}(1^n, 1^m, q)$ is statistically close to a uniformly random matrix. Specifically, if $(\mathbf{A}, \mathsf{td}_\mathbf{A}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$ and $\mathbf{A}' \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, then the*

statistical distance between the distributions of $\mathbf{A}$ *and* $\mathbf{A}'$ *is at most* 2^{-n}*. The trapdoor* $\mathsf{td}_{\mathbf{A}}$ *serves as* τ*-trapdoor where* $\tau = O(\sqrt{n \log q})$*. Denote* $\chi_0 = \tau \cdot \omega(\sqrt{\log n})$*.*

- **Preimage sampling:** *Suppse* $\mathsf{td}_{\mathbf{A}}$ *is a* τ*-trapdoor for* $\mathbf{A}$*. For all* $\sigma \geq \chi_0$ *and all target vectors* $\mathbf{y} \in \mathbb{Z}_q^n$*, the statistical distance between the following two distributions is at most* 2^{-n}*:*

$$\{\mathbf{x} \leftarrow \mathsf{SamplePre}(\mathbf{A}, \mathbf{T}, \mathbf{y}, \sigma)\} \text{ and } \{\mathbf{x} \leftarrow \mathbf{A}_\sigma^{-1}(\mathbf{y})\}.$$

Corollary 1 ([WWW22], adapted)**.** *Let* $\mathbf{H} \in \mathbb{Z}_q^{k \times t}$ *be a full row rank matrix (*$k \leq t$*). Let* $\mathbf{A} \in \mathbb{Z}_q^{kn \times m'}$ *and* $\mathbf{R} \in \mathbb{Z}_q^{m' \times mt}$*. Suppose* $\mathbf{AR} = \mathbf{H} \otimes \mathbf{G}$*. Then* $(\mathbf{R}, \mathbf{H})$ *can be used as a* τ*-trapdoor for* $\mathbf{A}$*, where* $\tau \leq \sqrt{kmm'} \cdot mt \|\mathbf{R}\|$*.*

Succinct LWE **assumption.** Our proposed CP-ABE scheme relies on the succinct LWE assumption [Wee24].

Assumption 1 (ℓ-succinct LWE [Wee24])**.** *Let* λ *be a security parameter. Let* n, m *be such that* $m \geq 2n \log q$*. The* (ℓ, σ)*-succinct* LWE *says*

$$(\mathbf{B}, \mathbf{s}^\top \mathbf{B} + \mathbf{e}^\top, \mathbf{W}, \mathbf{T}) \overset{c}{\approx} (\mathbf{B}, \mathbf{c}, \mathbf{W}, \mathbf{T})$$

where

$$\mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}, \mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \mathbf{e} \leftarrow D_{\mathbb{Z},\chi}^m, \mathbf{c} \xleftarrow{\$} \mathbb{Z}_q^m$$
$$\mathbf{W} \xleftarrow{\$} \mathbb{Z}_q^{\ell n \times m}, \mathbf{T} \leftarrow [\mathbf{I}_\ell \otimes \mathbf{B} \mid \mathbf{W}]_\sigma^{-1}(\mathbf{I}_\ell \otimes \mathbf{G})$$

Lemma 4 (Leftover hash lemma, [ABB10])**.** *Let* λ *be a security parameter. Let* $n = n(\lambda), q = q(\lambda)$ *be some lattice parameters. Let* $m > (n+1)\log q + \omega(\log n)$*, and* $k = k(n)$ *be some polynomial. Then the following two distributions are statistically indistinguishable:*

$$\{(\mathbf{A}, \mathbf{AR}) : \mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}, \mathbf{R} \xleftarrow{\$} \{-1,1\}^{m \times k}\} \text{ and } \{(\mathbf{A}, \mathbf{S}) : \mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}, \mathbf{S} \xleftarrow{\$} \mathbb{Z}_q^{n \times k}\}.$$

Lemma 5 (Leftover hash lemma with trapdoor)**.** *Let* λ *be a security parameter. Let* $n = n(\lambda), q = q(\lambda)$ *be some lattice parameters. Let* $m > 2(n+1)\log q + \omega(\log n)$*. Let* $k = k(n)$ *be some polyomial. Let* $m > m_0, \chi, \sigma > \chi_0$*, where* m_0, χ_0 *are polynomials given in Lemma 3. Then for any adversary* $\mathcal{A}$*, there exists a negligible function* $\mathrm{negl}(\lambda)$*, such that for all* $n \in \mathbb{N}$*,*

$$\left| \Pr\left[\mathsf{Exp}_{\mathcal{A}}^{\mathsf{LHL\text{-}Trap},q,\sigma,\chi}(\lambda) = 1 \right] - 1/2 \right| \leq \mathrm{negl}(\lambda),$$

where the experiment is defined below.

Setup phase: The adversary $\mathcal{A}$ sends $1^n, 1^m$ to the challenger. The challenger flips a random bit $b \xleftarrow{\$} \{0,1\}$, and proceeds as follows:

1. It samples $(\mathbf{B}, \mathsf{td}_{\mathbf{B}}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$, $\mathbf{W} \xleftarrow{\$} \mathbb{Z}_q^{2m^2 n \times m}$, $\mathbf{T} \leftarrow \mathsf{SamplePre}([\mathbf{I}_{2m^2} \otimes \mathbf{B} \mid \mathbf{W}], \begin{bmatrix} \mathbf{I}_{2m^2} \otimes \mathsf{td}_{\mathbf{B}} \\ \mathbf{0} \end{bmatrix}, \mathbf{I}_{2m^2} \otimes \mathbf{G}, \sigma)$ and sets $\mathsf{pp} := (\mathbf{B}, \mathbf{W}, \mathbf{T})$. It then samples $\mathbf{R} \xleftarrow{\$} \{-1,1\}^{m \times k}$, and $\mathbf{S}_0 \leftarrow \mathbf{BR}$. It then samples $\mathbf{S}_1 \xleftarrow{\$} \mathbb{Z}_q^{n \times k}$.
2. It sends $(\mathsf{pp}, \mathbf{S}_b)$ to the adversary $\mathcal{A}$.

Query Phase: The adversary $\mathcal{A}$ makes polynomially many pre-image queries of the form $\mathbf{z} \in \mathbb{Z}_q^n$. The challenger responds to each query by sampling $\mathbf{s} \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathsf{td}_{\mathbf{B}}, \mathbf{z}, \chi)$ and sends $\mathbf{s}$ to $\mathcal{A}$.

Response Phase: The adversary outputs its guess $b' \in \{0,1\}$ for b. The experiments outputs 1 if and only if $b = b'$.

Experiment $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{LHL\text{-}Trap},q,\sigma,\chi}$

Proof. The proof proceeds via a sequence of hybrid experiments. We define games $\mathsf{H}_0, \ldots, \mathsf{H}_6$ and show that each consecutive pair of hybrids is statistically indistinguishable. The detailed arguments are given in Lemmas 7–11 below.

Game H_0 This corresponds to the original game with $b = 0$.

1) **Setup Phase:** The challenger receives 1^n and 1^m.
   a) It samples $(\mathbf{B}, \mathsf{td}_{\mathbf{B}}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$.
   b) It samples $\mathbf{W} \xleftarrow{\$} \mathbb{Z}_q^{2m^2 n \times m}$.
   c) $\mathbf{T} \leftarrow \mathsf{SamplePre}([\mathbf{I}_{2m^2} \otimes \mathbf{B} \mid \mathbf{W}], \begin{bmatrix} \mathbf{I}_{2m^2} \otimes \mathsf{td}_{\mathbf{B}} \\ \mathbf{0} \end{bmatrix}, \mathbf{I}_{2m^2} \otimes \mathbf{G}, \sigma)$ and sets $\mathsf{pp} := (\mathbf{B}, \mathbf{W}, \mathbf{T})$.
   d) It then samples $\mathbf{R} \xleftarrow{\$} \{-1, 1\}^{m \times k}$, and $\mathbf{S} \leftarrow \mathbf{B}\mathbf{R}$.
   e) It sends $(\mathsf{pp}, \mathbf{S})$ to the adversary $\mathcal{A}$.
2) **Query Phase:** The adversary makes polynomially many queries $\mathbf{z} \in \mathbb{Z}_q^n$. The challenger responds to each query as follows:
   a) It samples $\mathbf{s} \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathsf{td}_{\mathbf{B}}, \mathbf{z}, \chi)$ and sends $\mathbf{s}$ to $\mathcal{A}$.
3) The adversary outputs a bit b'.

Game H_1 This game is defined identically to Game H_0 except for the way to generate the matrices $\mathbf{B}$, $\mathbf{T}$ and the responses to the pre-image queries.

1) **Setup Phase:** The challenger receives 1^n and 1^m.
   a) It samples $\boxed{(\mathbf{B}_1, \mathsf{td}_{\mathbf{B}_1}) \leftarrow \mathsf{TrapGen}(1^n, 1^{m/2}, q), (\mathbf{B}_2, \mathsf{td}_{\mathbf{B}_2}) \leftarrow \mathsf{TrapGen}(1^n, 1^{m/2}, q)}$ and sets $\mathbf{B} = [\mathbf{B}_1 \mid \mathbf{B}_2]$.
   b) It samples $\mathbf{W} = \begin{bmatrix} \mathbf{W}_1 \\ \vdots \\ \mathbf{W}_{2m^2} \end{bmatrix} \xleftarrow{\$} \mathbb{Z}_q^{2m^2 n \times m}$, where $\mathbf{W}_i \in \mathbb{Z}_q^{n \times m}$ for each $i \in [2m^2]$.
   c) It samples $\mathbf{T}_{2m^2+1} \leftarrow D_{\mathbb{Z},\sigma}^{m \times 2m^3}$. For each $i \in [2m^2]$, it samples $\mathbf{T}_i^{\mathsf{up}} \leftarrow D_{\mathbb{Z},\sigma}^{m/2 \times 2m^3}$, and $\mathbf{T}_i^{\mathsf{down}} \leftarrow \mathsf{SamplePre}(\mathbf{B}_2, \mathsf{td}_{\mathbf{B}_2}, \mathbf{e}_i^\top \otimes \mathbf{G} - \mathbf{W}_i \mathbf{T}_{2m^2+1} - \mathbf{B}_1 \mathbf{T}_i^{\mathsf{up}}, \sigma)$, where $\mathbf{e}_i \in \{0,1\}^{2m^2}$ represents the vector with the i-th entry 1 and the other entries 0. It sets $\mathbf{T}_i = \begin{bmatrix} \mathbf{T}_i^{\mathsf{up}} \\ \mathbf{T}_i^{\mathsf{down}} \end{bmatrix}$ and $\boxed{\mathbf{T} = \begin{bmatrix} \mathbf{T}_1 \\ \vdots \\ \mathbf{T}_{2m^2} \\ \mathbf{T}_{2m^2+1} \end{bmatrix}}$, and sets $\mathsf{pp} := (\mathbf{B}, \mathbf{W}, \mathbf{T})$.
   d) It then samples $\mathbf{R}_1, \mathbf{R}_2 \xleftarrow{\$} \{-1, 1\}^{m/2 \times k}$, and $\boxed{\mathbf{S} \leftarrow \mathbf{B}_1 \mathbf{R}_1 + \mathbf{B}_2 \mathbf{R}_2}$.
   e) It sends $(\mathsf{pp}, \mathbf{S})$ to the adversary.
2) **Query Phase:** The adversary makes polynomially many queries $\mathbf{z} \in \mathbb{Z}_q^n$. The challenger responds to each query as follows:
   a) It samples $\mathbf{s}_1 \leftarrow \chi^{m/2}$, and samples $\mathbf{s}_2 \leftarrow \mathsf{SamplePre}(\mathbf{B}_2, \mathsf{td}_{\mathbf{B}_2}, \mathbf{z} - \mathbf{B}_1 \mathbf{s}_1, \chi)$ and sends $\boxed{\mathbf{s} = \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix}}$ to $\mathcal{A}$.
3) The adversary outputs a bit b'.

Game H_2 This game is defined identically to Game H_1 except that the challenger samples the matrix $\mathbf{B}_1$ uniformly at random instead of using $\mathsf{TrapGen}$.

1) **Setup Phase:** The challenger receives 1^n and 1^m.
   a) It samples $\boxed{\mathbf{B}_1 \xleftarrow{\$} \mathbb{Z}_q^{n \times m/2}}$, $(\mathbf{B}_2, \mathsf{td}_{\mathbf{B}_2}) \leftarrow \mathsf{TrapGen}(1^n, 1^{m/2}, q)$ and sets $\mathbf{B} = [\mathbf{B}_1 \mid \mathbf{B}_2]$.

Game H_3 This game is defined identically to Game H_2 except that we replace the term $\mathbf{B}_1 \mathbf{R}_1$ with a

uniformly random matrix $\mathbf{S}'$.

1) **Setup Phase:** The challenger receives 1^n and 1^m.
   d) It then samples $\mathbf{S}' \xleftarrow{\$} \mathbb{Z}_q^{n\times k}$ and $\mathbf{R}_2 \xleftarrow{\$} \{-1,1\}^{m/2\times k}$, and $\boxed{\mathbf{S} \leftarrow \mathbf{S}' + \mathbf{B}_2\mathbf{R}_2}$.

Game H_4 This game is defined identically to Game H_3 except that we do not add the term $\mathbf{B}_2\mathbf{R}_2$ to $\mathbf{S}$, but instead sample $\mathbf{S}$ uniformly at random.

1) **Setup Phase:** The challenger receives 1^n and 1^m.
   d) It then samples $\boxed{\mathbf{S} \xleftarrow{\$} \mathbb{Z}_q^{n\times k}}$.

Game H_5 This game is defined identically to Game H_4 except that the challenger samples $\mathbf{B}_1$ using the algorithm $\mathsf{TrapGen}$ instead of sampling it uniformly at random.

1) **Setup Phase:** The challenger receives 1^n and 1^m.
   a) It samples $\boxed{(\mathbf{B}_1, \mathsf{td}_{\mathbf{B}_1}) \leftarrow \mathsf{TrapGen}(1^n, 1^{m/2}, q)}$, $(\mathbf{B}_2, \mathsf{td}_{\mathbf{B}_2}) \leftarrow \mathsf{TrapGen}(1^n, 1^{m/2}, q)$ and sets $\mathbf{B} = [\mathbf{B}_1 \mid \mathbf{B}_2]$.

Game H_6 This game is defined identically to Game H_5 except that we sample $(\mathbf{B}, \mathsf{td}_{\mathbf{B}})$ and $\mathbf{T}$ as in Game H_0, and answer preimage queries directly using $\mathsf{td}_{\mathbf{B}}$. This game corresponds to the original game with $b = 1$.

1) **Setup Phase:** The challenger receives 1^n and 1^m.
   a) It samples $\boxed{(\mathbf{B}, \mathsf{td}_{\mathbf{B}}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)}$.
   c) $\boxed{\mathbf{T} \leftarrow \mathsf{SamplePre}([\mathbf{I}_{2m^2} \otimes \mathbf{B} \mid \mathbf{W}], \begin{bmatrix} \mathbf{I}_{2m^2} \otimes \mathsf{td}_{\mathbf{B}} \\ \mathbf{0} \end{bmatrix}, \mathbf{I}_{2m^2} \otimes \mathbf{G}, \sigma)}$ and sets $\mathsf{pp} := (\mathbf{B}, \mathbf{W}, \mathbf{T})$.
2) **Query Phase:** The adversary makes polynomially many queries $\mathbf{z} \in \mathbb{Z}_q^n$. The challenger responds to each query as follows:
   a) It samples $\boxed{\mathbf{s} \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathsf{td}_{\mathbf{B}}, \mathbf{z}, \chi)}$ and sends $\mathbf{s}$ to $\mathcal{A}$.

Lemma 6. *Suppose that $\chi, \sigma \geq \chi_0$, where χ_0 is the polynomial given in Lemma 3. We have $\mathsf{H}_0 \overset{s}{\approx} \mathsf{H}_1$.*

Proof. First we consider the matrix $\mathbf{B}$ in two games. In Game H_0, the challenger samples $(\mathbf{B}, \mathsf{td}_{\mathbf{B}}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$. In Game H_1, it samples $(\mathbf{B}_1, \mathsf{td}_{\mathbf{B}_1}), (\mathbf{B}_2, \mathsf{td}_{\mathbf{B}_2}) \leftarrow \mathsf{TrapGen}(1^n, 1^{m/2}, q)$ and sets $\mathbf{B} = [\mathbf{B}_1 \mid \mathbf{B}_2]$ instead. Since $\mathsf{TrapGen}$ outputs a matrix that is statistically close to a uniform one together with a trapdoor, the resulting matrix $\mathbf{B}$ in both games is statistically indistinguishable from each other.

Next we consider the matrix $\mathbf{T}$. It suffices to consider each block $\mathbf{T}_i$. In Game H_0, the block $\mathbf{T}_i$ is distributed as a discrete Gaussian distribution with parameter σ conditioned on

$$\mathbf{B}\mathbf{T}_i + \mathbf{W}_i\mathbf{T}_{2m^2+1} = \mathbf{e}_i \otimes \mathbf{G}. \tag{3}$$

In Game H_1, the matrices $\mathbf{T}_i^{\mathsf{up}}$ and $\mathbf{T}_i^{\mathsf{down}}$ are sampled from discrete Gaussian distribution with parameter σ satisfying $\mathbf{B}_2\mathbf{T}_i^{\mathsf{down}} = \mathbf{e}_i \otimes \mathbf{G} - \mathbf{W}_i\mathbf{T}_{2m^2+1} - \mathbf{B}_1\mathbf{T}_i^{\mathsf{up}}$, which aligns with (3) if we write $\mathbf{B} = [\mathbf{B}_1 \mid \mathbf{B}_2]$ and $\mathbf{T}_i = \begin{bmatrix} \mathbf{T}_i^{\mathsf{up}} \\ \mathbf{T}_i^{\mathsf{down}} \end{bmatrix}$.

We now consider the matrix $\mathbf{S}$ in two games. In Game H_0, the challenger samples $\mathbf{R} \leftarrow \{-1,1\}^{m\times k}$ and sets $\mathbf{S} = \mathbf{B}\mathbf{R}$. In Game H_1, it samples $\mathbf{R}_1, \mathbf{R}_2 \leftarrow \{-1,1\}^{m/2\times k}$ independently and sets $\mathbf{S} = \mathbf{B}_1\mathbf{R}_1 + \mathbf{B}_2\mathbf{R}_2$. If we write $\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{bmatrix} \in \{-1,1\}^{m\times k}$, then the distribution of $\mathbf{S} = \mathbf{B}\mathbf{R}$ in Game H_1 is identical to that in Game H_0.

Finally, it suffices to consider the answers to preimage queries. In Game H_0, on input $\mathbf{z} \in \mathbb{Z}_q^n$ the challenger returns $\mathbf{s} \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathsf{td}_{\mathbf{B}}, \mathbf{z}, \chi)$, which is a discrete Gaussian preimage of $\mathbf{z}$ under $\mathbf{B}$. In Game H_1, on input the same $\mathbf{z}$ it samples $\mathbf{s}_1 \leftarrow \chi^{m/2}$ and then

$$\mathbf{s}_2 \leftarrow \mathsf{SamplePre}(\mathbf{B}_2, \mathsf{td}_{\mathbf{B}_2}, \mathbf{z} - \mathbf{B}_1\mathbf{s}_1, \chi),$$

and outputs $\mathbf{s} = \begin{bmatrix}\mathbf{s}_1 \\ \mathbf{s}_2\end{bmatrix}$. By the preimage-sampling process and the well-sampleness of the preimage with respect to $\mathsf{SamplePre}$, this procedure yields exactly the same discrete Gaussian distribution over all solutions $\mathbf{s} \in \mathbb{Z}_q^m$ to $\mathbf{B}\mathbf{s} = \mathbf{z} \bmod q$, which is identical to that in H_0.

Putting everything together, we see that the joint distribution of $(\mathsf{pp}, \mathbf{S})$ and all oracle answers in Game H_1 is statistically indistinguishable from that in Game H_0. □

Lemma 7. *Suppose that $m \geq m_0$, where m_0 is the polynomials given in Lemma 3. We have $\mathsf{H}_1 \overset{s}{\approx} \mathsf{H}_2$.*

Proof. This follows directly from the well-sampleness of $\mathsf{TrapGen}$: in Game H_1 the matrix $\mathbf{B}_1$ output by $\mathsf{TrapGen}$ is statistically indistinguishable from a uniform matrix in $\mathbb{Z}_q^{n \times m/2}$, and its trapdoor $\mathsf{td}_{\mathbf{B}_1}$ is neither used nor revealed. All other values are generated identically in the two games. Moreover, all subsequent steps depend on $\mathbf{B}_1$ itself or $(\mathbf{B}_2, \mathsf{td}_{\mathbf{B}_2})$ in the same way in both games. Hence the lemma holds. □

Lemma 8. *Suppose that $m \geq 2n \log q + \omega(\log n)$. Let $k = k(n)$ be some polynomial. We have $\mathsf{H}_2 \overset{s}{\approx} \mathsf{H}_3$.*

Proof. Suppose there exists an adversary $\mathcal{A}$ that distinguishes between Hybrid H_2 and H_3 with non-negligible probability. We construct an adversary $\mathcal{B}$ that violates the leftover hash lemma (Lemma 4). The algorithm $\mathcal{B}$ proceeds as follows:

- The algorithm $\mathcal{B}$ receives a challenge $(\mathbf{B}_1, \mathbf{S}')$, where $\mathbf{B}_1 \xleftarrow{\$} \mathbb{Z}_q^{n \times m/2}$ and $\mathbf{S}'$ is either sampled from $\mathbf{S}' = \mathbf{B}_1\mathbf{R}_1$ with $\mathbf{R}_1 \xleftarrow{\$} \{-1, 1\}^{m/2 \times k}$ or $\mathbf{S}' \xleftarrow{\$} \mathbb{Z}_q^{n \times k}$.
- It samples $(\mathbf{B}_2, \mathsf{td}_{\mathbf{B}_2}) \leftarrow \mathsf{TrapGen}(1^n, 1^{m/2}, q)$ and sets $\mathbf{B} = [\mathbf{B}_1 \mid \mathbf{B}_2]$.
- It samples $\mathbf{W} = \begin{bmatrix}\mathbf{W}_1 \\ \vdots \\ \mathbf{W}_{2m^2}\end{bmatrix} \xleftarrow{\$} \mathbb{Z}_q^{2m^2 n \times m}$, where $\mathbf{W}_i \in \mathbb{Z}_q^{n \times m}$ for each $i \in [2m^2]$.
- It samples $\mathbf{T}_{2m^2+1} \leftarrow D_{\mathbb{Z},\sigma}^{m \times 2m^3}$. For each $i \in [2m^2]$, it samples $\mathbf{T}_i^{\mathsf{up}} \leftarrow D_{\mathbb{Z},\sigma}^{m/2 \times 2m^3}$, and $\mathbf{T}_i^{\mathsf{down}} \leftarrow \mathsf{SamplePre}(\mathbf{B}_2, \mathsf{td}_{\mathbf{B}_2}, \mathbf{e}_i \otimes \mathbf{G} - \mathbf{W}_i\mathbf{T}_{2m^2+1} - \mathbf{B}_1\mathbf{T}_i^{\mathsf{up}}, \sigma)$, where $\mathbf{e}_i \in \{0,1\}^{2m^2}$ represents the vector with the i-th entry 1 and the other entries 0. It sets $\mathbf{T}_i = \begin{bmatrix}\mathbf{T}_i^{\mathsf{up}} \\ \mathbf{T}_i^{\mathsf{down}}\end{bmatrix}$ and $\mathbf{T} = \begin{bmatrix}\mathbf{T}_1 \\ \vdots \\ \mathbf{T}_{2m^2}\end{bmatrix}$, and sets $\mathsf{pp} := (\mathbf{B}, \mathbf{W}, \mathbf{T})$.
- It then samples $\mathbf{R}_2 \xleftarrow{\$} \{-1, 1\}^{m/2 \times k}$, and $\mathbf{S} \leftarrow \mathbf{S}' + \mathbf{B}_2\mathbf{R}_2$.
- It sends $(\mathsf{pp}, \mathbf{B}, \mathbf{S})$ to the adversary $\mathcal{A}$.
- It responds to each query $\mathbf{z} \in \mathbb{Z}_q^n$ given by the adversary $\mathcal{A}$ as follows:
  - It samples $\mathbf{s}_1 \leftarrow \chi^{m/2}$, and samples $\mathbf{s}_2 \leftarrow \mathsf{SamplePre}(\mathbf{B}_2, \mathsf{td}_{\mathbf{B}_2}, \mathbf{z} - \mathbf{B}_1\mathbf{s}_1, \chi)$ and answers with $\mathbf{s} = [\begin{smallmatrix}\mathbf{s}_1 \\ \mathbf{s}_2\end{smallmatrix}]$.
- It outputs whatever the adversary $\mathcal{A}$ outputs.

From the definition of the algorithm $\mathcal{B}$, we notice that when $\mathbf{S}' = \mathbf{B}_1\mathbf{R}_1$ with $\mathbf{R}_1 \xleftarrow{\$} \{-1, 1\}^{m/2 \times k}$, the adversary $\mathcal{B}$ simulates the challenger in Game H_2 perfectly, when $\mathbf{S}' \xleftarrow{\$} \mathbb{Z}_q^{n \times k}$, the adversary $\mathcal{B}$ simulates the challenger in Game H_3 perfectly. Hence the adversary $\mathbf{B}$ in the leftover hash lemma game has the same advantage as that of the adversary $\mathcal{A}$ of distinguish between Game H_2 and H_3. By the leftover hash lemma (Lemma 4) and the assumption $m/2 \overset{s}{\geq} 2n \log q + \omega(\log n)$, this advantage must be negligible, contradicting our assumption on $\mathcal{A}$. Hence $\mathsf{H}_2 \overset{s}{\approx} \mathsf{H}_3$. □

Lemma 9. *We have $\mathsf{H}_3 \equiv \mathsf{H}_4$.*

Proof. The difference between Hybrid H_3 and Hybrid H_4 is merely syntactic. In Game H_3, $\mathbf{S}' \xleftarrow{\$} \mathbb{Z}_q^{n\times k}$ is uniform and $\mathbf{B}_2\mathbf{R}_2$ is independent of S', so $\mathbf{S} = \mathbf{S}' + \mathbf{B}_2\mathbf{R}_2$ is also uniform over $\mathbb{Z}_q^{n\times k}$. In Game H_4, $\mathbf{S}$ is sampled directly and uniformly from $\mathbb{Z}_q^{n\times k}$. Thus $\mathbf{S}$ has exactly the same distribution in both games, and all other variables are generated identically. Hence $\mathsf{H}_3 \equiv \mathsf{H}_4$. □

Lemma 10. *Suppose that $m \geq m_0$, where m_0 is the polynomials given in Lemma 3. We have $\mathsf{H}_4 \overset{s}{\approx} \mathsf{H}_5$.*

Proof. The lemma follows the same reason as Lemma 7. □

Lemma 11. *Suppose that $\chi, \sigma \geq \chi_0$, where χ_0 is the polynomial given in Lemma 3. We have $\mathsf{H}_5 \overset{s}{\approx} \mathsf{H}_6$.*

Proof. The lemma follows the same reason as Lemma 6. □

By combining Lemmas 6–11, we can conclude that $\mathsf{H}_0 \overset{s}{\approx} \mathsf{H}_6$, which completes the proof of the theorem. □

A. Matrix commitment

Our scheme relies on the matrix commitment presented by [Wee25]. We refer to the formal description of the algorithm to [Wee25]. Roughly, the matrix commitment scheme first gives the public parameters $\mathsf{pp} := (\mathbf{B}, \mathbf{W}, \mathbf{T})$ which are sampled according to the same procedure as in the succinct LWE assumption. It then commits to a matrix $\mathbf{M} \in \mathbb{Z}_q^{n\times L}$ and the commitment is a matrix $\mathbf{C} \in \mathbb{Z}_q^{n\times m}$. The verification matrix $\mathbf{V}_L \in \mathbb{Z}_q^{n\times L}$ only depends on the width of the matrix $\mathbf{M}$. The opening matrix $\mathbf{Z} \in \mathbb{Z}_q^{n\times L}$ is such that

$$\mathbf{C} \cdot \mathbf{V}_L = \mathbf{M} - \mathbf{B} \cdot \mathbf{Z}.$$

Lemma 12 (Matrix Commitment [Wee25])**.** *Let n, m, q be lattice parameters with $m \geq 2n \log q$. There exists a tuple of deterministic efficient algorithms $(\mathsf{Com}^{\mathsf{mx}}, \mathsf{Ver}^{\mathsf{mx}}, \mathsf{Open}^{\mathsf{mx}})$ with the following syntax:*

- $\mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{M}) \to \mathbf{C}$*: On input the public parameter* pp*, and a matrix $\mathbf{M} \in \mathbb{Z}_q^{n\times L}$, the algorithm outputs a matrix $\mathbf{C} \in \mathbb{Z}_q^{n\times m}$.*
- $\mathsf{Ver}^{\mathsf{mx}}(\mathsf{pp}, 1^L) \to \mathbf{V}_L$*: On input the public parameters* pp *and the length parameter L, the algorithm outputs a matrix $\mathbf{V}_L \in \mathbb{Z}_q^{m\times L}$.*
- $\mathsf{Open}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{M}) \to \mathbf{Z}$*: On input the public parameters* pp*, and a matrix $\mathbf{M} \in \mathbb{Z}_q^{n\times L}$, the algorithm outputs a matrix $\mathbf{Z} \in \mathbb{Z}_q^{m\times L}$.*

For all $\mathsf{pp} = (\mathbf{B}, \mathbf{W}, \mathbf{T})$, where $\mathbf{B} \in \mathbb{Z}_q^{n\times m}, \mathbf{W} \in \mathbb{Z}_q^{2m^2 n\times m}, \mathbf{T} \in \mathbb{Z}_q^{(2m^2+1)m\times 2m^3}$ are such that $[\mathbf{I}_{2m^2} \otimes \mathbf{B} \mid \mathbf{W}] \cdot \mathbf{T} = \mathbf{I}_{2m^2} \otimes \mathbf{G}$, all $L \in \mathbb{N}$, all matrices $\mathbf{M} \in \mathbb{Z}_q^{n\times L}$, the matrices $\mathbf{C} \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{M}), \mathbf{V}_L \leftarrow \mathsf{Ver}^{\mathsf{mx}}(\mathsf{pp}, 1^L), \mathbf{Z} \leftarrow \mathsf{Open}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{M})$ satisfy

$$\begin{aligned} &\mathbf{C} \cdot \mathbf{V}_L = \mathbf{M} - \mathbf{B} \cdot \mathbf{Z}, \\ &\|\mathbf{V}_L\| \leq O(\|\mathbf{T}\|^4 \cdot m^4 \log q), \quad \|\mathbf{Z}\| \leq O(\|\mathbf{T}\| \cdot m^7 \log q \log L). \end{aligned}$$

IV. LSSS AND CP-ABE FOR LSSS-REALIZABLE ACCESS STRUCTURES

In this section, we first introduce the notion of Linear Secret Sharing Scheme (LSSS) which we use to describe the access structure in our construction, and the ciphertext-policy attribute-based encryption (CP-ABE).

Definition 13 (Access Structure, [Bei96])**.** *Let S be a set and 2^S be the power set of S, i.e., the collection of all subsets of S. An* access structure *on S is a set $\mathbb{A} \subseteq 2^S \setminus \{\varnothing\}$, consisting of some non-empty subsets of S. A subset $A \in 2^S$ is called* authorized *if $A \in \mathbb{A}$, and* unauthorized *otherwise.*

An access structure is called monotone *if it satisfies the following condition: for all subsets $B, C \in 2^S$, if $B \in \mathbb{A}$ and $B \subseteq C$, then $C \in \mathbb{A}$. In other words, adding more elements to an authorized subset does not invalidate its authorization.*

In the following, we present a result given by [DKW21], which shows that there exists a non-monotone $\{0,1\}$-LSSS for access structure represented by a Boolean formula.

Lemma 14 ([DKW21])**.** *For any access structure $\mathbb{A}$ which is described by a Boolean formula, there is a deterministic polynomial time algorithm for generating $(\mathbf{M}, \rho)$, where $\mathbf{M} \in \{-1,0,1\}^{\ell \times d}$ and $\rho : [\ell] \to [2n]$ satisfying that*

1) $(\mathbf{M}, \rho)$ yields a non-monotone $\{0,1\}$-LSSS for $\mathbb{A}$, namely
- *For $S \subseteq [n]$, let $\hat{S} = S \cup \{i \in [n+1, \ldots, 2n] \mid i - n \notin S\} \subseteq [2n]$. Let $\mathbf{M}_{\hat{S}}$ be the submatrix that consists of all the rows of $\mathbf{M}$ with row indices in $\rho^{-1}(\hat{S})$. For any authorized set of parties $S \subseteq [n]$, there is a linear combination of the rows of $\mathbf{M}_{\hat{S}}$ that gives $(1,0,\ldots,0) \in \mathbb{Z}_q^d$. Moreover, the coefficients in this linear combination are from $\{0,1\}$.*
- *For any unauthorized set of parties $S \subseteq [n]$, there is no linear combination of the rows of $\mathbf{M}_{\hat{S}}$ that gives $(1,0,\ldots,0) \in \mathbb{Z}_q^d$. Moreover, there exists a vector $\mathbf{d} \in \mathbb{Z}_q^d$, such that its first component $d_1 = 1$ and $\mathbf{M}_{\hat{S}}\mathbf{d} = \mathbf{0}$.*

2) For any unauthorized set of parties $S \subseteq [n]$, all of the rows of $\mathbf{M}_{\hat{S}}$ are linearly independent.

Here we present the notion of ciphertext-policy attribute-based encryption (CP-ABE) for LSSS.

Definition 15 (CP-ABE for LSSS)**.** *Let λ be the security parameter, and $n = n(\lambda), q = q(\lambda)$ be lattice parameters. A* ciphertext-policy attribute-based encryption (CP-ABE) scheme *for access structures captured by a* linear secret sharing scheme (LSSS) *over some finite field $\mathbb{Z}_q$ is defined as four efficient algorithms $\Pi_{\mathsf{CP\text{-}ABE}} = (\mathsf{Setup}, \mathsf{Keygen}, \mathsf{Enc}, \mathsf{Dec})$. These algorithms proceed as follows:*
- $\mathsf{Setup}(1^\lambda, \mathbb{U}) \to (\mathsf{pk}, \mathsf{msk})$*: The setup algorithm takes as input the security parameter λ and an attribute universe $\mathbb{U}$, and outputs the public parameters* pk *and a master secret key* msk*. We assume that* pk *specifies the attribute universe $\mathbb{U}$.*
- $\mathsf{Keygen}(\mathsf{msk}, U) \to \mathsf{sk}_U$*: The key generation algorithm takes as input the master secret key* msk *and an attribute set $U \subseteq \mathbb{U}$. It outputs a secret key sk_U associated with attribute set U. Without loss of generality, we assume that the secret key sk_U implicitly contains the attribute set U.*
- $\mathsf{Enc}(\mathsf{pk}, \mathsf{msg}, (\mathbf{M}, \rho)) \to \mathsf{ct}$*: The encryption algorithm takes as input the public parameters* pk*, a message* msg*, and an* LSSS *access policy $(\mathbf{M}, \rho)$ such that $\mathbf{M}$ is a matrix over $\mathbb{Z}_q$ and ρ is a row-labeling function that assigns each row of $\mathbf{M}$ to an attribute in $\mathbb{U}$. It outputs a ciphertext* ct*.*
- $\mathsf{Dec}(\mathsf{pk}, \mathsf{sk}, (\mathbf{M}, \rho), \mathsf{ct}) \to \mathsf{msg}' / \perp$*: The decryption algorithm takes as input the public parameters* pk*, a secret key* sk*, an access structure $(\mathbf{M}, \rho)$, and a ciphertext* ct*. It outputs a value $\mathsf{msg}' \in \mathbb{Z}_q$ when the attributes in U satisfy the* LSSS *structure $(\mathbf{M}, \rho)$, i.e., the vector $(1,0,\ldots,0)$ lies in the span of the rows of $\mathbf{M}$ whose labels are in U. Otherwise, the decryption outputs $\perp$, indicating the failure of the decryption.*

Correctness. A CP-ABE scheme for LSSS-realizable access structures is said to be *correct* if for every $\lambda \in \mathbb{N}$, every attribute universe $\mathbb{U}$, every message msg, every LSSS access policy $(\mathbf{M}, \rho)$, and every attribute set $U \subseteq \mathbb{U}$ that satisfies the access policy $(\mathbf{M}, \rho)$, it holds that

$$\Pr\left[\mathsf{msg}' = \mathsf{msg} \;\middle|\; \begin{array}{c} (\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathbb{U}); \\ \mathsf{sk}_U \leftarrow \mathsf{Keygen}(\mathsf{msk}, U); \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{msg}, (\mathbf{M}, \rho)); \\ \mathsf{msg}' \leftarrow \mathsf{Dec}(\mathsf{pk}, \mathsf{sk}_U, (\mathbf{M}, \rho), \mathsf{ct}) \end{array}\right] \geq 1 - \mathrm{negl}(\lambda).$$

Selective Security. Here we define the selective security of a CP-ABE for LSSS-realizable access structures by a game between a challenger and an adversary. For a security parameter $\lambda \in \mathbb{N}$, the game proceeds as follows:
- **Setup Phase:** The adversary first sends an LSSS access policy $(\mathbf{M}, \rho)$. The challenger runs algorithm $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}$ and sends the public parameters pk to the adversary.

- **Key Query Phase 1:** The adversary can make polynomially many secret-key queries to the challenger. Each secret-key query is of the form $U \subseteq \mathbb{U}$ such that U does not satisfy the LSSS access policy $(\mathbf{M}, \rho)$. The challenger then runs $\mathsf{sk}_U \leftarrow \mathsf{Keygen}(\mathsf{msk}, U)$ and sends sk_U to the adversary.
- **Challenge Phase:** The challenger flips a coin $b \xleftarrow{\$} \{0,1\}$ and encrypts the message b with respect to the LSSS access policy $(\mathbf{M}, \rho)$ as $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, b, (\mathbf{M}, \rho))$. Then the challenger sends the ciphertext ct to the adversary.
- **Key Query Phase 2:** This phase is the same as Key Query Phase 1.
- **Guess Phase:** The adversary outputs a guess $b' \in \{0,1\}$ for the value of b.

The *advantage* of the adversary $\mathcal{A}$ in this game is defined by

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{CP\text{-}ABE}}(\lambda) := |\Pr[b = b'] - 1/2|.$$

Definition 16 (Selective security for CP-ABE for LSSS)**.** *A* CP-ABE *scheme for* LSSS*-realizable access structures is* selectively secure *if for any* PPT *adversary $\mathcal{A}$, there exists a negligible function* $\mathrm{negl}(\cdot)$ *such that* $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{CP\text{-}ABE}}(\lambda) \leq \mathrm{negl}(\lambda)$ *for all* $\lambda \in \mathbb{N}$.

V. CP-ABE FOR LSSS FROM THE ℓ-SUCCINCT LWE ASSUMPTION

In this section, we present our construction of a CP-ABE scheme supporting access structures represented by NC^1 circuits. As discussed in [DKW21], we achieve our goal by presenting a CP-ABE scheme supporting access structures realized by LSSS. More precisely, we construct a CP-ABE scheme for LSSS access policies $(\mathbf{M}, \rho)$, where every entry of $\mathbf{M}$ lies in $\{-1, 0, 1\}$ and the reconstruction coefficients are in $\{0, 1\}$. We then prove our construction achieves selective security as in Definition 16, which in turn yields a CP-ABE schemes for access structures represented by NC^1 circuits. We assume that all LSSS access policies in the scheme correspond to matrices $\mathbf{M}$ with $s_{\max}$ columns and an injective row-labeling function ρ. The bound $s_{\max}$ can also be viewed as a bound on the circuit size of the corresponding NC^1 circuits.

Construction 1. *Let λ be the security parameter, and $n = n(\lambda), q = q(\lambda), m = m(\lambda)$ be lattice parameters. We describe the algorithms $\Pi_{\mathsf{CP\text{-}ABE}} = (\mathsf{Setup}, \mathsf{Keygen}, \mathsf{Enc}, \mathsf{Dec})$ as follows:*

- $\mathsf{Setup}(1^\lambda, \mathbb{U}, s_{\max})$*: The global setup algorithm takes as input the security parameter λ, an attribute universe $\mathbb{U}$, and the maximum width of the* LSSS *matrix $s_{\max} = s_{\max}(\lambda)$. Let $N := |\mathbb{U}|$. We fix an arbitrary ordering of $\mathbb{U}$ and henceforth identify it with the index set $[N] := \{1, 2, \ldots, N\}$. With a slight abuse of notation, we write $u \in [N]$ to denote the u-th attribute in this ordering. First, it chooses the distributions $\sigma, \chi, \chi_1, \chi_s$. Next, it samples $(\mathbf{B}, \mathsf{td}_{\mathbf{B}}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q), \mathbf{W} \xleftarrow{\$} \mathbb{Z}_q^{2m^2 n \times m}$, and*

$$\mathbf{T} \leftarrow \mathsf{SamplePre}([\mathbf{I}_{2m^2} \otimes \mathbf{B} \mid \mathbf{W}], \begin{bmatrix} \mathbf{I}_{2m^2} \otimes \mathsf{td}_{\mathbf{B}} \\ \mathbf{0} \end{bmatrix}, \mathbf{I}_{2m^2} \otimes \mathbf{G}, \sigma).$$

  It then samples $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, $\mathbf{B}_i \xleftarrow{\$} \mathbb{Z}_q^{n \times (m+1)}$ for each $i \in \{2, \ldots, s_{\max}\}$, $\mathbf{D}_u \xleftarrow{\$} \mathbb{Z}_q^{n \times (m+1)}$ for each $u \in [N]$, $\mathbf{Q}_u \xleftarrow{\$} \mathbb{Z}_q^{n \times (m+1)}$ for each $u \in [N]$, and a vector $\mathbf{y} \xleftarrow{\$} \mathbb{Z}_q^n$. It sets $\mathsf{pp} := (\mathbf{B}, \mathbf{W}, \mathbf{T})$. It outputs the public parameters

$$\mathsf{pk} = (\mathsf{pp}, n, m, q, \sigma, \chi, \chi_1, \chi_s, \mathbf{A}, \{\mathbf{B}_i\}_{i \in \{2, \ldots, s_{\max}\}}, \{\mathbf{D}_u\}_{u \in [N]}, \{\mathbf{Q}_u\}_{u \in [N]}, \mathbf{y})$$

  and a master secret key $\mathsf{msk} = (\mathsf{pk}, \mathsf{td}_{\mathbf{B}})$.
- $\mathsf{Keygen}(\mathsf{msk}, U)$*: The key generation algorithm takes as input the master secret key* msk *and an attribute set $U \subseteq \mathbb{U}$. It first computes $\mathbf{V} \leftarrow \mathsf{Ver}^{\mathsf{mx}}(\mathsf{pp}, 1^{(m+1)N}) \in \mathbb{Z}_q^{m \times (m+1)N}$, and parses it as $\mathbf{V} = [\mathbf{V}_1 \mid \cdots \mid \mathbf{V}_N]$ with $\mathbf{V}_u \in \mathbb{Z}_q^{m \times (m+1)}$ for all $u \in [N]$. Then it samples a vector $\hat{\mathbf{t}} \leftarrow D_{\mathbb{Z},\chi}^m$ and sets the vector $\mathbf{t} = (1, \hat{\mathbf{t}}^\top)^\top \in \mathbb{Z}_q^{m+1}$. For each attribute $u \in U$, the algorithm samples $\hat{\mathbf{k}}_u \leftarrow D_{\mathbb{Z},\chi_s}^m$ and*

$$\tilde{\mathbf{k}}_u \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathsf{td}_{\mathbf{B}}, (\mathbf{A}\mathbf{V}_u + \mathbf{Q}_u)\mathbf{t} - \mathbf{B}\hat{\mathbf{k}}_u, \chi_1),$$

and sets $\mathbf{k}_u = \hat{\mathbf{k}}_u + \tilde{\mathbf{k}}_u$. *It outputs* $\mathsf{sk} = (\{\mathbf{k}_u\}_{u\in U}, \mathbf{t})$.

- $\mathsf{Enc}(\mathsf{pk}, \mathsf{msg}, (\mathbf{M}, \rho))$*: The encryption algorithm takes as input the public parameters* pk*, a message* $\mathsf{msg} \in \{0,1\}$*, and an* LSSS *access policy* $(\mathbf{M}, \rho)$*, where* $\mathbf{M} := (M_{i,j}) \in \{-1, 0, 1\}^{\ell \times s_{\max}}$ *and* $\rho : [\ell] \to [N]$*. The function* ρ *associates rows of* $\mathbf{M}$ *to attributes in* $\mathbb{U}$*. Assume that* ρ *is an injective function (hence* $\ell \le N = |\mathbb{U}|$*). The algorithm first computes*

$$\mathbf{U}_{\rho(i)} \leftarrow M_{i,1}(\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}) + \sum_{2\le j\le s_{\max}} M_{i,j}\mathbf{B}_j + \mathbf{Q}_{\rho(i)} \in \mathbb{Z}_q^{n\times(m+1)}, \qquad \forall i \in [\ell],$$
$$\mathbf{U}_u \leftarrow \mathbf{Q}_u + \mathbf{D}_u \in \mathbb{Z}_q^{n\times(m+1)}, \qquad \forall u \in [N] \setminus \rho([\ell]),$$

and sets $\mathbf{U} \leftarrow [\mathbf{U}_1 \mid \cdots \mid \mathbf{U}_N] \in \mathbb{Z}_q^{n\times(m+1)N}$*. It then computes*

$$\mathbf{C} \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{U}) \in \mathbb{Z}_q^{n\times m}$$

and samples $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$*,* $\mathbf{e}_1 \leftarrow D_{\mathbb{Z},\chi}^m, \mathbf{e}_2 \leftarrow D_{\mathbb{Z},\chi_s}^m, e_3 \leftarrow D_{\mathbb{Z},\chi_s}$*. It then computes ciphertext components* $\mathbf{c}_1 \in \mathbb{Z}_q^m, \mathbf{c}_2 \in \mathbb{Z}_q^m$ *and* $c_3 \in \mathbb{Z}_q$ *as follows:*

$$\mathbf{c}_1^\top \leftarrow \mathbf{s}^\top\mathbf{B} + \mathbf{e}_1^\top,$$
$$\mathbf{c}_2^\top \leftarrow \mathbf{s}^\top(\mathbf{A} + \mathbf{C}) + \mathbf{e}_2^\top,$$
$$c_3 \leftarrow \mathbf{s}^\top\mathbf{y} + \mathsf{msg} \cdot \lceil q/2 \rfloor + e_3.$$

Then the algorithm outputs the ciphertext $\mathsf{ct} = (\mathbf{c}_1, \mathbf{c}_2, c_3)$.

- $\mathsf{Dec}(\mathsf{pk}, \mathsf{sk}, (\mathbf{M}, \rho), \mathsf{ct})$*: The decryption algorithm takes as input the public parameters* pk*, a secret key* $\mathsf{sk} = (\{\mathbf{k}_u\}_{u\in U}, \mathbf{t})$ *for an attribute subset* $U \subseteq \mathbb{U}$*. If* $(1, 0, \ldots, 0)$ *is not in the span of the rows of* $\mathbf{M}$ *associated with* U*, then the decryption fails and outputs* $\perp$*. Otherwise, "let* $I \subseteq [\ell]$ *be the set of row indices such that* $I = \{i \in [\ell] \mid \rho(i) \in U\}$*. First, the algorithm finds the reconstruction coefficients* $\{w_i\}_{i\in I}$ *such that* $(1, 0, \ldots, 0) = \sum_{i\in I} w_i\mathbf{M}_i$*, where* $\mathbf{M}_i$ *is the* i*-th row vector of* $\mathbf{M}$*. It then computes*

$$\mathbf{V} \leftarrow \mathsf{Ver}^{\mathsf{mx}}(\mathsf{pp}, 1^{(m+1)N})$$

and parses it as $\mathbf{V} = [\mathbf{V}_1 \mid \mathbf{V}_2 \mid \cdots \mid \mathbf{V}_N]$ *where* $\mathbf{V}_i \in \mathbb{Z}_q^{m\times(m+1)}$*.*

$$\mathbf{U}_{\rho(i)} \leftarrow M_{i,1}(\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}) + \sum_{2\le j\le s_{\max}} M_{i,j}\mathbf{B}_j + \mathbf{Q}_{\rho(i)} \in \mathbb{Z}_q^{n\times(m+1)}, \qquad \forall i \in [\ell],$$
$$\mathbf{U}_u \leftarrow \mathbf{Q}_u + \mathbf{D}_u \in \mathbb{Z}_q^{n\times(m+1)}, \qquad \forall u \in [N] \setminus \rho([\ell]).$$

and sets $\mathbf{U} \leftarrow [\mathbf{U}_1 \mid \cdots \mid \mathbf{U}_N]] \in \mathbb{Z}_q^{n\times(m+1)N}$*. It then computes*

$$\mathbf{Z} \leftarrow \mathsf{Open}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{U}) \in \mathbb{Z}_q^{m\times(m+1)N},$$

and parses $\mathbf{Z} := [\mathbf{Z}_1 \mid \cdots \mid \mathbf{Z}_N]$ *where* $\mathbf{Z}_i \in \mathbb{Z}_q^{m\times(m+1)}$*. Finally, it computes*

$$\mu \leftarrow c_3 - \sum_{i\in I} w_i[(\mathbf{c}_2^\top\mathbf{V}_{\rho(i)} + \mathbf{c}_1^\top\mathbf{Z}_{\rho(i)})\mathbf{t} - \mathbf{c}_1^\top\mathbf{k}_{\rho(i)}] \bmod q,$$

and it outputs 0 *if* $-q/4 < \mu < q/4$ *and 1 otherwise.*

Parameters. The parameters are selected as follows:

- $q/4 > N \cdot \mathrm{poly}(\lambda, \log N, m) \cdot \sigma \cdot (\chi + \chi_s)$. (for correctness)
- $m > 2n\log q + \omega(\log n)$. (for leftover hash lemma)
- $\sigma = \mathrm{poly}(m, \lambda)$. (for succinct LWE)
- $\chi_1 > \mathrm{poly}(m, N, \sigma, \log q) \cdot \omega(\sqrt{\log n})$. (for preimage sampling)
- $\chi > \sqrt{m}\chi_1 \cdot \lambda^{\omega(1)}$. (for noise-smudging/security)

- $\chi_s \geq \chi \cdot (\chi_1 + \sigma) \cdot \log q \cdot \log N \cdot \mathrm{poly}(m) \cdot \lambda^{\omega(1)}$. (for noise-smudging/security)

For example, fix $0 < \varepsilon < 1$, where $2m^2$-succinct LWE is hard for 2^{n^ε} modulus-to-noise ratio. Specifically, we set

$$n = \mathrm{poly}(\lambda), \quad m = n \cdot \mathrm{poly}(\lambda), \quad \chi = \mathrm{poly}(n, N, \lambda) \cdot \lambda^{\omega(1)}, \quad q = \mathrm{poly}(n, N, \lambda) \cdot \chi \cdot \lambda^{\omega(1)}.$$

This gives the parameters of our $\mathsf{CP\text{-}ABE}$ scheme:

$$|\mathsf{pk}| = O(s_{\max}), \quad |\mathsf{ct}| = O(1), \quad |\mathsf{sk}| = O(N),$$

where $O(\cdot)$ hides the polynomial in λ. Note that N scales with the input length of the NC^1 circuit and $s_{\max}$ scales with the size of the circuit.

A. Correctness

Theorem 17 (Correctness)**.** *Suppose that $q > N \cdot \mathrm{poly}(\lambda, \log N, m) \cdot \sigma \cdot (\chi + \chi_s)$. The Construction 1 is correct as a* $\mathsf{CP\text{-}ABE}$ *scheme.*

Proof. We begin the proof by analyzing the term $\sum_{i \in I} w_i[(\mathbf{c}_2^\top \mathbf{V}_{\rho(i)} + \mathbf{c}_1^\top \mathbf{Z}_{\rho(i)})\mathbf{t} - \mathbf{c}_1^\top \mathbf{k}_{\rho(i)}]$. First, by the commitment scheme, we know for all $u \in [N]$, we have $\mathbf{C}\mathbf{V}_u = \mathbf{U}_u - \mathbf{B}\mathbf{Z}_u$, then

$$\begin{aligned}(\mathbf{c}_2^\top \mathbf{V}_{\rho(i)} + \mathbf{c}_1^\top \mathbf{Z}_{\rho(i)})\mathbf{t} &\approx \mathbf{s}^\top(\mathbf{A}\mathbf{V}_{\rho(i)} + \mathbf{C}\mathbf{V}_{\rho(i)} + \mathbf{B}\mathbf{Z}_{\rho(i)})\mathbf{t} \\ &= \mathbf{s}^\top(\mathbf{A}\mathbf{V}_{\rho(i)} + \mathbf{U}_{\rho(i)})\mathbf{t} \\ &= \mathbf{s}^\top(\mathbf{A}\mathbf{V}_{\rho(i)} + M_{i,1}(\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}) + \sum_{2 \leq j \leq s_{\max}} M_{i,j}\mathbf{B}_j + \mathbf{Q}_{\rho(i)})\mathbf{t},\end{aligned} \tag{4}$$

where "$\approx$" hides the error term. By the Keygen algorithm, we have for all $u \in U$, we have that $\mathbf{k}_u = \hat{\mathbf{k}}_u + \tilde{\mathbf{k}}_u$ and $\mathbf{B}\tilde{\mathbf{k}}_u = (\mathbf{A}\mathbf{V}_u + \mathbf{Q}_u)\mathbf{t} - \mathbf{B}\hat{\mathbf{k}}_u$. Therefore, for all $i \in I$, we have

$$\mathbf{c}_1^\top \mathbf{k}_{\rho(i)} \approx \mathbf{s}^\top \mathbf{B}\mathbf{k}_{\rho(i)} = \mathbf{s}^\top(\mathbf{A}\mathbf{V}_{\rho(i)} + \mathbf{Q}_{\rho(i)})\mathbf{t}. \tag{5}$$

Combining (4) and (5), we obtain that

$$\sum_{i \in I} w_i[(\mathbf{c}_2^\top \mathbf{V}_{\rho(i)} + \mathbf{c}_1^\top \mathbf{Z}_{\rho(i)})\mathbf{t} - \mathbf{c}_1^\top \mathbf{k}_{\rho(i)}] \approx \sum_{i \in I} \mathbf{s}^\top(w_i M_{i,1}(\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}) + \sum_{2 \leq j \leq s_{\max}} w_i M_{i,j}\mathbf{B}_j)\mathbf{t}. \tag{6}$$

From the LSSS access structure, if the rows of indices in I satisfy the policy $(\mathbf{M}, \rho)$, we have $\sum_{i \in I} w_i M_{i,1} = 1$ and $\sum_{i \in I} w_i M_{i,j} = 0$ for all $j \in \{2, \ldots, s_{\max}\}$. Since $\mathbf{t} = (1 \mid \hat{\mathbf{t}}^\top)^\top$, we have

$$\sum_{i \in I} w_i[(\mathbf{c}_2^\top \mathbf{V}_{\rho(i)} + \mathbf{c}_1^\top \mathbf{Z}_{\rho(i)})\mathbf{t} - \mathbf{c}_1^\top \mathbf{k}_{\rho(i)}] \approx \mathbf{s}^\top \mathbf{y}.$$

It follows that

$$c_3 - \sum_{i \in I} w_i[(\mathbf{c}_2^\top \mathbf{V}_{\rho(i)} + \mathbf{c}_1^\top \mathbf{Z}_{\rho(i)})\mathbf{t} - \mathbf{c}_1^\top \mathbf{k}_{\rho(i)}] \approx \mathsf{msg} \cdot \lceil q/2 \rfloor,$$

where the error term is given by

$$e_3 - \sum_{i \in I} w_i((\mathbf{e}_2^\top \mathbf{V}_{\rho(i)} + \mathbf{e}_1^\top \mathbf{Z}_{\rho(i)})\mathbf{t} - \mathbf{e}_1^\top \mathbf{k}_{\rho(i)}).$$

The following bounds hold except with negligible probability:

- $\|\mathbf{k}_{\rho(i)}\| \leq \sqrt{m}\chi_s$ for all $i \in I$: Since $\tilde{\mathbf{k}}_u \leftarrow D^m_{\mathbb{Z},\chi_1}$ conditioned on $(\mathbf{A}\mathbf{V}_u + \mathbf{Q}_u)\mathbf{t} - \mathbf{B}\hat{\mathbf{k}}_u = \mathbf{B}\tilde{\mathbf{k}}_u$, $\hat{\mathbf{k}}_u \leftarrow D^m_{\mathbb{Z},\chi_s}$, and by the noise-smudging lemma (Lemma 2), we obtain that $\hat{\mathbf{k}}_u + \tilde{\mathbf{k}}_u \overset{s}{\approx} \hat{\mathbf{k}}_u$, and hence $\|\mathbf{k}_u\| \leq \sqrt{m}\chi_s$ for all $u \in U$.
- $\|\mathbf{e}_1\| \leq \sqrt{m}\chi, \|\mathbf{e}_2\| \leq \sqrt{m}\chi_s, \|e_3\| \leq \chi_s$: It follows from the fact that $\mathbf{e}_1 \leftarrow D^m_{\mathbb{Z},\chi}, \mathbf{e}_2 \leftarrow D^m_{\mathbb{Z},\chi_s}, e_3 \leftarrow D_{\mathbb{Z},\chi_s}$ and Lemma 1.

- $\|\mathbf{V}_u\| \leq O(\sigma \cdot \mathrm{poly}(m) \cdot \log q), \|\mathbf{Z}_u\| \leq O(\sigma \cdot \mathrm{poly}(m) \cdot \log q \cdot \log N)$: It follows from the matrix commitment (Lemma 12) and the matrix $\mathbf{T}$ is sampled as $\mathbf{T} \leftarrow \mathsf{SamplePre}([\mathbf{I}_{2m^2} \otimes \mathbf{B} \mid \mathbf{W}], \begin{bmatrix} \mathbf{I}_{2m^2} \otimes \mathsf{td}_{\mathbf{B}} \\ \mathbf{0} \end{bmatrix}, \mathbf{I}_{2m^2} \otimes \mathbf{G}, \sigma)$.
- $\|\mathbf{t}\| \leq \sqrt{m}\chi$: It follows from the fact that $\hat{\mathbf{t}} \leftarrow D^m_{\mathbb{Z},\chi}$ and $\mathbf{t} \leftarrow (1 \mid \hat{\mathbf{t}}^\top)^\top$.

By combining the results above and the fact that $w_i \in \{0,1\}$ by our restriction on LSSS, we have that the error term is bounded by

$$N\,\mathrm{poly}(\lambda, \log N, m) \cdot \sigma \cdot (\chi + \chi_s).$$

Therefore, the correctness holds under the condition presented in the theorem. □

B. Selective Security

Theorem 18. *Suppose the $2m^2$-succinct LWE assumption holds, then Construction 1 is selectively secure.*

Proof. We prove the selective security of Construction 1 by giving a sequence of hybrid games. We claim that each adjacent pair of games is indistinguishable. Game H_0 corresponds to the real selective-security game (defined in 16), while in the final game the challenge bit is information-theoretically hidden, hence the adversary's advantage is 0. Therefore, the adversary's advantage in the real game is negligible.

Throughout the hybrid games, the adversary $\mathcal{A}$ first commits to an access policy $(\mathbf{M}, \rho)$, in which $\mathbf{M} = (M_{i,j}) \in \{-1, 0, 1\}^{\ell \times s_{\max}}$ and $\rho : [\ell] \to [N]$ is an injective row-labeling function for $\mathbf{M}$. Then the challenger sends the public parameters to $\mathcal{A}$. In the key query phase, the adversary can query secret keys corresponding to the attribute sets U such that the row indices in $\rho^{-1}(U)$ are not authorized with respect to the access policy $(\mathbf{M}, \rho)$. In the challenge phase, the challenger tosses a coin $b \xleftarrow{\$} \{0,1\}$ and provides the ciphertext of the bit b under the committed policy $(\mathbf{M}, \rho)$. Finally, in the guess phase, the adversary outputs a guess bit $\hat{b}$ for b.

Game H_0. This game corresponds to the real selective security game for our CP-ABE scheme.

Setup phase:

1. $(\mathbf{B}, \mathsf{td}_{\mathbf{B}}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$.
2. $\mathbf{W} \xleftarrow{\$} \mathbb{Z}_q^{2m^2 n \times m}$.
3. $\mathbf{T} \leftarrow \mathsf{SamplePre}([\mathbf{I}_{2m^2} \otimes \mathbf{B} \mid \mathbf{W}], \begin{bmatrix} \mathbf{I}_{2m^2} \otimes \mathsf{td}_{\mathbf{B}} \\ \mathbf{0} \end{bmatrix}, \mathbf{I}_{2m^2} \otimes \mathbf{G}, \sigma)$.
4. $\mathsf{pp} := (\mathbf{B}, \mathbf{W}, \mathbf{T})$.
5. $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$.
6. $\mathbf{y} \xleftarrow{\$} \mathbb{Z}_q^n$.
7. $\{\mathbf{B}_i\}_{i \in \{2,\ldots,s_{\max}\}} \xleftarrow{\$} \mathbb{Z}_q^{n \times (m+1)}$.
8. $\{\mathbf{D}_u\}_{u \in [N]} \xleftarrow{\$} \mathbb{Z}_q^{n \times (m+1)}$.
9. $\{\mathbf{Q}_u\}_{u \in [N]} \xleftarrow{\$} \mathbb{Z}_q^{n \times (m+1)}$.
10. $\mathsf{pk} = (\mathsf{pp}, n, m, q, \sigma, \chi, \chi_1, \chi_s, \mathbf{A}, \{\mathbf{B}_i\}_{i \in \{2,\ldots,s_{\max}\}}, \{\mathbf{D}_u\}_{u \in [N]}, \{\mathbf{Q}_u\}_{u \in [N]}, \mathbf{y})$.

Secret-key query for attribute set U:

1. $\mathbf{V} = [\mathbf{V}_1 \mid \cdots \mid \mathbf{V}_N] \leftarrow \mathsf{Ver}^{\mathsf{mx}}(\mathsf{pp}, 1^{(m+1)N})$.
2. $\hat{\mathbf{t}} \leftarrow D_{\mathbb{Z},\chi}^m$.
3. $\mathbf{t} = (1, \hat{\mathbf{t}}^\top)^\top \in \mathbb{Z}_q^{m+1}$.
4. $\{\hat{\mathbf{k}}_u\}_{u \in U} \leftarrow D_{\mathbb{Z},\chi_s}^m$.
5. $\forall u \in U$: $\tilde{\mathbf{k}}_u \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathsf{td}_{\mathbf{B}}, (\mathbf{A}\mathbf{V}_u + \mathbf{Q}_u)\mathbf{t} - \mathbf{B}\hat{\mathbf{k}}_u, \chi_1)$.
6. $\forall u \in U$: $\mathbf{k}_u \leftarrow \hat{\mathbf{k}}_u + \tilde{\mathbf{k}}_u$.
7. $\mathsf{sk} \leftarrow (\{\mathbf{k}_u\}_{u \in U}, \mathbf{t})$.

Challenge phase:

1. $\mathsf{msg} \xleftarrow{\$} \{0,1\}$.
2. $\forall i \in [\ell] : \mathbf{U}_{\rho(i)} \leftarrow M_{i,1}(\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}) + \sum_{2 \le j \le s_{\max}} M_{i,j}\mathbf{B}_j + \mathbf{Q}_{\rho(i)}$.
3. $\forall u \in [N] \setminus \rho([\ell]) : \mathbf{U}_u \leftarrow \mathbf{Q}_u + \mathbf{D}_u$.
4. $\mathbf{U} \leftarrow [\mathbf{U}_1 \mid \cdots \mid \mathbf{U}_N]$.
5. $\mathbf{C} \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{U})$.
6. $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$.
7. $\mathbf{e}_1 \leftarrow D_{\mathbb{Z},\chi}^m$.
8. $\mathbf{e}_2 \leftarrow D_{\mathbb{Z},\chi_s}^m$.
9. $e_3 \leftarrow D_{\mathbb{Z},\chi_s}$.
10. $\mathbf{c}_1^\top \leftarrow \mathbf{s}^\top\mathbf{B} + \mathbf{e}_1^\top$.
11. $\mathbf{c}_2^\top \leftarrow \mathbf{s}^\top(\mathbf{A} + \mathbf{C}) + \mathbf{e}_2^\top$.
12. $c_3 \leftarrow \mathbf{s}^\top\mathbf{y} + \mathsf{msg} \cdot \lceil q/2 \rfloor + e_3$.
13. $\mathsf{ct} \leftarrow (\mathbf{c}_1, \mathbf{c}_2, c_3)$.

Game H_1. This game is defined identically to Game H_0, except for how the matrix $\mathbf{A}$ is sampled and how the matrices $\{\mathbf{U}_u\}_{u\in[N]}$ and $\{\mathbf{Q}_u\}_{u\in[N]}$ are generated during the setup phase. The indistinguishability between Game H_0 and Game H_1 (Lemma 19) follows from the fact that this change is merely syntactic.

Setup phase:

1. $(\mathbf{B}, \mathsf{td}_{\mathbf{B}}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$.
2. $\mathbf{W} \xleftarrow{\$} \mathbb{Z}_q^{2m^2 n \times m}$.
3. $\mathbf{T} \leftarrow \mathsf{SamplePre}([\mathbf{I}_{2m^2} \otimes \mathbf{B} \mid \mathbf{W}], \begin{bmatrix} \mathbf{I}_{2m^2} \otimes \mathsf{td}_{\mathbf{B}} \\ \mathbf{0} \end{bmatrix}, \mathbf{I}_{2m^2} \otimes \mathbf{G}, \sigma)$.
4. $\mathsf{pp} := (\mathbf{B}, \mathbf{W}, \mathbf{T})$.
5. $\boxed{\mathbf{A}' \xleftarrow{\$} \mathbb{Z}_q^{n\times m}}$.
6. $\mathbf{y} \xleftarrow{\$} \mathbb{Z}_q^n$.
7. $\{\mathbf{B}_i\}_{i\in\{2,\ldots,s_{\max}\}} \xleftarrow{\$} \mathbb{Z}_q^{n\times(m+1)}$.
8. $\boxed{\{\mathbf{U}_u\}_{u\in[N]} \xleftarrow{\$} \mathbb{Z}_q^{n\times(m+1)}}$.
9. $\{\mathbf{D}_u\}_{u\in[N]} \xleftarrow{\$} \mathbb{Z}_q^{n\times(m+1)}$.
10. $\forall u \in \rho([\ell]) : \boxed{\mathbf{Q}_u \leftarrow \mathbf{U}_u - (M_{\rho^{-1}(u),1}(\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}) + \sum_{2\le j\le s_{\max}} M_{\rho^{-1}(u),j}\mathbf{B}_j)}$.
11. $\forall u \in [N] \setminus \rho([\ell]) : \boxed{\mathbf{Q}_u \leftarrow \mathbf{U}_u - \mathbf{D}_u}$.
12. $\mathbf{U} \leftarrow [\mathbf{U}_1 \mid \cdots \mid \mathbf{U}_N]$.
13. $\mathbf{C} \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{U})$.
14. $\boxed{\mathbf{A} \leftarrow \mathbf{A}' - \mathbf{C} \in \mathbb{Z}_q^{n\times m}}$.
15. $\mathsf{pk} = (\mathsf{pp}, n, m, q, \sigma, \chi, \chi_1, \chi_s, \mathbf{A}, \{\mathbf{B}_i\}_{i\in\{2,\ldots,s_{\max}\}}, \{\mathbf{D}_u\}_{u\in[N]}, \{\mathbf{Q}_u\}_{u\in[N]}, \mathbf{y})$.

Secret-key query for attribute set U:

1. $\mathbf{V} = [\mathbf{V}_1 \mid \cdots \mid \mathbf{V}_N] \leftarrow \mathsf{Ver}^{\mathsf{mx}}(\mathsf{pp}, 1^{(m+1)N})$.
2. $\hat{\mathbf{t}} \leftarrow D_{\mathbb{Z},\chi}^m$.
3. $\mathbf{t} = (1, \hat{\mathbf{t}}^\top)^\top \in \mathbb{Z}_q^{m+1}$.
4. $\{\hat{\mathbf{k}}_u\}_{u\in U} \leftarrow D_{\mathbb{Z},\chi_s}^m$.
5. $\forall u \in U$: $\tilde{\mathbf{k}}_u \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathsf{td}_{\mathbf{B}}, (\mathbf{A}\mathbf{V}_u + \mathbf{Q}_u)\mathbf{t} - \mathbf{B}\hat{\mathbf{k}}_u, \chi_1)$.
6. $\forall u \in U$: $\mathbf{k}_u \leftarrow \hat{\mathbf{k}}_u + \tilde{\mathbf{k}}_u$.
7. $\mathsf{sk} \leftarrow (\{\mathbf{k}_u\}_{u\in U}, \mathbf{t})$.

Challenge phase:

1. $\mathsf{msg} \xleftarrow{\$} \{0,1\}$.
2. $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$.
3. $\mathbf{e}_1 \leftarrow D_{\mathbb{Z},\chi}^m$.
4. $\mathbf{e}_2 \leftarrow D_{\mathbb{Z},\chi_s}^m$.
5. $e_3 \leftarrow D_{\mathbb{Z},\chi_s}$.
6. $\mathbf{c}_1^\top \leftarrow \mathbf{s}^\top\mathbf{B} + \mathbf{e}_1^\top$.
7. $\mathbf{c}_2^\top \leftarrow \mathbf{s}^\top(\mathbf{A} + \mathbf{C}) + \mathbf{e}_2^\top$.
8. $c_3 \leftarrow \mathbf{s}^\top\mathbf{y} + \mathsf{msg} \cdot \lceil q/2 \rfloor + e_3$.
9. $\mathsf{ct} \leftarrow (\mathbf{c}_1, \mathbf{c}_2, c_3)$.

Game H_2. This game is defined identically to Game H_1, except for the way the challenger generates the matrices $\mathbf{A}'$, $\{\mathbf{U}_u\}_{u\in[N]}$ and the vector $\mathbf{y}$ during the setup phase. The indistinguishability between Games H_1 and H_2 (Lemma 20) follows from the leftover hash lemma with trapdoor (Lemma 5).

Setup phase:

1. $(\mathbf{B}, \mathsf{td}_{\mathbf{B}}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$.
2. $\mathbf{W} \xleftarrow{\$} \mathbb{Z}_q^{2m^2 n \times m}$.
3. $\mathbf{T} \leftarrow \mathsf{SamplePre}([\mathbf{I}_{2m^2} \otimes \mathbf{B} \mid \mathbf{W}], \begin{bmatrix} \mathbf{I}_{2m^2} \otimes \mathsf{td}_{\mathbf{B}} \\ \mathbf{0} \end{bmatrix}, \mathbf{I}_{2m^2} \otimes \mathbf{G}, \sigma)$.
4. $\mathsf{pp} := (\mathbf{B}, \mathbf{W}, \mathbf{T})$.
5. $\boxed{\mathbf{R} \xleftarrow{\$} \{-1,1\}^{m\times m}, \{\mathbf{R}_u\}_{u\in[N]} \xleftarrow{\$} \{-1,1\}^{m\times(m+1)}, \mathbf{r} \xleftarrow{\$} \{-1,1\}^m}$.
6. $\boxed{\mathbf{A}' \leftarrow \mathbf{BR}}$.
7. $\boxed{\mathbf{y} \leftarrow \mathbf{Br}}$.
8. $\{\mathbf{B}_i\}_{i\in\{2,\ldots,s_{\max}\}} \xleftarrow{\$} \mathbb{Z}_q^{n\times(m+1)}$.
9. $\forall u \in [N]: \boxed{\mathbf{U}_u \leftarrow \mathbf{BR}_u}$.
10. $\{\mathbf{D}_u\}_{u\in[N]} \xleftarrow{\$} \mathbb{Z}_q^{n\times(m+1)}$.
11. $\forall u \in \rho([\ell]): \mathbf{Q}_u \leftarrow \mathbf{U}_u - (M_{\rho^{-1}(u),1}(\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}) + \sum_{2\le j\le s_{\max}} M_{\rho^{-1}(u),j}\mathbf{B}_j)$.
12. $\forall u \in [N] \setminus \rho([\ell]): \mathbf{Q}_u \leftarrow \mathbf{U}_u - \mathbf{D}_u$.
13. $\mathbf{U} \leftarrow [\mathbf{U}_1 \mid \cdots \mid \mathbf{U}_N]$.
14. $\mathbf{C} \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{U})$.
15. $\mathbf{A} \leftarrow \mathbf{A}' - \mathbf{C} \in \mathbb{Z}_q^{n\times m}$.
16. $\mathsf{pk} = (\mathsf{pp}, n, m, q, \sigma, \chi, \chi_1, \chi_s, \mathbf{A}, \{\mathbf{B}_i\}_{i\in\{2,\ldots,s_{\max}\}}, \{\mathbf{D}_u\}_{u\in[N]}, \{\mathbf{Q}_u\}_{u\in[N]}, \mathbf{y})$.

Secret-key query for attribute set U:

1. $\mathbf{V} = [\mathbf{V}_1 \mid \cdots \mid \mathbf{V}_N] \leftarrow \mathsf{Ver}^{\mathsf{mx}}(\mathsf{pp}, 1^{(m+1)N})$.
2. $\hat{\mathbf{t}} \leftarrow D_{\mathbb{Z},\chi}^m$.
3. $\mathbf{t} = (1, \hat{\mathbf{t}}^\top)^\top \in \mathbb{Z}_q^{m+1}$.
4. $\{\hat{\mathbf{k}}_u\}_{u\in U} \leftarrow D_{\mathbb{Z},\chi_s}^m$.
5. $\forall u \in U$: $\tilde{\mathbf{k}}_u \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathsf{td}_{\mathbf{B}}, (\mathbf{AV}_u + \mathbf{Q}_u)\mathbf{t} - \mathbf{B}\hat{\mathbf{k}}_u, \chi_1)$.
6. $\forall u \in U$: $\mathbf{k}_u \leftarrow \hat{\mathbf{k}}_u + \tilde{\mathbf{k}}_u$.
7. $\mathsf{sk} \leftarrow (\{\mathbf{k}_u\}_{u\in U}, \mathbf{t})$.

Challenge phase:

1. $\mathsf{msg} \xleftarrow{\$} \{0,1\}$.
2. $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$.
3. $\mathbf{e}_1 \leftarrow D_{\mathbb{Z},\chi}^m$.
4. $\mathbf{e}_2 \leftarrow D_{\mathbb{Z},\chi_s}^m$.
5. $e_3 \leftarrow D_{\mathbb{Z},\chi_s}$.
6. $\mathbf{c}_1^\top \leftarrow \mathbf{s}^\top\mathbf{B} + \mathbf{e}_1^\top$.
7. $\mathbf{c}_2^\top \leftarrow \mathbf{s}^\top(\mathbf{A} + \mathbf{C}) + \mathbf{e}_2^\top$.
8. $c_3 \leftarrow \mathbf{s}^\top\mathbf{y} + \mathsf{msg} \cdot \lceil q/2 \rfloor + e_3$.
9. $\mathsf{ct} \leftarrow (\mathbf{c}_1, \mathbf{c}_2, c_3)$.

Game H_3. This game is identical to Game H_2, except for how the challenger generates the responses to secret-key queries. The indistinguishability between Games H_2 and H_3 (Lemma 21) follows from the noise-smudging lemma (Lemma 2).

Setup phase:

1. $(\mathbf{B}, \mathsf{td}_{\mathbf{B}}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$.
2. $\mathbf{W} \xleftarrow{\$} \mathbb{Z}_q^{2m^2 n \times m}$.
3. $\mathbf{T} \leftarrow \mathsf{SamplePre}([\mathbf{I}_{2m^2} \otimes \mathbf{B} \mid \mathbf{W}], \begin{bmatrix} \mathbf{I}_{2m^2} \otimes \mathsf{td}_{\mathbf{B}} \\ \mathbf{0} \end{bmatrix}, \mathbf{I}_{2m^2} \otimes \mathbf{G}, \sigma)$.
4. $\mathsf{pp} := (\mathbf{B}, \mathbf{W}, \mathbf{T})$.
5. $\mathbf{R} \xleftarrow{\$} \{-1,1\}^{m \times m}, \{\mathbf{R}_u\}_{u \in [N]} \xleftarrow{\$} \{-1,1\}^{m \times (m+1)}, \mathbf{r} \xleftarrow{\$} \{-1,1\}^m$.
6. $\mathbf{A}' \leftarrow \mathbf{B}\mathbf{R}$.
7. $\mathbf{y} \leftarrow \mathbf{B}\mathbf{r}$.
8. $\{\mathbf{B}_i\}_{i \in \{2,\ldots,s_{\max}\}} \xleftarrow{\$} \mathbb{Z}_q^{n \times (m+1)}$.
9. $\forall i \in \{2,\ldots,s_{\max}\}$, $\mathbf{B}_i := [\mathbf{b}'_i \mid \mathbf{B}'_i]$ where $\mathbf{b}'_i \in \mathbb{Z}_q^n$, $\mathbf{B}'_i \in \mathbb{Z}_q^{n \times m}$.
10. $\forall u \in \rho([\ell])$: $\mathbf{N}_u \leftarrow \sum_{2 \le j \le s_{\max}} M_{\rho^{-1}(u),j} \mathbf{B}'_j$.
11. $\forall u \in [N] : \mathbf{U}_u \leftarrow \mathbf{B}\mathbf{R}_u$.
12. $\{\mathbf{D}_u\}_{u \in [N]} \xleftarrow{\$} \mathbb{Z}_q^{n \times (m+1)}$.
13. $\forall u \in [N]$, $\mathbf{D}_u := [\mathbf{d}'_u \mid \mathbf{D}'_u]$ where $\mathbf{d}'_u \in \mathbb{Z}_q^n$, $\mathbf{D}'_u \in \mathbb{Z}_q^{n \times m}$.
14. $\forall u \in \rho([\ell]) : \mathbf{Q}_u \leftarrow \mathbf{U}_u - (M_{\rho^{-1}(u),1}(\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}) + \sum_{2 \le j \le s_{\max}} M_{\rho^{-1}(u),j} \mathbf{B}_j)$.
15. $\forall u \in [N] \setminus \rho([\ell]) : \mathbf{Q}_u \leftarrow \mathbf{U}_u - \mathbf{D}_u$.
16. $\mathbf{U} \leftarrow [\mathbf{U}_1 \mid \cdots \mid \mathbf{U}_N]$.
17. $\mathbf{C} \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{U})$.
18. $\mathbf{A} \leftarrow \mathbf{A}' - \mathbf{C} \in \mathbb{Z}_q^{n \times m}$.
19. $\mathsf{pk} = (\mathsf{pp}, n, m, q, \sigma, \chi, \chi_1, \chi_s, \mathbf{A}, \{\mathbf{B}_i\}_{i \in \{2,\ldots,s_{\max}\}}, \{\mathbf{D}_u\}_{u \in [N]}, \{\mathbf{Q}_u\}_{u \in [N]}, \mathbf{y})$.

Secret-key query for attribute set U:

1. $\mathbf{V} = [\mathbf{V}_1 \mid \cdots \mid \mathbf{V}_N] \leftarrow \mathsf{Ver}^{\mathsf{mx}}(\mathsf{pp}, 1^{(m+1)N})$.
2. $\boxed{\mathbf{Z} = [\mathbf{Z}_1 \mid \cdots \mid \mathbf{Z}_N] \leftarrow \mathsf{Open}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{U})}$.
3. $\hat{\mathbf{t}} \leftarrow D^m_{\mathbb{Z},\chi}$.
4. $\mathbf{t} = (1, \hat{\mathbf{t}}^\top)^\top \in \mathbb{Z}_q^{m+1}$.
5. $\{\hat{\mathbf{k}}_u\}_{u \in U} \leftarrow D^m_{\mathbb{Z},\chi_s}$.
6. $\forall u \in U \cap \rho([\ell])$:
$$\boxed{\tilde{\mathbf{k}}_u \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathsf{td}_{\mathbf{B}}, -M_{\rho^{-1}(u),1}\mathbf{y} - \sum_{2 \le j \le s_{\max}} M_{\rho^{-1}(u),j}\mathbf{b}'_j - \mathbf{N}_u\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_u, \chi_1)}.$$
7. $\forall u \in U \setminus \rho([\ell])$:
$$\boxed{\tilde{\mathbf{k}}_u \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathsf{td}_{\mathbf{B}}, -\mathbf{d}'_u - \mathbf{D}'_u\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_u, \chi_1)}.$$
8. $\forall u \in U$: $\boxed{\mathbf{k}_u \leftarrow \hat{\mathbf{k}}_u + \tilde{\mathbf{k}}_u + \mathbf{R}\mathbf{V}_u\mathbf{t} + \mathbf{Z}_u\mathbf{t}}$.
9. $\mathsf{sk} \leftarrow (\{\mathbf{k}_u\}_{u \in U}, \mathbf{t})$.

Challenge phase:

1. $\mathsf{msg} \xleftarrow{\$} \{0,1\}$.
2. $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$.
3. $\mathbf{e}_1 \leftarrow D^m_{\mathbb{Z},\chi}$.
4. $\mathbf{e}_2 \leftarrow D^m_{\mathbb{Z},\chi_s}$.
5. $e_3 \leftarrow D_{\mathbb{Z},\chi_s}$.
6. $\mathbf{c}_1^\top \leftarrow \mathbf{s}^\top\mathbf{B} + \mathbf{e}_1^\top$.
7. $\mathbf{c}_2^\top \leftarrow \mathbf{s}^\top(\mathbf{A} + \mathbf{C}) + \mathbf{e}_2^\top$.
8. $c_3 \leftarrow \mathbf{s}^\top\mathbf{y} + \mathsf{msg} \cdot \lceil q/2 \rfloor + e_3$.
9. $\mathsf{ct} \leftarrow (\mathbf{c}_1, \mathbf{c}_2, c_3)$.

Game H_4. This game is identical to Game H_3, except for how the challenger generates the responses to secret-key queries. More precisely, for each secret-key query in this game, the challenger samples $\hat{\mathbf{t}} \leftarrow D^m_{\mathbb{Z},\chi}$ and $\tilde{\mathbf{t}} \leftarrow D^m_{\mathbb{Z},\chi_1}$, and sets the last m entries of $\mathbf{t}$ to be $\hat{\mathbf{t}} + \tilde{\mathbf{t}}$. The indistinguishability between Game H_3 and Game H_4 (Lemma 22) follows from the noise-smudging lemma (Lemma 2).

Setup phase:

1. $(\mathbf{B}, \mathsf{td}_{\mathbf{B}}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$.
2. $\mathbf{W} \xleftarrow{\$} \mathbb{Z}_q^{2m^2 n \times m}$.
3. $\mathbf{T} \leftarrow \mathsf{SamplePre}([\mathbf{I}_{2m^2} \otimes \mathbf{B} \mid \mathbf{W}], \begin{bmatrix} \mathbf{I}_{2m^2} \otimes \mathsf{td}_{\mathbf{B}} \\ \mathbf{0} \end{bmatrix}, \mathbf{I}_{2m^2} \otimes \mathbf{G}, \sigma)$.
4. $\mathsf{pp} := (\mathbf{B}, \mathbf{W}, \mathbf{T})$.
5. $\mathbf{R} \xleftarrow{\$} \{-1,1\}^{m \times m}, \{\mathbf{R}_u\}_{u \in [N]} \xleftarrow{\$} \{-1,1\}^{m \times (m+1)}, \mathbf{r} \xleftarrow{\$} \{-1,1\}^m$.
6. $\mathbf{A}' \leftarrow \mathbf{B}\mathbf{R}$.
7. $\mathbf{y} \leftarrow \mathbf{B}\mathbf{r}$.
8. $\{\mathbf{B}_i\}_{i \in \{2,\ldots,s_{\max}\}} \xleftarrow{\$} \mathbb{Z}_q^{n \times (m+1)}$.
9. $\forall i \in \{2,\ldots,s_{\max}\}$, $\mathbf{B}_i := [\mathbf{b}'_i \mid \mathbf{B}'_i]$ where $\mathbf{b}'_i \in \mathbb{Z}_q^n$, $\mathbf{B}'_i \in \mathbb{Z}_q^{n \times m}$.
10. $\forall u \in \rho([\ell])$: $\mathbf{N}_u \leftarrow \sum_{2 \le j \le s_{\max}} M_{\rho^{-1}(u),j} \mathbf{B}'_j$.
11. $\forall u \in [N] : \mathbf{U}_u \leftarrow \mathbf{B}\mathbf{R}_u$.
12. $\{\mathbf{D}_u\}_{u \in [N]} \xleftarrow{\$} \mathbb{Z}_q^{n \times (m+1)}$.
13. $\forall u \in [N]$, $\mathbf{D}_u := [\mathbf{d}'_u \mid \mathbf{D}'_u]$ where $\mathbf{d}'_u \in \mathbb{Z}_q^n$, $\mathbf{D}'_u \in \mathbb{Z}_q^{n \times m}$.
14. $\forall u \in \rho([\ell]) : \mathbf{Q}_u \leftarrow \mathbf{U}_u - (M_{\rho^{-1}(u),1}(\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}) + \sum_{2 \le j \le s_{\max}} M_{\rho^{-1}(u),j} \mathbf{B}_j)$.
15. $\forall u \in [N] \setminus \rho([\ell]) : \mathbf{Q}_u \leftarrow \mathbf{U}_u - \mathbf{D}_u$.
16. $\mathbf{U} \leftarrow [\mathbf{U}_1 \mid \cdots \mid \mathbf{U}_N]$.
17. $\mathbf{C} \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{U})$.
18. $\mathbf{A} \leftarrow \mathbf{A}' - \mathbf{C} \in \mathbb{Z}_q^{n \times m}$.
19. $\mathsf{pk} = (\mathsf{pp}, n, m, q, \sigma, \chi, \chi_1, \chi_s, \mathbf{A}, \{\mathbf{B}_i\}_{i \in \{2,\ldots,s_{\max}\}}, \{\mathbf{D}_u\}_{u \in [N]}, \{\mathbf{Q}_u\}_{u \in [N]}, \mathbf{y})$.

Secret-key query for attribute set U:

1. $\mathbf{V} = [\mathbf{V}_1 \mid \cdots \mid \mathbf{V}_N] \leftarrow \mathsf{Ver}^{\mathsf{mx}}(\mathsf{pp}, 1^{(m+1)N})$.
2. $\mathbf{Z} = [\mathbf{Z}_1 \mid \cdots \mid \mathbf{Z}_N] \leftarrow \mathsf{Open}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{U})$.
3. $\hat{\mathbf{t}} \leftarrow D^m_{\mathbb{Z},\chi}$.
4. $\boxed{\tilde{\mathbf{t}} \leftarrow D^m_{\mathbb{Z},\chi_1}}$.
5. $\boxed{\mathbf{t} = (1, \hat{\mathbf{t}}^\top + \tilde{\mathbf{t}}^\top)^\top \in \mathbb{Z}_q^{m+1}}$.
6. $\{\hat{\mathbf{k}}_u\}_{u \in U} \leftarrow D^m_{\mathbb{Z},\chi_s}$.
7. $\forall u \in U \cap \rho([\ell])$:
$$\boxed{\tilde{\mathbf{k}}_u \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathsf{td}_{\mathbf{B}}, -M_{\rho^{-1}(u),1}\mathbf{y} - \sum_{2 \le j \le s_{\max}} M_{\rho^{-1}(u),j}\mathbf{b}'_j - \mathbf{N}_u\hat{\mathbf{t}} - \mathbf{N}_u\tilde{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_u, \chi_1)}.$$
8. $\forall u \in U \setminus \rho([\ell])$: $\boxed{\tilde{\mathbf{k}}_u \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathsf{td}_{\mathbf{B}}, -\mathbf{d}'_u - \mathbf{D}'_u\hat{\mathbf{t}} - \mathbf{D}'_u\tilde{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_u, \chi_1)}$.
9. $\forall u \in U$: $\mathbf{k}_u \leftarrow \hat{\mathbf{k}}_u + \tilde{\mathbf{k}}_u + \mathbf{R}\mathbf{V}_u\mathbf{t} + \mathbf{Z}_u\mathbf{t}$.
10. $\mathsf{sk} \leftarrow (\{\mathbf{k}_u\}_{u \in U}, \mathbf{t})$.

Challenge phase:

1. $\mathsf{msg} \xleftarrow{\$} \{0,1\}$.
2. $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$.
3. $\mathbf{e}_1 \leftarrow D^m_{\mathbb{Z},\chi}$.
4. $\mathbf{e}_2 \leftarrow D^m_{\mathbb{Z},\chi_s}$.
5. $e_3 \leftarrow D_{\mathbb{Z},\chi_s}$.
6. $\mathbf{c}_1^\top \leftarrow \mathbf{s}^\top\mathbf{B} + \mathbf{e}_1^\top$.
7. $\mathbf{c}_2^\top \leftarrow \mathbf{s}^\top(\mathbf{A} + \mathbf{C}) + \mathbf{e}_2^\top$.
8. $c_3 \leftarrow \mathbf{s}^\top\mathbf{y} + \mathsf{msg} \cdot \lceil q/2 \rfloor + e_3$.
9. $\mathsf{ct} \leftarrow (\mathbf{c}_1, \mathbf{c}_2, c_3)$.

Game H_5. This game is identical to Game H_4, except for how the challenger generates the matrices $\{\mathbf{B}_i\}_{i\in\{2,\ldots,s_{\max}\}}$ and $\{\mathbf{D}_u\}_{u\in[N]}$, which are no longer sampled uniformly at random. In this game, the vector $\mathbf{u}_i \in \{0,1\}^{N+s_{\max}-1}$ represents the unit vector whose i-th entry is 1. The indistinguishability between Game H_4 and Game H_5 (Lemma 23) follows from the leftover hash lemma with trapdoor (Lemma 5).

Setup phase:

1. $(\mathbf{B}, \mathsf{td}_{\mathbf{B}}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$.
2. $\mathbf{W} \xleftarrow{\$} \mathbb{Z}_q^{2m^2 n\times m}$.
3. $\mathbf{T} \leftarrow \mathsf{SamplePre}([\mathbf{I}_{2m^2}\otimes\mathbf{B} \mid \mathbf{W}], \begin{bmatrix}\mathbf{I}_{2m^2}\otimes\mathsf{td}_{\mathbf{B}}\\ \mathbf{0}\end{bmatrix}, \mathbf{I}_{2m^2}\otimes\mathbf{G}, \sigma)$.
4. $\mathsf{pp} := (\mathbf{B}, \mathbf{W}, \mathbf{T})$.
5. $\mathbf{R} \xleftarrow{\$} \{-1,1\}^{m\times m}, \{\mathbf{R}_u\}_{u\in[N]} \xleftarrow{\$} \{-1,1\}^{m\times(m+1)}, \mathbf{r} \xleftarrow{\$} \{-1,1\}^m$, $\boxed{\{\mathbf{R}'_i\}_{i\in\{2,\ldots,s_{\max}\}} \xleftarrow{\$} \{-1,1\}^{m\times m}}$, $\boxed{\{\mathbf{R}''_u\}_{u\in[N]} \xleftarrow{\$} \{-1,1\}^{m\times m}}$.
6. $\mathbf{A}' \leftarrow \mathbf{B}\mathbf{R}$.
7. $\mathbf{y} \leftarrow \mathbf{B}\mathbf{r}$.
8. $\forall i \in \{2,\ldots,s_{\max}\}$, $\boxed{\mathbf{C}_i \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{u}_{i-1}\otimes\mathbf{G})}$.
9. $\forall u \in [N]$, $\boxed{\mathbf{C}'_u \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{u}_{u+s_{\max}-1}\otimes\mathbf{G})}$.
10. $\{\mathbf{b}'_i\}_{i\in\{2,\ldots,s_{\max}\}} \xleftarrow{\$} \mathbb{Z}_q^n$.
11. $\forall i \in \{2,\ldots,s_{\max}\}$: $\boxed{\mathbf{B}'_i \leftarrow \mathbf{B}\mathbf{R}'_i + \mathbf{C}_i}$.
12. $\forall i \in \{2,\ldots,s_{\max}\}$, $\mathbf{B}_i := [\mathbf{b}'_i \mid \mathbf{B}'_i] \in \mathbb{Z}_q^{n\times(m+1)}$.
13. $\forall u \in \rho([\ell])$: $\mathbf{N}_u \leftarrow \sum_{2\le j\le s_{\max}} M_{\rho^{-1}(u),j}\mathbf{B}'_j$.
14. $\forall u \in [N]$: $\mathbf{U}_u \leftarrow \mathbf{B}\mathbf{R}_u$.
15. $\{\mathbf{d}'_u\}_{u\in[N]} \xleftarrow{\$} \mathbb{Z}_q^n$.
16. $\forall u \in [N]$: $\boxed{\mathbf{D}'_u \leftarrow \mathbf{B}\mathbf{R}''_u + \mathbf{C}'_u}$.
17. $\forall u \in [N]$, $\mathbf{D}_u := [\mathbf{d}'_u \mid \mathbf{D}'_u] \in \mathbb{Z}_q^{n\times(m+1)}$.
18. $\forall u \in \rho([\ell])$: $\mathbf{Q}_u \leftarrow \mathbf{U}_u - (M_{\rho^{-1}(u),1}(\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}) + \sum_{2\le j\le s_{\max}} M_{\rho^{-1}(u),j}\mathbf{B}_j)$.
19. $\forall u \in [N] \setminus \rho([\ell])$: $\mathbf{Q}_u \leftarrow \mathbf{U}_u - \mathbf{D}_u$.
20. $\mathbf{U} \leftarrow [\mathbf{U}_1 \mid \cdots \mid \mathbf{U}_N]$.
21. $\mathbf{C} \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{U})$.
22. $\mathbf{A} \leftarrow \mathbf{A}' - \mathbf{C} \in \mathbb{Z}_q^{n\times m}$.
23. $\mathsf{pk} = (\mathsf{pp}, n, m, q, \sigma, \chi, \chi_1, \chi_s, \mathbf{A}, \{\mathbf{B}_i\}_{i\in\{2,\ldots,s_{\max}\}}, \{\mathbf{D}_u\}_{u\in[N]}, \{\mathbf{Q}_u\}_{u\in[N]}, \mathbf{y})$.

Secret-key query for attribute set U:

1. $\mathbf{V} = [\mathbf{V}_1 \mid \cdots \mid \mathbf{V}_N] \leftarrow \mathsf{Ver}^{\mathsf{mx}}(\mathsf{pp}, 1^{(m+1)N})$.
2. $\mathbf{Z} = [\mathbf{Z}_1 \mid \cdots \mid \mathbf{Z}_N] \leftarrow \mathsf{Open}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{U})$.
3. $\hat{\mathbf{t}} \leftarrow D^m_{\mathbb{Z},\chi}$.
4. $\tilde{\mathbf{t}} \leftarrow D^m_{\mathbb{Z},\chi_1}$.
5. $\mathbf{t} = (1, \hat{\mathbf{t}}^\top + \tilde{\mathbf{t}}^\top)^\top \in \mathbb{Z}_q^{m+1}$.
6. $\{\hat{\mathbf{k}}_u\}_{u\in U} \leftarrow D^m_{\mathbb{Z},\chi_s}$.
7. $\forall u \in U \cap \rho([\ell])$: $\tilde{\mathbf{k}}_u \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathsf{td}_{\mathbf{B}}, -M_{\rho^{-1}(u),1}\mathbf{y} - \sum_{2\le j\le s_{\max}} M_{\rho^{-1}(u),j}\mathbf{b}'_j - \mathbf{N}_u\hat{\mathbf{t}} - \mathbf{N}_u\tilde{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_u, \chi_1)$.
8. $\forall u \in U \setminus \rho([\ell])$: $\tilde{\mathbf{k}}_u \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathsf{td}_{\mathbf{B}}, -\mathbf{d}'_u - \mathbf{D}'_u\hat{\mathbf{t}} - \mathbf{D}'_u\tilde{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_u, \chi_1)$.
9. $\forall u \in U$: $\mathbf{k}_u \leftarrow \hat{\mathbf{k}}_u + \tilde{\mathbf{k}}_u + \mathbf{R}\mathbf{V}_u\mathbf{t} + \mathbf{Z}_u\mathbf{t}$.
10. $\mathsf{sk} \leftarrow (\{\mathbf{k}_u\}_{u\in U}, \mathbf{t})$.

Challenge phase:

1. $\mathsf{msg} \xleftarrow{\$} \{0,1\}$.
2. $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$.
3. $\mathbf{e}_1 \leftarrow D^m_{\mathbb{Z},\chi}$.
4. $\mathbf{e}_2 \leftarrow D^m_{\mathbb{Z},\chi_s}$.
5. $e_3 \leftarrow D_{\mathbb{Z},\chi_s}$.
6. $\mathbf{c}_1^\top \leftarrow \mathbf{s}^\top\mathbf{B} + \mathbf{e}_1^\top$.
7. $\mathbf{c}_2^\top \leftarrow \mathbf{s}^\top(\mathbf{A}+\mathbf{C}) + \mathbf{e}_2^\top$.
8. $c_3 \leftarrow \mathbf{s}^\top\mathbf{y} + \mathsf{msg}\cdot\lceil q/2\rfloor + e_3$.
9. $\mathsf{ct} \leftarrow (\mathbf{c}_1, \mathbf{c}_2, c_3)$.

Game H_6. This game is identical to Game H_5, except for how the challenger generates the responses to secret-key queries. More precisely, the challenger answers secret-key queries without using the trapdoor $\mathsf{td}_{\mathbf{B}}$. Instead, in this game we use a trapdoor td for the matrix $[\mathbf{I}_{g+h} \otimes \mathbf{B} \mid \mathbf{W}']$ in order to invoke $\mathsf{SamplePre}$ in the key-generation algorithm. This trapdoor is obtained from $(\mathbf{B}, \mathbf{W}, \mathbf{T})$ by the procedure described in the proof of Lemma 24, which is based on the Wee25 commitment scheme. The indistinguishability between Games H_5 and H_6 (Lemma 24) follows from the well-sampleness of the preimage distribution for $\mathsf{SamplePre}$ (Lemma 3) and the noise-smudging lemma (Lemma 2).

Setup phase:

1. $(\mathbf{B}, \mathsf{td}_{\mathbf{B}}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$.
2. $\mathbf{W} \xleftarrow{\$} \mathbb{Z}_q^{2m^2 n \times m}$.
3. $\mathbf{T} \leftarrow \mathsf{SamplePre}([\mathbf{I}_{2m^2} \otimes \mathbf{B} \mid \mathbf{W}], \begin{bmatrix} \mathbf{I}_{2m^2} \otimes \mathsf{td}_{\mathbf{B}} \\ \mathbf{0} \end{bmatrix}, \mathbf{I}_{2m^2} \otimes \mathbf{G}, \sigma)$.
4. $\mathsf{pp} := (\mathbf{B}, \mathbf{W}, \mathbf{T})$.
5. $\mathbf{R} \xleftarrow{\$} \{-1,1\}^{m \times m}, \{\mathbf{R}_u\}_{u \in [N]} \xleftarrow{\$} \{-1,1\}^{m \times (m+1)}, \mathbf{r} \xleftarrow{\$} \{-1,1\}^m, \{\mathbf{R}'_i\}_{i \in \{2,\ldots,s_{\max}\}} \xleftarrow{\$} \{-1,1\}^{m \times m}, \{\mathbf{R}''_u\}_{u \in [N]} \xleftarrow{\$} \{-1,1\}^{m \times m}$.
6. $\mathbf{A}' \leftarrow \mathbf{B}\mathbf{R}$.
7. $\mathbf{y} \leftarrow \mathbf{B}\mathbf{r}$.
8. $\forall i \in \{2,\ldots,s_{\max}\}$, $\mathbf{C}_i \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{u}_{i-1} \otimes \mathbf{G})$.
9. $\forall u \in [N]$, $\mathbf{C}'_u \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{u}_{u+s_{\max}-1} \otimes \mathbf{G})$.
10. $\{\mathbf{b}'_i\}_{i \in \{2,\ldots,s_{\max}\}} \xleftarrow{\$} \mathbb{Z}_q^n$.
11. $\forall i \in \{2,\ldots,s_{\max}\}$: $\mathbf{B}'_i \leftarrow \mathbf{B}\mathbf{R}'_i + \mathbf{C}_i$.
12. $\forall i \in \{2,\ldots,s_{\max}\}$, $\mathbf{B}_i := [\mathbf{b}'_i \mid \mathbf{B}'_i] \in \mathbb{Z}_q^{n \times (m+1)}$.
13. $\forall u \in \rho([\ell])$: $\mathbf{N}_u \leftarrow \sum_{2 \le j \le s_{\max}} M_{\rho^{-1}(u),j} \mathbf{B}'_j$.
14. $\forall u \in \rho([\ell])$: $\mathbf{n}_u \leftarrow M_{\rho^{-1}(u),1}\mathbf{y} + \sum_{2 \le j \le s_{\max}} M_{\rho^{-1}(u),j} \mathbf{b}'_j$.
15. $\forall u \in [N] : \mathbf{U}_u \leftarrow \mathbf{B}\mathbf{R}_u$.
16. $\{\mathbf{d}'_u\}_{u \in [N]} \xleftarrow{\$} \mathbb{Z}_q^n$.
17. $\forall u \in [N]$: $\mathbf{D}'_u \leftarrow \mathbf{B}\mathbf{R}''_u + \mathbf{C}'_u$.
18. $\forall u \in [N]$, $\mathbf{D}_u := [\mathbf{d}'_u \mid \mathbf{D}'_u] \in \mathbb{Z}_q^{n \times (m+1)}$.
19. $\forall u \in \rho([\ell]) : \mathbf{Q}_u \leftarrow \mathbf{U}_u - (M_{\rho^{-1}(u),1}(\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}) + \sum_{2 \le j \le s_{\max}} M_{\rho^{-1}(u),j} \mathbf{B}_j)$.
20. $\forall u \in [N] \setminus \rho([\ell]) : \mathbf{Q}_u \leftarrow \mathbf{U}_u - \mathbf{D}_u$.
21. $\mathbf{U} \leftarrow [\mathbf{U}_1 \mid \cdots \mid \mathbf{U}_N]$.
22. $\mathbf{C} \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{U})$.
23. $\mathbf{A} \leftarrow \mathbf{A}' - \mathbf{C} \in \mathbb{Z}_q^{n \times m}$.
24. $\mathsf{pk} = (\mathsf{pp}, n, m, q, \sigma, \chi, \chi_1, \chi_s, \mathbf{A}, \{\mathbf{B}_i\}_{i \in \{2,\ldots,s_{\max}\}}, \{\mathbf{D}_u\}_{u \in [N]}, \{\mathbf{Q}_u\}_{u \in [N]}, \mathbf{y})$.

Secret-key query for attribute set U:

1. $\mathbf{V} = [\mathbf{V}_1 \mid \cdots \mid \mathbf{V}_N] \leftarrow \mathsf{Ver}^{\mathsf{mx}}(\mathsf{pp}, 1^{(m+1)N})$.
2. $\mathbf{Z} = [\mathbf{Z}_1 \mid \cdots \mid \mathbf{Z}_N] \leftarrow \mathsf{Open}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{U})$.
3. $\hat{\mathbf{t}} \leftarrow D^m_{\mathbb{Z},\chi}$.
4. Denote $U \cap \rho([\ell]) = \{u_1, \ldots, u_g\}$ and $U \setminus \rho([\ell]) = \{u'_1, \ldots, u'_h\}$.
5. $\mathbf{W}' \leftarrow [\mathbf{N}_{u_1}^\top \mid \cdots \mid \mathbf{N}_{u_g}^\top \mid \mathbf{D}_{u'_1}'^\top \mid \cdots \mid \mathbf{D}_{u'_h}'^\top]^\top$.
6. $\{\hat{\mathbf{k}}_u\}_{u \in U} \leftarrow D^m_{\mathbb{Z},\chi_s}$.
7. $\begin{bmatrix} \tilde{\mathbf{k}}_{u_1} \\ \vdots \\ \tilde{\mathbf{k}}_{u_g} \\ \tilde{\mathbf{k}}_{u'_1} \\ \vdots \\ \tilde{\mathbf{k}}_{u'_h} \\ \tilde{\mathbf{t}} \end{bmatrix} \leftarrow \mathsf{SamplePre}(\left[\mathbf{I}_{g+h} \otimes \mathbf{B} \mid \mathbf{W}'\right], \mathsf{td}, \begin{bmatrix} -\mathbf{n}_{u_1} - \mathbf{N}_{u_1}\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_{u_1} \\ \vdots \\ -\mathbf{n}_{u_g} - \mathbf{N}_{u_g}\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_{u_g} \\ -\mathbf{d}'_{u'_1} - \mathbf{D}'_{u'_1}\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_{u'_1} \\ \vdots \\ -\mathbf{d}'_{u'_h} - \mathbf{D}'_{u'_h}\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_{u'_h} \end{bmatrix}, \chi_1)$.
8. $\mathbf{t} = (1, \hat{\mathbf{t}}^\top + \tilde{\mathbf{t}}^\top)^\top \in \mathbb{Z}_q^{m+1}$.
9. $\forall u \in U$: $\mathbf{k}_u \leftarrow \hat{\mathbf{k}}_u + \tilde{\mathbf{k}}_u + \mathbf{R}\mathbf{V}_u\mathbf{t} + \mathbf{Z}_u\mathbf{t}$.
10. $\mathsf{sk} \leftarrow (\{\mathbf{k}_u\}_{u \in U}, \mathbf{t})$.

Challenge phase:

1. $\mathsf{msg} \xleftarrow{\$} \{0,1\}$.
2. $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$.
3. $\mathbf{e}_1 \leftarrow D^m_{\mathbb{Z},\chi}$.
4. $\mathbf{e}_2 \leftarrow D^m_{\mathbb{Z},\chi_s}$.
5. $e_3 \leftarrow D_{\mathbb{Z},\chi_s}$.
6. $\mathbf{c}_1^\top \leftarrow \mathbf{s}^\top\mathbf{B} + \mathbf{e}_1^\top$.
7. $\mathbf{c}_2^\top \leftarrow \mathbf{s}^\top(\mathbf{A} + \mathbf{C}) + \mathbf{e}_2^\top$.
8. $c_3 \leftarrow \mathbf{s}^\top\mathbf{y} + \mathsf{msg} \cdot \lceil q/2 \rfloor + e_3$.
9. $\mathsf{ct} \leftarrow (\mathbf{c}_1, \mathbf{c}_2, c_3)$.

Game H_7. This game is defined identically to Game H_6 except for the way the challenger generates the challenge ciphertext components $\mathbf{c}_2$ and c_3. The indistinguishability between Game H_6 and Game H_7 (Lemma 25) follows from the noise-smudging lemma (Lemma 2).

Setup phase:

1. $(\mathbf{B}, \mathsf{td}_{\mathbf{B}}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$.
2. $\mathbf{W} \xleftarrow{\$} \mathbb{Z}_q^{2m^2 n \times m}$.
3. $\mathbf{T} \leftarrow \mathsf{SamplePre}([\mathbf{I}_{2m^2} \otimes \mathbf{B} \mid \mathbf{W}], \begin{bmatrix} \mathbf{I}_{2m^2} \otimes \mathsf{td}_{\mathbf{B}} \\ \mathbf{0} \end{bmatrix}, \mathbf{I}_{2m^2} \otimes \mathbf{G}, \sigma)$.
4. $\mathsf{pp} := (\mathbf{B}, \mathbf{W}, \mathbf{T})$.
5. $\mathbf{R} \xleftarrow{\$} \{-1,1\}^{m \times m}, \{\mathbf{R}_u\}_{u \in [N]} \xleftarrow{\$} \{-1,1\}^{m \times (m+1)}, \mathbf{r} \xleftarrow{\$} \{-1,1\}^m, \{\mathbf{R}'_i\}_{i \in \{2,\ldots,s_{\max}\}} \xleftarrow{\$} \{-1,1\}^{m \times m}, \{\mathbf{R}''_u\}_{u \in [N]} \xleftarrow{\$} \{-1,1\}^{m \times m}$.
6. $\mathbf{A}' \leftarrow \mathbf{B}\mathbf{R}$.
7. $\mathbf{y} \leftarrow \mathbf{B}\mathbf{r}$.
8. $\forall i \in \{2, \ldots, s_{\max}\}$, $\mathbf{C}_i \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{u}_{i-1} \otimes \mathbf{G})$.
9. $\forall u \in [N]$, $\mathbf{C}'_u \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{u}_{u+s_{\max}-1} \otimes \mathbf{G})$.
10. $\{\mathbf{b}'_i\}_{i \in \{2,\ldots,s_{\max}\}} \xleftarrow{\$} \mathbb{Z}_q^n$.
11. $\forall i \in \{2, \ldots, s_{\max}\}$: $\mathbf{B}'_i \leftarrow \mathbf{B}\mathbf{R}'_i + \mathbf{C}_i$.
12. $\forall i \in \{2, \ldots, s_{\max}\}$, $\mathbf{B}_i := [\mathbf{b}'_i \mid \mathbf{B}'_i] \in \mathbb{Z}_q^{n \times (m+1)}$.
13. $\forall u \in \rho([\ell])$: $\mathbf{N}_u \leftarrow \sum_{2 \le j \le s_{\max}} M_{\rho^{-1}(u),j} \mathbf{B}'_j$.
14. $\forall u \in \rho([\ell])$: $\mathbf{n}_u \leftarrow M_{\rho^{-1}(u),1}\mathbf{y} + \sum_{2 \le j \le s_{\max}} M_{\rho^{-1}(u),j} \mathbf{b}'_j$.
15. $\forall u \in [N] : \mathbf{U}_u \leftarrow \mathbf{B}\mathbf{R}_u$.
16. $\{\mathbf{d}'_u\}_{u \in [N]} \xleftarrow{\$} \mathbb{Z}_q^n$.
17. $\forall u \in [N]$: $\mathbf{D}'_u \leftarrow \mathbf{B}\mathbf{R}''_u + \mathbf{C}'_u$.
18. $\forall u \in [N]$, $\mathbf{D}_u := [\mathbf{d}'_u \mid \mathbf{D}'_u] \in \mathbb{Z}_q^{n \times (m+1)}$.
19. $\forall u \in \rho([\ell]) : \mathbf{Q}_u \leftarrow \mathbf{U}_u - (M_{\rho^{-1}(u),1}(\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}) + \sum_{2 \le j \le s_{\max}} M_{\rho^{-1}(u),j} \mathbf{B}_j)$.
20. $\forall u \in [N] \setminus \rho([\ell]) : \mathbf{Q}_u \leftarrow \mathbf{U}_u - \mathbf{D}_u$.
21. $\mathbf{U} \leftarrow [\mathbf{U}_1 \mid \cdots \mid \mathbf{U}_N]$.
22. $\mathbf{C} \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{U})$.
23. $\mathbf{A} \leftarrow \mathbf{A}' - \mathbf{C} \in \mathbb{Z}_q^{n \times m}$.
24. $\mathsf{pk} = (\mathsf{pp}, n, m, q, \sigma, \chi, \chi_1, \chi_s, \mathbf{A}, \{\mathbf{B}_i\}_{i \in \{2,\ldots,s_{\max}\}}, \{\mathbf{D}_u\}_{u \in [N]}, \{\mathbf{Q}_u\}_{u \in [N]}, \mathbf{y})$.

Secret-key query for attribute set U:

1. $\mathbf{V} = [\mathbf{V}_1 \mid \cdots \mid \mathbf{V}_N] \leftarrow \mathsf{Ver}^{\mathsf{mx}}(\mathsf{pp}, 1^{(m+1)N})$.
2. $\mathbf{Z} = [\mathbf{Z}_1 \mid \cdots \mid \mathbf{Z}_N] \leftarrow \mathsf{Open}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{U})$.
3. $\hat{\mathbf{t}} \leftarrow D^m_{\mathbb{Z},\chi}$.
4. Denote $U \cap \rho([\ell]) = \{u_1, \ldots, u_g\}$ and $U \setminus \rho([\ell]) = \{u'_1, \ldots, u'_h\}$.
5. $\mathbf{W}' \leftarrow [\mathbf{N}_{u_1}^\top \mid \cdots \mid \mathbf{N}_{u_g}^\top \mid \mathbf{D}_{u'_1}'^\top \mid \cdots \mid \mathbf{D}_{u'_h}'^\top]^\top$.
6. $\{\hat{\mathbf{k}}_u\}_{u \in U} \leftarrow D^m_{\mathbb{Z},\chi_s}$.
7. $\begin{bmatrix} \tilde{\mathbf{k}}_{u_1} \\ \vdots \\ \tilde{\mathbf{k}}_{u_g} \\ \tilde{\mathbf{k}}_{u'_1} \\ \vdots \\ \tilde{\mathbf{k}}_{u'_h} \\ \tilde{\mathbf{t}} \end{bmatrix} \leftarrow \mathsf{SamplePre}(\left[\mathbf{I}_{g+h} \otimes \mathbf{B} \mid \mathbf{W}'\right], \mathsf{td}, \begin{bmatrix} -\mathbf{n}_{u_1} - \mathbf{N}_{u_1}\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_{u_1} \\ \vdots \\ -\mathbf{n}_{u_g} - \mathbf{N}_{u_g}\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_{u_g} \\ -\mathbf{d}'_{u'_1} - \mathbf{D}'_{u'_1}\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_{u'_1} \\ \vdots \\ -\mathbf{d}'_{u'_h} - \mathbf{D}'_{u'_h}\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_{u'_h} \end{bmatrix}, \chi_1)$.
8. $\mathbf{t} = (1, \hat{\mathbf{t}}^\top + \tilde{\mathbf{t}}^\top)^\top \in \mathbb{Z}_q^{m+1}$.
9. $\forall u \in U$: $\mathbf{k}_u \leftarrow \hat{\mathbf{k}}_u + \tilde{\mathbf{k}}_u + \mathbf{R}\mathbf{V}_u\mathbf{t} + \mathbf{Z}_u\mathbf{t}$.
10. $\mathsf{sk} \leftarrow (\{\mathbf{k}_u\}_{u \in U}, \mathbf{t})$.

Challenge phase:

1. $\mathsf{msg} \xleftarrow{\$} \{0,1\}$.
2. $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$.
3. $\mathbf{e}_1 \leftarrow D^m_{\mathbb{Z},\chi}$.
4. $\mathbf{e}_2 \leftarrow D^m_{\mathbb{Z},\chi_s}$.
5. $e_3 \leftarrow D_{\mathbb{Z},\chi_s}$.
6. $\mathbf{c}_1^\top \leftarrow \mathbf{s}^\top\mathbf{B} + \mathbf{e}_1^\top$.
7. $\boxed{\mathbf{c}_2^\top \leftarrow \mathbf{c}_1^\top\mathbf{R} + \mathbf{e}_2^\top}$.
8. $\boxed{c_3 \leftarrow \mathbf{c}_1^\top\mathbf{r} + \mathsf{msg} \cdot \lceil q/2 \rfloor + e_3}$.

Game H_8. This game is defined identically to Game H_7, except for the way the challenger generates the ciphertext component $\mathbf{c}_1$. In Game H_8, $\mathbf{c}_1$ is generated uniformly and independently at random. The indistinguishability between Games H_7 and H_8 (Lemma 26) follows from the $(2m^2, \sigma)$-succinct LWE assumption.

Setup phase:

1. $(\mathbf{B}, \mathsf{td}_{\mathbf{B}}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$.
2. $\mathbf{W} \xleftarrow{\$} \mathbb{Z}_q^{2m^2 n \times m}$.
3. $\mathbf{T} \leftarrow \mathsf{SamplePre}([\mathbf{I}_{2m^2} \otimes \mathbf{B} \mid \mathbf{W}], \begin{bmatrix} \mathbf{I}_{2m^2} \otimes \mathsf{td}_{\mathbf{B}} \\ \mathbf{0} \end{bmatrix}, \mathbf{I}_{2m^2} \otimes \mathbf{G}, \sigma)$.
4. $\mathsf{pp} := (\mathbf{B}, \mathbf{W}, \mathbf{T})$.
5. $\mathbf{R} \xleftarrow{\$} \{-1,1\}^{m \times m}, \{\mathbf{R}_u\}_{u \in [N]} \xleftarrow{\$} \{-1,1\}^{m \times (m+1)}, \mathbf{r} \xleftarrow{\$} \{-1,1\}^m, \{\mathbf{R}'_i\}_{i \in \{2,\ldots,s_{\max}\}} \xleftarrow{\$} \{-1,1\}^{m \times m}, \{\mathbf{R}''_u\}_{u \in [N]} \xleftarrow{\$} \{-1,1\}^{m \times m}$.
6. $\mathbf{A}' \leftarrow \mathbf{B}\mathbf{R}$.
7. $\mathbf{y} \leftarrow \mathbf{B}\mathbf{r}$.
8. $\forall i \in \{2, \ldots, s_{\max}\}$, $\mathbf{C}_i \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{u}_{i-1} \otimes \mathbf{G})$.
9. $\forall u \in [N]$, $\mathbf{C}'_u \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{u}_{u+s_{\max}-1} \otimes \mathbf{G})$.
10. $\{\mathbf{b}'_i\}_{i \in \{2,\ldots,s_{\max}\}} \xleftarrow{\$} \mathbb{Z}_q^n$.
11. $\forall i \in \{2, \ldots, s_{\max}\}$: $\mathbf{B}'_i \leftarrow \mathbf{B}\mathbf{R}'_i + \mathbf{C}_i$.
12. $\forall i \in \{2, \ldots, s_{\max}\}$, $\mathbf{B}_i := [\mathbf{b}'_i \mid \mathbf{B}'_i] \in \mathbb{Z}_q^{n \times (m+1)}$.
13. $\forall u \in \rho([\ell])$: $\mathbf{N}_u \leftarrow \sum_{2 \le j \le s_{\max}} M_{\rho^{-1}(u),j} \mathbf{B}'_j$.
14. $\forall u \in \rho([\ell])$: $\mathbf{n}_u \leftarrow M_{\rho^{-1}(u),1}\mathbf{y} + \sum_{2 \le j \le s_{\max}} M_{\rho^{-1}(u),j} \mathbf{b}'_j$.
15. $\forall u \in [N] : \mathbf{U}_u \leftarrow \mathbf{B}\mathbf{R}_u$.
16. $\{\mathbf{d}'_u\}_{u \in [N]} \xleftarrow{\$} \mathbb{Z}_q^n$.
17. $\forall u \in [N]$: $\mathbf{D}'_u \leftarrow \mathbf{B}\mathbf{R}''_u + \mathbf{C}'_u$.
18. $\forall u \in [N]$, $\mathbf{D}_u := [\mathbf{d}'_u \mid \mathbf{D}'_u] \in \mathbb{Z}_q^{n \times (m+1)}$.
19. $\forall u \in \rho([\ell]) : \mathbf{Q}_u \leftarrow \mathbf{U}_u - (M_{\rho^{-1}(u),1}(\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}) + \sum_{2 \le j \le s_{\max}} M_{\rho^{-1}(u),j} \mathbf{B}_j)$.
20. $\forall u \in [N] \setminus \rho([\ell]) : \mathbf{Q}_u \leftarrow \mathbf{U}_u - \mathbf{D}_u$.
21. $\mathbf{U} \leftarrow [\mathbf{U}_1 \mid \cdots \mid \mathbf{U}_N]$.
22. $\mathbf{C} \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{U})$.
23. $\mathbf{A} \leftarrow \mathbf{A}' - \mathbf{C} \in \mathbb{Z}_q^{n \times m}$.
24. $\mathsf{pk} = (\mathsf{pp}, n, m, q, \sigma, \chi, \chi_1, \chi_s, \mathbf{A}, \{\mathbf{B}_i\}_{i \in \{2,\ldots,s_{\max}\}}, \{\mathbf{D}_u\}_{u \in [N]}, \{\mathbf{Q}_u\}_{u \in [N]}, \mathbf{y})$.

Secret-key query for attribute set U:

1. $\mathbf{V} = [\mathbf{V}_1 \mid \cdots \mid \mathbf{V}_N] \leftarrow \mathsf{Ver}^{\mathsf{mx}}(\mathsf{pp}, 1^{(m+1)N})$.
2. $\mathbf{Z} = [\mathbf{Z}_1 \mid \cdots \mid \mathbf{Z}_N] \leftarrow \mathsf{Open}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{U})$.
3. $\hat{\mathbf{t}} \leftarrow D^m_{\mathbb{Z},\chi}$.
4. Denote $U \cap \rho([\ell]) = \{u_1, \ldots, u_g\}$ and $U \setminus \rho([\ell]) = \{u'_1, \ldots, u'_h\}$.
5. $\mathbf{W}' \leftarrow [\mathbf{N}_{u_1}^\top \mid \cdots \mid \mathbf{N}_{u_g}^\top \mid \mathbf{D}_{u'_1}'^\top \mid \cdots \mid \mathbf{D}_{u'_h}'^\top]^\top$.
6. $\{\hat{\mathbf{k}}_u\}_{u \in U} \leftarrow D^m_{\mathbb{Z},\chi_s}$.
7. $\begin{bmatrix} \tilde{\mathbf{k}}_{u_1} \\ \vdots \\ \tilde{\mathbf{k}}_{u_g} \\ \tilde{\mathbf{k}}_{u'_1} \\ \vdots \\ \tilde{\mathbf{k}}_{u'_h} \\ \tilde{\mathbf{t}} \end{bmatrix} \leftarrow \mathsf{SamplePre}(\left[\mathbf{I}_{g+h} \otimes \mathbf{B} \mid \mathbf{W}'\right], \mathsf{td}, \begin{bmatrix} -\mathbf{n}_{u_1} - \mathbf{N}_{u_1}\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_{u_1} \\ \vdots \\ -\mathbf{n}_{u_g} - \mathbf{N}_{u_g}\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_{u_g} \\ -\mathbf{d}'_{u'_1} - \mathbf{D}'_{u'_1}\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_{u'_1} \\ \vdots \\ -\mathbf{d}'_{u'_h} - \mathbf{D}'_{u'_h}\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_{u'_h} \end{bmatrix}, \chi_1)$.
8. $\mathbf{t} = (1, \hat{\mathbf{t}}^\top + \tilde{\mathbf{t}}^\top)^\top \in \mathbb{Z}_q^{m+1}$.
9. $\forall u \in U$: $\mathbf{k}_u \leftarrow \hat{\mathbf{k}}_u + \tilde{\mathbf{k}}_u + \mathbf{R}\mathbf{V}_u\mathbf{t} + \mathbf{Z}_u\mathbf{t}$.
10. $\mathsf{sk} \leftarrow (\{\mathbf{k}_u\}_{u \in U}, \mathbf{t})$.

Challenge phase:

1. $\mathsf{msg} \xleftarrow{\$} \{0,1\}$.
2. $\mathbf{e}_2 \leftarrow D^m_{\mathbb{Z},\chi_s}$.
3. $e_3 \leftarrow D_{\mathbb{Z},\chi_s}$.
4. $\boxed{\mathbf{c}_1 \xleftarrow{\$} \mathbb{Z}_q^m}$.
5. $\mathbf{c}_2^\top \leftarrow \mathbf{c}_1^\top \mathbf{R} + \mathbf{e}_2^\top$.
6. $c_3 \leftarrow \mathbf{c}_1^\top \mathbf{r} + \mathsf{msg} \cdot \lceil q/2 \rfloor + e_3$.

Game H_9. The game is defined identically to Game H_8 except for the process for generating the ciphertext component c_3. In this game, the element c_3 is generated uniformly and independently at random instead. The advantage of the adversary $\mathcal{A}$ in this game is 0, since the message msg is information-theoretically hidden. The indistinguishability between Game H_8 and H_9 (Lemma 27) follows from the leftover hash lemma (Lemma 4).

Setup phase:

1. $(\mathbf{B}, \mathsf{td}_{\mathbf{B}}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$.
2. $\mathbf{W} \xleftarrow{\$} \mathbb{Z}_q^{2m^2 n \times m}$.
3. $\mathbf{T} \leftarrow \mathsf{SamplePre}([\mathbf{I}_{2m^2} \otimes \mathbf{B} \mid \mathbf{W}], \begin{bmatrix} \mathbf{I}_{2m^2} \otimes \mathsf{td}_{\mathbf{B}} \\ \mathbf{0} \end{bmatrix}, \mathbf{I}_{2m^2} \otimes \mathbf{G}, \sigma)$.
4. $\mathsf{pp} := (\mathbf{B}, \mathbf{W}, \mathbf{T})$.
5. $\mathbf{R} \xleftarrow{\$} \{-1,1\}^{m \times m}, \{\mathbf{R}_u\}_{u \in [N]} \xleftarrow{\$} \{-1,1\}^{m \times (m+1)}, \mathbf{r} \xleftarrow{\$} \{-1,1\}^m, \{\mathbf{R}'_i\}_{i \in \{2,\ldots,s_{\max}\}} \xleftarrow{\$} \{-1,1\}^{m \times m}, \{\mathbf{R}''_u\}_{u \in [N]} \xleftarrow{\$} \{-1,1\}^{m \times m}$.
6. $\mathbf{A}' \leftarrow \mathbf{B}\mathbf{R}$.
7. $\mathbf{y} \leftarrow \mathbf{B}\mathbf{r}$.
8. $\forall i \in \{2,\ldots,s_{\max}\}$, $\mathbf{C}_i \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{u}_{i-1} \otimes \mathbf{G})$.
9. $\forall u \in [N]$, $\mathbf{C}'_u \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{u}_{u+s_{\max}-1} \otimes \mathbf{G})$.
10. $\{\mathbf{b}'_i\}_{i \in \{2,\ldots,s_{\max}\}} \xleftarrow{\$} \mathbb{Z}_q^n$.
11. $\forall i \in \{2,\ldots,s_{\max}\}$: $\mathbf{B}'_i \leftarrow \mathbf{B}\mathbf{R}'_i + \mathbf{C}_i$.
12. $\forall i \in \{2,\ldots,s_{\max}\}$, $\mathbf{B}_i := [\mathbf{b}'_i \mid \mathbf{B}'_i] \in \mathbb{Z}_q^{n \times (m+1)}$.
13. $\forall u \in \rho([\ell])$: $\mathbf{N}_u \leftarrow \sum_{2 \le j \le s_{\max}} M_{\rho^{-1}(u),j} \mathbf{B}'_j$.
14. $\forall u \in \rho([\ell])$: $\mathbf{n}_u \leftarrow M_{\rho^{-1}(u),1}\mathbf{y} + \sum_{2 \le j \le s_{\max}} M_{\rho^{-1}(u),j} \mathbf{b}'_j$.
15. $\forall u \in [N] : \mathbf{U}_u \leftarrow \mathbf{B}\mathbf{R}_u$.
16. $\{\mathbf{d}'_u\}_{u \in [N]} \xleftarrow{\$} \mathbb{Z}_q^n$.
17. $\forall u \in [N]$: $\mathbf{D}'_u \leftarrow \mathbf{B}\mathbf{R}''_u + \mathbf{C}'_u$.
18. $\forall u \in [N]$, $\mathbf{D}_u := [\mathbf{d}'_u \mid \mathbf{D}'_u] \in \mathbb{Z}_q^{n \times (m+1)}$.
19. $\forall u \in \rho([\ell]) : \mathbf{Q}_u \leftarrow \mathbf{U}_u - (M_{\rho^{-1}(u),1}(\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}) + \sum_{2 \le j \le s_{\max}} M_{\rho^{-1}(u),j} \mathbf{B}_j)$.
20. $\forall u \in [N] \setminus \rho([\ell]) : \mathbf{Q}_u \leftarrow \mathbf{U}_u - \mathbf{D}_u$.
21. $\mathbf{U} \leftarrow [\mathbf{U}_1 \mid \cdots \mid \mathbf{U}_N]$.
22. $\mathbf{C} \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{U})$.
23. $\mathbf{A} \leftarrow \mathbf{A}' - \mathbf{C} \in \mathbb{Z}_q^{n \times m}$.
24. $\mathsf{pk} = (\mathsf{pp}, n, m, q, \sigma, \chi, \chi_1, \chi_s, \mathbf{A}, \{\mathbf{B}_i\}_{i \in \{2,\ldots,s_{\max}\}}, \{\mathbf{D}_u\}_{u \in [N]}, \{\mathbf{Q}_u\}_{u \in [N]}, \mathbf{y})$.

Secret-key query for attribute set U:

1. $\mathbf{V} = [\mathbf{V}_1 \mid \cdots \mid \mathbf{V}_N] \leftarrow \mathsf{Ver}^{\mathsf{mx}}(\mathsf{pp}, 1^{(m+1)N})$.
2. $\mathbf{Z} = [\mathbf{Z}_1 \mid \cdots \mid \mathbf{Z}_N] \leftarrow \mathsf{Open}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{U})$.
3. $\hat{\mathbf{t}} \leftarrow D^m_{\mathbb{Z},\chi}$.
4. Denote $U \cap \rho([\ell]) = \{u_1, \ldots, u_g\}$ and $U \setminus \rho([\ell]) = \{u'_1, \ldots, u'_h\}$.
5. $\mathbf{W}' \leftarrow [\mathbf{N}_{u_1}^\top \mid \cdots \mid \mathbf{N}_{u_g}^\top \mid \mathbf{D}_{u'_1}'^\top \mid \cdots \mid \mathbf{D}_{u'_h}'^\top]^\top$.
6. $\{\hat{\mathbf{k}}_u\}_{u \in U} \leftarrow D^m_{\mathbb{Z},\chi_s}$.
7. $\begin{bmatrix} \tilde{\mathbf{k}}_{u_1} \\ \vdots \\ \tilde{\mathbf{k}}_{u_g} \\ \tilde{\mathbf{k}}_{u'_1} \\ \vdots \\ \tilde{\mathbf{k}}_{u'_h} \\ \tilde{\mathbf{t}} \end{bmatrix} \leftarrow \mathsf{SamplePre}(\left[\mathbf{I}_{g+h} \otimes \mathbf{B} \mid \mathbf{W}'\right], \mathsf{td}, \begin{bmatrix} -\mathbf{n}_{u_1} - \mathbf{N}_{u_1}\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_{u_1} \\ \vdots \\ -\mathbf{n}_{u_g} - \mathbf{N}_{u_g}\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_{u_g} \\ -\mathbf{d}'_{u'_1} - \mathbf{D}'_{u'_1}\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_{u'_1} \\ \vdots \\ -\mathbf{d}'_{u'_h} - \mathbf{D}'_{u'_h}\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_{u'_h} \end{bmatrix}, \chi_1)$.
8. $\mathbf{t} = (1, \hat{\mathbf{t}}^\top + \tilde{\mathbf{t}}^\top)^\top \in \mathbb{Z}_q^{m+1}$.
9. $\forall u \in U$: $\mathbf{k}_u \leftarrow \hat{\mathbf{k}}_u + \tilde{\mathbf{k}}_u + \mathbf{R}\mathbf{V}_u\mathbf{t} + \mathbf{Z}_u\mathbf{t}$.
10. $\mathsf{sk} \leftarrow (\{\mathbf{k}_u\}_{u \in U}, \mathbf{t})$.

Challenge phase:

1. $\mathbf{e}_2 \leftarrow D^m_{\mathbb{Z},\chi_s}$.
2. $\mathbf{c}_1 \xleftarrow{\$} \mathbb{Z}_q^m$.
3. $\mathbf{c}_2^\top \leftarrow \mathbf{c}_1^\top \mathbf{R} + \mathbf{e}_2^\top$.
4. $\boxed{c_3 \xleftarrow{\$} \mathbb{Z}_q.}$

Lemma 19. *We have* $\mathsf{H}_0 \equiv \mathsf{H}_1$.

Proof. The only difference between Games H_0 and H_1 is the way the matrices $\mathbf{A}$, $\{\mathbf{U}_u\}_{u \in [N]}$ and $\{\mathbf{Q}_u\}_{u \in [N]}$ are generated. In Game H_0, the matrices $\mathbf{A}, \{\mathbf{Q}_u\}_{u \in [N]}$ are generated uniformly and independently at

random. Then, for all $i \in [\ell]$, it sets $\mathbf{U}_{\rho(i)} \leftarrow M_{i,1}(\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}) + \sum_{2 \le j \le s_{\max}} M_{i,j}\mathbf{B}_j + \mathbf{Q}_{\rho(i)}$, and for all $u \in [N] \backslash \rho([\ell])$, it sets $\mathbf{U}_u \leftarrow \mathbf{Q}_u + \mathbf{D}_u$. In Game H_1, the matrix $\mathbf{A}$ is obtained by first sampling $\mathbf{A}' \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$ and then setting $\mathbf{A} \leftarrow \mathbf{A}' - \mathbf{C}$, where $\mathbf{C}$ is a matrix independent of $\mathbf{A}'$ ($\mathbf{C}$ is not sent to the adversary, and in both games it is computed as $\mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{U})$ from the same distribution of $\mathbf{U}$). Then the challenger instead samples $\{\mathbf{U}_u\}_{u \in [N]} \xleftarrow{\$} \mathbb{Z}_q^{n \times (m+1)}$ and sets $\mathbf{Q}_u \leftarrow \mathbf{U}_u - (M_{\rho^{-1}(u),1}(\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}) + \sum_{2 \le j \le s_{\max}} M_{\rho^{-1}(u),j}\mathbf{B}_j)$ for all $u \in \rho([\ell])$ and $\mathbf{Q}_u \leftarrow \mathbf{U}_u - \mathbf{D}_u$ for all $u \in [N] \setminus \rho([\ell])$.

In Game H_1 the matrix $\mathbf{A}$ is obtained as a fixed shift of the uniform matrix $\mathbf{A}'$, so $\mathbf{A}$ itself is uniform and independent of $\{\mathbf{U}_u\}$ and $\{\mathbf{Q}_u\}$, exactly as in Game H_0. For each $u \in \rho([\ell])$, the pair $(\mathbf{Q}_u, \mathbf{U}_u)$ in Game H_0 is uniformly distributed over $\mathbb{Z}_q^{n \times (m+1)} \times \mathbb{Z}_q^{n \times (m+1)}$ subject to

$$\mathbf{U}_u - \mathbf{Q}_u = M_{\rho^{-1}(u),1}(\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}) + \sum_{2 \le j \le s_{\max}} M_{\rho^{-1}(u),j}\mathbf{B}_j,$$

and exactly the same relation holds in Game H_1. Similarly, for each $u \in [N] \backslash \rho([\ell])$, the triple $(\mathbf{Q}_u, \mathbf{D}_u, \mathbf{U}_u)$ is uniformly distributed subject to $\mathbf{U}_u = \mathbf{Q}_u + \mathbf{D}_u$ in both games. Hence the joint distribution of $(\mathbf{A}, \{\mathbf{Q}_u\}, \{\mathbf{U}_u\})$ is exactly the same in Games H_0 and H_1. Observe that all the changes between Game H_0 and H_1 are merely syntactic. Since all subsequent uses of these matrices in the query phase and challenge phase are identical in both games, we conclude that $\mathsf{H}_0 \equiv \mathsf{H}_1$. □

Lemma 20. *Let q be a prime, and let n, m be such that $m > 2n \log q + \omega(\log n)$. Let $m \ge m_0(n, q)$ and $\sigma, \chi_1 \ge \chi_0(n, q)$, where m_0 and χ_0 are polynomials given in Lemma 3. We have $\mathsf{H}_1 \overset{s}{\approx} \mathsf{H}_2$.*

Proof. This lemma follows from the leftover hash lemma with trapdoor (Lemma 5). Suppose there exists an adversary $\mathcal{A}$ that can distinguish between Game H_1 and Game H_2 with non-negligible advantage. Then we can construct an adversary $\mathcal{B}$ that can win the game $\mathsf{Exp}_{\mathcal{B}}^{\mathsf{LHL\text{-}Trap},q,\sigma,\chi_1}$ with non-negligible probability, which leads to a contradiction. The algorithm $\mathcal{B}$ proceeds as follows:

Setup Phase. The algorithm $\mathcal{B}$ receives $1^\lambda, q, \sigma, \chi_1$ from its challenger. Then it invokes $\mathcal{A}$ and receives an access policy $(\mathbf{M}, \rho)$, where $\mathbf{M} \in \{-1, 0, 1\}^{\ell \times s_{\max}}$ and $\rho : [\ell] \to [N]$ is an injective function. Then it proceeds as follows:

1) It sends $1^n, 1^m, 1^{s_{\max}}$ to its challenger and receives a challenge $\{\mathsf{pp} = (\mathbf{B}, \mathbf{W}, \mathbf{T}), \mathbf{S}\}$. It then parses $\mathbf{S} := [\mathbf{S}_1 \mid \cdots \mid \mathbf{S}_N \mid \mathbf{S}_0 \mid \mathbf{s}']$ where $\mathbf{S}_i \in \mathbb{Z}_q^{n \times (m+1)}$ for each $i \in [N]$, $\mathbf{S}_0 \in \mathbb{Z}_q^{n \times m}$, and $\mathbf{s}' \in \mathbb{Z}_q^n$.
2) It sets $\mathbf{A}' \leftarrow \mathbf{S}_0$ and $\mathbf{y} \leftarrow \mathbf{s}'$.
3) It samples $\{\mathbf{B}_i\}_{i \in \{2, \ldots, s_{\max}\}} \xleftarrow{\$} \mathbb{Z}_q^{n \times (m+1)}$.
4) It sets $\mathbf{U}_u \leftarrow \mathbf{S}_u$ for all $u \in [N]$.
5) It samples $\{\mathbf{D}_u\}_{u \in [N]} \xleftarrow{\$} \mathbb{Z}_q^{n \times (m+1)}$.
6) For all $u \in \rho([\ell])$, it sets $\mathbf{Q}_u \leftarrow \mathbf{U}_u - (M_{\rho^{-1}(u),1}(\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}) + \sum_{2 \le j \le s_{\max}} M_{\rho^{-1}(u),j}\mathbf{B}_j)$.
7) For all $u \in [N] \setminus \rho([\ell])$, it sets $\mathbf{Q}_u \leftarrow \mathbf{U}_u - \mathbf{D}_u$.
8) It sets $\mathbf{U} \leftarrow [\mathbf{U}_1 \mid \cdots \mid \mathbf{U}_N]$ and computes $\mathbf{C} \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{U})$.
9) It sets $\mathbf{A} \leftarrow \mathbf{A}' - \mathbf{C}$.
10) It sends the public parameters

$$\mathsf{pk} = (\mathsf{pp}, n, m, q, \sigma, \chi, \chi_1, \chi_s, \mathbf{A}, \{\mathbf{B}_i\}_{i \in \{2, \ldots, s_{\max}\}}, \{\mathbf{D}_u\}_{u \in [N]}, \{\mathbf{Q}_u\}_{u \in [N]}, \mathbf{y})$$

to the adversary $\mathcal{A}$.

Query Phase. To respond to a secret-key query for an attribute set $U \subseteq \mathbb{U}$, the algorithm $\mathcal{B}$ proceeds as follows:

1) It first computes $\mathbf{V} = [\mathbf{V}_1 \mid \cdots \mid \mathbf{V}_N] \leftarrow \mathsf{Ver}^{\mathsf{mx}}(\mathsf{pp}, 1^{(m+1)N})$.
2) It samples $\hat{\mathbf{t}} \leftarrow D_{\mathbb{Z},\chi}^m$ and sets $\mathbf{t} = (1, \hat{\mathbf{t}}^\top)^\top \in \mathbb{Z}_q^{m+1}$.
3) It samples $\{\hat{\mathbf{k}}_u\}_{u \in U} \leftarrow D_{\mathbb{Z},\chi_s}^m$.

4) For all $u \in U$, $\mathcal{B}$ sends $(\mathbf{AV}_u + \mathbf{Q}_u)\mathbf{t} - \mathbf{B}\hat{\mathbf{k}}_u$ to its challenger in $\mathsf{Exp}_{\mathcal{B}}^{\mathsf{LHL\text{-}Trap},q,\sigma,\chi_1}$, and receives $\tilde{\mathbf{k}}_u$.
5) For all $u \in U$, it sets $\mathbf{k}_u \leftarrow \hat{\mathbf{k}}_u + \tilde{\mathbf{k}}_u$.
6) Then it provides $\mathsf{sk} \leftarrow (\{\mathbf{k}_u\}_{u\in U}, \mathbf{t})$ to the adversary $\mathcal{A}$.

Challenge Phase. In this phase, $\mathcal{B}$ proceeds exactly as the challenger in Games H_1 and H_2. Precisely,

1) It samples $\mathsf{msg} \xleftarrow{\$} \{0,1\}$.
2) It samples $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e}_1 \leftarrow D_{\mathbb{Z},\chi}^m$, $\mathbf{e}_2 \leftarrow D_{\mathbb{Z},\chi_s}^m$, and $e_3 \leftarrow D_{\mathbb{Z},\chi_s}$.
3) It computes $\mathbf{c}_1^\top \leftarrow \mathbf{s}^\top\mathbf{B} + \mathbf{e}_1^\top$, $\mathbf{c}_2^\top \leftarrow \mathbf{s}^\top(\mathbf{A}+\mathbf{C}) + \mathbf{e}_2^\top$, and $c_3 \leftarrow \mathbf{s}^\top\mathbf{y} + \mathsf{msg}\cdot\lceil q/2\rfloor + e_3$.
4) Finally, it provides the challenge ciphertext $\mathsf{ct} \leftarrow (\mathbf{c}_1, \mathbf{c}_2, c_3)$ to the adversary $\mathcal{A}$.

Guess Phase. The algorithm $\mathcal{B}$ outputs whatever the adversary $\mathcal{A}$ outputs.

It is straightforward that the algorithm $\mathcal{B}$ simulates either the game H_1 or H_2 perfectly depending on whether $\mathbf{S} \xleftarrow{\$} \mathbb{Z}_q^{n\times(m+1)(N+1)}$ or $\mathbf{S} \leftarrow \mathbf{BR}$ for some $\mathbf{R} \xleftarrow{\$} \{-1,1\}^{m\times(m+1)(N+1)}$. Therefore, the advantage of $\mathcal{B}$ in $\mathsf{Exp}_{\mathcal{B}}^{\mathsf{LHL\text{-}Trap},q,\sigma,\chi_1}$ is at least the advantage of $\mathcal{A}$ in distinguishing between the game H_1 and H_2. This completes the proof. □

Lemma 21. *Suppose that* $\chi_s \geq \chi\cdot(\chi_1+\sigma)\cdot\log q\cdot\log N\cdot\mathrm{poly}(m)\cdot\lambda^{\omega(1)}$. *Then we have* $\mathsf{H}_2 \overset{s}{\approx} \mathsf{H}_3$.

Proof. The only difference between the two games lies in the responses to the secret-key queries. In Game H_2, the secret-key responses $\{\mathbf{k}_u\}_{u\in U}$ are computed as follows: the challenger sets $\mathbf{k}_u \leftarrow \hat{\mathbf{k}}_u + \tilde{\mathbf{k}}_u$, where $\hat{\mathbf{k}}_u \leftarrow D_{\mathbb{Z},\chi_s}^m$ and samples $\tilde{\mathbf{k}}_u \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathsf{td}_{\mathbf{B}}, (\mathbf{AV}_u + \mathbf{Q}_u)\mathbf{t} - \mathbf{B}\hat{\mathbf{k}}_u, \chi_1)$, where $\mathbf{t} = (1, \hat{\mathbf{t}}^\top)^\top$ with $\hat{\mathbf{t}} \leftarrow D_{\mathbb{Z},\chi}^m$. By the construction above, we have

$$\mathbf{Bk}_u = \mathbf{B}\hat{\mathbf{k}}_u + \mathbf{B}\tilde{\mathbf{k}}_u = (\mathbf{AV}_u + \mathbf{Q}_u)\mathbf{t}, \quad \forall u \in U. \tag{7}$$

Moreover, $\tilde{\mathbf{k}}_u$ is sampled from a discrete Gaussian $D_{\mathbb{Z},\chi_1}^m$ conditioned on equation 7. Since $\chi_s \geq \sqrt{m}\cdot\chi_1\cdot\lambda^{\omega(1)}$ by our choice of parameters, the noise-smudging lemma (Lemma 2) implies that in Game H_2 each $\mathbf{k}_u$ is statistically indistinguishable from a fresh sample from $D_{\mathbb{Z},\chi_s}^m$ conditioned on (7).

In Game H_3, the secret-key responses $\{\mathbf{k}_u\}_{u\in U}$ are generated as follows. First the challenger samples $\hat{\mathbf{t}} \leftarrow D_{\mathbb{Z},\chi}^m$ and sets $\mathbf{t} \leftarrow (1, \hat{\mathbf{t}}^\top)^\top$. For each $u \in U \cap \rho([\ell])$, the challenger sets $\mathbf{k}_u \leftarrow \hat{\mathbf{k}}_u + \tilde{\mathbf{k}}_u + \mathbf{RV}_u\mathbf{t} + \mathbf{Z}_u\mathbf{t}$, where $\hat{\mathbf{k}}_u \leftarrow D_{\mathbb{Z},\chi_s}^m$ and

$$\tilde{\mathbf{k}}_u \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathsf{td}_{\mathbf{B}}, -M_{\rho^{-1}(u),1}\mathbf{y} - \sum_{2\leq j\leq s_{\max}} M_{\rho^{-1}(u),j}\mathbf{b}'_j - \mathbf{N}_u\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_u, \chi_1).$$

By bounding the norm of $\tilde{\mathbf{k}}_u + \mathbf{RV}_u\mathbf{t} + \mathbf{Z}_u\mathbf{t}$, we obtain

$$\|\tilde{\mathbf{k}}_u + \mathbf{RV}_u\mathbf{t} + \mathbf{Z}_u\mathbf{t}\| \leq \chi\cdot(\chi_1+\sigma)\log q\cdot\log N\cdot\mathrm{poly}(m).$$

Hence, by the noise-smudging lemma (Lemma 2), the distribution of $\mathbf{k}_u$ is again statistically indistinguishable from that of $\hat{\mathbf{k}}_u$. Moreover, in Game H_3, we have

$$\begin{aligned}
\mathbf{Bk}_u &= \mathbf{B}(\hat{\mathbf{k}}_u + \tilde{\mathbf{k}}_u + \mathbf{RV}_u\mathbf{t} + \mathbf{Z}_u\mathbf{t}) \\
&= \mathbf{B}\hat{\mathbf{k}}_u - M_{\rho^{-1}(u),1}\mathbf{y} - \sum_{2\leq j\leq s_{\max}} M_{\rho^{-1}(u),j}\mathbf{b}'_j - \mathbf{N}_u\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_u + \mathbf{BRV}_u\mathbf{t} + \mathbf{BZ}_u\mathbf{t} \\
&= (-M_{\rho^{-1}(u),1}(\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}) - \sum_{2\leq j\leq s_{\max}} M_{\rho^{-1}(u),j}\mathbf{B}_j + \mathbf{A}'\mathbf{V}_u + \mathbf{BZ}_u)\mathbf{t} \\
&= (-M_{\rho^{-1}(u),1}(\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}) - \sum_{2\leq j\leq s_{\max}} M_{\rho^{-1}(u),j}\mathbf{B}_j + \mathbf{AV}_u + \mathbf{CV}_u + \mathbf{BZ}_u)\mathbf{t} \\
&= (-M_{\rho^{-1}(u),1}(\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}) - \sum_{2\leq j\leq s_{\max}} M_{\rho^{-1}(u),j}\mathbf{B}_j + \mathbf{AV}_u + \mathbf{U}_u)\mathbf{t} \\
&= (\mathbf{AV}_u + \mathbf{Q}_u)\mathbf{t},
\end{aligned}$$

which matches the equation (7). For each $u \in U \setminus \rho([\ell])$, the challenger sets $\mathbf{k}_u \leftarrow \hat{\mathbf{k}}_u + \tilde{\mathbf{k}}_u + \mathbf{R}\mathbf{V}_u\mathbf{t} + \mathbf{Z}_u\mathbf{t}$, where $\hat{\mathbf{k}}_u \leftarrow D^m_{\mathbb{Z},\chi_s}$ and

$$\tilde{\mathbf{k}}_u \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathsf{td}_{\mathbf{B}}, -\mathbf{d}'_u - \mathbf{D}'_u\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_u, \chi_1).$$

By bounding the norm of $\mathbf{R}\mathbf{V}_u\mathbf{t} + \mathbf{Z}_u\mathbf{t}$ in the same way as above, we obtain that the distribution of $\mathbf{k}_u$ is statistically indistinguishable from that of $\hat{\mathbf{k}}_u$. Furthermore, we have

$$\begin{aligned}
\mathbf{B}\mathbf{k}_u &= \mathbf{B}(\hat{\mathbf{k}}_u + \tilde{\mathbf{k}}_u + \mathbf{R}\mathbf{V}_u\mathbf{t} + \mathbf{Z}_u\mathbf{t}) \\
&= \mathbf{B}\hat{\mathbf{k}}_u - \mathbf{d}'_u - \mathbf{D}'_u\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_u + \mathbf{B}\mathbf{R}\mathbf{V}_u\mathbf{t} + \mathbf{B}\mathbf{Z}_u\mathbf{t} \\
&= \mathbf{B}\hat{\mathbf{k}}_u - \mathbf{D}_u\mathbf{t} - \mathbf{B}\hat{\mathbf{k}}_u + \mathbf{A}\mathbf{V}_u\mathbf{t} + \mathbf{U}_u\mathbf{t} \\
&= (\mathbf{A}\mathbf{V}_u + \mathbf{Q}_u)\mathbf{t},
\end{aligned}$$

which matches the key equation (7) as well. In summary, for every attribute u and every secret-key query, the distribution of $\mathbf{k}_u$ in Game H_3 is statistically indistinguishable from a sample from $D^m_{\mathbb{Z},\chi_s}$ (and hence from that of $\mathbf{k}_u$ in Game H_2), and the key equation $\mathbf{B}\mathbf{k}_u = (\mathbf{A}\mathbf{V}_u + \mathbf{Q}_u)\mathbf{t}$ holds in both games.

Since the adversary makes only polynomially many secret-key queries, a union bound shows that the joint distribution of all secret-key answers in H_2 and H_3 is statistically indistinguishable. All other parts of the two games are identical. Therefore, $\mathsf{H}_2 \overset{s}{\approx} \mathsf{H}_3$. □

Lemma 22. *Suppose that $\chi > \sqrt{m}\chi_1 \cdot \lambda^{\omega(1)}$. We have $\mathsf{H}_3 \overset{s}{\approx} \mathsf{H}_4$.*

Proof. This lemma follows directly from the noise-smudging lemma (Lemma 2). The only difference between Games H_3 and H_4 lies in the way that the challenger generates the vector component $\mathbf{t}$ of each secret-key queries. In Game H_3, for each secret-key query, the challenger samples $\hat{\mathbf{t}} \leftarrow D^m_{\mathbb{Z},\chi}$ and sets $\mathbf{t} = (1, \hat{\mathbf{t}}^\top)^\top$. In Game H_4, for each secret-key query, the challenger samples $\hat{\mathbf{t}} \leftarrow D^m_{\mathbb{Z},\chi}, \tilde{\mathbf{t}} \leftarrow D^m_{\mathbb{Z},\chi_1}$, and sets $\mathbf{t} \leftarrow (1, \hat{\mathbf{t}}^\top + \tilde{\mathbf{t}}^\top)^\top$ instead. Since $\chi > \sqrt{m}\chi_1 \cdot \lambda^{\omega(1)}$, the noise-smudging lemma (Lemma 2) implies that in Game H_4 we have $\hat{\mathbf{t}} + \tilde{\mathbf{t}} \overset{s}{\approx} \hat{\mathbf{t}}$. Therefore, the distributions of the secret keys in Games H_3 and H_4 are statistically indistinguishable, and hence $\mathsf{H}_3 \overset{s}{\approx} \mathsf{H}_4$. □

Lemma 23. *Let q be a prime, and let n, m be such that $m > 2n\log q + \omega(\log n)$. Let $m \geq m_0(n,q)$ and $\sigma, \chi_1 \geq \chi_0(n,q)$, where m_0 and χ_0 are polynomials given in Lemma 3. We have $\mathsf{H}_4 \overset{s}{\approx} \mathsf{H}_5$.*

Proof. This lemma follows from the leftover hash lemma with trapdoor (Lemma 5). Suppose there exists an adversary $\mathcal{A}$ that can distinguish between Game H_4 and Game H_5 with non-negligible advantage, we can construct an adversary $\mathcal{B}$ that can win the game $\mathsf{Exp}^{\mathsf{LHL\text{-}Trap},q,\sigma,\chi_1}_{\mathcal{B}}$ with non-negligible advantage, which leads to a contradiction. The algorithm $\mathcal{B}$ proceeds as follows:

Setup Phase. The algorithm $\mathcal{B}$ receives $1^\lambda, q, \sigma, \chi_1$ from its challenger. Then it invokes $\mathcal{A}$ and receives an access policy $(\mathbf{M}, \rho)$, where $\mathbf{M} \in \{-1,0,1\}^{\ell \times s_{\max}}$ and $\rho : [\ell] \to [N]$ is an injective function. Then it proceeds as follows:

1) It sends $1^n, 1^m, 1^{s_{\max}}$ to its challenger and receives a challenge $\{\mathsf{pp} = (\mathbf{B}, \mathbf{W}, \mathbf{T}), \mathbf{S}\}$. It then parses $\mathbf{S} := [\mathbf{S}'_2 \mid \cdots \mid \mathbf{S}'_{s_{\max}} \mid \mathbf{S}''_1 \mid \cdots \mid \mathbf{S}''_N]$ where $\mathbf{S}'_i \in \mathbb{Z}_q^{n\times m}$ for each $i \in \{2, \ldots, s_{\max}\}$ and $\mathbf{S}''_i \in \mathbb{Z}_q^{n\times m}$ for each $i \in [N]$.
2) It samples $\mathbf{R} \xleftarrow{\$} \{-1,1\}^{m\times m}$, $\{\mathbf{R}_u\}_{u\in[N]} \xleftarrow{\$} \{-1,1\}^{m\times(m+1)}$, and $\mathbf{r} \xleftarrow{\$} \{-1,1\}^m$.
3) It sets $\mathbf{A}' \leftarrow \mathbf{B}\mathbf{R}$ and $\mathbf{y} \leftarrow \mathbf{B}\mathbf{r}$.
4) It computes $\mathbf{C}_i \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{u}_{i-1} \otimes \mathbf{G})$ for all $i \in \{2, \ldots, s_{\max}\}$, and $\mathbf{C}'_u \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{u}_{u+s_{\max}-1} \otimes \mathbf{G})$ for all $u \in [N]$.
5) It samples $\{\mathbf{b}'_i\}_{i\in\{2,\ldots,s_{\max}\}} \xleftarrow{\$} \mathbb{Z}_q^n$.
6) For each $i \in \{2, \ldots, s_{\max}\}$, it sets $\mathbf{B}'_i \leftarrow \mathbf{S}'_i + \mathbf{C}_i$ and sets $\mathbf{B}_i \leftarrow [\mathbf{b}'_i \mid \mathbf{B}'_i]$.
7) It computes $\mathbf{N}_u \leftarrow \sum_{2\leq j\leq s_{\max}} M_{\rho^{-1}(u),j}\mathbf{B}'_j$ for each $u \in \rho([\ell])$.

8) It sets $\mathbf{U}_u \leftarrow \mathbf{B}\mathbf{R}_u$ for all $u \in [N]$.
9) It samples $\{\mathbf{d}'_u\}_{u\in[N]} \xleftarrow{\$} \mathbb{Z}_q^n$, computes $\mathbf{D}'_u \leftarrow \mathbf{S}''_u + \mathbf{C}'_u$ and sets $\mathbf{D}_u := [\mathbf{d}'_u \mid \mathbf{D}'_u]$ for all $u \in [N]$.
10) For all $u \in \rho([\ell])$, it sets $\mathbf{Q}_u \leftarrow \mathbf{U}_u - (M_{\rho^{-1}(u),1}(\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}) + \sum_{2\le j\le s_{\max}} M_{\rho^{-1}(u),j}\mathbf{B}_j)$.
11) For all $u \in [N] \setminus \rho([\ell])$, it sets $\mathbf{Q}_u \leftarrow \mathbf{U}_u - \mathbf{D}_u$.
12) It sets $\mathbf{U} \leftarrow [\mathbf{U}_1 \mid \cdots \mid \mathbf{U}_N]$ and computes $\mathbf{C} \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{U})$.
13) It sets $\mathbf{A} \leftarrow \mathbf{A}' - \mathbf{C}$.
14) It sends the public parameters $\begin{array}{l}\mathsf{pk} = (\mathsf{pp}, n, m, q, \sigma, \chi, \chi_1, \chi_s,\\ \mathbf{A}, \{\mathbf{B}_i\}_{i\in\{2,\ldots,s_{\max}\}}, \{\mathbf{D}_u\}_{u\in[N]}, \{\mathbf{Q}_u\}_{u\in[N]}, \mathbf{y})\end{array}$ to the adversary $\mathcal{A}$.

Query Phase. To respond to a secret-key query for an attribute set $U \subseteq \mathbb{U}$, the algorithm $\mathcal{B}$ proceeds as follows:

1) It first computes $\mathbf{V} = [\mathbf{V}_1 \mid \cdots \mid \mathbf{V}_N] \leftarrow \mathsf{Ver}^{\mathsf{mx}}(\mathsf{pp}, 1^{(m+1)N})$ and $\mathbf{Z} = [\mathbf{Z}_1 \mid \cdots \mid \mathbf{Z}_N] \leftarrow \mathsf{Open}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{U})$.
2) It samples $\hat{\mathbf{t}} \leftarrow D^m_{\mathbb{Z},\chi}$, $\tilde{\mathbf{t}} \leftarrow D^m_{\mathbb{Z},\chi_1}$ and sets $\mathbf{t} = (1, \hat{\mathbf{t}}^\top + \tilde{\mathbf{t}}^\top)^\top \in \mathbb{Z}_q^{m+1}$.
3) It samples $\{\hat{\mathbf{k}}_u\}_{u\in U} \leftarrow D^m_{\mathbb{Z},\chi_s}$.
4) For all $u \in U \cap \rho([\ell])$, $\mathcal{B}$ sends $-M_{\rho^{-1}(u),1}\mathbf{y} - \sum_{2\le j\le s_{\max}} M_{\rho^{-1}(u),j}\mathbf{b}'_j - \mathbf{N}_u\hat{\mathbf{t}} - \mathbf{N}_u\tilde{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_u$ to its challenger in $\mathsf{Exp}_{\mathcal{B}}^{\mathsf{LHL\text{-}Trap},q,\sigma,\chi_1}$, and receives $\tilde{\mathbf{k}}_u$.
5) For all $u \in U \setminus \rho([\ell])$, $\mathcal{B}$ sends $-\mathbf{d}'_u - \mathbf{D}'_u\hat{\mathbf{t}} - \mathbf{D}'_u\tilde{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_u$ to its challenger in $\mathsf{Exp}_{\mathcal{B}}^{\mathsf{LHL\text{-}Trap},q,\sigma,\chi_1}$, and receives $\tilde{\mathbf{k}}_u$.
6) For all $u \in U$, it sets $\mathbf{k}_u \leftarrow \hat{\mathbf{k}}_u + \tilde{\mathbf{k}}_u + \mathbf{R}\mathbf{V}_u\mathbf{t} + \mathbf{Z}_u\mathbf{t}$.
7) Then it provides $\mathsf{sk} \leftarrow (\{\mathbf{k}_u\}_{u\in U}, \mathbf{t})$ to the adversary $\mathcal{A}$.

Challenge Phase. In this phase, $\mathcal{B}$ proceeds exactly as the challenger in Game H_4 and Game H_5 does. Precisely,

1) It samples $\mathsf{msg} \xleftarrow{\$} \{0,1\}$, $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e}_1 \leftarrow D^m_{\mathbb{Z},\chi}$, $\mathbf{e}_2 \leftarrow D^m_{\mathbb{Z},\chi_s}$, and $e_3 \leftarrow D_{\mathbb{Z},\chi_s}$.
2) It computes $\mathbf{c}_1^\top \leftarrow \mathbf{s}^\top\mathbf{B} + \mathbf{e}_1^\top$, $\mathbf{c}_2^\top \leftarrow \mathbf{s}^\top(\mathbf{A} + \mathbf{C}) + \mathbf{e}_2^\top$, and $c_3 \leftarrow \mathbf{s}^\top\mathbf{y} + \mathsf{msg} \cdot \lceil q/2 \rfloor + e_3$.
3) Finally, it provides the challenge ciphertext $\mathsf{ct} \leftarrow (\mathbf{c}_1, \mathbf{c}_2, c_3)$ to the adversary $\mathcal{A}$.

Guess Phase. The algorithm $\mathcal{B}$ outputs whatever the adversary $\mathcal{A}$ outputs.

It is straightforward that the algorithm $\mathcal{B}$ simulates either the game H_4 or H_5 perfectly depending on whether $\mathbf{S} \xleftarrow{\$} \mathbb{Z}_q^{n\times m(N+s_{\max}-1)}$ or $\mathbf{S} \leftarrow \mathbf{B}\mathbf{R}$ for some $\mathbf{R} \xleftarrow{\$} \{-1,1\}^{m\times m(N+s_{\max}-1)}$. Therefore, the advantage of $\mathcal{B}$ in $\mathsf{Exp}_{\mathcal{B}}^{\mathsf{LHL\text{-}Trap},q,\sigma,\chi_1}$ is no less than the advantage of $\mathcal{A}$ in distinguishing the game H_4 and H_5. This completes the proof. □

Lemma 24. *Suppose that* $\chi_s \ge \chi\cdot(\chi_1+\sigma)\log q\cdot\log N\cdot\mathrm{poly}(m)$ *and* $\chi_1 > \mathrm{poly}(m, N, \sigma, \log q)\cdot\omega(\sqrt{\log n})$. *We have* $\mathsf{H}_5 \overset{s}{\approx} \mathsf{H}_6$.

Proof. The only difference between Games H_5 and H_6 lies in the way the challenger responds to secret-key queries. In Game H_5, the challenger first samples $\hat{\mathbf{t}} \leftarrow D^m_{\mathbb{Z},\chi}$ and $\tilde{\mathbf{t}} \leftarrow D^m_{\mathbb{Z},\chi_1}$, and sets $\mathbf{t} \leftarrow (1 \mid \hat{\mathbf{t}}^\top + \tilde{\mathbf{t}}^\top)^\top$. For each $u \in U \cap \rho([\ell])$, it samples

$$\tilde{\mathbf{k}}_u \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathsf{td}_{\mathbf{B}}, -M_{\rho^{-1}(u),1}\mathbf{y} - \sum_{2\le j\le s_{\max}} M_{\rho^{-1}(u),j}\mathbf{b}'_j - \mathbf{N}_u\hat{\mathbf{t}} - \mathbf{N}_u\tilde{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_u, \chi_1).$$

For each $u \in U \setminus \rho([\ell])$, it samples

$$\tilde{\mathbf{k}}_u \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathsf{td}_{\mathbf{B}}, -\mathbf{d}'_u - \mathbf{D}'_u\hat{\mathbf{t}} - \mathbf{D}'_u\tilde{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_u, \chi_1).$$

For each $u \in U$, it sets $\mathbf{k}_u \leftarrow \hat{\mathbf{k}}_u + \tilde{\mathbf{k}}_u + \mathbf{R}\mathbf{V}_u\mathbf{t} + \mathbf{Z}_u\mathbf{t}$. In Game H_6, the challenger first samples $\hat{\mathbf{t}} \leftarrow D^m_{\mathbb{Z},\chi}$ and samples

$$\begin{pmatrix} \tilde{\mathbf{k}}_{u_1} \\ \vdots \\ \tilde{\mathbf{k}}_{u_g} \\ \tilde{\mathbf{k}}_{u'_1} \\ \vdots \\ \tilde{\mathbf{k}}_{u'_h} \\ \tilde{\mathbf{t}} \end{pmatrix} = \mathsf{SamplePre}(\left[\mathbf{I}_{g+h} \otimes \mathbf{B} \mid \mathbf{W}'\right], \mathsf{td}, \begin{bmatrix} -\mathbf{n}_{u_1} - \mathbf{N}_{u_1}\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_{u_1} \\ \vdots \\ -\mathbf{n}_{u_g} - \mathbf{N}_{u_g}\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_{u_g} \\ -\mathbf{d}'_{u'_1} - \mathbf{D}'_{u'_1}\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_{u'_1} \\ \vdots \\ -\mathbf{d}'_{u'_h} - \mathbf{D}'_{u'_h}\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_{u'_h} \end{bmatrix}, \chi_1)$$

and sets $\mathbf{t} \leftarrow (1, \hat{\mathbf{t}}^\top + \tilde{\mathbf{t}}^\top)^\top$.

In Game H_6, for each $u_i \in U \cap \rho([\ell])$, from the preimage sampling process, we obtain that

$$\mathbf{B}\tilde{\mathbf{k}}_{u_i} + \mathbf{N}_{u_i}\tilde{\mathbf{t}} = -\mathbf{n}_{u_i} - \mathbf{N}_{u_i}\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_{u_i},$$

which matches the secret-key equation in Game H_5. For each $u'_j \in U \setminus \rho([\ell])$, we have

$$\mathbf{B}\tilde{\mathbf{k}}_{u'_j} + \mathbf{D}'_{u'_j}\tilde{\mathbf{t}} = -\mathbf{d}'_{u'_j} - \mathbf{D}'_{u'_j}\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_{u'_j},$$

which also matches the corresponding secret-key equation in Game H_5.

By the noise-smudging lemma, in either game, since $\chi > \sqrt{m}\chi_1 \cdot \lambda^{\omega(1)}$, the distribution of $\hat{\mathbf{t}} + \tilde{\mathbf{t}}$ is statistically indistinguishable from a vector sampled from $D^m_{\mathbb{Z},\chi}$. By bounding the norm of $\tilde{\mathbf{k}}_u + \mathbf{R}\mathbf{V}_u\mathbf{t} + \mathbf{Z}_u\mathbf{t}$, we have

$$\|\tilde{\mathbf{k}}_u + \mathbf{R}\mathbf{V}_u\mathbf{t} + \mathbf{Z}_u\mathbf{t}\| \le \chi \cdot (\chi_1 + \sigma) \log q \cdot \log N \cdot \mathrm{poly}(m).$$

According to the selection of the parameters, we know each $\mathbf{k}_u$ in either game is statistically indistinguishable from a sample from $D^m_{\mathbb{Z},\chi_s}$ conditioned on $\mathbf{B}\mathbf{k}_u = (\mathbf{A}\mathbf{V}_u + \mathbf{Q}_u)\mathbf{t}$.

It suffices to clarify that the challenger can construct the trapdoor for the matrix $[\mathbf{I}_{g+h} \otimes \mathbf{B} \mid \mathbf{W}']$ efficiently. Denote $s = N + s_{\max} - 1$. From the correctness of the commitment scheme, we have

$$\begin{aligned} \mathbf{C}_i\mathbf{V}_{ns} &= \mathbf{u}_{i-1} \otimes \mathbf{G} - \mathbf{B}\mathbf{Z}_i, \quad \forall i \in \{2, \ldots, s_{\max}\}, \\ \mathbf{C}'_i\mathbf{V}_{ns} &= \mathbf{u}_{i+s_{\max}-1} \otimes \mathbf{G} - \mathbf{B}\mathbf{Z}'_i, \quad \forall i \in [N]. \end{aligned}$$

If we write $\mathbf{C}' = [\mathbf{C}_2^\top \mid \cdots \mid \mathbf{C}_{s_{\max}}^\top \mid \mathbf{C}_1'^\top \mid \cdots \mid \mathbf{C}_N'^\top]^\top$ and $\mathbf{Z}' = [\mathbf{Z}_2^\top \mid \cdots \mid \mathbf{Z}_{s_{\max}}^\top \mid \mathbf{Z}_1'^\top \mid \cdots \mid \mathbf{Z}_N'^\top]^\top$, we have

$$[\mathbf{I}_s \otimes \mathbf{B} \mid \mathbf{C}'] \begin{bmatrix} \mathbf{Z}' \\ \mathbf{V}_{ns} \end{bmatrix} = \mathbf{I}_s \otimes \mathbf{G} \tag{8}$$

Then by the norm bound of the matrices $\mathbf{Z}'$ and $\mathbf{V}_{ns}$, the matrix $\begin{bmatrix} \mathbf{Z}' \\ \mathbf{V}_{ns} \end{bmatrix}$ serves as a trapdoor for $[\mathbf{I}_s \otimes \mathbf{B} \mid \mathbf{C}']$.

Denote the set $I = \rho^{-1}(U) \cap [\ell]$ and $\mathbf{M}_I \in \{-1, 0, 1\}^{|I| \times (s_{\max}-1)}$ the submatrix of $\mathbf{M}$ containing all the rows with indices in I and removing the first column. Let $\mathbf{e}_i \in \{0,1\}^N$ be the unit vector whose i-th entry is 1.

Let $\mathbf{M}_U = \begin{bmatrix} \mathbf{M}_I & \\ & \mathbf{e}_{u'_1}^\top \\ & \mathbf{e}_{u'_2}^\top \\ & \vdots \\ & \mathbf{e}_{u'_h}^\top \end{bmatrix} \in \{-1, 0, 1\}^{(g+h) \times s}$. Now we prove the matrix $\mathbf{M}_U$ has full row rank. By the restriction on secret-key queries submitted by the adversary, the rows of the access matrix in $\rho^{-1}(U)$ are required to be linearly independent and unauthorized. This means that no non-zero linear combination

over $\mathbb{Z}_q$ of the vectors obtained by removing the first entries of the rows of $\mathbf{M}$ with indices in $\rho^{-1}(U)$ can span the zero vector in dimension $s_{\max}-1$, i.e., $\mathbf{M}_I$ has full row rank. Since $u'_1,\ldots,u'_h$ are different from each other, it also follows that $\mathbf{e}_{u'_1},\ldots,\mathbf{e}_{u'_h}$ are linearly independent. Therefore, we can conclude $\mathbf{M}_U$ has full row rank.

Let $\mathbf{R}' = [\mathbf{R}'^{\top}_2 \mid \cdots \mid \mathbf{R}'^{\top}_{s_{\max}} \mid \mathbf{R}''^{\top}_1 \mid \cdots \mid \mathbf{R}''^{\top}_N]^{\top}$. We prove

$$\mathbf{W}' = (\mathbf{M}_U \otimes \mathbf{I}_n)((\mathbf{I}_s \otimes \mathbf{B})\mathbf{R}' + \mathbf{C}').$$

It follows from that

$$(\mathbf{M}_U \otimes \mathbf{I}_n)((\mathbf{I}_s \otimes \mathbf{B})\mathbf{R}' + \mathbf{C}') = (\mathbf{M}_U \otimes \mathbf{I}_n)\begin{bmatrix} \mathbf{B}\mathbf{R}'_2 + \mathbf{C}_2 \\ \vdots \\ \mathbf{B}\mathbf{R}'_{s_{\max}} + \mathbf{C}_{s_{\max}} \\ \mathbf{B}\mathbf{R}''_1 + \mathbf{C}'_1 \\ \vdots \\ \mathbf{B}\mathbf{R}''_N + \mathbf{C}'_N \end{bmatrix} = (\mathbf{M}_U \otimes \mathbf{I}_n)\begin{bmatrix} \mathbf{B}'_2 \\ \vdots \\ \mathbf{B}'_{s_{\max}} \\ \mathbf{D}'_1 \\ \vdots \\ \mathbf{D}'_N \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{2\le j\le s_{\max}} M_{\rho^{-1}(u_1),j}\mathbf{B}'_j \\ \vdots \\ \sum_{2\le j\le s_{\max}} M_{\rho^{-1}(u_g),j}\mathbf{B}'_j \\ \mathbf{D}'_{u'_1} \\ \vdots \\ \mathbf{D}'_{u'_h} \end{bmatrix} = \begin{bmatrix} \mathbf{N}_{u_1} \\ \vdots \\ \mathbf{N}_{u_g} \\ \mathbf{D}'_{u'_1} \\ \vdots \\ \mathbf{D}'_{u'_h} \end{bmatrix} = \mathbf{W}'$$

We claim that $\mathsf{td} := \begin{bmatrix} (\mathbf{M}_U \otimes \mathbf{I}_m)(\mathbf{Z}' - \mathbf{R}'\mathbf{V}_{ns}) \\ \mathbf{V}_{ns} \end{bmatrix}$ can serve as a trapdoor for the matrix $[\mathbf{I}_{g+h} \otimes \mathbf{B} \mid \mathbf{W}']$.

$$\begin{aligned}
&[\mathbf{I}_{g+h} \otimes \mathbf{B} \mid \mathbf{W}'] \begin{bmatrix} (\mathbf{M}_U \otimes \mathbf{I}_m)(\mathbf{Z}' - \mathbf{R}'\mathbf{V}_{ns}) \\ \mathbf{V}_{ns} \end{bmatrix} \\
&= (\mathbf{I}_{g+h} \otimes \mathbf{B})(\mathbf{M}_U \otimes \mathbf{I}_m)(\mathbf{Z}' - \mathbf{R}'\mathbf{V}_{ns}) + (\mathbf{M}_U \otimes \mathbf{I}_n)((\mathbf{I}_s \otimes \mathbf{B})\mathbf{R}' + \mathbf{C}')\mathbf{V}_{ns} \\
&= (\mathbf{M}_U \otimes \mathbf{I}_n)(\mathbf{I}_s \otimes \mathbf{B})(\mathbf{Z}' - \mathbf{R}'\mathbf{V}_{ns}) + (\mathbf{M}_U \otimes \mathbf{I}_n)((\mathbf{I}_s \otimes \mathbf{B})\mathbf{R}' + \mathbf{C}')\mathbf{V}_{ns} \\
&= (\mathbf{M}_U \otimes \mathbf{I}_n)[\mathbf{I}_s \otimes \mathbf{B} \mid \mathbf{C}'] \begin{bmatrix} \mathbf{Z}' \\ \mathbf{V}_{ns} \end{bmatrix} = (\mathbf{M}_U \otimes \mathbf{I}_n)(\mathbf{I}_s \otimes \mathbf{G}) = \mathbf{M}_U \otimes \mathbf{G}.
\end{aligned}$$

Since $\mathbf{M}_U$ has full row rank, we know td serves as a trapdoor for $[\mathbf{I}_{g+h} \otimes \mathbf{B} \mid \mathbf{W}']$ by Lemma 1. □

Lemma 25. *Suppose that* $\chi_s > m^2\chi \cdot \lambda^{\omega(1)}$. *We have* $\mathsf{H}_6 \overset{s}{\approx} \mathsf{H}_7$.

Proof. The only difference between Games H_6 and H_7 lies in the way the ciphertext components $\mathbf{c}_2$ and c_3 are sampled. In Game H_6, the challenger first samples $\mathbf{s} \overset{\$}{\leftarrow} \mathbb{Z}_q^n, \mathbf{e}_1 \leftarrow D^m_{\mathbb{Z},\chi}, \mathbf{e}_2 \leftarrow D^m_{\mathbb{Z},\chi_s}, e_3 \leftarrow D_{\mathbb{Z},\chi_s}$, and computes $\mathbf{c}_1^{\top} \leftarrow \mathbf{s}^{\top}\mathbf{B} + \mathbf{e}_1^{\top}$, $\mathbf{c}_2^{\top} \leftarrow \mathbf{s}^{\top}(\mathbf{A}+\mathbf{C}) + \mathbf{e}_2^{\top}$ and $c_3 \leftarrow \mathbf{s}^{\top}\mathbf{y} + \mathsf{msg} \cdot \lceil q/2 \rfloor + e_3$. In Game H_7, the challenger generates $\mathbf{c}_1$ exactly the same as in Game H_6, samples $\mathbf{e}_2 \leftarrow D^m_{\mathbb{Z},\chi_s}, e_3 \leftarrow D_{\mathbb{Z},\chi_s}$, and computes $\mathbf{c}_2^{\top} \leftarrow \mathbf{c}_1^{\top}\mathbf{R} + \mathbf{e}_2^{\top}$ and $c_3 \leftarrow \mathbf{c}_1^{\top}\mathbf{r} + \mathsf{msg} \cdot \lceil q/2 \rfloor + e_3$. In Game H_7, using the relations $\mathbf{A}' = \mathbf{B}\mathbf{R}$ and $\mathbf{A} = \mathbf{A}' - \mathbf{C}$, we have that

$$\mathbf{c}_2^{\top} \leftarrow \mathbf{c}_1^{\top}\mathbf{R} + \mathbf{e}_2^{\top} = \mathbf{s}^{\top}\mathbf{B}\mathbf{R} + \mathbf{e}_1^{\top}\mathbf{R} + \mathbf{e}_2^{\top} = \mathbf{s}^{\top}\mathbf{A}' + \mathbf{e}_1^{\top}\mathbf{R} + \mathbf{e}_2^{\top} = \mathbf{s}^{\top}(\mathbf{A}+\mathbf{C}) + \mathbf{e}_1^{\top}\mathbf{R} + \mathbf{e}_2^{\top}.$$

By bounding the error term, we have $\|\mathbf{e}_1^{\top}\mathbf{R}\| \le m^{3/2} \cdot \chi$ with overwhelming probabiliry. By the noise-smudging lemma, we obtain that $\mathbf{e}_1^{\top}\mathbf{R} + \mathbf{e}_2^{\top}$ is statistically indistinguishable from $\mathbf{e}_2^{\top}$. Therefore, the ciphertexts $\mathbf{c}_2$ generated in two games are statistically indistinguishable. An analogous argument applies

to c_3. Since $(\mathbf{c}_2, c_3)$ are the only ciphertext components that differ between the two games, the full ciphertext distributions in Games H_6 and H_7 are statistically indistinguishable. Therefore, $\mathsf{H}_6 \overset{s}{\approx} \mathsf{H}_7$. □

Lemma 26. *Suppose that* $(2m^2, \sigma)$*-succinct* LWE *assumption holds. We have* $\mathsf{H}_7 \overset{c}{\approx} \mathsf{H}_8$.

Proof. Suppose that there exists an PPT adversary $\mathcal{A}$ that can distinguish between Game H_7 and H_8 with non-negligible advantage. We construct an PPT adversary $\mathcal{B}$ that can break the $(2m^2, \sigma)$-succinct LWE assumption with non-negligible advantage. The adversary $\mathcal{B}$ proceeds as follows:

Setup Phase. The algorithm $\mathcal{B}$ receives from its challenger a succinct LWE challenge $(\mathbf{B}, \mathbf{W}, \mathbf{T}, \mathbf{c})$, where $\mathbf{B} \in \mathbb{Z}_q^{n\times m}, \mathbf{W} \in \mathbb{Z}_q^{2m^2 n\times m}, \mathbf{T} \in \mathbb{Z}_q^{(2m^2+1)m\times 2m^3}$, and $\mathbf{c}$ is sampled as $\mathbf{c}^\top \leftarrow \mathbf{s}^\top\mathbf{B} + \mathbf{e}^\top$ for some $\mathbf{s} \in \mathbb{Z}_q^n, \mathbf{e} \leftarrow D_{\mathbb{Z},\chi}^m$ or $\mathbf{c} \overset{\$}{\leftarrow} \mathbb{Z}_q^m$. It sets $\mathsf{pp} := (\mathbf{B}, \mathbf{W}, \mathbf{T})$.

1) It samples $\mathbf{r} \overset{\$}{\leftarrow} \{-1,1\}^m, \{\mathbf{R}_u\}_{u\in[N]} \overset{\$}{\leftarrow} \{-1,1\}^{m\times(m+1)}$, and $\mathbf{R}, \{\mathbf{R}'_i\}_{i\in\{2,\dots,s_{\max}\}}, \{\mathbf{R}''_u\}_{u\in[N]} \overset{\$}{\leftarrow} \{-1,1\}^{m\times m}$.
2) It sets $\mathbf{A}' \leftarrow \mathbf{BR}$ and $\mathbf{y} \leftarrow \mathbf{Br}$.
3) For all $i \in \{2,\dots,s_{\max}\}$, it computes $\mathbf{C}_i \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{u}_{i-1} \otimes \mathbf{G})$. For all $u \in [N]$, it computes $\mathbf{C}'_u \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{u}_{u+s_{\max}-1} \otimes \mathbf{G})$.
4) It samples $\{\mathbf{b}'_i\}_{2,\dots,s_{\max}} \overset{\$}{\leftarrow} \mathbb{Z}_q^n$ and $\{\mathbf{d}'_u\}_{u\in[N]} \overset{\$}{\leftarrow} \mathbb{Z}_q^n$.
5) For all $i \in \{2,\dots,s_{\max}\}$, it sets $\mathbf{B}'_i \leftarrow \mathbf{BR}'_i + \mathbf{C}_i$ and $\mathbf{B}_i \leftarrow [\mathbf{b}'_i \mid \mathbf{B}'_i]$.
6) For all $u \in [N]$, it sets $\mathbf{D}'_u \leftarrow \mathbf{BR}''_u + \mathbf{C}'_u$ and $\mathbf{D}_u \leftarrow [\mathbf{d}'_u \mid \mathbf{D}'_u]$.
7) For all $u \in \rho([\ell])$, it sets $\mathbf{N}_u \leftarrow \sum_{2\le j\le s_{\max}} M_{\rho^{-1}(u),j}\mathbf{B}'_j$ and $\mathbf{n}_u \leftarrow M_{\rho^{-1}(u),1}\mathbf{y} + \sum_{2\le j\le s_{\max}} M_{\rho^{-1}(u),j}\mathbf{b}'_j$.
8) For all $u \in [N]$ set $\mathbf{U}_u \leftarrow \mathbf{BR}_u$, and set $\mathbf{U} \leftarrow [\mathbf{U}_1 \mid \cdots \mid \mathbf{U}_N]$.
9) For all $u \in \rho([\ell])$, it sets $\mathbf{Q}_u \leftarrow \mathbf{U}_u - (M_{\rho^{-1}(u),1}(\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}) + \sum_{2\le j\le s_{\max}} M_{\rho^{-1}(u),j}\mathbf{B}_j)$.
10) For all $u \in [N] \setminus \rho([\ell])$, it sets $\mathbf{Q}_u \leftarrow \mathbf{U}_u - \mathbf{D}_u$.
11) It computes $\mathbf{C} \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{U})$ and sets $\mathbf{A} \leftarrow \mathbf{A}' - \mathbf{C} \in \mathbb{Z}_q^{n\times m}$.
12) It sets the public key

$$\mathsf{pk} = (\mathsf{pp}, n, m, q, \sigma, \chi, \chi_1, \chi_s, \mathbf{A}, \{\mathbf{B}_i\}_{i\in\{2,\dots,s_{\max}\}}, \{\mathbf{D}_u\}_{u\in[N]}, \{\mathbf{Q}_u\}_{u\in[N]}, \mathbf{y})$$

and sends it to $\mathcal{A}$.

Query Phase. To respond to a secret-key query for an attribute set $U \subseteq \mathbb{U} = [N]$, the adversary $\mathcal{B}$ proceeds exactly as in Games H_7 and H_8:

1) It first computes $\mathbf{V} = [\mathbf{V}_1 \mid \cdots \mid \mathbf{V}_N] \leftarrow \mathsf{Ver}^{\mathsf{mx}}(\mathsf{pp}, 1^{(m+1)N})$ and $\mathbf{Z} = [\mathbf{Z}_1 \mid \cdots \mid \mathbf{Z}_N] \leftarrow \mathsf{Open}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{U})$.
2) It samples $\hat{\mathbf{t}} \leftarrow D_{\mathbb{Z},\chi}^m$.
3) Denote $U \cap \rho([\ell]) = \{u_1, \dots, u_g\}$ and $U \setminus \rho([\ell]) = \{u'_1, \dots, u'_h\}$ and set $\mathbf{W}' \leftarrow [\mathbf{N}_{u_1}^\top \mid \cdots \mid \mathbf{N}_{u_g}^\top \mid \mathbf{D}_{u'_1}'^\top \mid \cdots \mid \mathbf{D}_{u'_h}'^\top]^\top$.
4) It samples $\{\hat{\mathbf{k}}_u\}_{u\in U} \leftarrow D_{\mathbb{Z},\chi_s}^m$.
5) It computes $\begin{pmatrix} \tilde{\mathbf{k}}_{u_1} \\ \vdots \\ \tilde{\mathbf{k}}_{u_g} \\ \tilde{\mathbf{k}}_{u'_1} \\ \vdots \\ \tilde{\mathbf{k}}_{u'_h} \\ \tilde{\mathbf{t}} \end{pmatrix} = \mathsf{SamplePre}(\left[\mathbf{I}_{g+h} \otimes \mathbf{B} \mid \mathbf{W}'\right], \mathsf{td}, \begin{bmatrix} -\mathbf{n}_{u_1} - \mathbf{N}_{u_1}\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_{u_1} \\ \vdots \\ -\mathbf{n}_{u_g} - \mathbf{N}_{u_g}\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_{u_g} \\ -\mathbf{d}'_{u'_1} - \mathbf{D}'_{u'_1}\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_{u'_1} \\ \vdots \\ -\mathbf{d}'_{u'_h} - \mathbf{D}'_{u'_h}\hat{\mathbf{t}} - \mathbf{B}\hat{\mathbf{k}}_{u'_h} \end{bmatrix}, \chi_1)$ using the trapdoor td for $[\mathbf{I}_{g+h} \otimes \mathbf{B} \mid \mathbf{W}']$. The construction of td is given in the proof of Lemma 24.
6) It sets $\mathbf{t} = (1, \hat{\mathbf{t}}^\top + \tilde{\mathbf{t}}^\top)^\top \in \mathbb{Z}_q^{m+1}$.
7) For all $u \in U$, it computes $\mathbf{k}_u \leftarrow \hat{\mathbf{k}}_u + \tilde{\mathbf{k}}_u + \mathbf{RV}_u\mathbf{t} + \mathbf{Z}_u\mathbf{t}$.

8) Then it provides the secret-key response $\mathsf{sk} \leftarrow (\{\mathbf{k}_u\}_{u\in U}, \mathbf{t})$ to the adversary $\mathcal{A}$.

Challenge Phase.

1) It sets $\mathbf{c}_1 \leftarrow \mathbf{c}$.
2) It samples $\mathsf{msg} \xleftarrow{\$} \{0,1\}$, $\mathbf{e}_2 \leftarrow D^m_{\mathbb{Z},\chi_s}$, and $e_3 \leftarrow D_{\mathbb{Z},\chi_s}$.
3) It computes $\mathbf{c}_2^\top \leftarrow \mathbf{c}_1^\top \mathbf{R} + \mathbf{e}_2^\top$, and $c_3 \leftarrow \mathbf{c}_1^\top \mathbf{r} + \mathsf{msg} \cdot \lceil q/2 \rfloor + e_3$.
4) Finally, it provides the challenge ciphertext $\mathsf{ct} \leftarrow (\mathbf{c}_1, \mathbf{c}_2, c_3)$ to the adversary $\mathcal{A}$.

Guess Phase. The algorithm $\mathcal{B}$ outputs whatever the adversary $\mathcal{A}$ outputs.

It is straightforward that the algorithm $\mathcal{B}$ simulates either Game H_7 or Game H_8 perfectly, depending on whether $\mathbf{c}^\top = \mathbf{s}^\top \mathbf{B} + \mathbf{e}_1^\top$ for some $\mathbf{s} \in \mathbb{Z}_q^n$ and $\mathbf{e}_1 \leftarrow D^m_{\mathbb{Z},\chi}$, or $\mathbf{c} \xleftarrow{\$} \mathbb{Z}_q^m$. Therefore, the advantage of $\mathcal{B}$ in breaking the succinct LWE assumption is no less than the advantage of $\mathcal{A}$ in distinguishing Games H_7 and H_8, which leads to a contradiction. Hence, we complete the proof. □

Lemma 27. *Suppose that* $m > 2n \log q + \omega(\log n)$. *We have* $\mathsf{H}_8 \overset{s}{\approx} \mathsf{H}_9$.

Proof. This lemma follows from the leftover hash lemma (Lemma 4). Suppose there exists an adversary $\mathcal{A}$ that can distinguish between Game H_8 and Game H_9 with non-negligible advantage, we can construct an adversary $\mathcal{B}$ that can break leftover hash lemma with non-negligible advantage, which leads to a contradiction. The algorithm $\mathcal{B}$ proceeds as follows:

Setup Phase. The algorithm $\mathcal{B}$ first invokes $\mathcal{A}$ and receives an access policy $(\mathbf{M}, \rho)$, where $\mathbf{M} \in \{-1,0,1\}^{\ell \times s_{\max}}$ and $\rho : [\ell] \to \mathbb{U}$ is an injective function. Then it proceeds as follows:

1) It receives a challenge $(\mathbf{b}_0^\top, s_0)$, where $\mathbf{b}_0 \in \mathbb{Z}_q^m$, $s_0 \in \mathbb{Z}_q$.
2) It samples $(\mathbf{B}, \mathsf{td}_\mathbf{B}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$ and $\mathbf{W} \xleftarrow{\$} \mathbb{Z}_q^{2m^2 n \times m}$.
3) Then it samples $\mathbf{T} \leftarrow \mathsf{SamplePre}([\mathbf{I}_{2m^2} \otimes \mathbf{B} \mid \mathbf{W}], \begin{bmatrix} \mathbf{I}_{2m^2} \otimes \mathsf{td}_\mathbf{B} \\ \mathbf{0} \end{bmatrix}, \mathbf{I}_{2m^2} \otimes \mathbf{G}, \sigma)$.
4) It sets $\mathsf{pp} := (\mathbf{B}, \mathbf{W}, \mathbf{T})$.
5) It samples $\mathbf{r} \xleftarrow{\$} \{-1,1\}^m, \{\mathbf{R}_u\}_{u\in[N]} \xleftarrow{\$} \{-1,1\}^{m\times(m+1)}$, and $\mathbf{R}, \{\mathbf{R}'_i\}_{i\in\{2,\ldots,s_{\max}\}}, \{\mathbf{R}''_u\}_{u\in[N]} \xleftarrow{\$} \{-1,1\}^{m\times m}$.
6) It sets $\mathbf{A}' \leftarrow \mathbf{B}\mathbf{R}$ and $\mathbf{y} \leftarrow \mathbf{B}\mathbf{r}$.
7) For all $i \in \{2,\ldots,s_{\max}\}$, it computes $\mathbf{C}_i \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{u}_{i-1} \otimes \mathbf{G})$. For all $u \in [N]$, it computes $\mathbf{C}'_u \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{u}_{u+s_{\max}-1} \otimes \mathbf{G})$.
8) It samples $\{\mathbf{b}'_i\}_{i=2}^{s_{\max}} \xleftarrow{\$} \mathbb{Z}_q^n$ and $\{\mathbf{d}'_u\}_{u\in[N]} \xleftarrow{\$} \mathbb{Z}_q^n$.
9) For all $i \in \{2,\ldots,s_{\max}\}$, it sets $\mathbf{B}'_i \leftarrow \mathbf{B}\mathbf{R}'_i + \mathbf{C}_i$ and $\mathbf{B}_i \leftarrow [\mathbf{b}'_i \mid \mathbf{B}'_i]$.
10) For all $u \in [N]$, it sets $\mathbf{D}'_u \leftarrow \mathbf{B}\mathbf{R}''_u + \mathbf{C}'_u$ and $\mathbf{D}_u \leftarrow [\mathbf{d}'_u \mid \mathbf{D}'_u]$.
11) For all $u \in \rho([\ell])$, it sets $\mathbf{N}_u \leftarrow \sum_{2\le j\le s_{\max}} M_{\rho^{-1}(u),j}\mathbf{B}'_j$ and $\mathbf{n}_u \leftarrow M_{\rho^{-1}(u),1}\mathbf{y} + \sum_{2\le j\le s_{\max}} M_{\rho^{-1}(u),j}\mathbf{b}'_j$.
12) For all $u \in [N] : \mathbf{U}_u \leftarrow \mathbf{B}\mathbf{R}_u$ and sets $\mathbf{U} \leftarrow [\mathbf{U}_1 \mid \cdots \mid \mathbf{U}_N]$.
13) For all $u \in \rho([\ell])$, it sets $\mathbf{Q}_u \leftarrow \mathbf{U}_u - (M_{\rho^{-1}(u),1}(\mathbf{y} \mid \mathbf{0} \mid \cdots \mid \mathbf{0}) + \sum_{2\le j\le s_{\max}} M_{\rho^{-1}(u),j}\mathbf{B}_j)$.
14) For all $u \in [N] \setminus \rho([\ell])$, it sets $\mathbf{Q}_u \leftarrow \mathbf{U}_u - \mathbf{D}_u$.
15) It computes $\mathbf{C} \leftarrow \mathsf{Com}^{\mathsf{mx}}(\mathsf{pp}, \mathbf{U})$ and sets $\mathbf{A} \leftarrow \mathbf{A}' - \mathbf{C} \in \mathbb{Z}_q^{n\times m}$.
16) The challenger sets the public key

$$\mathsf{pk} = (\mathsf{pp}, n, m, q, \sigma, \chi, \chi_1, \chi_s, \mathbf{A}, \{\mathbf{B}_i\}_{i\in\{2,\ldots,s_{\max}\}}, \{\mathbf{D}_u\}_{u\in[N]}, \{\mathbf{Q}_u\}_{u\in[N]}, \mathbf{y})$$

and sends it to $\mathcal{A}$.

Query Phase. To respond to a secret-key query for an attribute set $U \subseteq \mathbb{U} = [N]$, the adversary $\mathcal{B}$ proceeds exactly as in Games H_8 and H_9.

Challenge Phase. Precisely,

1) It samples $\mathsf{msg} \xleftarrow{\$} \{0,1\}$, $\mathbf{e}_2 \leftarrow D^m_{\mathbb{Z},\chi_s}, e_3 \leftarrow D_{\mathbb{Z},\chi_s}$.
2) It samples $\mathbf{c}_1 \leftarrow \mathbf{b}_0$ and computes $\mathbf{c}_2^\top \leftarrow \mathbf{c}_1^\top \mathbf{R} + \mathbf{e}_2^\top$, and $c_3 \leftarrow s_0 + \mathsf{msg} \cdot \lceil q/2 \rfloor + e_3$.
3) Finally, it provides the challenge ciphertext $\mathsf{ct} \leftarrow (\mathbf{c}_1, \mathbf{c}_2, c_3)$ to the adversary $\mathcal{A}$.

Guess Phase. The algorithm $\mathcal{B}$ outputs whatever the adversary $\mathcal{A}$ outputs.

It is straightforward that the algorithm $\mathcal{B}$ simulates either the game H_8 or H_9 depending on whether $s_0 = \mathbf{b}_0^\top \mathbf{r}$ for some $\mathbf{r} \xleftarrow{\$} \{-1,1\}^m$ or $s_0 \xleftarrow{\$} \mathbb{Z}_q$. We know $\mathcal{B}$ simulates either Game H_8 or Game H_9 up to negligible statistical distance. Therefore, the advantage of the algorithm $\mathcal{B}$ of breaking leftover hash lemma has negligible difference from the advantage of the adversary $\mathcal{A}$ of distinguishing between Game H_8 and H_9.

Hence $\mathsf{H}_8 \overset{s}{\approx} \mathsf{H}_9$. □

□

C. Broadcast Encryption

As discussed in [BV22, Wee22, Wee24], CP-ABE for circuits gives a broadcast encryption for $N = \ell$ users. In the broadcast encryption, we can use a circuit of size $O(N \log N)$ and depth $O(\log N)$ to check the membership of the users. More precisely,

Corollary 2 (Broadcast encryption)**.** *Under the* $\mathrm{poly}(\lambda)$*-succinct* LWE *assumption, we have a broadcast encryption scheme for* N *users with parameters*

$$|\mathsf{pk}| = N \cdot \mathrm{poly}(\lambda, \log N), \quad |\mathsf{sk}| = \mathrm{poly}(\lambda, \log N), \quad |\mathsf{ct}| = \mathrm{poly}(\lambda).$$

References

[ABB10] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (h)ibe in the standard model. In *Advances in Cryptology – EUROCRYPT 2010*, pages 553–572, 2010.

[AFV11] Shweta Agrawal, David Mandell Freeman, and Vinod Vaikuntanathan. Functional encryption for inner product predicates from learning with errors. In *Advances in Cryptology – ASIACRYPT 2011*, pages 21–40, 2011.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 99–108, 1996.

[AKY24] Shweta Agrawal, Simran Kumari, and Shota Yamada. Attribute based encryption for turing machines from lattices. In *Advances in Cryptology – CRYPTO 2024*, pages 352–386, 2024.

[AWY20] Shweta Agrawal, Daniel Wichs, and Shota Yamada. Optimal broadcast encryption from lwe and pairings in the standard model. In *Theory of Cryptography*, pages 149–178, 2020.

[AY20] Shweta Agrawal and Shota Yamada. Optimal broadcast encryption from pairings and lwe. In *Advances in Cryptology – EUROCRYPT 2020*, pages 13–43, 2020.

[Ban93] Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Math. Annalen*, 296(4):625–635, 1993.

[BDE+18] Jonathan Bootle, Claire Delaplace, Thomas Espitau, Pierre-Alain Fouque, and Mehdi Tibouchi. LWE without modular reduction and improved side-channel attacks against BLISS. In *Advances in Cryptology - ASIACRYPT 2018*, pages 494–524, 2018.

[Bei96] Amos Beimel. *Secure schemes for secret sharing and key distribution*. PhD thesis, Technion - Israel Institute of Technology, Israel, 1996.

[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 *Advances in Cryptology – EUROCRYPT 2014*, pages 533–556, 2014.

[Boy13] Xavier Boyen. Attribute-based functional encryption on lattices. In *Theory of Cryptography*, pages 122–142, 2013.

[BTVW17] Zvika Brakerski, Rotem Tsabary, Vinod Vaikuntanathan, and Hoeteck Wee. Private constrained PRFs (and more) from LWE. In *Theory of Cryptography - 15th International Conference, TCC 2017*, pages 264–302, 2017.

[BV16] Zvika Brakerski and Vinod Vaikuntanathan. Circuit-abe from lwe: Unbounded attributes and semi-adaptive security. In *Advances in Cryptology – CRYPTO 2016*, pages 363–384, 2016.

[BV22] Zvika Brakerski and Vinod Vaikuntanathan. Lattice-inspired broadcast encryption and succinct ciphertext-policy ABE. In *13th Innovations in Theoretical Computer Science Conference, ITCS 2022*, volume 215, pages 28:1–28:20, 2022.

[CW23] Valerio Cini and Hoeteck Wee. ABE for circuits with $\mathrm{poly}(\lambda)$-sized keys from LWE. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 435–446, 2023.

[DKW21] Pratish Datta, Ilan Komargodski, and Brent Waters. Decentralized multi-authority abe for dnfs from lwe. In *Advances in Cryptology – EUROCRYPT 2021*, pages 177–209, 2021.

[GGH+13] Sanjam Garg, Craig Gentry, Shai Halevi, Amit Sahai, and Brent Waters. Attribute-based encryption for circuits from multilinear maps. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013*, pages 479–499, 2013.

[GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006*, pages 89–98, 2006.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proc. 41st Annu. ACM Symp. Theory Comput.*, pages 197–206, 2008.

[GV15] Sergey Gorbunov and Dhinakaran Vinayagamurthy. Riding on asymmetry: Efficient abe for branching programs. In *Advances in Cryptology – ASIACRYPT 2015*, pages 550–574, 2015.

[HLL24] Yao-Ching Hsieh, Huijia Lin, and Ji Luo. A general framework for lattice-based ABE using evasive inner-product functional encryption. In *Advances in Cryptology – EUROCRYPT 2024*, pages 433–464, 2024.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Advances in Cryptology - EUROCRYPT 2012*, pages 700–718, 2012.

[MR07] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007.

[SW05] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *Advances in Cryptology – EUROCRYPT 2005*, pages 457–473, 2005.

[Tsa19] Rotem Tsabary. Fully secure attribute-based encryption for t-cnf from lwe. In *Advances in Cryptology – CRYPTO 2019*, pages 62–85, 2019.

[Wee22] Hoeteck Wee. Optimal broadcast encryption and CP-ABE from evasive lattice assumptions. In *Advances in Cryptology – EUROCRYPT 2022*, pages 217–241, 2022.

[Wee24] Hoeteck Wee. Circuit ABE with $\mathrm{poly}(\mathsf{depth}, \lambda)$-sized ciphertexts and keys from lattices. In *Advances in Cryptology – CRYPTO 2024*, pages 178–209, 2024.

[Wee25] Hoeteck Wee. Almost optimal KP and CP-ABE for circuits from Succinct LWE. In *Advances in Cryptology – EUROCRYPT 2025*, pages 34–62, 2025.

[WWW22] Brent Waters, Hoeteck Wee, and David J. Wu. Multi-authority ABE from lattices without random oracles. In *Theory of Cryptography - 20th International Conference, TCC 2022*, pages 651–679, 2022.