Unbounded Broadcast and KP-ABE with Sublinear Ciphertext from Pairings

Junichi Tomida and Hoeteck Wee

NTT Research, CA, USA.

Abstract. We present the first pairing-based unbounded broadcast encryption and key-policy attribute-based encryption (KP-ABE) with sublinear ciphertext size. Here, unbounded means set-up and the public parameters do not impose a bound on the size of the broadcast set, attribute length, or policy size.

- Our broadcast encryption scheme supports an unbounded number of users, and achieves

$$|\mathsf{mpk}| = O(1), |\mathsf{ct}| = O(\sqrt{N}), |\mathsf{sk}| = O(\sqrt{N})$$

  where N denotes an upper bound on the size of the broadcast set.
- Our KP-ABE supports boolean formula and span programs, and achieves

$$|\mathsf{mpk}| = O(1), |\mathsf{ct}| = O(\sqrt{N}), |\mathsf{sk}| = O(\sqrt{N} \cdot |f|)$$

  where N is the attribute length and $|f|$ the policy size.

We prove adaptive security for the broadcast encryption and selective security for the KP-ABE, based on the k-Lin assumption in the standard model without random oracles.

Keywords: Attribute-based encryption, Broadcast encryption, Pairings

Table of Contents

1 Introduction . . . 3
1.1 Our Results . . . 3
2 Technical Overview . . . 4
2.1 Overview of our constructions . . . 4
2.2 Security overview . . . 6
2.3 Instantiation in prime-order groups . . . 8
2.4 Unbounded KP-ABE . . . 9
2.5 Paper Organization . . . 10
3 Preliminaries . . . 10
3.1 Bilinear Entropy Expansion Lemma . . . 12
4 Unbounded ABE from Linear CDS . . . 15
4.1 Construction . . . 15
4.2 Security . . . 16
5 Unbounded Broadcast Encryption . . . 21
5.1 From Bounded to Unbounded Security. . . . 21
5.2 Unbounded ABE for Quadratic Polynomials . . . 22
5.3 Unbounded Broadcast Encryption . . . 24
6 Our Unbounded KP-ABE for Span Programs from CDS . . . 25
6.1 Preliminaries . . . 25
6.2 Construction . . . 26
6.3 Security . . . 27
6.4 Instantiations . . . 28
References . . . 29

1 Introduction

In this work, we study broadcast encryption [FN94] as well as key-policy attribute-based encryption (KP-ABE) schemes [SW05, GPSW06], where ciphertexts ct and keys sk are associated with descriptive values which determine whether decryption is possible. In broadcast encryption, ciphertexts are associated with a set X and a message M and key with a value y, and decryption returns M when $y \in X$. In KP-ABE, ciphertexts are associated with an attribute x and a message M and keys with a policy f, and decryption returns M when x satisfies f. Throughout, we require security against unbounded collusions, so that an adversary that sees a ciphertext along with an arbitrary number of secret keys learns nothing about the message as long as none of the keys can decrypt the ciphertext.

Unbounded broadcast and KP-ABE. Following [LW11], we consider unbounded broadcast and KP-ABE, where-in set-up and the public parameters mpk do not impose an a priori bound on the size of the broadcast set, attribute length, or policy size. We note that such unbounded schemes follow from recent constructions of "optimal" broadcast and KP-ABE [AY20, AWY20, BV22, Wee22, Wee24, AKY24, Wee25, WW25]; however these schemes either rely on LWE plus generic groups, or non-standard variants of the LWE assumption. In this work, we focus on schemes based on standard assumptions, and in the standard model without random oracles. Here, we do know unbounded broadcast and KP-ABE from either the standard k-Lin assumption in pairing groups [LW11,Lew12,OT12,RW13,Att14,KL15,Att16,AC17,CGKW18,KW19,Att19,AT20], or the LWE assumption [BV16, GKW16, CW24], but they all have ciphertext size that is linear in the size of the broadcast set or the attribute length.

1.1 Our Results

In this work, we present the first unbounded broadcast encryption and key-policy KP-ABE with sublinear ciphertext size based on standard assumptions.

- Our broadcast encryption scheme supports an unbounded number of users[1], and achieves

$$|\mathsf{mpk}| = O(1), |\mathsf{ct}| = O(\sqrt{N}), |\mathsf{sk}| = O(\sqrt{N})$$

  where N denotes an upper bound on the size of the broadcast set.
- Our KP-ABE supports boolean formula and span programs, and achieves

$$|\mathsf{mpk}| = O(1), |\mathsf{ct}| = O(\sqrt{N}), |\mathsf{sk}| = O(\sqrt{N} \cdot |f|)$$

  where N is the attribute length and $|f|$ the policy size.

We prove adaptive security for the broadcast encryption and selective security for the KP-ABE, based on the k-Lin assumption in the standard model without random oracles.

Comparison with prior works. We provide a brief comparison with prior pairing-based schemes, including constructions based on non-standard assumptions, or even the generic group model.

- All prior pairing-based broadcast encryption e.g. [BGW05, Wee21] requires $|\mathsf{mpk}| \cdot |\mathsf{ct}| = \Omega(N^{2/3})$ (see Fig 1); we beat this bound, with $|\mathsf{mpk}| \cdot |\mathsf{ct}| = O(\sqrt{N})$. Moreover, the only pairing-based unbounded broadcast is the trivial one that uses an identity-based encryption (IBE) scheme to encrypt to every user in the set, where $|\mathsf{ct}| = O(N)$.[2]
- All prior pairing-based KP-ABE e.g. [GPSW06, ALdP11, AHM+16] requires $|\mathsf{mpk}| \cdot |\mathsf{ct}| = \Omega(N)$; we beat this bound, with $|\mathsf{mpk}| \cdot |\mathsf{ct}| = O(\sqrt{N})$. Prior unbounded KP-ABE for boolean formula and span programs [LW11, Lew12, OT12, RW13, Att14, KL15, Att16, AC17, CGKW18, KW19, Att19, AT20] achieve $|\mathsf{ct}| = O(N), |\mathsf{sk}| = O(|f|)$.

[1] sometimes referred to as identity-based broadcast encryption in the literature.

[2] [Att19] provided a direct unbounded broadcast construction from pairings, but its asymptotic efficiency is the same as the trivial construction from identity-based encryption.

Scheme	$\|\mathsf{mpk}\|$	$\|\mathsf{ct}\|$	$\|\mathsf{sk}\|$	Assumption	Security
[BGW05, Del07, GW09]	N	1	1 †	q-type	selective
	$\sqrt{N}$	$\sqrt{N}$	1 †		
[Wee16, CMM16]	N	1	1 †	subgroup decision	selective
[GKW18]	N^2	1	1 †	k-Lin, $k \geq 1$	adaptive
[BW06, LVW17]	$\sqrt{N}$	$\sqrt{N}$	$\sqrt{N}$	k-Lin, $k \geq 1$	adaptive
[Wee21]	$N^{1/3}$	$N^{1/3}$	$N^{1/3}$	bi-k-Lin*, $k \geq 2$	adaptive
this work	1	$\sqrt{N}$	$\sqrt{N}$	k-Lin, $k \geq 1$	adaptive

Fig 1: Comparison with prior pairing-based broadcast encryption schemes for broadcast sets of size at most N, where $|\mathsf{mpk}|, |\mathsf{ct}|, |\mathsf{sk}|$ refer to number of group elements, ignoring $O(1)$ factors. Note that $|\mathsf{ct}|$ ignores the contribution from the set S, which is "public".
† Here, decryption requires knowledge of mpk in addition to sk.
* Here, bi-k-Lin (bilateral k-Lin) is a strengthening of k-Lin.

Brief overview. The main technical novelty of this work lies in the broadcast encryption scheme, notably, one achieving $|\mathsf{mpk}| = O(1), |\mathsf{ct}| = O(\sqrt{N})$. It is helpful to think about the construction abstractly. We adopt the framework of ABE for general predicates P, where ciphertexts are associated with attributes x and keys with policies y, and decryption succeeds when $\mathsf{P}(x, y) = 1$. Broadcast encryption corresponds to set membership (with x encoding the recipient set), while KP-ABE captures general satisfiability (where y specifies the policy).

At a high level, we construct a new compiler that transforms a pairing-based ABE scheme from a specific class —namely, those derived from conditional disclosure of secrets (CDS) [GIKM98, Att14, Wee14][3]— into an unbounded ABE for the same predicate with parameters

$$|\mathsf{mpk}'| = O(1),\ |\mathsf{ct}'| = |\mathsf{ct}| + O(|\mathsf{mpk}|),\ |\mathsf{sk}'| = |\mathsf{sk}| + O(|\mathsf{mpk}|)$$

Crucially, this transformation compresses the public parameters to constant size while increasing ciphertext and key sizes only additively in $|\mathsf{mpk}|$. Instantiating this compiler with the broadcast encryption of [LVW17] with $|\mathsf{mpk}| = |\mathsf{ct}| = |\mathsf{sk}| = O(\sqrt{N})$ yields our new unbounded broadcast encryption scheme.

We start with a compiler implicit in prior unbounded KP-ABE [LW11, OT12, CGKW18]. This compiler preserves the decryption functionality only if we start schemes whose ciphertext structure corresponds to projection "in the exponent". This restriction excludes broadcast encryption schemes where ciphertext components encode more general linear relations. Our main technical advance is overcoming this barrier by developing a compiler that supports all linear functions in the exponent, thereby enabling parameter compression in settings previously out of reach. Our compiler introduces another additive $O(|\mathsf{mpk}|)$ overhead to both the ciphertext and the secret key to facilitate decryption, while preserving security.

2 Technical Overview

We use boldface lower case to denote vectors, and boldface upper case to denote matrices. We rely on a bilinear group (G_1, G_2, G_T, e) of prime order p where $e : G_1 \times G_2 \to G_T$. We use $[\cdot]_1, [\cdot]_2, [\cdot]_T$ to denote component-wise exponentiations in respective groups G_1, G_2, G_T. The k-Lin assumption in G_1 asserts that $([\mathbf{A}]_1, [\mathbf{A}\mathbf{s}^\top]_1) \approx_c ([\mathbf{A}]_1, [\mathbf{u}^\top]_1)$ where $\mathbf{s} \leftarrow \mathbb{Z}_p^k, \mathbf{A} \leftarrow \mathbb{Z}_p^{(k+1)\times k}, \mathbf{u} \leftarrow \mathbb{Z}_p^{k+1}$.

2.1 Overview of our constructions

We provide an overview of our constructions, focusing on the unbounded broadcast encryption.

[3] This does not include the $N^{1/3}$-sized broadcast encryption in [Wee21].

ABE from linear CDS. We start by describing a broad class of pairing-based ABE derived from conditional disclosure of secrets (CDS) [GIKM98, Att14, Wee14]. A CDS for a predicate P comprises three functions $c = c_x, k = k_y, d = d_{x,y}$ such that

- for all x, y such that $\mathsf{P}(x, y) = 1$, for all $\mathbf{w}, \alpha$, we have $d(c(\mathbf{w}), k(\alpha, \mathbf{w})) = \alpha$;
- for all x, y such that $\mathsf{P}(x, y) = 0$, the joint distribution $(c(\mathbf{w}), k(\alpha, \mathbf{w}))$ perfectly hide α for a random $\mathbf{w}$.

We say that a CDS is linear, if the three functions c, k, d are linear over $\mathbb{Z}_p$. Starting from a linear CDS with $\mathbf{w} = (w_1, \ldots, w_n) \in \mathbb{Z}_p^n$, we can derive an ABE for the same predicate as follows [Att14, Wee14]:[4]

$$\begin{aligned} \mathsf{mpk} &= [w_1, \ldots, w_n]_1, [\alpha]_T \\ \mathsf{ct} &= [c(s_0 w_1, \ldots, s_0 w_n),\ s_0]_1, [s_0 \alpha]_T \cdot M \\ \mathsf{sk} &= [k(\alpha, w_1 r_0, \ldots, w_n r_0),\ r_0]_2 \end{aligned} \tag{1}$$

where c, k are linear functions. Decryption recovers $[s_0\alpha]_T$ using

$$d\big(\overbrace{[c(s_0 w_1, \ldots, s_0 w_n) \cdot r_0]_T}^{=c(s_0 w_1 r_0, \ldots, s_0 w_n r_0)}, \overbrace{[s_0 \cdot k(\alpha, w_1 r_0, \ldots, w_n r_0)]_T}^{=k(s_0\alpha, s_0 w_1 r_0, \ldots, s_0 w_n r_0)}\big)$$

where d is computed "in the exponent".

Looking ahead, we show how to construct an unbounded ABE for P with

$$|\mathsf{mpk}| = O(1),\ |\mathsf{ct}| = O(|c| + n),\ |\mathsf{sk}| = O(|k| + n)$$

where $|c|, |k|$ denote the output lengths of c, k respectively. Combining this with the linear CDS for broadcast in [LVW17] where $|c| = |k| = n = O(\sqrt{N})$ yields our unbounded broadcast encryption.

Previous compiler. The previous works of [LW11, OT12, CGKW18] show how to reduce the size of mpk to a constant for restricted classes of c. The compiler implicit in these works is sufficient for constructing KP-ABE schemes for boolean formula where c is the projection function, but not broadcast encryption where c exploits the full power of linearity to obtain small ciphertexts. The compiler works as follows: replace

$$\begin{aligned} [s_0 w_i]_1 &\mapsto [s_0 v + s_i(u_1 + i u_2)]_1 \\ [w_i r_0]_2 &\mapsto [v r_i]_2 \end{aligned}$$

and append $[s_i]_1$ to ct and $[r_i,\ (u_1 + i u_2) r_i]_2$ to sk.

$$\begin{aligned} \mathsf{mpk} &= [v,\ u_1,\ u_2]_1, [\alpha]_T \\ \mathsf{ct} &= [c(s_0 v + s_1(u_1 + u_2), \ldots, s_0 v + s_n(u_1 + n u_2)), s_0,\ s_1, \ldots, s_n]_1, [s_0\alpha]_T \cdot M \\ \mathsf{sk} &= [k(\alpha, v r_1, \ldots, v r_n),\ \{\, r_i,\ (u_1 + i u_2) r_i \,\}_{i \in [n]}]_2 \end{aligned}$$

Here, decryption recovers $[s_0\alpha]_T$ using

$$d\big([c(s_0 v r_1, \ldots, s_0 v r_n)]_T, \overbrace{[k(s_0\alpha, s_0 v r_1, \ldots, s_0 v r_n)]_T}^{=e([s_0]_1, [k(\alpha, r_!, \ldots, r_n)]_2)}\big)$$

In order to compute the first input to d in the above equation, we need to first compute $[c(\{\, (s_0 v + s_i(u_1 + i u_2)) r_i \,\}_{i \in [n]})]_T$. This steps only works when c is the projection function, so that we can compute $[c(t_1 r_1, \ldots, t_n r_n)]_T$ given just $[c(t_1, \ldots, t_n)]_1, \{\, [r_i]_2 \,\}_{i \in [n]}$, where $t_i = s_0 v + s_i(u_1 + i u_2)$.[5] However, the function c underlying the linear CDS for size $O(\sqrt{N})$ broadcast [LVW17] is not a projection function.

[4] Readers may interpret a linear CDS as a pair encoding [Att14, Wee14] if the latter is more familiar.

[5] As a concrete example, we can compute $(t_1 r_1, t_2 r_2)$ given $(t_1, t_2), r_1, r_2$ but we cannot compute $t_1 r_1 + t_2 r_2$ given $t_1 + t_2, r_1, r_2$.

Our compiler. We present a new transformation that works for any linear function c. As a warm-up, we present a simple transformation that is insecure: we start with (1) and replace

$$[s_0 w_i]_1 \mapsto [s_i v_1]_1$$
$$[w_i r_0]_2 \mapsto [v_2 r_i]_2$$

and append $[s_i]_1$ and $[s_0 v_2]_1$ to ct and $[r_i]_2$ and $[v_1 r_0]_2$ to sk.

$$\begin{aligned}\mathsf{mpk} &= [v_1,\ v_2]_1, [\alpha]_T \\ \mathsf{ct} &= [c(s_1 v_1, \ldots, s_n v_1),\ s_0,\ s_1, \ldots, s_n,\ s_0 v_2]_1, [s_0\alpha]_T \cdot M \\ \mathsf{sk} &= [k(\alpha, v_2 r_1, \ldots, v_2 r_n),\ r_0,\ r_1, \ldots, r_n,\ v_1 r_0]_2\end{aligned}$$

Decryption first computes

$$\begin{aligned}&[c(s_0 v_2 r_1, \ldots, s_0 v_2 r_n)]_T \\ &= e([c(s_1 v_1, \ldots, s_n v_1)]_1,\ [r_0]_2) \cdot c(\{\ [s_0 v_2 \cdot r_i - s_i \cdot v_1 r_0]_T\ \}_{i\in[n]})\end{aligned}$$

and then computes $[s_0\alpha]_T = d([c(s_0 v_2 r_1, \ldots, s_0 v_2 r_n)]_T,\ [k(s_0\alpha, s_0 v_2 r_1, \ldots, s_0 v_2 r_n)]_T)$. Note that decryption now works for any linear function c. However, the scheme is insecure because an adversary can compute the expression on the LHS of the preceding equation for any reordering of the inputs $s_0 v_2 r_1, \ldots, s_0 v_2 r_n$ to c. The attack arises because an adversary can compute $[s_0 v_2 \cdot r_j - s_i \cdot v_1 r_0]_T$ for any i, j. To prevent this attack, we introduce masking terms similar to the previous compiler where-in we further replace

$$[s_0 v_2]_1 \mapsto \{\ [s_0 v_2 + s_i(u_1 + i u_2)]_1\ \}_{i\in[n]}$$
$$[v_1 r_0]_2 \mapsto \{\ [v_1 r_0 + (u_1 + i u_2) r_i]_2\ \}_{i\in[n]}$$

Decryption proceeds as before, except we compute

$$s_0 v_2 \cdot r_i - s_i v_1 \cdot r_0 = \overbrace{(s_0 v_2 + s_i(u_1 + i u_2))}^{\mathsf{ct}} \cdot \overbrace{r_i}^{\mathsf{sk}} - \overbrace{s_i}^{\mathsf{ct}} \cdot \overbrace{(v_1 r_0 + (u_1 + i u_2) r_i)}^{\mathsf{sk}}$$

That is, our new transformation is given by:

$$\begin{aligned}\mathsf{mpk} &= [v_1,\ v_2,\ u_1,\ u_2]_1, [\alpha]_T \\ \mathsf{ct} &= [c(s_1 v_1, \ldots, s_n v_1),\ s_0,\ \{\ s_i,\ s_0 v_2 + s_i(u_1 + i u_2)\ \}_{i\in[n]}]_1, [s_0\alpha]_T \cdot M \\ \mathsf{sk} &= [k(\alpha, v_2 r_1, \ldots, v_2 r_n),\ r_0,\ \{\ r_i,\ v_1 r_0 + (u_1 + i u_2) r_i\ \}_{i\in[n]}]_2\end{aligned} \tag{2}$$

Observe that we cannot efficiently compute $[s_0 v_2 \cdot r_j - s_i v_1 \cdot r_0]_T$ for $i \neq j$ from the above elements. The underlying security mechanism here is essentially the same as Boneh-Boyen IBE [BB04] (also used in the previous compiler), where $[s_0\alpha]_T$ is not efficiently computable from,

$$\mathsf{ct} = [s_0,\ s_0(u_1 + i u_2)]_1, \quad \mathsf{sk} = [r_0, \alpha + r_0(u_1 + j u_2)]_2$$

for $i \neq j$. As with the previous compiler, we increase both ciphertext and key size by $O(n)$ group elements. We note that correctness works even if $v_1 = v_2$; the reason we use independent v_1, v_2 is for security.

2.2 Security overview

The security proof follows the dual system encryption methodology, which comprises of (i) computational steps that rely on the k-Lin assumption and are independent of the linear CDS c, k, d, as well as (ii) an information theoretic step that exploits the security guarantee of the linear CDS.[6] We begin by outlining the key computational and information theoretic arguments that are novel in this work.

[6] The reason for working with information-theoretic security here is that (perfect) selective security implies adaptive security.

Entropy expansion. We can start with the proof techniques in [CGKW18], where they showed statements of the form:

$$\left\{ \begin{array}{l} [s_0,\ s_i,\ s_0 v + s_i(u_1 + iu_2)]_1, \\ [r_i,\ vr_i,\ (u_1 + iu_2)r_i]_2 \end{array} \right\}_{i\in[N]} \approx_c \left\{ \begin{array}{l} [s_0,\ s_i,\ s_0 w_i + s_i u_i]_1, \\ [r_i,\ w_i r_i,\ u_i r_i]_2 \end{array} \right\}_{i\in[N]}$$

where on the RHS $u_1, \ldots, u_N, w_1, \ldots, w_N \leftarrow \mathbb{Z}_p$. We note that the distributions $[s_0, \{\ s_i, c_i\ \}_{i\in[n]}]_1, [\{\ r_i, d'_i, d_i\ \}_{i\in[n]}]_2$ on both sides satisfy the consistency check $c_i \cdot r_i = s_i \cdot d_i + s_0 \cdot d'_i$ (which can be verified using a pairing).

In our setting, we want a statement with extra terms involving v_2, r_0 on the LHS:

$$\left\{ \begin{array}{l} [s_0,\ s_i,\ s_i v_1, s_0 v_2 + s_i(u_1 + iu_2)]_1, \\ [r_0,\ r_i,\ v_2 r_i,\ v_1 r_0 + (u_1 + iu_2)r_i]_2 \end{array} \right\}_{i\in[N]} \approx_c \left\{ \begin{array}{l} [s_0,\ s_i,\ s_i w_i,\ s_0 w'_i + s_i u'_i]_1, \\ [r_0,\ r_i,\ w'_i r_i,\ w_i r_0 + u'_i r_i]_2 \end{array} \right\}_{i\in[N]} \tag{3}$$

Two remarks to help the reader parse the above statement:

- The distributions $[s_0, \{\ s_i, c'_i, c_i\ \}_{i\in[n]}]_1, [r_0, \{\ r_i, d'_i, d_i\ \}_{i\in[n]}]_2$ on both sides satisfy the consistency check $c_i \cdot r_i + c'_i \cdot r_0 = s_i \cdot d_i + s_0 \cdot d'_i$.[7]
- Setting (u'_i, w_i, w'_i) on the RHS to be $(u_1 + iu_2, v_1, v_2)$ yields the LHS.

Applying the original entropy expansion lemma to the LHS of (3), we have:

$$\text{LHS of (3)} \approx_c \left\{ \begin{array}{l} [s_0,\ s_i,\ s_i v_1,\ s_0 w'_i + s_i u'_i]_1, \\ [r_0,\ r_i,\ w'_i r_i,\ v_1 r_0 + u'_i r_i]_2 \end{array} \right\}_{i\in[N]}$$

Next, we have by DDH[8] $[s_i, s_i v_1]_1 \approx_c [s_i, s_i w_i]_1$ to obtain the RHS of (3).

Information-theoretic step. In order to motivate the information-theoretic step, observe that applying (3) to our scheme in (2) yields:

$$(\mathsf{ct}, \mathsf{sk}) \approx_c \begin{array}{l} [c(s_1 w_1, \ldots, s_n w_n),\ s_0,\ \{\ s_i,\ s_0 w'_i + s_i u'_i\ \}_{i\in[n]}]_1, [s_0 \alpha]_T \cdot M \\ [k(\alpha, w'_1 r_1, \ldots, w'_n r_n),\ r_0,\ \{\ r_i,\ w_i r_0 + u'_i r_i\ \}_{i\in[n]}]_2 \end{array} \tag{4}$$

Note that this above statement holds for any $(\mathsf{ct}, \mathsf{sk})$ pair (even if $\mathsf{P}(x, y) = 1$) and only uses the fact that c, k compute linear functions.

Next, we want to argue that the RHS of (4) perfectly hides M whenever $\mathsf{P}(x, y) = 0$. It suffices to show that for all $s_0, s_1, \ldots, s_n, r_0, r_1, \ldots, r_n \neq 0$, for all α and for all x, y s.t. $\mathsf{P}(x, y) = 0$, the distributions

$$\begin{array}{l} c(s_1 w_1, \ldots, s_n w_n), \{\ s_0 w'_i + s_i u'_i\ \}_{i\in[n]} \\ k(\alpha, w'_1 r_1, \ldots, w'_n r_n),\ \{\ w_i r_0 + u'_i r_i\ \}_{i\in[n]} \\ w_i, w'_i, u'_i \leftarrow \mathbb{Z}_p, i \in [n] \end{array}$$

perfectly hide α. As a sanity check, observe that if $s_0 = s_1 = \cdots = s_n = r_0 = r_1 = \cdots = r_n = 1$ and $w_i = w'_i$, then the statement follows readily from linear CDS security, since u'_i perfectly masks $w_i = w'_i$. In the more general setting, it is sufficient to show that the following distributions hide α:

$$\begin{array}{l} c(s_1 w_1, \ldots, s_n w_n) \cdot r_0, \{\ (s_0 w'_i + s_i u'_i) \cdot r_i\ \}_{i\in[n]} \\ s_0 \cdot k(\alpha, w'_1 r_1, \ldots, w'_n r_n),\ \{\ s_i \cdot (w_i r_0 + u'_i r_i)\ \}_{i\in[n]} \end{array}$$

[7] If we set $v_1 = v_2 = v$, then there would be $O(n^2)$ additional consistency checks given by $c'_i \cdot r_j = s_i \cdot d'_j$ for all i, j, which are violated on the RHS.

[8] Here, we need an extra "change of variables" step as in [CGKW18] to account for the fact that v_1 also appears in $[v_1 r_0 + u'_i r_i]_2$.

With additional linear algebra (similar to that used for decryption), and also using w_i, u'_i to mask w'_i, this boils down to arguing that α is perfectly hidden given

$$c(s_0 w'_1 r_1, \ldots, s_0 w'_n r_n)$$
$$k(s_0\alpha, s_0 w'_1 r_1, \ldots, s_0 w'_n r_n)$$

The latter follows readily from linear CDS security.

2.3 Instantiation in prime-order groups

Following [CGW15, GDCC16, CGKW18], we start with our ABE scheme in (2), sample $\mathbf{A}_1 \leftarrow \mathbb{Z}_p^{3k\times k}, \mathbf{B} \leftarrow \mathbb{Z}_p^{k\times(k+1)}$, and carry out the following substitutions:

$$\begin{array}{llll} \alpha & \mapsto \mathbf{k} \in \mathbb{Z}_p^{3k} & v_1, v_2, u_1, u_2 \mapsto \mathbf{V}_1, \mathbf{V}_2, \mathbf{U}_1, \mathbf{U}_2 \in \mathbb{Z}_p^{(k+1)\times 3k} & \\ s_0, s_i & \mapsto \mathbf{s}_0, \mathbf{s}_i \in \mathbb{Z}_p^k, & r_0, r_i \mapsto \mathbf{r}_0, \mathbf{r}_i \in \mathbb{Z}_p^k & \\ [s_i]_1 & \mapsto [\mathbf{A}_1\mathbf{s}_i^\top]_1, & [r_i]_2 \mapsto [\mathbf{r}_i\mathbf{B}]_2 & \\ [s_i v_j]_1 & \mapsto [\mathbf{V}_j\mathbf{A}_1\mathbf{s}_i^\top]_1, & [v_j r_i]_2 \mapsto [\mathbf{r}_i\mathbf{B}\mathbf{V}_j]_2, & [s_0\alpha]_T \mapsto [\mathbf{k}\mathbf{A}\mathbf{s}_0^\top]_T \end{array} \tag{5}$$

The scheme is as follows:

$$\begin{aligned} \mathsf{mpk} &= ([\mathbf{A}_1, \mathbf{U}_1\mathbf{A}_1, \mathbf{U}_2\mathbf{A}_1, \mathbf{V}_1\mathbf{A}_1, \mathbf{V}_2\mathbf{A}_1]_1, [\mathbf{k}\mathbf{A}_1]_T) \\ \mathsf{ct} &= \begin{pmatrix} c([\mathbf{V}_1\mathbf{A}_1\mathbf{s}_1^\top, \ldots, \mathbf{V}_1\mathbf{A}_1\mathbf{s}_n^\top]_1) \\ [\mathbf{A}_1\mathbf{s}_0^\top, \{\mathbf{A}_1\mathbf{s}_i^\top, \mathbf{V}_2\mathbf{A}_1\mathbf{s}_0^\top + (\mathbf{U}_1 + i\mathbf{U}_2)\mathbf{A}_1\mathbf{s}_i^\top\}_{i\in[n]}]_1,\ [\mathbf{k}\mathbf{A}\mathbf{s}_0^\top]_T \cdot M \end{pmatrix} \\ \mathsf{sk} &= (k([\mathbf{k}, \mathbf{r}_1\mathbf{B}\mathbf{V}_2, \ldots, \mathbf{r}_n\mathbf{B}\mathbf{V}_2]_2), [\mathbf{r}_0\mathbf{B}, \{\mathbf{r}_i\mathbf{B}, \mathbf{r}_0\mathbf{B}\mathbf{V}_1 + \mathbf{r}_i\mathbf{B}(\mathbf{U}_1 + i\mathbf{U}_2)\}_{i\in[n]}]_2) \end{aligned}$$

Proof overview. The security proof then uses the following sequence of games. We additionally sample $\mathbf{A}_2, \mathbf{A}_2^\parallel \leftarrow \mathbb{Z}_p^{3k\times k}$ conditioned on $\mathbf{A}_2^{\parallel\top}\mathbf{A}_1 = \mathbf{O}$ and $\mathbf{A}_2^{\parallel\top}\mathbf{A}_2 = \mathbf{I}_k$.

- H_0: ABE security game where the adversary sees one ciphertext and Q secret keys.
- H_1: we switch the ciphertext and Q secret keys to the "expanded" distribution where we replace

$$\begin{aligned} \mathbf{A}_1\mathbf{s}_0^\top, \ldots, \mathbf{A}\mathbf{s}_n^\top &\mapsto \mathbf{c}_0^\top, \ldots, \mathbf{c}_n^\top \leftarrow \mathrm{span}(\mathbf{A}_1, \mathbf{A}_2) \\ \mathbf{U}_1 + i\mathbf{U}_2 &\mapsto \mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}'_i \\ \mathbf{V}_1, \mathbf{V}_2 &\mapsto \mathbf{V}_1 + \mathbf{W}_i, \mathbf{V}_2 + \mathbf{W}'_i \end{aligned}$$

  where $\mathbf{U}'_i, \mathbf{W}_i, \mathbf{W}'_i \leftarrow \mathrm{span}(\mathbf{A}_2^{\parallel\top})$. That is,

$$\begin{aligned} \mathsf{ct} &= \begin{pmatrix} c([(\mathbf{V}_1 + \mathbf{W}_1)\mathbf{c}_1^\top, \ldots, (\mathbf{V}_1 + \mathbf{W}_n)\mathbf{c}_n^\top]_1) \\ [\mathbf{c}_0^\top, \{\mathbf{c}_i^\top, (\mathbf{V}_2 + \mathbf{W}'_i)\mathbf{c}_0^\top + (\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}'_i)\mathbf{c}_i^\top\}_{i\in[n]}]_1,\ [\mathbf{k}\mathbf{c}_0^\top]_T \cdot M \end{pmatrix} \\ \mathsf{sk} &= \begin{pmatrix} k([\mathbf{k}, \mathbf{r}_1\mathbf{B}(\mathbf{V}_2 + \mathbf{W}'_1), \ldots, \mathbf{r}_n\mathbf{B}(\mathbf{V}_2 + \mathbf{W}'_n)]_2) \\ [\mathbf{r}_0\mathbf{B}, \{\mathbf{r}_i\mathbf{B}, \mathbf{r}_0\mathbf{B}(\mathbf{V}_1 + \mathbf{W}_i) + \mathbf{r}_i\mathbf{B}(\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}'_i)\}_{i\in[n]}]_2 \end{pmatrix} \end{aligned}$$

- H_2: we switch the Q secret keys to the semi-functional distribution where we replace $\mathbf{k}$ in with $\mathbf{k} + \mathbf{z}$ where $\mathbf{z} \leftarrow \mathrm{span}(\mathbf{A}_2^{\parallel\top})$ is freshly sampled across all Q keys. That is,

$$\mathsf{sk} = \begin{pmatrix} k([\mathbf{k} + \mathbf{z}, \mathbf{r}_1\mathbf{B}(\mathbf{V}_2 + \mathbf{W}'_1), \ldots, \mathbf{r}_n\mathbf{B}(\mathbf{V}_2 + \mathbf{W}'_n)]_2) \\ [\mathbf{r}_0\mathbf{B}, \{\mathbf{r}_i\mathbf{B}, \mathbf{r}_0\mathbf{B}(\mathbf{V}_1 + \mathbf{W}_i) + \mathbf{r}_i\mathbf{B}(\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}'_i)\}_{i\in[n]}]_2 \end{pmatrix}$$

$\mathsf{H}_0 \approx_c \mathsf{H}_1$ relies on our new entropy expansion lemma. Then, $\mathsf{H}_1 \approx_c \mathsf{H}_2$ is essentially the proof of the bounded scheme (except we don't need to switch the ciphertext to a "semi-functional" distribution). Concretely, we switch the keys one by one from expanded to semi-functional:

$$\begin{aligned}
&\begin{pmatrix} k([\mathbf{k}, \mathbf{r}_1\mathbf{B}(\mathbf{V}_2 + \mathbf{W}'_1), \ldots, \mathbf{r}_n\mathbf{B}(\mathbf{V}_2 + \mathbf{W}'_n)]_2) \\ [\mathbf{r}_0\mathbf{B}, \{\mathbf{r}_i\mathbf{B}, \mathbf{r}_0\mathbf{B}(\mathbf{V}_1 + \mathbf{W}_i) + \mathbf{r}_i\mathbf{B}(\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}'_i)\}_{i\in[n]}]_2 \end{pmatrix} \\
\approx_c &\begin{pmatrix} k([\mathbf{k}, \mathbf{d}_1(\mathbf{V}_2 + \mathbf{W}'_1), \ldots, \mathbf{d}_n(\mathbf{V}_2 + \mathbf{W}'_n)]_2) \\ [\mathbf{d}_0, \{\mathbf{d}_i, \mathbf{d}_0(\mathbf{V}_1 + \mathbf{W}_i) + \mathbf{d}_i(\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}'_i)\}_{i\in[n]}]_2 \end{pmatrix} \\
\approx_s &\begin{pmatrix} k([\mathbf{k} + \mathbf{z}, \mathbf{d}_1(\mathbf{V}_2 + \mathbf{W}'_1), \ldots, \mathbf{d}_n(\mathbf{V}_2 + \mathbf{W}'_n)]_2) \\ [\mathbf{d}_0, \{\mathbf{d}_i, \mathbf{d}_0(\mathbf{V}_1 + \mathbf{W}_i) + \mathbf{d}_i(\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}'_i)\}_{i\in[n]}]_2 \end{pmatrix} \\
\approx_c &\begin{pmatrix} k([\mathbf{k} + \mathbf{z}, \mathbf{r}_1\mathbf{B}(\mathbf{V}_2 + \mathbf{W}'_1), \ldots, \mathbf{r}_n\mathbf{B}(\mathbf{V}_2 + \mathbf{W}'_n)]_2) \\ [\mathbf{r}_0\mathbf{B}, \{\mathbf{r}_i\mathbf{B}, \mathbf{r}_0\mathbf{B}(\mathbf{V}_1 + \mathbf{W}_i) + \mathbf{r}_i\mathbf{B}(\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}'_i)\}_{i\in[n]}]_2 \end{pmatrix} \\
&\text{where } \mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_n \leftarrow \mathbb{Z}_p^{k+1}
\end{aligned}$$

The two $\approx_c$ above follows from the k-Lin assumption; the $\approx_s$ uses the information-theoretic argument outlined earlier. Finally, we can argue that in H_2, the message M is perfectly hidden.

2.4 Unbounded KP-ABE

We construct unbounded KP-ABE for span program following previous schemes, e.g. [GPSW06]. Here, we adopt the simpler notation used in Section 2.1. The idea is that we will compute shares $\{\alpha_j\}$ of α during key generation, and during decryption, we will first compute $[s_0\alpha_j]_T$ for all authorized shares α_j using our new broadcast encryption, upon which we can recover $[s_0\alpha]_T$.

In more detail, we extend our compiler to obtain unbounded KP-ABE for span program from a linear CDS for broadcast encryption. In the KP-ABE, a ciphertext for an attribute set $X \subseteq [N]$ can be decrypted by a secret key for span program $f = (\mathbf{M}, \rho)$ when f accepts X. Recall that ρ maps the rows of $\mathbf{M}$ to $[N]$, and f accepts X when $(1, 0, \ldots, 0)$ lies in the space spanned by all the i-th rows of $\mathbf{M}$ such that $\rho(i) \in X$. The KP-ABE from our compiler achieves

$$|\mathsf{mpk}'| = O(1),\ |\mathsf{ct}'| = |\mathsf{ct}| + O(|\mathsf{mpk}|),\ |\mathsf{sk}'| = (|\mathsf{sk}| + O(|\mathsf{mpk}|)) \cdot |f|$$

where $|\mathsf{mpk}|, |\mathsf{ct}|, |\mathsf{sk}|$ are the sizes of the corresponding elements in the underlying ABE scheme from the linear CDS for broadcast. We describe our KP-ABE for span program:

- mpk and ct are the same as those for the original compiler in (2).
- To generate a KP-ABE secret key for $f = (\mathbf{M}, \rho)$, we first compute shares $(\sigma_1, \ldots, \sigma_{|f|})$ of α for the span program f (if f accepts X, then we can recover α given the shares $\{\sigma_j\}_{j:\rho(j)\in X}$). The KP-ABE key comprises $|f|$ keys for our compiler in (2) corresponding to master secret key σ_j and index $\rho(j)$, for all $j \in [|f|]$.
- Decryption computes $[s_0\sigma_j]_T$ for all j satisfying $\rho(j) \in X$, upon which it can reconstruct $[s_0\alpha]_T$ "in the exponent".

Concretely, we have

$$\begin{aligned}
\mathsf{mpk} &= [v_1,\ v_2,\ u_1,\ u_2]_1, [\alpha]_T \\
\mathsf{ct} &= [c_X(s_1v_1, \ldots, s_nv_1), s_0,\ \{\, s_i,\ s_0v_2 + s_i(u_1 + iu_2)\,\}_{i\in[n]}]_1, [s_0\alpha]_T \cdot M \\
\mathsf{sk} &= \left\{\, [k_{\rho(j)}(\sigma_j, v_2r_{1,j}, \ldots, v_2r_{n,j}),\ \{\, r_{i,j},\ v_1r_{0,j} + (u_1 + iu_2)r_{i,j}\,\}_{i\in[n]}]_2 \,\right\}_{j\in[|f|]}
\end{aligned}$$

Instantiating the above scheme with the linear CDS for broadcast with size $O(\sqrt{N})$ by [LVW17] results in our unbounded KP-ABE for span program with

$$|\mathsf{mpk}| = O(1),\ |\mathsf{ct}| = O(\sqrt{N}),\ |\mathsf{sk}| = O(\sqrt{N} \cdot |f|)$$

2.5 Paper Organization

Section 3 provides preliminaries, which includes our new bilinear entropy expansion lemma and the definition of linear CDS. Section 4 provides how to compile a linear CDS scheme with unbounded security to an unbounded ABE scheme. Section 5 provides how to convert a linear CDS scheme with bounded security to one with unbounded security. The section also shows how to instantiate our new broadcast encryption scheme:

1. We apply the conversion to the linear CDS scheme for quadratic functions with bounded security by [LVW17] to obtain the linear CDS scheme with unbounded security.
2. Then, we apply the compiler in Section 4 to the linear CDS for quadratic functions with unbounded security to achieve the unbounded ABE scheme for quadratic functions.
3. Finally, we achieve our broadcast encryption scheme from the unbounded ABE scheme for quadratic functions by a standard embedding technique.

Section 6 shows how to extend our broadcast encryption scheme to the KP-ABE scheme in a semi-generic way.

3 Preliminaries

Notations. For integers $m \geq 0, n \geq 1$, $[n]$ denotes $\{1, \ldots, n\}$ and $[m, n]$ denotes $\{m, \ldots, n\}$. For a set of column vectors $\mathbf{v}_1^\top, \ldots, \mathbf{v}_n^\top$ over $\mathbb{Z}_p$, $\mathsf{ColSpan}(\mathbf{v}_1^\top, \ldots, \mathbf{v}_n^\top) = \{\mathbf{v}^\top \mid \exists c_1, \ldots, c_n \in \mathbb{Z}_p, \mathbf{v}^\top = \sum_{i\in[n]} c_i \mathbf{v}_i^\top\}$. We sometimes denote a set of n columns of matrix $\mathbf{V}$ with width n by just $\mathbf{V}$. $\mathsf{RowSpan}$ is similarly defined for a set of row vectors. For matrices $\mathbf{M}, \mathbf{N}$ over $\mathbb{Z}_p$ such that $\mathbf{NM}$ is defined, $e([\mathbf{M}]_1, [\mathbf{N}]_2)$ is the pairing operation to compute $[\mathbf{NM}]_T$. For distributions X, Y indexed by the security parameter λ, $X \approx_c Y$ and $X \approx_s Y$ means X and Y are computationally and statistically indistinguishable, respectively. $X \equiv Y$ means X and Y are distributed the same.

Definition 3.1 (Bilinear Groups). A description of bilinear groups $\mathbb{G}=(p, G_1, G_2, G_T, g_1, g_2, e)$ consist of a prime p, cyclic groups G_1, G_2, G_T of order p, generators g_1 and g_2 of G_1 and G_2 respectively, and a bilinear map $e : G_1 \times G_2 \to G_T$, which has two properties.

- (Bilinearity): $\forall h_1 \in G_1, h_2 \in G_2, a, b \in \mathbb{Z}_p, e(h_1^a, h_2^b) = e(h_1, h_2)^{ab}$.
- (Non-degeneracy): For generators g_1, g_2; $g_T = e(g_1, g_2)$ is a generator of G_T.

A bilinear group generator $\mathcal{G}_{\mathsf{BG}}(1^\lambda)$ takes a security parameter 1^λ and outputs a description of bilinear groups $\mathbb{G}$ with a $\Omega(\lambda)$-bit prime p.

Definition 3.2 (Matrix Decisional Diffie-Hellman (MDDH) Assumption [EHK+17]). Let $\eta \in \{1, 2\}$, $j, k, \ell \in \mathbb{N}$ such that $j > k$. We consider the following distribution: $\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda)$, $\mathbf{A} \leftarrow \mathbb{Z}_p^{j\times k}$, $\mathbf{T}_0 \leftarrow \mathsf{ColSpan}(\mathbf{A})^{1\times\ell}$, $\mathbf{T}_1 \leftarrow \mathbb{Z}_p^{j\times\ell}$. We say that the MDDH_k assumption holds with respect to $\mathcal{G}_{\mathsf{BG}}$ if $(\mathbb{G}, [\mathbf{A}, \mathbf{T}_0]_\eta) \approx_c (\mathbb{G}, [\mathbf{A}, \mathbf{T}_1]_\eta)$ for all η, j, ℓ. Throughout the paper, we use k only for the MDDH parameter and often omit k, i.e., we just denote the MDDH assumption. The k-lin assumption implies the MDDH_k assumption.

Definition 3.3 (Unbounded Attribute-Based Encryption). Let $\mathsf{P} = \{\mathsf{P}_L\}_{L\in\mathcal{L}}$ be a predicate family where $\mathsf{P}_L : \mathcal{X}_L \times \mathcal{Y}_L \to \{0, 1\}$, and $\mathcal{L}$ is a parameter space. [9] An unbounded attribute-based encryption (ABE) scheme for a predicate P consists of the four algorithms:

$\mathsf{Setup}(1^\lambda)$**:** It takes a security parameter 1^λ and outputs a public key mpk and a master secret key msk.
$\mathsf{Enc}(\mathsf{mpk}, (x, L), M)$**:** It takes mpk, an attribute $x \in \mathcal{X}_L$, the parameter L, and a message $M \in \mathcal{M}$ as inputs, and outputs a ciphertext ct. (Note that we let $\mathcal{M}$ be specified in mpk.)
$\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, (y, L))$**:** It takes mpk, msk, an attribute $y \in \mathcal{Y}_L$, and L as inputs, and outputs a secret key sk.

[9] L specifies such as the attribute length in ABE for span program, the user set size in broadcast encryption (namely N in Section 1.1).

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{ct}, \mathsf{sk}, x, y, L)$: It takes $\mathsf{mpk}, \mathsf{ct}, \mathsf{sk}, x, y, L$ as inputs, and outputs a message M' or a symbol $\perp$.

Correctness. An ABE scheme is correct if it satisfies the following condition. For all $\lambda \in \mathbb{N}$, $L \in \mathcal{L}$, $x \in \mathcal{X}_L$, $y \in \mathcal{Y}_L$ such that $\mathsf{P}_L(x, y) = 1$, and $M \in \mathcal{M}$, we have

$$\Pr\left[M = M' : \begin{array}{l} (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, (x, L), M) \\ \mathsf{sk} \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, (y, L)) \\ M' = \mathsf{Dec}(\mathsf{mpk}, \mathsf{ct}, \mathsf{sk}, x, y, L) \end{array}\right] = 1$$

Security. An ABE scheme satisfies adaptive unbounded security if it satisfies the following condition. That is, the advantage of $\mathcal{A}$ defined as follows is negligible in λ for all stateful PPT adversary $\mathcal{A}$:

$$\mathsf{Adv}^{\mathsf{ABE}}_{\mathcal{A}} = \left|\Pr\left[\beta = \beta' : \begin{array}{l} \beta \leftarrow \{0,1\} \\ (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda) \\ ((x, L^*), M_0, M_1) \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{mpk},\mathsf{msk},\cdot)}(\mathsf{mpk}) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, (x, L^*), M_\beta) \\ \beta' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{mpk},\mathsf{msk},\cdot)}(\mathsf{ct}) \end{array}\right] - \frac{1}{2}\right|$$

where $x \in \mathcal{X}_{L^*}$, and all $(y_j, L_j)_{j\in[Q]}$ on which $\mathcal{A}$ queries KeyGen must satisfy the following:

- $y_j \in \mathcal{Y}_{L_j}$;
- $L^* \neq L_j$ or $L^* = L_j \wedge \mathsf{P}_{L_j}(x, y_j) = 0$.

Semi-adaptive unbounded security is similarly defined as adaptive unbounded security except that the KeyGen oracle is not available before $\mathcal{A}$ outputs the challenge $((x, L^*), M_0, M_1)$.[10]

Remark 3.1. Definition 3.3 is slightly different from the standard ABE definition (e.g. [GPSW06,Att14]).Specifically, in the standard ABE definition, Setup takes L and includes it in mpk. Enc and KeyGen take only $x \in \mathcal{X}_L$ and $y \in \mathcal{Y}_L$ as attribute, respectively; we then allow the adversary to specify different L's in the security game. That is, we capture unbounded broadcast encryption and KP-ABE where-in the adversary asks for secret keys corresponding to different values of N (i.e., different bounds on the size of the recipient set, or policies with different input lengths), some of which may be incompatible with the value of N specified in the challenge ciphertext. Our definition is more general than the standard definition, and thus an ABE scheme that satisfies Definition 3.3 can be easily converted to the scheme that satisfies the standard ABE definition. We use our definition to capture ABE schemes where Setup solely depends on 1^λ but not on L.

We define linear conditional disclosure of secrets (linear CDS) [GIKM98] as follows, which is essentially the same as predicate encoding [Wee14] and the special case of pair encoding [Att14].

Definition 3.4 (Linear Conditional Disclosure of Secrets (Linear CDS)). Let $\mathsf{P} = \{\mathsf{P}_L\}_{L\in\mathcal{L}}$ be a family of predicates $\mathsf{P}_L : \mathcal{X}_L \times \mathcal{Y}_L \to \{0,1\}$ parametrized by $L \in \mathcal{L}$, where $\mathcal{L}$ is a parameter space. A linear CDS for P consists of a function $\mathsf{p} : \mathcal{L} \to \mathbb{N}$ and the three deterministic polynomial-time algorithms:

- $\mathsf{EncC}(x, L) \to c_x$. On input $x \in \mathcal{X}_L$ and $L \in \mathcal{L}$, EncC outputs a linear function $c_x : \mathbb{Z}_p^{\mathsf{p}(L)} \to \mathbb{Z}_p^{m_1}$. That is, c_x specifies coefficients $c_{i,j} \in \mathbb{Z}_p$ for $(i, j) \in [\mathsf{p}(L)] \times [m_1]$ such that

$$c_x(\mathbf{w}) = \left\{\sum_{i\in[\mathsf{p}(L)]} c_{i,j} w_i\right\}_{j\in[m_1]}$$

  where $\mathbf{w} = (w_1, \ldots, w_{\mathsf{p}(L)}) \in \mathbb{Z}_p^{\mathsf{p}(L)}$.

[10] Semi-adaptive security is slightly stronger than selective security [CW14, GKW16].

– $\mathsf{EncK}(y, L) \to k_y$. On input $y \in \mathcal{Y}_L$ and L, EncK outputs a linear function $k_y : \mathbb{Z}_p^{\mathsf{p}(L)+1} \to \mathbb{Z}_p^{m_2}$. That is, k_y specifies coefficients $k_{i,j} \in \mathbb{Z}_p$ for $(i, j) \in [0, \mathsf{p}(L)] \times [m_2]$ such that

$$k_y(\alpha, \mathbf{w}) = \left\{ k_{0,j}\alpha + \sum_{i \in [\mathsf{p}(L)]} k_{i,j} w_i \right\}_{j \in [m_2]}$$

where $\alpha \in \mathbb{Z}_p$, $\mathbf{w} \in \mathbb{Z}_p^{\mathsf{p}(L)}$.

– $\mathsf{cRec}(x, y, L) \to d_{x,y}$. On input L, $x \in \mathcal{X}_L$, and $y \in \mathcal{Y}_L$, cRec outputs a linear function $d_{x,y} : \mathbb{Z}_p^{m_1+m_2} \to \mathbb{Z}_p$. That is, $d_{x,y}$ specifies coefficients $d_1, \ldots, d_{m_1}, d'_1, \ldots, d'_{m_2} \in \mathbb{Z}_p$, and we have

$$d_{x,y}(\mathbf{c}, \mathbf{k}) = \sum_{i \in [m_1]} d_i c_i + \sum_{i \in [m_2]} d'_i k_i$$

where $\mathbf{c} = (c_1, \ldots, c_{m_1}) \in \mathbb{Z}_p^{m_1}$, $\mathbf{k} = (k_1, \ldots, k_{m_2}) \in \mathbb{Z}_p^{m_2}$.

Correctness. For $x \in \mathcal{X}_L$ and $y \in \mathcal{Y}_L$, let $c_x \leftarrow \mathsf{EncC}(x, L)$, $k_y \leftarrow \mathsf{EncK}(y, L)$, and $d_{x,y} \leftarrow \mathsf{cRec}(x, y, L)$. If $\mathsf{P}_L(x, y) = 1$, then it symbolically holds that $d_{x,y}(c_x(\mathbf{w}), k_y(\alpha, \mathbf{w})) = \alpha$.

Bounded Security. Let x, y, c_x, k_y be as above. If $\mathsf{P}_L(x, y) = 0$, the following distributions are identical:

$$(c_x(\mathbf{w}), k_y(\alpha, \mathbf{w})) \quad \text{and} \quad (c_x(\mathbf{w}), k_y(0, \mathbf{w}))$$

where $\alpha \leftarrow \mathbb{Z}_p, \mathbf{w} \leftarrow \mathbb{Z}_p^{\mathsf{p}(L)}$.

Unbounded Security. For $L, L' \in \mathcal{L}$, $x \in \mathcal{X}_L$ and $y \in \mathcal{Y}_{L'}$, let $c_x \leftarrow \mathsf{EncC}(x, L)$, $k_y \leftarrow \mathsf{EncK}(y, L')$. If $L \neq L'$ or $L = L' \wedge \mathsf{P}_L(x, y) = 0$, the following distributions are identical:

$$(c_x(\mathbf{w}), k_y(\alpha, \mathbf{w}')) \quad \text{and} \quad (c_x(\mathbf{w}), k_y(0, \mathbf{w}'))$$

where $\alpha \leftarrow \mathbb{Z}_p, \mathbf{w} \leftarrow \mathbb{Z}_p^{\mathsf{p}(L)}$, and $\mathbf{w}'$ consists of the first $\mathsf{p}(L')$ elements of $\mathbf{w}$ if $\mathsf{p}(L) \geq \mathsf{p}(L')$, and otherwise, $\alpha \leftarrow \mathbb{Z}_p, \mathbf{w}' \leftarrow \mathbb{Z}_p^{\mathsf{p}(L')}$, and $\mathbf{w}$ consists of the first $\mathsf{p}(L)$ elements of $\mathbf{w}'$.

3.1 Bilinear Entropy Expansion Lemma

We recall the bilinear entropy expansion lemma [CGKW18] and introduce new variants to make it applicable to our constructions.

Definition 3.5 (Dual Matrices). Let $\ell, \ell_1, \ell_2, \ell_3$ be positive integers such that $\ell = \ell_1 + \ell_2 + \ell_3$. Let $\mathbf{A}_1 \in \mathbb{Z}_p^{\ell \times \ell_1}$, $\mathbf{A}_2 \in \mathbb{Z}_p^{\ell \times \ell_2}$, $\mathbf{A}_3 \in \mathbb{Z}_p^{\ell \times \ell_3}$ be matrices such that $\mathbf{A} = (\mathbf{A}_1 || \mathbf{A}_2 || \mathbf{A}_3) \in \mathbb{Z}_p^{\ell \times \ell}$ is invertible. We define $\mathbf{A}_1^{\parallel} \in \mathbb{Z}_p^{\ell \times \ell_1}$, $\mathbf{A}_2^{\parallel} \in \mathbb{Z}_p^{\ell \times \ell_2}$, $\mathbf{A}_3^{\parallel} \in \mathbb{Z}_p^{\ell \times \ell_3}$ as $(\mathbf{A}_1^{\parallel} || \mathbf{A}_2^{\parallel} || \mathbf{A}_3^{\parallel}) = (\mathbf{A}^\top)^{-1}$. Then, we have

$$\mathbf{A}_j^{\parallel \top} \mathbf{A}_i = \begin{cases} \mathbf{O} & (i \neq j) \\ \mathbf{I}_{\ell_i} & (i = j) \end{cases}$$

Lemma 3.1 (Bilinear Entropy Expansion Lemma [CGKW18, Lemma 8]). *Let $n, \ell_1, \ell_2, \ell_3, \ell_W, \ell$ be positive integers such that $\ell_1, \ell_3, \ell_W \geq k$, and $\ell = \ell_1 + \ell_2 + \ell_3$. Sample $\mathbf{A}_i \leftarrow \mathbb{Z}_p^{\ell \times \ell_i}$ and define $\mathbf{A}_i^{\parallel}$ for $i \in \{1, 2, 3\}$ following Definition 3.5. The original entropy expansion lemma asserts that*

$$\left\{ \begin{array}{l} \mathsf{aux} : \mathbb{G}, [\mathbf{A}_1, \mathbf{U}_1\mathbf{A}_1, \mathbf{U}_2\mathbf{A}_1, \mathbf{V}_2\mathbf{A}_1]_1 \\ \mathsf{ct} : [\mathbf{c}_0^\top]_1, \{[\mathbf{c}_i^\top, \mathbf{V}_2\mathbf{c}_0^\top + (\mathbf{U}_1 + i\mathbf{U}_2)\mathbf{c}_i^\top]_1\}_{i \in [n]} \\ \mathsf{sk} : \{[\mathbf{D}_i, \mathbf{D}_i\mathbf{V}_2, \mathbf{D}_i(\mathbf{U}_1 + i\mathbf{U}_2)]_2\}_{i \in [n]} \end{array} \right\}$$

$$\approx_c \left\{ \begin{array}{l} \mathsf{aux} : \mathbb{G}, [\mathbf{A}_1, \mathbf{U}_1\mathbf{A}_1, \mathbf{U}_2\mathbf{A}_1, \mathbf{V}_2\mathbf{A}_1]_1 \\ \mathsf{ct} : [\mathbf{c}_0^\top]_1, \{[\mathbf{c}_i^\top, (\mathbf{V}_2 + \mathbf{W}'_i)\mathbf{c}_0^\top + (\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}'_i)\mathbf{c}_i^\top]_1\}_{i \in [n]} \\ \mathsf{sk} : \{[\mathbf{D}_i, \mathbf{D}_i(\mathbf{V}_2 + \mathbf{W}'_i), \mathbf{D}_i(\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}'_i)]_2\}_{i \in [n]} \end{array} \right\}$$

where $\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda), \mathbf{U}_1, \mathbf{U}_2, \mathbf{V}_2 \leftarrow \mathbb{Z}_p^{\ell_W \times \ell}, \mathbf{D}_i \leftarrow \mathbb{Z}_p^{\ell_W \times \ell_W}$, *and* $\mathbf{W}'_i, \mathbf{U}'_i \leftarrow \mathsf{RowSpan}(\mathbf{A}_2^{\parallel^\top})^{\ell_1 \times 1}$ *for* $i \in [n]$. *In the LHS,* $\mathbf{c}_0^\top, \mathbf{c}_i \leftarrow \mathsf{ColSpan}(\mathbf{A}_1)$ *while in the RHS,* $\mathbf{c}_0^\top, \mathbf{c}_i^\top \leftarrow \mathsf{ColSpan}(\mathbf{A}_1, \mathbf{A}_2)$. *This lemma is implied by the MDDH assumption.*

For our compiler, we use the following variant of the entropy expansion lemma.

Lemma 3.2 (Our Bilinear Entropy Expansion Lemma). *Let* $n, \ell_1, \ell_2, \ell_3, \ell_W, \ell$ *be positive integers such that* $\ell_1, \ell_2, \ell_3, \ell_W \geq k$, *and* $\ell = \ell_1 + \ell_2 + \ell_3$. *Sample* $\mathbf{A}_i \leftarrow \mathbb{Z}_p^{\ell \times \ell_i}$ *and define* $\mathbf{A}_i^\parallel$ *for* $i \in \{1, 2, 3\}$ *following Definition 3.5. Our entropy expansion lemma asserts that*

$$\left\{\begin{array}{l}\mathsf{aux} : \mathbb{G}, [\mathbf{A}_1, \mathbf{U}_1\mathbf{A}_1, \mathbf{U}_2\mathbf{A}_1, \mathbf{V}_1\mathbf{A}_1, \mathbf{V}_2\mathbf{A}_1]_1 \\ \mathsf{ct} : [\mathbf{c}_0^\top]_1, \{[\mathbf{c}_i^\top, \mathbf{V}_1\mathbf{c}_i^\top, \mathbf{V}_2\mathbf{c}_0^\top + (\mathbf{U}_1 + i\mathbf{U}_2)\mathbf{c}_i^\top]_1\}_{i\in[n]} \\ \mathsf{sk} : [\mathbf{D}_0]_2, \{[\mathbf{D}_i, \mathbf{D}_i\mathbf{V}_2, \mathbf{D}_0\mathbf{V}_1 + \mathbf{D}_i(\mathbf{U}_1 + i\mathbf{U}_2)]_2\}_{i\in[n]}\end{array}\right\}$$
$$\approx_c \left\{\begin{array}{l}\mathsf{aux} : \mathbb{G}, [\mathbf{A}_1, \mathbf{U}_1\mathbf{A}_1, \mathbf{U}_2\mathbf{A}_1, \mathbf{V}_1\mathbf{A}_1, \mathbf{V}_2\mathbf{A}_1]_1 \\ \mathsf{ct} : [\mathbf{c}_0^\top]_1, \{[\mathbf{c}_i^\top, (\mathbf{V}_1 + \mathbf{W}_i)\mathbf{c}_i^\top, (\mathbf{V}_2 + \mathbf{W}'_i)\mathbf{c}_0^\top + (\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}'_i)\mathbf{c}_i^\top]_1\}_{i\in[n]} \\ \mathsf{sk} : [\mathbf{D}_0]_2, \{[\mathbf{D}_i, \mathbf{D}_i(\mathbf{V}_2 + \mathbf{W}'_i), \mathbf{D}_0(\mathbf{V}_1 + \mathbf{W}_i) + \mathbf{D}_i(\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}'_i)]_2\}_{i\in[n]}\end{array}\right\} \tag{6}$$

where $\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda), \mathbf{U}_1, \mathbf{U}_2, \mathbf{V}_1, \mathbf{V}_2 \leftarrow \mathbb{Z}_p^{\ell_W \times \ell}, \mathbf{D}_0, \mathbf{D}_i \leftarrow \mathbb{Z}_p^{\ell_W \times \ell_W}$, *and* $\mathbf{W}_i, \mathbf{W}'_i, \mathbf{U}'_i \leftarrow \mathsf{RowSpan}(\mathbf{A}_2^{\parallel^\top})^{\ell_1 \times 1}$ *for* $i \in [n]$. *In the LHS,* $\mathbf{c}_0^\top, \mathbf{c}_i \leftarrow \mathsf{ColSpan}(\mathbf{A}_1)$ *while in the RHS,* $\mathbf{c}_0^\top, \mathbf{c}_i^\top \leftarrow \mathsf{ColSpan}(\mathbf{A}_1, \mathbf{A}_2)$. *The lemma is implied by the MDDH assumption.*

Proof. We prove Lemma 3.2 as follows:

$$\begin{aligned}&[\text{The LHS of Eq. (6)}]\\ \approx_c &\left\{\begin{array}{l}\mathsf{aux} : \mathbb{G}, [\mathbf{A}_1, \mathbf{U}_1\mathbf{A}_1, \mathbf{U}_2\mathbf{A}_1, \mathbf{V}_1\mathbf{A}_1, \mathbf{V}_2\mathbf{A}_1]_1 \\ \mathsf{ct} : [\mathbf{c}_0^\top]_1, \{[\mathbf{c}_i^\top, \mathbf{V}_1\mathbf{c}_i^\top, (\mathbf{V}_2 + \mathbf{W}'_i)\mathbf{c}_0^\top + (\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}'_i)\mathbf{c}_i^\top]_1\}_{i\in[n]} \\ \mathsf{sk} : [\mathbf{D}_0]_2, \{[\mathbf{D}_i, \mathbf{D}_i(\mathbf{V}_2 + \mathbf{W}'_i), \mathbf{D}_0\mathbf{V}_1 + \mathbf{D}_i(\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}'_i)]_2\}_{i\in[n]}\end{array}\right\}\\ \approx_c &\left\{\begin{array}{l}\mathsf{aux} : \mathbb{G}, [\mathbf{A}_1, \mathbf{U}_1\mathbf{A}_1, \mathbf{U}_2\mathbf{A}_1, \mathbf{V}_1\mathbf{A}_1, \mathbf{V}_2\mathbf{A}_1]_1 \\ \mathsf{ct} : [\mathbf{c}_0^\top]_1, \{[\mathbf{c}_i^\top, (\mathbf{V}_1 + \mathbf{W}_i)\mathbf{c}_i^\top, (\mathbf{V}_2 + \mathbf{W}'_i)\mathbf{c}_0^\top + (\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}'_i)\mathbf{c}_i^\top]_1\}_{i\in[n]} \\ \mathsf{sk} : [\mathbf{D}_0]_2, \{[\mathbf{D}_i, \mathbf{D}_i(\mathbf{V}_2 + \mathbf{W}'_i), \mathbf{D}_0(\mathbf{V}_1 + \mathbf{W}_i) + \mathbf{D}_i(\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}'_i)]_2\}_{i\in[n]}\end{array}\right\}\\ =&[\text{The RHS of Eq. (6)}]\end{aligned} \tag{7}$$

where $\mathbf{c}_0^\top, \mathbf{c}_i^\top \leftarrow \mathsf{ColSpan}(\mathbf{A}_1, \mathbf{A}_2)$ in the second and the third expressions. The first indistinguishability directly follows from the original entropy expansion lemma (Lemma 3.1) by letting the reduction sample $\mathbf{V}_1, \mathbf{D}_0$. To prove the second indistinguishability, we consider the following programming:

$$\mathbf{V}_1 = \widehat{\mathbf{V}}_1 + \mathbf{T}\mathbf{A}_2^{\parallel^\top}, \quad \mathbf{U}'_i = \widehat{\mathbf{U}}'_i - \mathbf{M}_i^{-1}\mathbf{T}\mathbf{A}_2^{\parallel^\top}$$

where $\widehat{\mathbf{V}}_1 \leftarrow \mathbb{Z}_p^{\ell_W \times \ell}$, $\mathbf{T} \leftarrow \mathbb{Z}_p^{\ell_W \times \ell_2}$, $\widehat{\mathbf{U}}'_i \leftarrow \mathsf{RowSpan}(\mathbf{A}_2^{\parallel^\top})^{\ell_1 \times 1}$, and $\mathbf{M} = \mathbf{D}_0^{-1}\mathbf{D}_i$. Then, we have

[The second line of Eq.(7)]

$$\equiv \left\{ \begin{array}{l} \mathsf{aux} : \mathbb{G}, [\mathbf{A}_1, \mathbf{U}_1\mathbf{A}_1, \mathbf{U}_2\mathbf{A}_1, \widehat{\mathbf{V}}_1\mathbf{A}_1, \mathbf{V}_2\mathbf{A}_1]_1 \\ \mathsf{ct} : [\mathbf{c}_0^\top]_1, \{[\mathbf{c}_i^\top, (\widehat{\mathbf{V}}_1 + \mathbf{T}\mathbf{A}_2^{\parallel^\top})\mathbf{c}_i^\top, (\mathbf{V}_2 + \mathbf{W}'_i)\mathbf{c}_0^\top + (\mathbf{U}_1 + i\mathbf{U}_2 + \widehat{\mathbf{U}}'_i - \mathbf{M}_i^{-1}\mathbf{T}\mathbf{A}_2^{\parallel^\top})\mathbf{c}_i^\top]_1\}_{i\in[n]} \\ \mathsf{sk} : [\mathbf{D}_0]_2, \{[\mathbf{D}_i, \mathbf{D}_i(\mathbf{V}_2 + \mathbf{W}'_i), \mathbf{D}_0\widehat{\mathbf{V}}_1 + \mathbf{D}_i(\mathbf{U}_1 + i\mathbf{U}_2 + \widehat{\mathbf{U}}'_i)]_2\}_{i\in[n]} \end{array} \right\}$$

$$\approx_c \left\{ \begin{array}{l} \mathsf{aux} : \mathbb{G}, [\mathbf{A}_1, \mathbf{U}_1\mathbf{A}_1, \mathbf{U}_2\mathbf{A}_1, \widehat{\mathbf{V}}_1\mathbf{A}_1, \mathbf{V}_2\mathbf{A}_1]_1 \\ \mathsf{ct} : [\mathbf{c}_0^\top]_1, \{[\mathbf{c}_i^\top, (\widehat{\mathbf{V}}_1 + \mathbf{T}_i\mathbf{A}_2^{\parallel^\top})\mathbf{c}_i^\top, (\mathbf{V}_2 + \mathbf{W}'_i)\mathbf{c}_0^\top + (\mathbf{U}_1 + i\mathbf{U}_2 + \widehat{\mathbf{U}}'_i - \mathbf{M}_i^{-1}\mathbf{T}_i\mathbf{A}_2^{\parallel^\top})\mathbf{c}_i^\top]_1\}_{i\in[n]} \\ \mathsf{sk} : [\mathbf{D}_0]_2, \{[\mathbf{D}_i, \mathbf{D}_i(\mathbf{V}_2 + \mathbf{W}'_i), \mathbf{D}_0\widehat{\mathbf{V}}_1 + \mathbf{D}_i(\mathbf{U}_1 + i\mathbf{U}_2 + \widehat{\mathbf{U}}'_i)]_2\}_{i\in[n]} \end{array} \right\}$$

$$\equiv \left\{ \begin{array}{l} \mathsf{aux} : \mathbb{G}, [\mathbf{A}_1, \mathbf{U}_1\mathbf{A}_1, \mathbf{U}_2\mathbf{A}_1, \widehat{\mathbf{V}}_1\mathbf{A}_1, \mathbf{V}_2\mathbf{A}_1]_1 \\ \mathsf{ct} : [\mathbf{c}_0^\top]_1, \{[\mathbf{c}_i^\top, (\widehat{\mathbf{V}}_1 + \mathbf{T}_i\mathbf{A}_2^{\parallel^\top})\mathbf{c}_i^\top, (\mathbf{V}_2 + \mathbf{W}'_i)\mathbf{c}_0^\top + (\mathbf{U}_1 + i\mathbf{U}_2 + \widetilde{\mathbf{U}}'_i z)\mathbf{c}_i^\top]_1\}_{i\in[n]} \\ \mathsf{sk} : [\mathbf{D}_0]_2, \{[\mathbf{D}_i, \mathbf{D}_i(\mathbf{V}_2 + \mathbf{W}'_i), \mathbf{D}_0(\mathbf{V}_1 + \mathbf{T}_i\mathbf{A}_2^{\parallel^\top}) + \mathbf{D}_i(\mathbf{U}_1 + i\mathbf{U}_2 + \widetilde{\mathbf{U}}'_i)]_2\}_{i\in[n]} \end{array} \right\}$$

≡[The third line of Eq.(7)]

where $\mathbf{T}_i \leftarrow \mathbb{Z}_p^{\ell_W \times \ell_2}$ and $\widetilde{\mathbf{U}}'_i \leftarrow \mathsf{RowSpan}(\mathbf{A}_2^{\parallel^\top})^{\ell_1 \times 1}$. By definition, the first and the third equivalences are trivial. The second one directly follows by programming $\widehat{\mathbf{U}}'_i = \widetilde{\mathbf{U}}'_i + \mathbf{M}_i^{-1}\mathbf{T}_i\mathbf{A}_2^{\parallel^\top}$. The computational indistinguishability above can be shown under MDDH as follows. Recall that we can write $\mathbf{c}_i^\top = [\mathbf{A}_1||\mathbf{A}_2]\begin{pmatrix}\mathbf{s}_{1,i}^\top \\ \mathbf{s}_{2,i}^\top\end{pmatrix}$. Then, we have

$$\begin{aligned} \{[\mathbf{T}\mathbf{A}_2^{\parallel^\top}\mathbf{c}_i^\top, \mathbf{c}_i]_1\}_{i\in[n]} &= \{[\mathbf{T}\mathbf{s}_{2,i}^\top, \mathbf{c}_i]_1\}_{i\in[n]} \approx_c \{[\mathbf{r}_i^\top, \mathbf{c}_i]_1\}_{i\in[n]} \\ \equiv \{[\mathbf{r}_i^\top + \widehat{\mathbf{T}}_i\mathbf{A}_2^{\parallel^\top}\mathbf{c}_i^\top, \mathbf{c}_i]_1\}_{i\in[n]} &\approx_c \{[\underbrace{(\mathbf{T} + \widehat{\mathbf{T}}_i)}_{\mathbf{T}_i}\mathbf{A}_2^{\parallel^\top}\mathbf{c}_i^\top, \mathbf{c}_i]_1\}_{i\in[n]} \end{aligned}$$

where $\mathbf{r}_i \leftarrow \mathbb{Z}_p^{\ell_W}$ and $\widehat{\mathbf{T}}_i \leftarrow \mathbb{Z}_p^{\ell_W \times \ell_2}$. The first and second computational indistinguishability directly follows from the MDDH assumption, which says that $\{[\mathbf{T}\mathbf{s}_{2,i}^\top, \mathbf{s}_{2,i}]_1\}_{i\in[n]} \approx_c \{[\mathbf{r}_i^\top, \mathbf{s}_{2,i}]_1\}_{i\in[n]}$ if $\ell_2 \geq k$. This concludes the proof. □

In contrast to [CGKW18, Lemma 8] (i.e., Lemma 3.1), we need a Q-fold version of Lemma 3.2 for our ABE constructions. Intuitively, when we plug the lemma into the dual system proof, the reduction needs to simulate Q secret keys from sk in the instance of the entropy expansion lemma. In the case of [CGKW18, Lemma 8], generating Q sets of random vectors $\{\mathbf{d}_{0,j}, \mathbf{d}_{i,j}\}_{j\in[Q]}$ from $\mathbf{D}_0, \mathbf{D}_i$ is easy by multiplying random vectors $\mathbf{r}_{0,j}, \mathbf{r}_{i,j}$ from the left, respectively, for $j \in [Q]$. In our case, however, this does not work due to the term $[\mathbf{D}_0(\mathbf{V}_1 + \mathbf{W}_i) + \mathbf{D}_i(\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}'_i)]_2$ in sk. This is why we formally define and prove the Q-fold version of Lemma 3.2.

Lemma 3.3 (Our Bilinear Entropy Expansion Lemma (Q-fold)). *Let $n, \ell_1, \ell_2, \ell_3, \ell_W, \ell, Q$ be positive integers such that $\ell_1, \ell_2, \ell_3, \ell_W \geq k$, and $\ell = \ell_1 + \ell_2 + \ell_3$. Sample $\mathbf{A}_i \leftarrow \mathbb{Z}_p^{\ell \times \ell_i}$ and define $\mathbf{A}_i^{\parallel}$ for $i \in \{1, 2, 3\}$ following Definition 3.5. The Q-fold entropy expansion lemma asserts that*

$$\begin{aligned} &\left\{ \begin{array}{l} \mathsf{aux} : \mathbb{G}, [\mathbf{A}_1, \mathbf{U}_1\mathbf{A}_1, \mathbf{U}_2\mathbf{A}_1, \mathbf{V}_1\mathbf{A}_1, \mathbf{V}_2\mathbf{A}_1]_1 \\ \mathsf{ct} : [\mathbf{c}_0^\top]_1, \{[\mathbf{c}_i^\top, \mathbf{V}_1\mathbf{c}_i^\top, \mathbf{V}_2\mathbf{c}_0^\top + (\mathbf{U}_1 + i\mathbf{U}_2)\mathbf{c}_i^\top]_2\}_{i\in[n]} \\ \mathsf{sk} : [\mathbf{D}_0]_2, \{[\mathbf{D}_i, \mathbf{D}_i\mathbf{V}_2, \mathbf{D}_0\mathbf{V}_1 + \mathbf{D}_i(\mathbf{U}_1 + i\mathbf{U}_2)]_2\}_{i\in[n]} \end{array} \right\} \\ \approx_c &\left\{ \begin{array}{l} \mathsf{aux} : \mathbb{G}, [\mathbf{A}_1, \mathbf{U}_1\mathbf{A}_1, \mathbf{U}_2\mathbf{A}_1, \mathbf{V}_1\mathbf{A}_1, \mathbf{V}_2\mathbf{A}_1]_1 \\ \mathsf{ct} : [\mathbf{c}_0^\top]_1, \{[\mathbf{c}_i^\top, (\mathbf{V}_1 + \mathbf{W}_i)\mathbf{c}_i^\top, (\mathbf{V}_2 + \mathbf{W}'_i)\mathbf{c}_0^\top + (\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}'_i)\mathbf{c}_i^\top]_1\}_{i\in[n]} \\ \mathsf{sk} : [\mathbf{D}_0]_2, \{[\mathbf{D}_i, \mathbf{D}_i(\mathbf{V}_2 + \mathbf{W}'_i), \mathbf{D}_0(\mathbf{V}_1 + \mathbf{W}_i) + \mathbf{D}_i(\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}'_i)]_2\}_{i\in[n]} \end{array} \right\} \end{aligned} \tag{8}$$

where $\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda), \mathbf{U}_1, \mathbf{U}_2, \mathbf{V}_1, \mathbf{V}_2 \leftarrow \mathbb{Z}_p^{\ell_W \times \ell}, \mathbf{D}_0, \mathbf{D}_i \leftarrow \mathbb{Z}_p^{Q \times \ell_W}$, *and* $\mathbf{W}_i, \mathbf{W}'_i, \mathbf{U}'_i \leftarrow \mathsf{RowSpan}(\mathbf{A}_2^{\parallel \top})^{\ell_1 \times 1}$ *for* $i \in [n]$. *In the LHS,* $\mathbf{c}_0^\top, \mathbf{c}_i \leftarrow \mathsf{ColSpan}(\mathbf{A}_1)$ *while in the RHS,* $\mathbf{c}_0^\top, \mathbf{c}_i^\top \leftarrow \mathsf{ColSpan}(\mathbf{A}_1, \mathbf{A}_2)$. *The lemma is implied by the MDDH assumption.*

Proof. We can prove Lemma 3.3 as follows:

$$
\begin{aligned}
&[\text{The LHS of Eq.(8)}] \\
\approx_c &\left\{\begin{array}{l} \mathsf{aux} : \mathbb{G}, [\mathbf{A}_1, \mathbf{U}_1\mathbf{A}_1, \mathbf{U}_2\mathbf{A}_1, \mathbf{V}_1\mathbf{A}_1, \mathbf{V}_2\mathbf{A}_1]_1 \\ \mathsf{ct} : [\mathbf{c}_0^\top]_1, \{[\mathbf{c}_i^\top, \mathbf{V}_1\mathbf{c}_i^\top, \mathbf{V}_2\mathbf{c}_0^\top + (\mathbf{U}_1 + i\mathbf{U}_2)\mathbf{c}_i^\top]_2\}_{i\in[n]} \\ \mathsf{sk} : [\mathbf{R}\widehat{\mathbf{D}}_0]_2, \{[\mathbf{R}\widehat{\mathbf{D}}_i, \mathbf{R}\widehat{\mathbf{D}}_i\mathbf{V}_2, \mathbf{R}\widehat{\mathbf{D}}_0\mathbf{V}_1 + \mathbf{R}\widehat{\mathbf{D}}_i(\mathbf{U}_1 + i\mathbf{U}_2)]_2\}_{i\in[n]} \end{array}\right\} \\
\approx_c &\left\{\begin{array}{l} \mathsf{aux} : \mathbb{G}, [\mathbf{A}_1, \mathbf{U}_1\mathbf{A}_1, \mathbf{U}_2\mathbf{A}_1, \mathbf{V}_1\mathbf{A}_1, \mathbf{V}_2\mathbf{A}_1]_1 \\ \mathsf{ct} : [\mathbf{c}_0^\top]_1, \{[\mathbf{c}_i^\top, (\mathbf{V}_1 + \mathbf{W}_i)\mathbf{c}_i^\top, (\mathbf{V}_2 + \mathbf{W}'_i)\mathbf{c}_0^\top + (\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}'_i)\mathbf{c}_i^\top]_1\}_{i\in[n]} \\ \mathsf{sk} : [\mathbf{R}\widehat{\mathbf{D}}_0]_2, \{[\mathbf{R}\widehat{\mathbf{D}}_i, \mathbf{R}\widehat{\mathbf{D}}_i(\mathbf{V}_2 + \mathbf{W}'_i), \mathbf{R}\widehat{\mathbf{D}}_0(\mathbf{V}_1 + \mathbf{W}_i) + \mathbf{R}\widehat{\mathbf{D}}_i(\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}'_i)]_2\}_{i\in[n]} \end{array}\right\} \\
\approx_c &[\text{The RHS of Eq.(8)}]
\end{aligned}
$$

where $\mathbf{R} \leftarrow \mathbb{Z}_p^{Q\times \ell_W}$, $\widehat{\mathbf{D}}_0, \widehat{\mathbf{D}}_i \leftarrow \mathbb{Z}_p^{\ell_W \times \ell_W}$, and in the second expression, $\mathbf{c}_0^\top, \mathbf{c}_i \leftarrow \mathsf{ColSpan}(\mathbf{A}_1)$ while in the third expression, $\mathbf{c}_0^\top, \mathbf{c}_i^\top \leftarrow \mathsf{ColSpan}(\mathbf{A}_1, \mathbf{A}_2)$. The first and the third indistinguishability follows from MDDH, which tells us $[\{\mathbf{D}_i\}_{i\in[0,n]}]_2 \approx_c [\{\mathbf{R}\widehat{\mathbf{D}}_i\}_{i\in[0,n]}]_2$ if $\ell_W \geq k$. The second indistinguishability directly follows from Lemma 3.2. □

4 Unbounded ABE from Linear CDS

In this section, we present our compiler from a linear CDS to an unbounded ABE scheme. Our compiler is generic in a sense that it works with any linear CDS. Our ABE scheme is secure if the underlying linear CDS satisfies unbounded security.

4.1 Construction

Let $\mathsf{P} = \{\mathsf{P}_L\}_{L\in\mathcal{L}}$ be a predicate family and $\Gamma = (\mathsf{p}, \mathsf{EncC}, \mathsf{EncK}, \mathsf{cRec})$ be a linear CDS for $\mathsf{P}_L : \mathcal{X}_L \times \mathcal{Y}_L \to \{0,1\}$ where $L \in \mathcal{L}$.

$\mathsf{Setup}(1^\lambda)$**:** It outputs mpk and msk as follows.

$$
\begin{aligned}
&\mathbf{A}_1 \leftarrow \mathbb{Z}_p^{3k\times k},\ \mathbf{B} \leftarrow \mathbb{Z}_p^{k\times(k+1)},\ \mathbf{U}_1, \mathbf{U}_2, \mathbf{V}_1, \mathbf{V}_2 \leftarrow \mathbb{Z}_p^{(k+1)\times 3k},\ \mathbf{k} \leftarrow \mathbb{Z}_p^{3k} \\
&\mathsf{mpk} = ([\mathbf{A}_1, \mathbf{U}_1\mathbf{A}_1, \mathbf{U}_2\mathbf{A}_1, \mathbf{V}_1\mathbf{A}_1, \mathbf{V}_2\mathbf{A}_1]_1, [\mathbf{k}\mathbf{A}_1]_T) \\
&\mathsf{msk} = (\mathbf{k}, \mathbf{B}, \mathbf{U}_1, \mathbf{U}_2, \mathbf{V}_1, \mathbf{V}_2)
\end{aligned}
$$

$\mathsf{Enc}(\mathsf{mpk}, (x, L), M)$**:** It takes mpk, $x \in \mathcal{X}_L$, $L \in \mathcal{L}$, and $M \in G_T$ as inputs, and outputs ct by computing as follows.

$$
\begin{aligned}
&n = \mathsf{p}(L),\ c \leftarrow \mathsf{EncC}(x, L),\ \mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_n \leftarrow \mathbb{Z}_p^k \\
&\mathsf{ct} = \left(\begin{array}{l} \underbrace{c([\mathbf{V}_1\mathbf{A}_1\mathbf{s}_1^\top, \ldots, \mathbf{V}_1\mathbf{A}_1\mathbf{s}_n^\top]_1)}_{\mathsf{ct}_1} \\ \underbrace{[\mathbf{A}_1\mathbf{s}_0^\top}_{\mathsf{ct}_2}, \{\underbrace{\mathbf{A}_1\mathbf{s}_i^\top}_{\mathsf{ct}_{3,i}}, \underbrace{\mathbf{V}_2\mathbf{A}_1\mathbf{s}_0^\top + (\mathbf{U}_1 + i\mathbf{U}_2)\mathbf{A}_1\mathbf{s}_i^\top}_{\mathsf{ct}_{4,i}}\}_{i\in[n]}]_1,\ \underbrace{[\mathbf{k}\mathbf{A}\mathbf{s}_0^\top]_T \cdot M}_{\mathsf{ct}_5} \end{array}\right)
\end{aligned}
\tag{9}
$$

$\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, (y, L))$: It takes mpk, msk, $y \in \mathcal{Y}_L$, and L as inputs, and outputs sk by computing as follows.

$$n = \mathsf{p}(L),\ k \leftarrow \mathsf{EncK}(y, L),\ \mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_n \leftarrow \mathbb{Z}_p^k$$
$$\mathsf{sk} = (\underbrace{k([\mathbf{k}, \mathbf{r}_1\mathbf{B}\mathbf{V}_2, \ldots, \mathbf{r}_n\mathbf{B}\mathbf{V}_2]_2)}_{\mathsf{sk}_1}, \underbrace{[\mathbf{r}_0\mathbf{B}}_{\mathsf{sk}_2}, \{\underbrace{\mathbf{r}_i\mathbf{B}}_{\mathsf{sk}_{3,i}}, \underbrace{\mathbf{r}_0\mathbf{B}\mathbf{V}_1 + \mathbf{r}_i\mathbf{B}(\mathbf{U}_1 + i\mathbf{U}_2)}_{\mathsf{sk}_{4,i}}\}_{i\in[n]}]_2) \tag{10}$$

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{ct}, \mathsf{sk}, x, y, L)$: It takes mpk, ct, sk, x, y, L such that $x \in \mathcal{X}_L, y \in \mathcal{Y}_L$, and $\mathsf{P}_L(x, y) = 1$. It computes $c \leftarrow \mathsf{EncC}(x, L)$, $k \leftarrow \mathsf{EncK}(y, L)$, $d \leftarrow \mathsf{cRec}(x, y, L)$ and

$$z_1 = e(\mathsf{ct}_1, \mathsf{sk}_2) - c(e(\mathsf{ct}_{3,1}, \mathsf{sk}_{4,1}), \ldots, e(\mathsf{ct}_{3,n}, \mathsf{sk}_{4,n}))$$
$$z_2 = e(\mathsf{ct}_2, \mathsf{sk}_1) - k([0]_T, e(\mathsf{ct}_{4,1}, \mathsf{sk}_{3,1}), \ldots, e(\mathsf{ct}_{4,n}, \mathsf{sk}_{3,n}))$$

where the operator $-$ denotes element-wise division of G_T elements. Then outputs $\mathsf{ct}_5/d(z_1, z_2)$.

Correctness. Recall $e([\mathbf{M}]_1, [\mathbf{N}]_2) = [\mathbf{N}\mathbf{M}]_T$. In decryption, we have

$$\begin{aligned} e(\mathsf{ct}_1, \mathsf{sk}_2) &= [\mathbf{r}_0\mathbf{B} \cdot c(\mathbf{V}_1\mathbf{A}_1\mathbf{s}_1^\top, \ldots, \mathbf{V}_1\mathbf{A}_1\mathbf{s}_n^\top)]_T \\ &= [c(\mathbf{r}_0\mathbf{B}\mathbf{V}_1\mathbf{A}_1\mathbf{s}_1^\top, \ldots, \mathbf{r}_0\mathbf{B}\mathbf{V}_1\mathbf{A}_1\mathbf{s}_n^\top)]_T \\ e(\mathsf{ct}_{3,i}, \mathsf{sk}_{4,i}) &= [(\mathbf{r}_0\mathbf{B}\mathbf{V}_1 + \mathbf{r}_i\mathbf{B}(\mathbf{U}_1 + i\mathbf{U}_2))\mathbf{A}_1\mathbf{s}_i^\top]_T \end{aligned}$$

Hence, we have

$$\begin{aligned} z_1 &= \begin{bmatrix} c(\mathbf{r}_0\mathbf{B}\mathbf{V}_1\mathbf{A}_1\mathbf{s}_1^\top, \ldots, \mathbf{r}_0\mathbf{B}\mathbf{V}_1\mathbf{A}_1\mathbf{s}_n^\top) \\ - c((\mathbf{r}_0\mathbf{B}\mathbf{V}_1 + \mathbf{r}_1\mathbf{B}(\mathbf{U}_1 + \mathbf{U}_2))\mathbf{A}_1\mathbf{s}_1^\top, \ldots, (\mathbf{r}_0\mathbf{B}\mathbf{V}_1 + \mathbf{r}_n\mathbf{B}(\mathbf{U}_1 + n\mathbf{U}_2))\mathbf{A}_1\mathbf{s}_n^\top) \end{bmatrix}_T \\ &= [c(-\mathbf{r}_1\mathbf{B}(\mathbf{U}_1 + \mathbf{U}_2)\mathbf{A}_1\mathbf{s}_1^\top, \ldots, -\mathbf{r}_n\mathbf{B}(\mathbf{U}_1 + n\mathbf{U}_2)\mathbf{A}_1\mathbf{s}_n^\top)]_T \end{aligned}$$

which follows from the fact the c is a linear function. Similarly, it also holds that

$$z_2 = [k(\mathbf{k}\mathbf{A}_1\mathbf{s}_0^\top, -\mathbf{r}_1\mathbf{B}(\mathbf{U}_1 + \mathbf{U}_2)\mathbf{A}_1\mathbf{s}_1^\top, \ldots, -\mathbf{r}_n\mathbf{B}(\mathbf{U}_1 + n\mathbf{U}_2)\mathbf{A}_1\mathbf{s}_n^\top)]_T$$

Hence, thanks to the correctness of linear CDS Γ, we have $d(z_1, z_2) = [\mathbf{k}\mathbf{A}_1\mathbf{s}_0^\top]_T$.

Parameter Sizes. In the above construction, the sizes of $\mathsf{mpk}, \mathsf{ct}, \mathsf{sk}$ are as follows:

$$|\mathsf{mpk}| = O(1), \quad |\mathsf{ct}| = O(n + m_1), \quad |\mathsf{sk}| = O(n + m_2)$$

where $n = \mathsf{p}(L)$, m_1, m_2 are the number of $\mathbb{Z}_p$ elements in $c(\mathbf{w}), k(\alpha, \mathbf{w})$, respectively. The above sizes hide $\mathsf{poly}(\lambda)$ factors.

4.2 Security

Theorem 4.1. *Suppose the MDDH assumption holds and the linear CDS Γ satisfies unbounded security. Then, our scheme in Section 4.1 satisfies adaptive unbounded security as per Definition 3.3.*

Proof. We define various distributions of ciphertext and secret key. As global parameters, we sample $\mathbf{A}_2, \mathbf{A}_3 \leftarrow \mathbb{Z}_p^{3k\times k}$, define $\mathbf{A}_1^\parallel, \mathbf{A}_2^\parallel, \mathbf{A}_3^\parallel$ following Definition 3.5, and also sample $\mathbf{W}_i, \mathbf{W}_i', \mathbf{U}_i' \leftarrow \mathsf{RowSpan}(\mathbf{A}_2^{\parallel\top})^{(k+1)\times 1}$ for $i \in [n']$ where n' is sufficiently large. Let $\mathbf{a}_{2,1}^{\parallel\top}$ be the first column of $\mathbf{A}_2^\parallel$. For a ciphertext,

- Normal: As defined in Eq.(9).

– Expanded normal (E-normal): It is generates as

$$n = \mathsf{p}(L),\ c \leftarrow \mathsf{EncC}(x, L),\ \mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_n \leftarrow \mathsf{ColSpan}(\mathbf{A}_1, \mathbf{A}_2)$$
$$\mathsf{ct} = \begin{pmatrix} c([(\mathbf{V}_1 + \mathbf{W}_1)\mathbf{c}_1^\top, \ldots, (\mathbf{V}_1 + \mathbf{W}_n)\mathbf{c}_n^\top]_1) \\ [\mathbf{c}_0^\top, \{\mathbf{c}_i^\top, (\mathbf{V}_2 + \mathbf{W}_i')\mathbf{c}_0^\top + (\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}_i')\mathbf{c}_i^\top\}_{i\in[n]}]_1,\ [\mathbf{k}\mathbf{c}_0^\top]_T \cdot M \end{pmatrix} \tag{11}$$

For a secret key,

– Normal: As defined in Eq.(10).
– E-normal: It is generated as

$$n = \mathsf{p}(L),\ k \leftarrow \mathsf{EncK}(y, L),\ \mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_n \leftarrow \mathbb{Z}_p^k$$
$$\mathsf{sk} = \begin{pmatrix} k([\mathbf{k}, \mathbf{r}_1\mathbf{B}(\mathbf{V}_2 + \mathbf{W}_1'), \ldots, \mathbf{r}_n\mathbf{B}(\mathbf{V}_2 + \mathbf{W}_n')]_2) \\ [\mathbf{r}_0\mathbf{B}, \{\mathbf{r}_i\mathbf{B}, \mathbf{r}_0\mathbf{B}(\mathbf{V}_1 + \mathbf{W}_i) + \mathbf{r}_i\mathbf{B}(\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}_i')\}_{i\in[n]}]_2 \end{pmatrix} \tag{12}$$

– Pseudo normal (P-normal): It is generated as

$$n = \mathsf{p}(L),\ k \leftarrow \mathsf{EncK}(y, L),\ \mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_n \leftarrow \mathbb{Z}_p^{k+1}$$
$$\mathsf{sk} = \begin{pmatrix} k([\mathbf{k}, \mathbf{d}_1(\mathbf{V}_2 + \mathbf{W}_1'), \ldots, \mathbf{d}_n(\mathbf{V}_2 + \mathbf{W}_n')]_2) \\ [\mathbf{d}_0, \{\mathbf{d}_i, \mathbf{d}_0(\mathbf{V}_1 + \mathbf{W}_i) + \mathbf{d}_i(\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}_i')\}_{i\in[n]}]_2 \end{pmatrix}$$

– Pseudo semi-functional (P-SF): It is generated as

$$n = \mathsf{p}(L),\ k \leftarrow \mathsf{EncK}(y, L),\ \mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_n \leftarrow \mathbb{Z}_p^{k+1},\ \mathbf{z} \leftarrow \mathsf{RowSpan}(\mathbf{a}_{2,1}^\parallel)$$
$$\mathsf{sk} = \begin{pmatrix} k([\mathbf{k} + \mathbf{z}, \mathbf{d}_1(\mathbf{V}_2 + \mathbf{W}_1'), \ldots, \mathbf{d}_n(\mathbf{V}_2 + \mathbf{W}_n')]_2) \\ [\mathbf{d}_0, \{\mathbf{d}_i, \mathbf{d}_0(\mathbf{V}_1 + \mathbf{W}_i) + \mathbf{d}_i(\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}_i')\}_{i\in[n]}]_2 \end{pmatrix}$$

– Semi-functional (SF): It is generated as

$$n = \mathsf{p}(L),\ k \leftarrow \mathsf{EncK}(y, L),\ \mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_n \leftarrow \mathbb{Z}_p^k,\ \mathbf{z} \leftarrow \mathsf{RowSpan}(\mathbf{a}_{2,1}^\parallel)$$
$$\mathsf{sk} = \begin{pmatrix} k([\mathbf{k} + \mathbf{z}, \mathbf{r}_1\mathbf{B}(\mathbf{V}_2 + \mathbf{W}_1'), \ldots, \mathbf{r}_n\mathbf{B}(\mathbf{V}_2 + \mathbf{W}_n')]_2) \\ [\mathbf{r}_0\mathbf{B}, \{\mathbf{r}_i\mathbf{B}, \mathbf{r}_0\mathbf{B}(\mathbf{V}_1 + \mathbf{W}_i) + \mathbf{r}_i\mathbf{B}(\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}_i')\}_{i\in[n]}]_2 \end{pmatrix} \tag{13}$$

Note that for every P-SF and SF key, we sample a fresh $\mathbf{z}$.

We prove the theorem via a series of hybrids. Let Q be the maximum number of secret key queries by the adversary.

– H_0: The original security game.
– H_1: Same as H_0 except that the challenge ciphertext and all secret keys given to the adversary are E-normal.
– $\mathsf{H}_{2,j,1}$ for $j \in [Q]$: Same as H_1 except that the first $j-1$ keys are SF, the j-th key is P-normal (the rest of the keys are E-normal).
– $\mathsf{H}_{2,j,2}$ for $j \in [Q]$: Same as $\mathsf{H}_{2,j,1}$ except that the j-th key is P-SF.
– $\mathsf{H}_{2,j,3}$ for $j \in [Q]$: Same as $\mathsf{H}_{2,j,2}$ except that the j-th key is SF.
– H_3: Same as $\mathsf{H}_{2,Q,3}$ except that ct_5 in the challenge ciphertext is randomly chosen from G_T.

The hybrids proceed as follows:

$$\mathsf{H}_0 \to \mathsf{H}_1 \to \mathsf{H}_{2,1,1} \to \mathsf{H}_{2,1,2} \to \mathsf{H}_{2,1,3} \to \mathsf{H}_{2,2,1} \to \cdots \to \mathsf{H}_{2,Q,3} \to \mathsf{H}_3$$

We prove the indistinguishability of each hybrid transition in Lemma 4.1 to Lemma 4.6. Observe that the adversary's view in H_3 is independent of the challenge bit β and its advantage is 0. This concludes the proof. □

Lemma 4.1. $\mathsf{H}_0 \approx_c \mathsf{H}_1$ *under the entropy expansion lemma.*

Proof. We use a slight variant of the entropy expansion lemma in Lemma 3.3, which is given as follows. The only difference from Lemma 3.3 is the distribution of $\mathbf{D}_0, \mathbf{D}_i$. Additionally, we set $\ell_1 = \ell_2 = \ell_3 = k$ and $\ell_W = k + 1$.

Lemma 4.2 (Bilinear Entropy Expansion Lemma (Variant)). *Let n and Q be a positive integer. Sample $\mathbf{A}_i \leftarrow \mathbb{Z}_p^{3k\times k}$ and define $\mathbf{A}_i^{\parallel}$ for $i \in \{1,2,3\}$ following Definition 3.5. Then, we have*

$$\left\{\begin{array}{l}\mathsf{aux} : \mathbb{G}, [\mathbf{A}_1, \mathbf{U}_1\mathbf{A}_1, \mathbf{U}_2\mathbf{A}_1, \mathbf{V}_1\mathbf{A}_1, \mathbf{V}_2\mathbf{A}_1]_1 \\ \mathsf{ct} : [\mathbf{c}_0^\top]_1, \{[\mathbf{c}_i^\top, \mathbf{V}_1\mathbf{c}_i^\top, \mathbf{V}_2\mathbf{c}_0^\top + (\mathbf{U}_1 + i\mathbf{U}_2)\mathbf{c}_i^\top]_2\}_{i\in[n]} \\ \mathsf{sk} : [\mathbf{D}_0]_2, \{[\mathbf{D}_i, \mathbf{D}_i\mathbf{V}_2, \mathbf{D}_0\mathbf{V}_1 + \mathbf{D}_i(\mathbf{U}_1 + i\mathbf{U}_2)]_2\}_{i\in[n]}\end{array}\right\}$$
$$\approx_c \left\{\begin{array}{l}\mathsf{aux} : \mathbb{G}, [\mathbf{A}_1, \mathbf{U}_1\mathbf{A}_1, \mathbf{U}_2\mathbf{A}_1, \mathbf{V}_1\mathbf{A}_1, \mathbf{V}_2\mathbf{A}_1]_1 \\ \mathsf{ct} : [\mathbf{c}_0^\top]_1, \{[\mathbf{c}_i^\top, (\mathbf{V}_1 + \mathbf{W}_i)\mathbf{c}_i^\top, (\mathbf{V}_2 + \mathbf{W}_i')\mathbf{c}_0^\top + (\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}_i')\mathbf{c}_i^\top]_1\}_{i\in[n]} \\ \mathsf{sk} : [\mathbf{D}_0]_2, \{[\mathbf{D}_i, \mathbf{D}_i(\mathbf{V}_2 + \mathbf{W}_i'), \mathbf{D}_0(\mathbf{V}_1 + \mathbf{W}_i) + \mathbf{D}_i(\mathbf{U}_1 + i\mathbf{U}_2 + \mathbf{U}_i')]_2\}_{i\in[n]}\end{array}\right\}$$

where $\mathbb{G} \leftarrow \mathcal{G}_{\mathsf{BG}}(1^\lambda)$, $\mathbf{U}_1, \mathbf{U}_2, \mathbf{V}_1, \mathbf{V}_2 \leftarrow \mathbb{Z}_p^{(k+1)\times 3k}$, $\mathbf{B} \leftarrow \mathbb{Z}_p^{k\times(k+1)}$, $\mathbf{D}_0, \mathbf{D}_i \leftarrow \mathsf{RowSpan}(\mathbf{B})^{Q\times 1}$, and $\mathbf{W}_i, \mathbf{W}_i', \mathbf{U}_i' \leftarrow \mathsf{RowSpan}(\mathbf{A}_2^{\parallel\top})^{3k\times 1}$ for $i \in [n]$. In the LHS, $\mathbf{c}_0^\top, \mathbf{c}_i \leftarrow \mathsf{ColSpan}(\mathbf{A}_1)$ while in the RHS, $\mathbf{c}_0^\top, \mathbf{c}_i^\top \leftarrow \mathsf{ColSpan}(\mathbf{A}_1, \mathbf{A}_2)$. Our lemma is also implied by the MDDH assumption.

The proof of the above lemma is the same as [CGKW18, Lemma 12]. In a nutshell, the MDDH assumption says that $[\{\mathbf{D}_i\}_{i\in[0,n]}]_2 \approx_c [\{\mathbf{R}_i\mathbf{B}\}_{i\in[0,n]}]_2$ where $\mathbf{D}_i \leftarrow \mathbb{Z}_p^{Q\times(k+1)}$, $\mathbf{R}_i \leftarrow \mathbb{Z}_p^{Q\times k}$, $\mathbf{B} \leftarrow \mathbb{Z}_p^{k\times(k+1)}$, and thus Lemma 3.3 readily implies Lemma 4.2.

We prove $\mathsf{H}_0 \approx_c \mathsf{H}_1$ under Lemma 4.2. To this end, we construct a reduction algorithm $\mathcal{B}$ that distinguishes Lemma 4.2 by internally runs $\mathcal{A}$ that distinguishes H_0 and H_1 as follows. Let (x, L^*) be the attribute for the challenge query, $(y^{(j)}, L^{(j)})$ be the attribute queried in the j-th secret key query, and $n_{\max} = \max(\mathsf{p}(L^*), \{\mathsf{p}(L^{(j)})\}_{j\in[Q]})$.

- $\mathcal{B}$ is given the instance of Lemma 4.2:

$$\left\{\begin{array}{l}\mathsf{aux} : \mathbb{G}, [\mathbf{A}_1, \mathbf{U}_1\mathbf{A}_1, \mathbf{U}_2\mathbf{A}_1, \mathbf{V}_1\mathbf{A}_1, \mathbf{V}_2\mathbf{A}_1]_1 \\ \mathsf{ct} : [\hat{\mathbf{c}}_0^\top]_1, \{[\hat{\mathbf{c}}_{1,i}^\top, \hat{\mathbf{c}}_{2,i}^\top, \hat{\mathbf{c}}_{3,i}^\top]_2\}_{i\in[n_{\max}]} \\ \mathsf{sk} : [\mathbf{K}_0]_2, \{[\mathbf{K}_{1,i}, \mathbf{K}_{2,i}, \mathbf{K}_{3,i}]_2\}_{i\in[n_{\max}]}\end{array}\right\}$$

- $\mathcal{B}$ samples $\mathbf{k} \leftarrow \mathbb{Z}_p^{3k}$ and gives $\mathsf{mpk} = ([\mathbf{A}_1, \mathbf{U}_1\mathbf{A}_1, \mathbf{U}_2\mathbf{A}_1, \mathbf{V}_1\mathbf{A}_1, \mathbf{V}_2\mathbf{A}_1]_1, [\mathbf{k}\mathbf{A}_1]_T)$ to $\mathcal{A}$.
- For the j-th secret key query, which is on (y, L), $\mathcal{B}$ generates sk as

$$n = \mathsf{p}(L),\ k \leftarrow \mathsf{EncK}(y, L),$$
$$\mathsf{sk} = (k([\mathbf{k}, \mathbf{k}_{2,1}^{(j)}, \ldots, \mathbf{k}_{2,n}^{(j)}]_2), [\mathbf{k}_0^{(j)}, \{\mathbf{k}_{1,i}^{(j)}, \mathbf{k}_{3,i}^{(j)}\}_{i\in[n]}]_2)$$

  where $\mathbf{k}_0^{(j)}$ and $\mathbf{k}_{\ell,i}^{(j)}$ are the j-th row of $\mathbf{K}_0$ and $\mathbf{K}_{\ell,i}$ for $\ell \in \{1,2,3\}$, respectively.[11]
- For the challenge ciphertext query for (x, L^*), M_0, M_1, $\mathcal{B}$ samples $\beta \leftarrow \{0,1\}$ and generates ct as

$$n^* = \mathsf{p}(L^*),\ c \leftarrow \mathsf{EncC}(x, L^*),$$
$$\mathsf{ct} = (c([\hat{\mathbf{c}}_{2,1}^\top, \ldots, \hat{\mathbf{c}}_{2,n^*}^\top]_1), [\hat{\mathbf{c}}_0^\top, \{\hat{\mathbf{c}}_{1,i}^\top, \hat{\mathbf{c}}_{3,i}^\top\}_{i\in[n^*]}]_1, [\mathbf{k}\hat{\mathbf{c}}_0^\top]_T \cdot M_\beta)$$

- When $\mathcal{A}$ outputs β', $\mathcal{B}$ outputs $\beta \stackrel{?}{=} \beta'$.

Observe that $\mathcal{A}$'s view corresponds to H_0 if the given instance is the LHS of Lemma 4.2, and H_1 otherwise. □

[11] Note that parameters such as L, n can be different in each secret key query, though we do not index them for conciseness.

Lemma 4.3. *Let* $\mathsf{H}_{2,0,3} = \mathsf{H}_1$. $\mathsf{H}_{2,j-1,3} \approx_c \mathsf{H}_{2,j,1}$ *for* $j \in [Q]$ *under MDDH.*

Proof. The difference between $\mathsf{H}_{2,j-1,3}$ and $\mathsf{H}_{2,j,1}$ is whether the j-th key is E-normal or P-normal. We prove $\mathsf{H}_{2,j-1,3} \approx_c \mathsf{H}_{2,j,1}$ under MDDH. To this end, we construct a reduction algorithm $\mathcal{B}$ solving MDDH that internally runs $\mathcal{A}$ that distinguishes $\mathsf{H}_{2,j-1,3}$ and $\mathsf{H}_{2,j,1}$ as follows.

- $\mathcal{B}$ is given a MDDH instance $[\mathbf{B}, \{\mathbf{d}_i\}_{i\in[0,n]}]_2$ where $\mathbf{B} \leftarrow \mathbb{Z}_p^{k\times(k+1)}$, and either $\mathbf{d}_i \leftarrow \mathsf{RowSpan}(\mathbf{B})$ or $\mathbf{d}_i \leftarrow \mathbb{Z}_p^{k+1}$ for all $i \in [0, n]$.
- $\mathcal{B}$ generates $\mathsf{mpk} = ([\mathbf{U}_1\mathbf{A}_1, \mathbf{U}_2\mathbf{A}_1, \mathbf{V}_1\mathbf{A}_1, \mathbf{V}_2\mathbf{A}_1]_1, [\mathbf{k}\mathbf{A}_1]_T)$ following Setup. It samples $\mathbf{A}_2 \leftarrow \mathbb{Z}_p^{3k\times k}$, $\mathbf{A}_2^{\parallel} \leftarrow \mathbb{Z}_p^{3k\times k}$ conditioned on $\mathbf{A}_2^{\parallel\top}\mathbf{A}_1 = \mathbf{O}$ and $\mathbf{A}_2^{\parallel\top}\mathbf{A}_2 = \mathbf{I}_k$, and $\mathbf{W}_i, \mathbf{W}'_i, \mathbf{U}'_i \leftarrow \mathsf{RowSpan}(\mathbf{A}_2^{\parallel\top})^{(k+1)\times 1}$ for $i \in [n]$. It gives mpk to $\mathcal{A}$.
- For the j'-th key query, which is on (y, L), $\mathcal{B}$ generates sk as

$$n = \mathsf{p}(L),\ k \leftarrow \mathsf{EncK}(y, L),\ \mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_n \leftarrow \mathbb{Z}_p^k,\ \mathbf{z} \leftarrow \mathsf{RowSpan}(\mathbf{a}_{2,1}^{\parallel})$$
$$\mathsf{sk} = \begin{cases} \begin{pmatrix} k([\mathbf{k}+\mathbf{z}, \mathbf{r}_1\mathbf{B}(\mathbf{V}_2+\mathbf{W}'_1), \ldots, \mathbf{r}_n\mathbf{B}(\mathbf{V}_2+\mathbf{W}'_n)]_2) \\ [\mathbf{r}_0\mathbf{B}, \{\mathbf{r}_i\mathbf{B}, \mathbf{r}_0\mathbf{B}(\mathbf{V}_1+\mathbf{W}_i) + \mathbf{r}_i\mathbf{B}(\mathbf{U}_1+i\mathbf{U}_2+\mathbf{U}'_i)\}_{i\in[n]}]_2 \end{pmatrix} & (j' < j) \\ \begin{pmatrix} k([\mathbf{k}, \mathbf{d}_1(\mathbf{V}_2+\mathbf{W}'_1), \ldots, \mathbf{d}_n(\mathbf{V}_2+\mathbf{W}'_n)]_2) \\ [\mathbf{d}_0, \{\mathbf{d}_i, \mathbf{d}_0(\mathbf{V}_1+\mathbf{W}_i) + \mathbf{d}_i(\mathbf{U}_1+i\mathbf{U}_2+\mathbf{U}'_i)\}_{i\in[n]}]_2 \end{pmatrix} & (j' = j) \\ \begin{pmatrix} k([\mathbf{k}, \mathbf{r}_1\mathbf{B}(\mathbf{V}_2+\mathbf{W}'_1), \ldots, \mathbf{r}_n\mathbf{B}(\mathbf{V}_2+\mathbf{W}'_n)]_2) \\ [\mathbf{r}_0\mathbf{B}, \{\mathbf{r}_i\mathbf{B}, \mathbf{r}_0\mathbf{B}(\mathbf{V}_1+\mathbf{W}_i) + \mathbf{r}_i\mathbf{B}(\mathbf{U}_1+i\mathbf{U}_2+\mathbf{U}'_i)\}_{i\in[n]}]_2 \end{pmatrix} & (j' > j) \end{cases}$$

- For the challenge ciphertext query for $(x, L^*), M_0, M_1$, $\mathcal{B}$ samples $\beta \leftarrow \{0, 1\}$ and generates ct as

$$n^* = \mathsf{p}(L^*),\ c \leftarrow \mathsf{EncC}(x, L^*),\ \mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_{n^*} \leftarrow \mathsf{ColSpan}(\mathbf{A}_1, \mathbf{A}_2)$$
$$\mathsf{ct} = \begin{pmatrix} c([(\mathbf{V}_1+\mathbf{W}_1)\mathbf{c}_1^\top, \ldots, (\mathbf{V}_1+\mathbf{W}_{n^*})\mathbf{c}_{n^*}^\top]_1) \\ [\mathbf{c}_0^\top, \{\mathbf{c}_i^\top, (\mathbf{V}_2+\mathbf{W}'_i)\mathbf{c}_0^\top + (\mathbf{U}_1+i\mathbf{U}_2+\mathbf{U}'_i)\mathbf{c}_i^\top\}_{i\in[n^*]}]_1,\ [\mathbf{k}\mathbf{c}_0^\top]_T \cdot M_\beta \end{pmatrix}$$

- When $\mathcal{A}$ outputs β', $\mathcal{B}$ outputs $\beta \stackrel{?}{=} \beta'$.

Observe that $\mathcal{A}$'s view corresponds to $\mathsf{H}_{2,j-1,3}$ if $\mathbf{d}_i$ in the MDDH instance is distributed as $\mathbf{d}_i \leftarrow \mathsf{RowSpan}(\mathbf{B})$, and $\mathsf{H}_{2,j,1}$ otherwise. □

Lemma 4.4. $\mathsf{H}_{2,j,1} \approx_s \mathsf{H}_{2,j,2}$ *for* $j \in [Q]$ *under the unbounded security of the linear CDS.*

Proof. To see these hybrids are statistically close, we consider the following programming

$$\mathbf{W}_i = \widehat{\mathbf{W}}_i + w_i\mathbf{b}^{\parallel\top}\mathbf{a}_{2,1}^{\parallel}, \quad \mathbf{W}'_i = \widehat{\mathbf{W}}'_i + w'_i\mathbf{b}^{\parallel\top}\mathbf{a}_{2,1}^{\parallel}, \quad \mathbf{U}'_i = \widehat{\mathbf{U}}'_i + u'_i\mathbf{b}^{\parallel\top}\mathbf{a}_{2,1}^{\parallel} \tag{14}$$

where $\widehat{\mathbf{W}}_i, \widehat{\mathbf{W}}'_i, \widehat{\mathbf{U}}'_i \leftarrow \mathsf{RowSpan}(\mathbf{A}_2^{\parallel\top})^{(k+1)\times 1}$, $w_i, w'_i, u'_i \leftarrow \mathbb{Z}_p$ for $i \in [n_{\max}]$, and $\mathbf{b}^{\parallel}$ is a non-zero vector such that $\mathbf{B}\mathbf{b}^{\parallel\top} = \mathbf{0}$. Observe that this programming does not change the distributions of $\mathbf{W}_i, \mathbf{W}'_i, \mathbf{U}'_i$ since $\mathbf{b}^{\parallel\top}\mathbf{a}_{2,1}^{\parallel} \in \mathsf{RowSpan}(\mathbf{A}_2^{\parallel\top})^{(k+1)\times 1}$. Due to the orthogonality, this programming does not affect the distributions of the master public key and all the secret keys except the j-th key. The challenge ciphertext and the j-th key in $\mathsf{H}_{2,j,1}$ can be written as

$$\mathsf{ct} = \begin{pmatrix} c([(\mathbf{V}_1+\widehat{\mathbf{W}}_1)\mathbf{c}_1^\top, \ldots, (\mathbf{V}_1+\widehat{\mathbf{W}}_{n^*})\mathbf{c}_{n^*}^\top]_1) + [c(s_1w_1, \ldots, s_{n^*}w_{n^*})\mathbf{b}^{\parallel\top}]_1 \\ [\mathbf{c}_0^\top, \{\mathbf{c}_i^\top, (\mathbf{V}_2+\widehat{\mathbf{W}}'_i)\mathbf{c}_0^\top + (\mathbf{U}_1+i\mathbf{U}_2+\widehat{\mathbf{U}}'_i)\mathbf{c}_i^\top + (s_0w'_i+s_iu'_i)\mathbf{b}^{\parallel\top}\}_{i\in[n^*]}]_1 \\ [\mathbf{k}\mathbf{c}_0^\top]_T \cdot M_\beta \end{pmatrix}$$
$$\mathsf{sk}_j = \begin{pmatrix} k([\mathbf{k}, \mathbf{d}_1(\mathbf{V}_2+\widehat{\mathbf{W}}'_1), \ldots, \mathbf{d}_n(\mathbf{V}_2+\widehat{\mathbf{W}}'_n)]_2) + [k(0, r_1w'_1, \ldots, r_nw'_n)\mathbf{a}_{2,1}^{\parallel}]_2 \\ [\mathbf{d}_0, \{\mathbf{d}_i, \mathbf{d}_0(\mathbf{V}_1+\widehat{\mathbf{W}}_i) + \mathbf{d}_i(\mathbf{U}_1+i\mathbf{U}_2+\widehat{\mathbf{U}}'_i) + (r_0w_i+r_iu'_i)\mathbf{a}_{2,1}^{\parallel}\}_{i\in[n]}]_2 \end{pmatrix} \tag{15}$$

where $s_i = \mathbf{a}_{2,1}^{\parallel}\mathbf{c}_i^\top$ and $r_i = \mathbf{d}_i\mathbf{b}^{\parallel^\top}$ for $i \in [0,n]$. By definition, s_i, r_i are uniformly distributed in $\mathbb{Z}_p$. We prove that the following:

$$\begin{aligned}&\left\{\begin{matrix} c(s_1w_1,\ldots,s_{n^*}w_{n^*}), \{s_0w'_i + s_iu'_i\}_{i\in[n^*]} \\ k(0, r_1w'_1,\ldots,r_nw'_n), \{r_0w_i + r_iu'_i\}_{i\in[n]}\end{matrix}\right\} \\ \equiv &\left\{\begin{matrix} c(s_1w_1,\ldots,s_{n^*}w_{n^*}), \{s_0w'_i + s_iu'_i\}_{i\in[n^*]} \\ k(\alpha, r_1w'_1,\ldots,r_nw'_n), \{r_0w_i + r_iu'_i\}_{i\in[n]}\end{matrix}\right\}\end{aligned} \tag{16}$$

where all valuables are randomly taken from $\mathbb{Z}_p$. To this end, we consider the following programming: $w_i = \hat{w}_i - \frac{r_iu'_i}{r_0},\ w'_i = \hat{w}'_i - \frac{s_iu'_i}{s_0}$ where $\hat{w}_i, \hat{w}'_i \leftarrow \mathbb{Z}_p$ for $i \in [\max(n, n^*)]$. Then, the LHS of Eq.(16) can be written as

$$\begin{aligned}&\left\{\begin{matrix} c(s_1\hat{w}_1,\ldots,s_{n^*}\hat{w}_{n^*}) - \dfrac{c(s_1r_1u'_1,\ldots,s_{n^*}r_{n^*}u'_{n^*})}{r_0}, \{\hat{w}'_i\}_{i\in[n^*]} \\ k(0, r_1\hat{w}'_1,\ldots,r_n\hat{w}'_n) - \dfrac{k(0, s_1r_1u'_1,\ldots,s_nr_nu'_n)}{s_0}, \{\hat{w}_i\}_{i\in[n]}\end{matrix}\right\} \\ \equiv &\left\{\begin{matrix} c(s_1\hat{w}_1,\ldots,s_{n^*}\hat{w}_{n^*}) - \dfrac{c(s_1r_1u'_1,\ldots,s_{n^*}r_{n^*}u'_{n^*})}{r_0}, \{\hat{w}'_i\}_{i\in[n^*]} \\ k(0, r_1\hat{w}'_1,\ldots,r_n\hat{w}'_n) - \dfrac{k(\alpha, s_1r_1u'_1,\ldots,s_nr_nu'_n)}{s_0}, \{\hat{w}_i\}_{i\in[n]}\end{matrix}\right\}\end{aligned}$$

where the equivalence follows from the unbounded security of the linear CDS as per Definition 3.4 and the admissibility of the adversary in unbounded ABE security game, which requires that $L \neq L^*$ or $\mathsf{P}_L(x,y) = 0$. The distribution of the RHS of the above equation is the same as that of the RHS of Eq.(16), which follows from undoing the programming. Thus, Eq.(16) holds. Thanks to Eq.(16), the distribution of Eq.(15) is the same as the following:

$$\begin{aligned}\mathsf{ct} = &\begin{pmatrix} c([(\mathbf{V}_1 + \widehat{\mathbf{W}}_1)\mathbf{c}_1^\top, \ldots, (\mathbf{V}_1 + \widehat{\mathbf{W}}_{n^*})\mathbf{c}_{n^*}^\top]_1) + [c(s_1w_1,\ldots,s_{n^*}w_{n^*})\mathbf{b}^{\parallel^\top}]_1 \\ [\mathbf{c}_0^\top, \{\mathbf{c}_i^\top, (\mathbf{V}_2 + \widehat{\mathbf{W}}'_i)\mathbf{c}_0^\top + (\mathbf{U}_1 + i\mathbf{U}_2 + \widehat{\mathbf{U}}'_i)\mathbf{c}_i^\top + (s_0w'_i + s_iu'_i)\mathbf{b}^{\parallel^\top}\}_{i\in[n^*]}]_1 \\ [\mathbf{k}\mathbf{c}_0^\top]_T \cdot M_\beta \end{pmatrix} \\ \mathsf{sk}_j = &\begin{pmatrix} k([\mathbf{k}, \mathbf{d}_1(\mathbf{V}_2 + \widehat{\mathbf{W}}'_1), \ldots, \mathbf{d}_n(\mathbf{V}_2 + \widehat{\mathbf{W}}'_n)]_2) + [k(\alpha, r_1w'_1,\ldots,r_nw'_n)\mathbf{a}_{2,1}^{\parallel}]_2 \\ [\mathbf{d}_0, \{\mathbf{d}_i, \mathbf{d}_0(\mathbf{V}_1 + \widehat{\mathbf{W}}_i) + \mathbf{d}_i(\mathbf{U}_1 + i\mathbf{U}_2 + \widehat{\mathbf{U}}'_i) + (r_0w_i + r_iu'_i)\mathbf{a}_{2,1}^{\parallel}\}_{i\in[n]}]_2 \end{pmatrix}\end{aligned}$$

By undoing the programming in Eq.(14), we can see that the above distribution corresponds to the challenge ciphertext and the j-th key in $\mathsf{H}_{2,j,2}$. Thus, $\mathsf{H}_{2,j,1}$ and $\mathsf{H}_{2,j,2}$ are statistically close. □

Lemma 4.5. $\mathsf{H}_{2,j,2} \approx_c \mathsf{H}_{2,j,3}$ *for* $j \in [Q]$ *under MDDH.*

The proof is similar to that of Lemma 4.3. Thus, we omit the proof.

Lemma 4.6. $\mathsf{H}_{2,Q,3} \approx_s \mathsf{H}_3$.

Proof. To see this, we program $\mathbf{k}$ as $\mathbf{k} = \hat{\mathbf{k}} + \gamma\mathbf{a}_{2,1}^{\parallel}$ where $\hat{\mathbf{k}} \leftarrow \mathbb{Z}_p^{3k}$ and $\gamma \leftarrow \mathbb{Z}_p$. Observe that mpk and all sk do not reveal any information on γ because $\mathbf{a}_{2,1}^{\parallel}\mathbf{A} = \mathbf{0}$ for mpk, and $\mathbf{k}$ in each sk is masked by a fresh random vector $\mathbf{z} \leftarrow \mathsf{RowSpan}(\mathbf{a}_{2,1}^{\parallel})$. For ct, $\mathbf{a}_{2,1}^{\parallel}\mathbf{c}_0^\top \neq 0$ with overwhelming probability, and thus $\mathbf{k}\mathbf{c}_0^\top = \hat{\mathbf{k}}\mathbf{c}_0^\top + \gamma\mathbf{a}_{2,1}^{\parallel}\mathbf{c}_0^\top$ is statistically close to uniform in $\mathbb{Z}_p$ and independent of other elements. Thus, $\mathsf{H}_{2,Q,3}$ and H_3 are statistically close. □

5 Unbounded Broadcast Encryption

In this section, we present how to instantiate an unbounded (indentity-based) broadcast encryption scheme with new parameters via our compiler in Section 4.1. First, we show that any linear CDS for P with bounded security can be converted to a linear CDS for P with unbounded security. Then, we apply the conversion to the linear CDS for non-zero quadratic polynomials with bounded security by [LVW17] to achieve unbounded security. Using the compiler in Section 4.1, we can obtain an unbounded ABE scheme for non-zero quadratic polynomials. Finally, we achieve unbounded broadcast encryption with new parameters from the unbounded ABE for non-zero quadratic polynomials with a standard embedding technique. Our unbounded identity-based broadcast encryption can be similarly obtained except that we start with a linear CDS for zero quadratic polynomials.

5.1 From Bounded to Unbounded Security.

A high level idea of the conversion is the conjunction with the IBE predicate [ABS17] so that only the ciphertext and key encodings for the same L can interact. Given a linear CDS $\Gamma = (\mathsf{p}, \mathsf{EncC}, \mathsf{EncK}, \mathsf{cRec})$ for P with bounded security, we can generically obtain a linear CDS $\Gamma' = (\mathsf{p}', \mathsf{EncC}', \mathsf{EncK}', \mathsf{cRec}')$ for P with unbounded security as follows (see Definition 3.4 for the security definition). We require that L is taken from a space $\mathcal{L}^*$ such that $|\mathcal{L}^*| < p$ and can be embedded in $\mathbb{Z}_p$ by an injective map $\phi : \mathcal{L}^* \to \mathbb{Z}_p$.[12]

- $\mathsf{p}'(L) = \mathsf{p}(L) + 2$.
- $\mathsf{EncC}'(x, L) \to c'$: It runs $c \leftarrow \mathsf{EncC}(x, L)$ where $c : \mathbb{Z}_p^{\mathsf{p}(L)} \to \mathbb{Z}_p^{m_1}$. It outputs $c' : \mathbb{Z}_p^{\mathsf{p}(L)+2} \to \mathbb{Z}_p^{m_1+1}$ such that

$$c'(\mathbf{w}, u_1, u_2) = (c(\mathbf{w}), u_1 + \phi(L)u_2)$$

- $\mathsf{EncK}'(y, L) \to k'$: It runs $k \leftarrow \mathsf{EncK}(y, L)$ where $k : \mathbb{Z}_p^{\mathsf{p}(L)+1} \to \mathbb{Z}_p^{m_2}$. It outputs $k' : \mathbb{Z}_p^{\mathsf{p}(L)+3} \to \mathbb{Z}_p^{m_2}$ such that

$$k'(\alpha, \mathbf{w}, u_1, u_2) = k(\alpha - (u_1 + \phi(L)u_2), \mathbf{w})$$

- $\mathsf{cRec}'(x, y, L) \to d'$: It runs $d \leftarrow \mathsf{cRec}(x, y, L)$ and outputs $d' : \mathbb{Z}_p^{m_1+1+m_2+1} \to \mathbb{Z}_p$ such that

$$d'(\mathbf{c}, c_{m_1+1}, \mathbf{k}) = d(\mathbf{c}, \mathbf{k}) + c_{m_1+1}$$

Lemma 5.1 (From Bounded to Unbounded Security). *If Γ satisfies correctness and bounded security, then Γ' satisfies correctness and unbounded security.*

Proof. Correctness is almost trivial from the definition, i.e.,

$$\begin{aligned} d'(\mathbf{c}, c_{m_1+1}, \mathbf{k}) =& d(\mathbf{c}, \mathbf{k}) + c_{m_1+1} \\ =& \alpha - (u_1 + \phi(L)u_2) + (u_1 + \phi(L)u_2) = \alpha \end{aligned}$$

For security, the unbounded security of Γ' for the case $L = L' \wedge \mathsf{P}_L(x, y) = 0$ directly follows from the bounded security of Γ, so we consider the case $L \neq L'$. What we need to prove is

$$\left\{ \begin{matrix} (c(\mathbf{w}), u_1 + \phi(L)u_2) \\ k(\alpha - (u_1 + \phi(L')u_2), \mathbf{w}') \end{matrix} \right\} \equiv \left\{ \begin{matrix} (c(\mathbf{w}), u_1 + \phi(L)u_2) \\ k(0 - (u_1 + \phi(L')u_2), \mathbf{w}') \end{matrix} \right\}$$

where all variables are randomly taken from $\mathbb{Z}_p$. In this case, we have $\phi(L) \neq \phi(L')$, and thus $(u_1 + \phi(L)u_2, u_1 + \phi(L')u_2)$ is uniformly distributed in $\mathbb{Z}_p^2$. Thus, the above distributions are identical. □

[12] If $|\mathcal{L}| > p$, we can use collision resistant hashing from $\mathcal{L}$ to $\mathcal{L}^*$ such that $|\mathcal{L}^*| < p$ when we instantiate an unbounded ABE scheme from a linear CDS.

5.2 Unbounded ABE for Quadratic Polynomials

To construct unbounded ABE for (non-)zero quadratic polynomials, we define the predicate for quadratic polynomials as follows.

Definition 5.1 (Quadratic Polynomials [LVW17]). A predicate family $\mathsf{P}^{\mathsf{NQP}} = \{\mathsf{P}^{\mathsf{NQP}}_L\}_{L\in\mathcal{L}}$ where $\mathsf{P}_L : \mathcal{X}_L \times \mathcal{Y}_L \to \{0,1\}$ for non-zero quadratic polynomials is given as follows.

$$\mathcal{L} = \mathbb{N}^2, \quad \mathcal{X}_{N_1,N_2} = \mathbb{Z}_p^{N_1\times N_2}, \quad \mathcal{Y}_{N_1,N_2} = \mathbb{Z}_p^{N_1} \times \mathbb{Z}_p^{N_2}$$
$$\mathsf{P}^{\mathsf{NQP}}_{N_1,N_2}(\mathbf{M},(\mathbf{x},\mathbf{y})) = 1 \Leftrightarrow \mathbf{x}\mathbf{M}\mathbf{y}^\top \neq 0$$

where $N_1, N_2 \in \mathbb{N}, \mathbf{M} \in \mathbb{Z}_p^{N_1\times N_2}, \mathbf{x} \in \mathbb{Z}_p^{N_1}, \mathbf{y} \in \mathbb{Z}_p^{N_2}$. A predicate family $\mathsf{P}^{\mathsf{ZQP}}$ for zero quadratic polynomials is the same as $\mathsf{P}^{\mathsf{NQP}}$ except that

$$\mathsf{P}^{\mathsf{ZQP}}_{N_1,N_2}(\mathbf{M},(\mathbf{x},\mathbf{y})) = 1 \Leftrightarrow \mathbf{x}\mathbf{M}\mathbf{y}^\top = 0$$

Lemma 5.2 (Linear CDS for $\mathsf{P}^{\mathsf{NQP}}$ [LVW17]). *A linear CDS $\Gamma = (\mathsf{p}, \mathsf{EncC}, \mathsf{EncK}, \mathsf{cRec})$ for P^{NQP} was given by [LVW17] as follows. This linear CDS satisfies correctness and bounded security as per Definition 3.4.*

- $\mathsf{p}(N_1, N_2) = N_1 + N_2$.
- $\mathsf{EncC}(\mathbf{M}, (N_1, N_2)) \to c$: *Output c such that*

$$c(\mathbf{w}) = \left\{ w_{i_1} + \sum_{i_2\in[N_2]} m_{i_1,i_2} w_{N_1+i_2} \right\}_{i_1\in[N_1]}$$

  where $\mathbf{w} = (w_1, \ldots, w_{N_1+N_2}) \in \mathbb{Z}_p^{N_1+N_2}$ *and* $\mathbf{M} = (m_{i_1,i_2})_{(i_1,i_2)\in[N_1]\times[N_2]}$.
- $\mathsf{EncK}((\mathbf{x},\mathbf{y}), (N_1, N_2)) \to k$: *Output k such that*

$$k(\alpha, \mathbf{w}) = \Big(\sum_{i_1\in[N_1]} x_{i_1} w_{i_1}, \{y_{i_2}\alpha + w_{N_1+i_2}\}_{i_2\in[N_2]} \Big)$$

  where $\alpha \in \mathbb{Z}_p$, $\mathbf{x} = (x_1, \ldots, x_{N_1})$, *and* $\mathbf{y} = (y_1, \ldots, y_{N_2})$.
- $\mathsf{cRec}(\mathbf{M}, (\mathbf{x},\mathbf{y}), (N_1, N_2)) \to d$: *Output d such that*

$$d(\mathbf{c}, \mathbf{k}) = \frac{k_0 - \sum_{i_1\in[N_1]} x_{i_1} c_{i_1} + \sum_{(i_1,i_2)\in[N_1]\times[N_2]} x_{i_1} m_{i_1,i_2} k_{i_2}}{\mathbf{x}\mathbf{M}\mathbf{y}^\top}$$

  where $\mathbf{c} = (c_1, \ldots, c_{N_1})$, $\mathbf{k} = (k_0, \ldots, k_{N_2})$.

We can generically obtain a linear CDS for $\mathsf{P}^{\mathsf{ZQP}}$ by applying the generic negation [ABS17] to the above linear CDS for $\mathsf{P}^{\mathsf{NQP}}$. However, the parameters of the converted linear CDS with $L = (N_1, N_2)$ will be

$$\mathsf{p}(N_1, N_2) = 2(N_1 + N_2), \quad m_1 = N_1 + N_2, \quad m_2 = N_1 + N_2 + 1$$

We directly construct a more efficient linear CDS for $\mathsf{P}^{\mathsf{ZQP}}$ with the following parameters by slightly modifying the above linear CDS:

$$\mathsf{p}(N_1, N_2) = N_1 + N_2 + 1, \quad m_1 = N_1, \quad m_2 = N_2 + 1$$

Lemma 5.3 (Linear CDS for $\mathsf{P}^{\mathsf{ZQP}}$). *A linear CDS $\Gamma = (\mathsf{p}, \mathsf{EncC}, \mathsf{EncK}, \mathsf{cRec})$ for P^{ZQP} can be constructed as follows. This linear CDS satisfies correctness and bounded security as per Definition 3.4.*

- $\mathsf{p}(N_1, N_2) = N_1 + N_2 + 1$.

– $\mathsf{EncC}(\mathbf{M}, (N_1, N_2)) \to c$: *Output* c *such that*

$$c(\mathbf{w}) = \left\{ w_{i_1} + \sum_{i_2 \in [N_2]} m_{i_1,i_2} w_{N_1+i_2} \right\}_{i_1 \in [N_1]}$$

where $\mathbf{w} = (w_1, \ldots, w_{N_1+N_2+1}) \in \mathbb{Z}_p^{N_1+N_2+1}$ *and* $\mathbf{M} = (m_{i_1,i_2})_{(i_1,i_2) \in [N_1] \times [N_2]}$.

– $\mathsf{EncK}((\mathbf{x}, \mathbf{y}), (N_1, N_2)) \to k$: *Output* k *such that*

$$k(\alpha, \mathbf{w}) = \left(\alpha + \sum_{i_1 \in [N_1]} x_{i_1} w_{i_1}, \{ y_{i_2} w_{N_1+N_2+1} + w_{N_1+i_2} \}_{i_2 \in [N_2]} \right)$$

where $\alpha \in \mathbb{Z}_p$, $\mathbf{x} = (x_1, \ldots, x_{N_1})$, *and* $\mathbf{y} = (y_1, \ldots, y_{N_2})$.

– $\mathsf{cRec}(\mathbf{M}, (\mathbf{x}, \mathbf{y}), (N_1, N_2)) \to d$: *Output* d *such that*

$$d(\mathbf{c}, \mathbf{k}) = k_0 - \sum_{i_1 \in [N_1]} x_{i_1} c_{i_1} + \sum_{(i_1,i_2) \in [N_1] \times [N_2]} x_{i_1} m_{i_1,i_2} k_{i_2}$$

where $\mathbf{c} = (c_1, \ldots, c_{N_1})$, $\mathbf{k} = (k_0, \ldots, k_{N_2})$.

Proof of Lemma 5.3. Correctness can be confirmed by doing a simple math. For security, we consider the following programming:

$$w_{N_1+i'_2} = \widetilde{w}_{N_1+i'_2} - y_{i'_2} w_{N_1+n_2+1}, \quad w_{i_1} = \widetilde{w}_{i_1} - \sum_{i_2 \in [N_2]} m_{i_1,i_2} w_{N_1+i_2}$$

for $(i_1, i'_2) \in [N_1] \times [N_2]$, where $\widetilde{w}_{N_1+i'_2}, \widetilde{w}_{i_1} \leftarrow \mathbb{Z}_p$. Then, we have

$$\begin{aligned} \{c(\mathbf{w}), k(\alpha, \mathbf{w})\} &= \{\{\widetilde{w}_{i_1}\}_{i_1 \in [N_1]}, (k_0, \{\widetilde{w}_{N_1+i_2}\}_{i_2 \in [N_2]}) \quad \} \\ &\equiv \{c(\mathbf{w}), k(0, \mathbf{w})\} \end{aligned}$$

where

$$k_0 = \alpha + \sum_{i_1 \in [N_1]} x_{i_1} \widetilde{w}_{i_1} - \sum_{(i_1,i_2) \in [N_1] \times [N_2]} x_{i_1} m_{i_1,i_2} (\widetilde{w}_{N_1+i_2} - y_{i_2} w_{N_1+n_2+1})$$

The equivalence follows from fact that $\sum_{(i_1,i_2) \in [N_1] \times [N_2]} x_{i_1} m_{i_1,i_2} y_{i_2} \neq 0$ and thus k_0 is randomly distributed in $\mathbb{Z}_p$ due to the entropy of $w_{N_1+N_2+1}$. □

We obtain unbounded ABE schemes for $\mathsf{P}^{\mathsf{NQP}}$ and $\mathsf{P}^{\mathsf{ZQP}}$ as follows.

Corollary 5.1 (Unbounded ABE for $\mathsf{P}^{\mathsf{NQP}}$ and $\mathsf{P}^{\mathsf{ZQP}}$). *We can construct unbounded ABE schemes (Definition 3.3) for* P^{NQP} *and* P^{ZQP} *(Definition 5.1) that satisfies adaptive unbounded security under the MDDH assumption. Furthermore, the sizes of* mpk, ct *for* $(\mathbf{M}, (N_1, N_2))$, *and* sk *for* $((\mathbf{x}, \mathbf{y}), (N_1, N_2))$ *of both schemes are as follows:*

$$|\mathsf{mpk}| = O(1), \quad |\mathsf{ct}| = O(N_1 + N_2), \quad |\mathsf{sk}| = O(N_1 + N_2)$$

Proof. Lemma 5.1 states that we can obtain a linear CDS for P with unbounded security from a linear CDS for P with bounded security without changing the asymptotic sizes of $\mathsf{p}(L), m_1, m_2$. By applying the conversion in Section 5.1 to the linear CDS schemes in Lemma 5.2 and Lemma 5.3, we obtain linear CDS schemes for $\mathsf{P}^{\mathsf{NQP}}$ and $\mathsf{P}^{\mathsf{ZQP}}$ with unbounded security such that

$$\mathsf{p}(N_1, N_2) = O(N_1 + N_2), \quad m_1 = O(N_1), \quad m_2 = O(N_2)$$

where m_1 and m_2 refer to the number of $\mathbb{Z}_p$ elements in the output of $c \leftarrow \mathsf{EncC}(\cdot, (N_1, N_2))$ and $k \leftarrow \mathsf{EncK}(\cdot, (N_1, N_2))$, respectively. By instantiating the unbounded ABE schemes in Section 4.1 with the linear CDS for $\mathsf{P}^{\mathsf{NQP}}$ and $\mathsf{P}^{\mathsf{ZQP}}$ with unbounded security, we can obtain the ABE schemes for $\mathsf{P}^{\mathsf{NQP}}$ and $\mathsf{P}^{\mathsf{ZQP}}$ described in Corollary 5.1. Unbounded adaptive security of the schemes follows from Theorem 4.1. □

5.3 Unbounded Broadcast Encryption

We describe how to construct our (identity-based) broadcast encryption from unbounded ABE for $\mathsf{P}^{\mathrm{NQP}}/\mathsf{P}^{\mathrm{ZQP}}$. We first recall unbounded (identity-based) broadcast encryption.

Definition 5.2 (Broadcast). A predicate family $\mathsf{P}^{\mathrm{BE}} = \{\mathsf{P}^{\mathrm{BE}}_L\}_{L\in\mathcal{L}}$ where $\mathsf{P}_L : \mathcal{X}_L \times \mathcal{Y}_L \to \{0,1\}$ for broadcast is given as follows.

$$\mathcal{L} = \mathbb{N}, \quad \mathcal{X}_N = 2^{[N]}, \quad \mathcal{Y}_N = [N]$$
$$\mathsf{P}_N(X, y) = 1 \Leftrightarrow y \in X$$

where $N \in \mathbb{N}$, $X \in 2^{[N]}$ and $y \in [N]$.

Definition 5.3 (Identity-Based Broadcast). A predicate family $\mathsf{P}^{\mathrm{IBBE}} = \{\mathsf{P}^{\mathrm{IBBE}}_L\}_{L\in\mathcal{L}}$ where $\mathsf{P}_L : \mathcal{X}_L \times \mathcal{Y}_L \to \{0,1\}$ for identity-based broadcast is given as follows.

$$\mathcal{L} = \mathbb{N}, \quad \mathcal{X}_N = \{S \subseteq \mathbb{Z}_p\backslash\{0\} \mid |S| \le N\}, \quad \mathcal{Y}_N = \mathbb{Z}_p\backslash\{0\}$$
$$\mathsf{P}_N(X, y) = 1 \Leftrightarrow y \in X$$

where $N \in \mathbb{N}$, p is an exponentially large modulus, $X \in \mathcal{X}_N$, and $y \in \mathcal{Y}_N$.[13]

Definition 5.4 (Unbounded Broadcast Encryption). Unbounded broadcast encryption (BE) is defined as unbounded ABE (Definition 3.3) for the predicate P^{BE}, and identity-based broadcast encryption (IBBE) as unbounded ABE for the predicate $\mathsf{P}^{\mathrm{IBBE}}$.

Unbounded Broadcast Encryption. P^{BE} can be embedded in $\mathsf{P}^{\mathrm{NQP}}$ (Definition 5.1), that is, a linear CDS for $\mathsf{P}^{\mathrm{NQP}}$ can also be used for P^{BE} as follows. For $N \in \mathbb{N}$, let $g(N) = \lceil\sqrt{N}\rceil$ be the minimum integer that is greater than or equal to $\sqrt{N}$ and $\psi : [N] \to [g(N)]^2$ be any injective function. (A direct CDS for P^{BE} also appears in [GKW15, Appendix C].)

- $\mathsf{P}^{\mathrm{BE}}_N$ is embedded in $\mathsf{P}^{\mathrm{NQP}}_{g(N),g(N)}$.
- $X \in \mathcal{X}^{\mathrm{BE}}_N = 2^{[N]}$ is translated to $\mathbf{M} \in \mathcal{X}^{\mathrm{NQP}}_{g(N),g(N)} = \mathbb{Z}_p^{g(N)\times g(N)}$ such that $m_{i,j} = 1$ if $\psi^{-1}(i,j) \in X$ and $m_{i,j} = 0$ otherwise.
- $y \in \mathcal{Y}^{\mathrm{BE}}_N = [N]$ is translated to $(\mathbf{x}, \mathbf{y}) \in \mathcal{Y}^{\mathrm{NQP}}_{g(N),g(N)} \in \mathbb{Z}_p^{g(N)} \times \mathbb{Z}_p^{g(N)}$ such that the i-th entry of $\mathbf{x}$ and the j-th entry of $\mathbf{y}$ for $\psi^{-1}(i,j) = y$ are 1, and all the other entries are 0.

In the above construction, we can see that $\mathbf{x}\mathbf{M}\mathbf{y}^\top \neq 0$ if and only if $y \in X$. Thanks to Corollary 5.1, we can construct an unbounded BE scheme for N users with the following parameters:

$$|\mathsf{mpk}| = O(1), \quad |\mathsf{ct}| = O(\sqrt{N}), \quad |\mathsf{sk}| = O(\sqrt{N})$$

Unbounded IBBE. $\mathsf{P}^{\mathrm{IBBE}}$ can be embedded in $\mathsf{P}^{\mathrm{ZQP}}$ (Definition 5.1), that is, a linear CDS for $\mathsf{P}^{\mathrm{ZQP}}$ can also be used for $\mathsf{P}^{\mathrm{IBBE}}$ as follows. For $N \in \mathbb{N}$, let $g(N) = \lceil\sqrt{N+1}\rceil$ be the minimum integer that is greater than or equal to $\sqrt{N+1}$ and $\psi : [g(N)]^2 \to [0, g(N)^2 - 1]$ be an injective function such that $\psi(i,j) = g(N)(j-1) + i - 1$.

- $\mathsf{P}^{\mathrm{IBBE}}_N$ is embedded in $\mathsf{P}^{\mathrm{ZQP}}_{g(N),g(N)}$.
- $X \in \mathcal{X}^{\mathrm{IBBE}}_N = \{S \subseteq \mathbb{Z}_p\backslash\{0\} \mid |S| \le N\}$ is translated to $\mathbf{M} = (m_{i,j})_{(i,j)\in[g(N)]^2} \in \mathcal{X}^{\mathrm{ZQP}}_{g(N),g(N)} = \mathbb{Z}_p^{g(N)\times g(N)}$ such that

$$z^{g(N)^2-1-|X|}\prod_{x\in X}(z-x) = \sum_{(i,j)\in[g(N)]^2} m_{i,j} z^{\psi(i,j)}$$

for a formal variable z.

[13] We exclude 0 from $\mathcal{X}_N$ and $\mathcal{Y}_N$ to use it as a dummy attribute for padding when embedding $\mathsf{P}^{\mathrm{IBBE}}$ in $\mathsf{P}^{\mathrm{ZQP}}$.

– $y \in \mathcal{Y}_N^{\mathrm{IBBE}} = \mathbb{Z}_p \backslash \{0\}$ is translated to $(\mathbf{x}, \mathbf{y}) \in \mathcal{Y}_{g(N),g(N)}^{\mathrm{NQP}} \in \mathbb{Z}_p^{g(N)} \times \mathbb{Z}_p^{g(N)}$ such that the i-th element of $\mathbf{x}$ is y^{i-1} and the j-th element of $\mathbf{y}$ is $y^{g(N)(j-1)}$.

In the above construction, we can see that $\mathbf{x}\mathbf{M}\mathbf{y}^\top = 0$ if and only if $y \in X$. Thanks to Corollary 5.1, we can construct an unbounded IBBE scheme with a maximum user set size of N and the following parameters:

$$|\mathsf{mpk}| = O(1), \quad |\mathsf{ct}| = O(\sqrt{N}), \quad |\mathsf{sk}| = O(\sqrt{N})$$

6 Our Unbounded KP-ABE for Span Programs from CDS

In this section, we present how to extend our compiler in Section 4.1 to construct a KP-ABE scheme for span programs from a linear CDS.

6.1 Preliminaries

Definition 6.1 (Unbounded KP-ABE for Span Programs over P). Let $\mathsf{P} = \{\mathsf{P}_L\}_{L\in\mathcal{L}}$ be a predicate family where $\mathsf{P}_L : \mathcal{X}_L \times \mathcal{Y}_L \to \{0,1\}$. We define a predicate family $\mathsf{P}^{\mathrm{KP}}[\mathsf{P}] = \{\mathsf{P}_L^{\mathrm{KP}}[\mathsf{P}]\}_{L\in\mathcal{L}}$, where $\mathsf{P}_L^{\mathrm{KP}}[\mathsf{P}] : \mathcal{X}_L^{\mathrm{KP}} \times \mathcal{Y}_L^{\mathrm{KP}} \to \{0,1\}$ as follows:

$$\mathcal{X}_L^{\mathrm{KP}} = \mathcal{X}_L, \quad \mathcal{Y}_L^{\mathrm{KP}} = \{(\mathbf{M}, \rho) : d, s \in \mathbb{N},\ \mathbf{M} \in \mathbb{Z}_p^{d\times s},\ \rho : [d] \to \mathcal{Y}_L\}$$
$$\mathsf{P}_L^{\mathrm{KP}}[\mathsf{P}](x, (\mathbf{M}, \rho)) = 1 \Leftrightarrow (1, 0, \ldots, 0) \in \mathsf{RowSpan}(\{\mathbf{m}_h\}_{h:\mathsf{P}(x,\rho(h))=1})$$

The unbounded KP-ABE for span programs over P refers to the unbounded ABE for $\mathsf{P}^{\mathrm{KP}}[\mathsf{P}]$. We call $f = (\mathbf{M}, \rho)$ a span program and say that f accepts x if $\mathsf{P}_L^{\mathrm{KP}}[\mathsf{P}](x, f) = 1$. The size of f is the height of $\mathbf{M}$, i.e., d.

In Section 6.4, we explain how the above definition captures standard notions of small and large-universe KP-ABE for span programs, namely $\mathsf{P}^{\mathrm{KP}}[\mathsf{P}^{\mathrm{BE}}]$ and $\mathsf{P}^{\mathrm{KP}}[\mathsf{P}^{\mathrm{IBBE}}]$ respectively (Recall $\mathsf{P}^{\mathrm{BE}}, \mathsf{P}^{\mathrm{IBBE}}$ correspond to broadcast and identity-based broadcast, cf Definitions 5.2 and 5.3). Note that span programs capture boolean formula.

Lemma 6.1 (LSSS for Span Programs). *Let $\gamma \in \mathbb{N}$. We can construct a linear secret sharing scheme (LSSS) for span programs consisting of a probabilistic algorithm* Share *and a deterministic algorithm* sRec *as follows.*

$\mathsf{Share}(\mathbf{M}, \mathbf{k}; \mathbf{r}_1, \ldots, \mathbf{r}_{s-1})$**:** *It takes a program $\mathbf{M} \in \mathbb{Z}_p^{d\times s}$, a secret $\mathbf{k} \in \mathbb{Z}_p^\gamma$, and randomness $\mathbf{r}_1, \ldots, \mathbf{r}_{s-1} \in \mathbb{Z}_p^\gamma$, and outputs shares $\mathbf{k}_1, \ldots, \mathbf{k}_d$ as*

$$\begin{pmatrix} \mathbf{k}_1 \\ \vdots \\ \mathbf{k}_d \end{pmatrix} = \mathbf{M} \begin{pmatrix} \mathbf{k} \\ \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_{s-1} \end{pmatrix}$$

$\mathsf{sRec}(\mathbf{M}, S, \{\mathbf{k}_h\}_{h\in S})$**:** *It takes $\mathbf{M} \in \mathbb{Z}_p^{d\times s}$, $S \subseteq [d]$ such that $(1, 0, \ldots, 0) \in \mathsf{RowSpan}(\{\mathbf{m}_h\}_{h\in S})$, and shares $\{\mathbf{k}_h\}_{h\in S}$. It outputs $\mathbf{k}' = \sum_{h\in S} a_h \mathbf{k}_h$ where $\{a_h\}$ are coefficients satisfying $\sum_{h\in S} a_h \mathbf{m}_h = (1, 0, \ldots, 0)$.*

The above LSSS satisfies the following properties:

Linearity: $\mathsf{Share}(\mathbf{M}, \mathbf{k}; \mathbf{r}_1, \ldots, \mathbf{r}_{s-1})$ *is linear in $(\mathbf{k}, \mathbf{r}_1, \ldots, \mathbf{r}_{s-1})$, and $\mathsf{sRec}(\mathbf{M}, S, \{\mathbf{k}_h\}_{h\in S})$ is linear in $\{\mathbf{k}_h\}_{h\in S}$.*

Correctness: *For all $\mathbf{M} \in \mathbb{Z}_p^{d\times s}$, $S \subseteq [d]$ s.t. $(1, 0, \ldots, 0) \in \mathsf{RowSpan}(\{\mathbf{m}_h\}_{h\in S})$, and $\mathbf{r}_1, \ldots, \mathbf{r}_{s-1} \in \mathbb{Z}_p^\gamma$, we have $\mathsf{sRec}(\mathbf{M}, S, \{\mathbf{k}_h\}_{h\in S}) = \mathbf{k}$, where $\mathbf{k}_1, \ldots, \mathbf{k}_d \leftarrow \mathsf{Share}(\mathbf{M}, \mathbf{k}; \mathbf{r}_1, \ldots, \mathbf{r}_{s-1})$.*

Security: *For all $\mathbf{M} \in \mathbb{Z}_p^{d\times s}$, $S \subseteq [d]$ s.t. $(1, 0, \ldots, 0) \notin \mathsf{RowSpan}(\{\mathbf{m}_h\}_{h\in S})$, and $\mathbf{k} \in \mathbb{Z}_p^\gamma$, $\{\mathbf{k}_h\}_{h\in S}$ and $\{\mathbf{k}'_h\}_{h\in S}$ are identically distributed, where $\mathbf{k}_1, \ldots, \mathbf{k}_d \leftarrow \mathsf{Share}(\mathbf{M}, \mathbf{k})$ and $\mathbf{k}'_1, \ldots, \mathbf{k}'_d \leftarrow \mathsf{Share}(\mathbf{M}, \mathbf{0})$ over the uniform choice of the random vectors.*

In the security proof of our unbounded KP-ABE scheme for span programs, we use the following lemma regarding the above LSSS.

Lemma 6.2. *For $\mathbf{a} \in \mathbb{Z}_p^\gamma$, $S \subseteq [d]$, let $D_{S,\mathbf{a}}$ be the uniform distribution in $\{(u_1, \ldots, u_d) \otimes \mathbf{a} \mid u_h \in \mathbb{Z}_p, h \in S \Rightarrow u_h = 0\}$. Then, for all $\mathbf{a}, \mathbf{k} \in \mathbb{Z}_p^\gamma$, $\mathbf{M} \in \mathbb{Z}_p^{d \times s}$, $S \subseteq [d]$ s.t. $(1, 0, \ldots, 0) \notin \mathsf{RowSpan}(\{\mathbf{m}_h\}_{h \in S})$, the following distributions are identical:*

$$\mathsf{Share}(\mathbf{M}, \mathbf{k}; \mathbf{r}_1, \ldots, \mathbf{r}_{s-1}) + D_{S,\mathbf{a}} \quad \textit{and} \quad \mathsf{Share}(\mathbf{M}, \mathbf{k} + u\mathbf{a}; \mathbf{r}_1, \ldots, \mathbf{r}_{s-1}) + D_{S,\mathbf{a}}$$

where $u \leftarrow \mathbb{Z}_p$, $\mathbf{r}_1, \ldots, \mathbf{r}_{s-1} \leftarrow \mathbb{Z}_p^\gamma$, and we see the output of $\mathsf{Share}(\mathbf{M}, \cdot)$ as an element in $\mathbb{Z}_p^{d\gamma}$.

Proof. We have

$$\begin{aligned}
&\mathsf{Share}(\mathbf{M}, \mathbf{k}; \mathbf{r}_1, \ldots, \mathbf{r}_{s-1}) + D_{S,\mathbf{a}} \\
\equiv&\mathsf{Share}(\mathbf{M}, \mathbf{k}; \mathbf{r}_1, \ldots, \mathbf{r}_{s-1}) + \mathsf{Share}(\mathbf{M}, \mathbf{0}; u_1\mathbf{a}, \ldots, u_{s-1}\mathbf{a}) + D_{S,\mathbf{a}} \\
=&\mathsf{Share}(\mathbf{M}, \mathbf{k}; \mathbf{r}_1, \ldots, \mathbf{r}_{s-1}) + \mathsf{Share}(\mathbf{M}, 0; u_1, \ldots, u_{s-1}) \otimes \mathbf{a} + D_{S,\mathbf{a}} \\
\equiv&\mathsf{Share}(\mathbf{M}, \mathbf{k}; \mathbf{r}_1, \ldots, \mathbf{r}_{s-1}) + \mathsf{Share}(\mathbf{M}, u; u_1, \ldots, u_{s-1}) \otimes \mathbf{a} + D_{S,\mathbf{a}} \\
=&\mathsf{Share}(\mathbf{M}, \mathbf{k}; \mathbf{r}_1, \ldots, \mathbf{r}_{s-1}) + \mathsf{Share}(\mathbf{M}, u\mathbf{a}; u_1\mathbf{a}, \ldots, u_{s-1}\mathbf{a}) + D_{S,\mathbf{a}} \\
\equiv&\mathsf{Share}(\mathbf{M}, \mathbf{k} + u\mathbf{a}; \mathbf{r}_1, \ldots, \mathbf{r}_{s-1}) + D_{S,\mathbf{a}}
\end{aligned}$$

where $u, u_1, \ldots, u_{s-1} \leftarrow \mathbb{Z}_p$. The first and the third equivalence, the first and the second equality follow from the linearity of Share and the fact that $\mathbf{r}_1, \ldots, \mathbf{r}_{s-1}$ are uniformly distributed in $\mathbb{Z}_p^\gamma$. The second equivalence follows from the security of the LSSS and the fact that $\{u_h\}_{h \notin S}$ are masked by $D_{S,\mathbf{a}}$. □

6.2 Construction

We construct our unbounded KP-ABE scheme for span programs over P, i.e., an unbounded ABE scheme for the predicate $\mathsf{P}_L^{\mathrm{KP}}[\mathsf{P}]$ (Definition 6.1) using the unbounded ABE scheme for P in Section 4.1 in a semi-generic way as follows. Let $(\mathsf{Share}, \mathsf{sRec})$ be a LSSS for span programs in Lemma 6.1. Let $(\mathsf{Setup}_0, \mathsf{Enc}_0, \mathsf{KeyGen}_0, \mathsf{Dec}_0)$ be the algorithms of the unbounded ABE scheme for P in Section 4.1 with the following modifications:

- $\mathsf{KeyGen}_0[\mathbf{k}']$ is the same as key generation in the unbounded ABE scheme except that it uses $\mathbf{k}'$ in place of $\mathbf{k}$ in msk;
- Dec_0 is the same as decryption in the unbounded ABE scheme except that it outputs $d(z_1, z_2)$ instead of $\mathsf{ct}_5/d(z_1, z_2)$ (thus Dec_0 works even if ct does not contain ct_5).

The vector/matrix symbols in the components of mpk, msk, ct, sk follow Section 4.1.

$\mathsf{Setup}(1^\lambda)$**:** It outputs $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}_0(1^\lambda)$.

$\mathsf{Enc}(\mathsf{mpk}, (x, L), M)$**:** It outputs $\mathsf{ct} = (\mathsf{ct}_0, \mathsf{ct}_5) \leftarrow \mathsf{Enc}_0(\mathsf{mpk}, (x, L), M)$ where ct_0 includes all the ciphertext elements except ct_5 as per Eq. (9).

$\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, (\mathbf{M}, \rho, L))$**:** It extracts $\mathbf{k}$ from msk, runs $\mathbf{k}_1, \ldots, \mathbf{k}_d \leftarrow \mathsf{Share}(\mathbf{M}, \mathbf{k})$, and outputs $\mathsf{sk} = \{\mathsf{sk}_h\}_{h \in [d]}$ where

$$\mathsf{sk}_h \leftarrow \mathsf{KeyGen}_0[\mathbf{k}_h](\mathsf{mpk}, \mathsf{msk}, (\rho(h), L))$$

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{ct}, \mathsf{sk}, x, (\mathbf{M}, \rho), L)$**:** It takes mpk, ct, sk, x, $(\mathbf{M}, \rho)$, L such that $(\mathbf{M}, \rho)$ accepts x, and parses $\mathsf{ct} = (\mathsf{ct}_0, \mathsf{ct}_5)$. Let $S = \{h \in [d] \mid \mathsf{P}(x, \rho(h)) = 1\}$. It computes

$$[z_h]_T \leftarrow \mathsf{Dec}_0(\mathsf{mpk}, \mathsf{ct}_0, \mathsf{sk}_h, x, \rho(h), L) \quad \text{for } h \in S$$

and outputs $\mathsf{ct}_5/\mathsf{sRec}(\mathbf{M}, S, \{[z_h]_T\}_{h \in S})$.

Correctness. In decryption, we have $z_h = \mathbf{k}_h \mathbf{A}_1 \mathbf{s}_0^\top$, which follows from the correctness of the unbounded ABE scheme in Section 4.1. Recall that we have $(1, 0, \ldots, 0) \in \mathsf{RowSpan}(\{\mathbf{m}_h\}_{h \in S})$, which follows from the condition that $(\mathbf{M}, \rho)$ accepts x. Thus, we have

$$\mathsf{ct}_5/\mathsf{sRec}(\mathbf{M}, S, \{[z_h]_T\}_{h \in S}) = [\mathbf{k}\mathbf{A}_1\mathbf{s}_0^\top]_T \cdot M/[\mathbf{k}\mathbf{A}_1\mathbf{s}_0^\top]_T = M$$

where the first equality follows from the correctness and linearity of the LSSS.

Parameter Sizes. In the above construction, the sizes of $\mathsf{mpk}, \mathsf{ct}, \mathsf{sk}$ are as follows:

$$|\mathsf{mpk}| = O(1), \quad |\mathsf{ct}| = O(n + m_1), \quad |\mathsf{sk}| = O(|f| \cdot (n + m_2))$$

where $n = \mathsf{p}(L), m_1, m_2$ are the number of $\mathbb{Z}_p$ elements in $c(\mathbf{w}), k(\alpha, \mathbf{w})$, respectively. The above sizes hide $\mathsf{poly}(\lambda)$ factors.

6.3 Security

Theorem 6.1. *Suppose the MDDH assumption holds and the linear CDS Γ satisfies unbounded security. Then, our scheme in Section 6.2 satisfies semi-adaptive unbounded security as per Definition 3.3.*

Proof. We define various distributions of ciphertext and secret key. As global parameters, we sample $\mathbf{A}_2, \mathbf{A}_3 \leftarrow \mathbb{Z}_p^{3k \times k}$, define $\mathbf{A}_1^{\parallel}, \mathbf{A}_2^{\parallel}, \mathbf{A}_3^{\parallel}$ following Definition 3.5, and also sample $\mathbf{W}_i, \mathbf{W}'_i, \mathbf{U}'_i \leftarrow \mathsf{RowSpan}(\mathbf{A}_2^{\parallel\top})^{(k+1)\times 1}$ for $i \in [n']$ where n' is sufficiently large. Let $\mathbf{a}_{2,1}^{\parallel\top}$ be the first column of $\mathbf{A}_2^{\parallel}$. For ciphertexts, we define Normal^{KP} and E-normal^{KP} in the same way as Normal and E-normal in Section 4.2. For a secret key on $((\mathbf{M}, \rho), L)$ such that $\mathbf{M} \in \mathbb{Z}_p^{d \times s}$ and $\rho : [d] \to \mathcal{Y}_L$, we consider the following distributions. Since we are considering semi-adaptive security, secret key distributions can depend on the challenge attribute (x, L^*). In the following, let $\mathsf{Cond}^*(L, h)$ be the condition such that $L \neq L^* \vee \mathsf{P}_L(x, \rho(h)) = 0$.

- Normal^{KP}: Same as the output of KeyGen in Section 6.2.
- E-normal^{KP}: Same as the output of KeyGen in Section 6.2 except that sk_h is generated as

$$\mathsf{sk}_h \leftarrow \mathsf{KeyGen}_0^{\mathsf{EN}}[\mathbf{k}_h](\mathsf{mpk}, \mathsf{msk}, (\rho(h), L)) \tag{17}$$

  for $h \in [d]$ where $\mathsf{KeyGen}_0^{\mathsf{EN}}[\mathbf{k}_h]$ outputs E-normal key distribution (Eq. (12)) with $\mathbf{k}_h$ in place of $\mathbf{k}$.
- Pre-SF^{KP}: It runs $\mathbf{z}_1, \dots, \mathbf{z}_d \leftarrow \mathsf{RowSpan}(\mathbf{a}_{2,1}^{\parallel})$, $\mathbf{k}_1, \dots, \mathbf{k}_d \leftarrow \mathsf{Share}(f, \mathbf{k})$, and outputs $\mathsf{sk} = \{\mathsf{sk}_h\}_{h \in [d]}$ where

$$\mathsf{sk}_h \leftarrow \begin{cases} \mathsf{KeyGen}_0^{\mathsf{EN}}[\mathbf{k}_h + \mathbf{z}_h](\mathsf{mpk}, \mathsf{msk}, (\rho(h), L)) & (\mathsf{Cond}^*(L, h)) \\ \mathsf{KeyGen}_0^{\mathsf{EN}}[\mathbf{k}_h](\mathsf{mpk}, \mathsf{msk}, (\rho(h), L)) & (\text{otherwise}) \end{cases} \tag{18}$$

- SF^{KP}: It runs $\mathbf{z}, \mathbf{z}_1, \dots, \mathbf{z}_d \leftarrow \mathsf{RowSpan}(\mathbf{a}_{2,1}^{\parallel})$, $\mathbf{k}_1, \dots, \mathbf{k}_d \leftarrow \mathsf{Share}(f, \mathbf{k} + \mathbf{z})$, and outputs $\mathsf{sk} = \{\mathsf{sk}_h\}_{h \in [d]}$ where

$$\mathsf{sk}_h \leftarrow \begin{cases} \mathsf{KeyGen}_0^{\mathsf{EN}}[\mathbf{k}_h + \mathbf{z}_h](\mathsf{mpk}, \mathsf{msk}, (\rho(h), L)) & (\mathsf{Cond}^*(L, h)) \\ \mathsf{KeyGen}_0^{\mathsf{EN}}[\mathbf{k}_h](\mathsf{mpk}, \mathsf{msk}, (\rho(h), L)) & (\text{otherwise}) \end{cases} \tag{19}$$

  Note that $\mathbf{z}, \mathbf{z}_1, \dots, \mathbf{z}_d$ are chosen independently in each key generation for Pre-SF and SF.

We prove the theorem via a series of hybrids. Let Q be the maximum number of secret key queries by the adversary.

- H_0: The original security game.
- H_1: Same as H_0 except that the challenge ciphertext and all the secret keys given to the adversary are E-normal^{KP}.
- H_2: Same as H_1 except that all the secret keys given to the adversary are Pre-SF^{KP}.
- H_3: Same as H_2 except that all the secret keys given to the adversary are SF^{KP}.
- H_4: Same as H_3 except that ct_5 in the challenge ciphertext is randomly chosen from G_T.

We prove the indistinguishability of each hybrid transition in Lemma 6.3 to Lemma 6.6. Observe that the adversary's view is independent of the challenge bit β and its advantage is 0 in H_4. This concludes the proof. □

Lemma 6.3. $\mathsf{H}_0 \approx_c \mathsf{H}_1$ *under the bilinear entropy expansion lemma.*

Proof. The proof is almost the same as Lemma 4.1. Observe that each key generation of Normal^{KP} just invokes $\mathsf{KeyGen}_0[\mathbf{k}_h]$ d times while that of E-normal^{KP} invokes $\mathsf{KeyGen}_0^{\mathsf{EN}}[\mathbf{k}_h]$ d times. Thus, it is sufficient to just change $\sum_{j\in[Q]} d^{(j)}$ keys from Normal (Eq.(10)) to E-normal (Eq.(12)) instead of Q keys similarly to Lemma 4.1. Here, we use the fact that the proof of Lemma 4.1 works even if $\mathbf{k}$ is different vectors in each secret key, since $\mathbf{k}$ is sampled by the reduction. □

Lemma 6.4. $\mathsf{H}_1 \approx_c \mathsf{H}_2$ *under MDDH.*

Proof. In this step, we change all the secret keys from E-normal^{KP} to Pre-SF^{KP}. Observe that E-normal^{KP} key generation only involves $\mathsf{KeyGen}_0^{\mathsf{EN}}[\mathbf{k}_h]$, which is distributed as E-normal (Eq. (12)), except that $\mathbf{k}$ is replaced with $\mathbf{k}_h$. Similarly, in Pre-SF^{KP} key generation, $\mathsf{KeyGen}_0^{\mathsf{EN}}[\mathbf{k}_h + \mathbf{z}_h]$ is distributed as SF (Eq. (13)), and $\mathsf{KeyGen}_0^{\mathsf{EN}}[\mathbf{k}_h]$ is distributed as E-normal (Eq.(12)). Thus, for changing all the keys from E-normal^{KP} to Pre-SF^{KP}, it is sufficient to change multiple keys of the underlying scheme that cannot decrypt the challenge ciphertext (i.e., sk_h that satisfies $\mathsf{Cond}^*(L,h)$) from E-normal (Eq. (12)) to SF (Eq. (13)) in the presence of E-normal keys that can decrypt the challenge ciphertext (i.e., sk_h that does not satisfy $\mathsf{Cond}^*(L,h)$).

We can show the indistinguishability of the above change similarly to the proof of $\mathsf{H}_1 \approx_c \mathsf{H}_{2,Q,3}$ in Theorem 4.1, where we change all the secret keys of the underlying scheme, which cannot decrypt the challenge ciphertext, from E-normal (Eq.(12)) to SF (Eq.(13)). During this process, the reduction can always simulate any E-normal keys that can decrypt the challenge ciphertext because any E-normal keys can be efficiently sampled even given $\mathbf{B}$ as a group element. Furthermore, this process works even if $\mathbf{k}$ used in each key generation is distinct vectors. Thus, the proof of Lemma 6.4 is essentially the same as the combination of Lemmas 4.3 to 4.5. □

Lemma 6.5. $\mathsf{H}_2 \equiv \mathsf{H}_3$.

Proof. The difference between H_2 and H_3 is whether all the keys are Pre-SF^{KP} or SF^{KP}. More specifically, in each key generation, the difference lies in whether $\mathbf{k}_1, \dots, \mathbf{k}_d$ are shares of $\mathbf{k}$ or $\mathbf{k}+\mathbf{z}$ where $\mathbf{z}$ is a fresh random vector in $\mathsf{RowSpan}(\mathbf{a}_{2,1}^{\parallel})$. Recall that, in each key generation, the shares $\mathbf{k}_1, \dots, \mathbf{k}_d$ are masked by a random vector $\mathbf{z}_h \leftarrow \mathsf{RowSpan}(\mathbf{a}_{2,1}^{\parallel})$ for all $h \in [d]$ such that $\mathsf{Cond}^*(L,h)$ is true, i.e., $L \neq L^* \vee \mathsf{P}_L(x,\rho(h)) = 0$.

For the secret keys for $(\mathbf{M},\rho,L)$ such that $L = L^*$, $\mathbf{k}_1, \dots, \mathbf{k}_d$ are masked by $\mathbf{z}_h$ for all $h : \mathsf{P}_L(x,\rho(h)) = 0$. Due to the admissibility of unbounded ABE game, we have $(1,0,\dots,0) \notin \mathsf{RowSpan}(\{\mathbf{m}_h\}_{h:\mathsf{P}_L(x,\rho(h))=1})$. Thus, thanks to Lemma 6.2 with $\mathbf{a} = \mathbf{a}_{2,1}^{\parallel}$ and $S = \{h \in [d] \mid \mathsf{P}_L(x,\rho(h)) = 1\}$, the distributions of $\mathsf{sk}_1, \dots, \mathsf{sk}_d$ of the keys of this case in H_2 and H_3 are equivalent.

For the secret keys $(\mathbf{M},\rho,L)$ such that $L \neq L^*$, $\mathbf{k}_1, \dots, \mathbf{k}_d$ are masked by $\mathbf{z}_h$ for all $h \in [d]$. Thanks to Lemma 6.2 with $\mathbf{a} = \mathbf{a}_{2,1}^{\parallel}$ and $S = \emptyset$, the distributions of $\mathsf{sk}_1, \dots, \mathsf{sk}_d$ of the keys of this case in H_2 and H_3 are equivalent. □

Lemma 6.6. $\mathsf{H}_3 \approx_s \mathsf{H}_4$.

Proof. The proof is similar to that of Lemma 4.6. To see this, we program $\mathbf{k}$ as $\mathbf{k} = \hat{\mathbf{k}} + \gamma \mathbf{a}_{2,1}^{\parallel}$ where $\hat{\mathbf{k}} \leftarrow \mathbb{Z}_p^{3k}$ and $\gamma \leftarrow \mathbb{Z}_p$. Observe that mpk and all sk do not reveal any information on γ because $\mathbf{a}_{2,1}^{\parallel}\mathbf{A} = \mathbf{0}$ for mpk, and $\mathbf{k}$ in each sk is masked by a fresh random vector $\mathbf{z}$. For ct, $\mathbf{a}_{2,1}^{\parallel}\mathbf{c}_0^{\top} \neq 0$ with overwhelming probability, and thus $\mathbf{k}\mathbf{c}_0^{\top} = \hat{\mathbf{k}}\mathbf{c}_0^{\top} + \gamma \mathbf{a}_{2,1}^{\parallel}\mathbf{c}_0^{\top}$ is statistically close to uniform in $\mathbb{Z}_p$ and independent of other elements. Thus, H_3 and H_4 are statistically close. □

6.4 Instantiations

We can obtain unbounded KP-ABE schemes for span programs with new parameters from the construction in Section 6.2.

Definition 6.2 (Unbounded KP-ABE for Span Programs). Small-universe unbounded KP-ABE for span programs is defined as unbounded ABE for $\mathsf{P}^{\text{S-KP}} = \{\mathsf{P}_L\}_{L\in\mathcal{L}}$ where $\mathsf{P}_L : \mathcal{X}_L \times \mathcal{Y}_L \to \{0,1\}$ such that

$$\mathcal{L} = \mathbb{N}, \quad \mathcal{X}_N = 2^{[N]}, \quad \mathcal{Y}_N = \{(\mathbf{M},\rho) : d, s \in \mathbb{N},\ \mathbf{M} \in \mathbb{Z}_p^{d\times s},\ \rho : [d] \to [N]\}$$
$$\mathsf{P}_N(X,(\mathbf{M},\rho)) = 1 \Leftrightarrow (1,0,\ldots,0) \in \mathsf{RowSpan}(\{\mathbf{m}_h\}_{h:\rho(h)\in X})$$

where $N \in \mathbb{N}$, $X \in \mathcal{X}_N$, $(\mathbf{M},\rho) \in \mathcal{Y}_N$. Similarly, large-universe unbounded KP-ABE for span programs is defined as unbounded ABE for $\mathsf{P}^{\text{L-KP}} = \{\mathsf{P}_L\}_{L\in\mathcal{L}}$ where $\mathsf{P}_L : \mathcal{X}_L \times \mathcal{Y}_L \to \{0,1\}$ such that

$$\mathcal{L} = \mathbb{N}, \quad \mathcal{X}_N = \{X \subseteq \mathbb{Z}_p\backslash\{0\} \mid |S| \le N\}$$
$$\mathcal{Y}_N = \{(\mathbf{M},\rho) : d, s \in \mathbb{N},\ \mathbf{M} \in \mathbb{Z}_p^{d\times s},\ \rho : [d] \to \mathbb{Z}_p\backslash\{0\}\}$$
$$\mathsf{P}_N(X,(\mathbf{M},\rho)) = 1 \Leftrightarrow (1,0,\ldots,0) \in \mathsf{RowSpan}(\{\mathbf{m}_h\}_{h:\rho(h)\in X})$$

where p is an exponentially large modulus.

Our Unbounded KP-ABE for Span Programs. It is not hard to see that $\mathsf{P}^{\text{S-KP}} = \mathsf{P}^{\text{KP}}[\mathsf{P}^{\text{BE}}]$ and $\mathsf{P}^{\text{L-KP}} = \mathsf{P}^{\text{KP}}[\mathsf{P}^{\text{IBBE}}]$ (see Definitions 5.2, 5.3 and 6.1 for the definitions of these predicates). Thus, we can instantiate small/large-universe unbounded ABE for span programs by applying the linear CDS for P^{NQP}/ P^{ZQP} to the KP-ABE scheme in Section 6.2, since we can embed P^{BE}/P^{IBBE} in P^{NQP}/P^{ZQP}, respectively. The resulting schemes have the following parameters:

$$|\mathsf{mpk}| = O(1), \quad |\mathsf{ct}| = O(\sqrt{N}), \quad |\mathsf{sk}| = O(\sqrt{N}\cdot|f|)$$

where $|f|$ is the height of $\mathbf{M}$ for $f = (\mathbf{M},\rho)$.

References

ABS17. Miguel Ambrona, Gilles Barthe, and Benedikt Schmidt. Generic transformations of predicate encodings: Constructions and applications. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 36–66. Springer, Cham, August 2017.

AC17. Shashank Agrawal and Melissa Chase. Simplifying design and analysis of complex predicate encryption schemes. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 627–656. Springer, Cham, April / May 2017.

AHM+16. Nuttapong Attrapadung, Goichiro Hanaoka, Tsutomu Matsumoto, Tadanori Teruya, and Shota Yamada. Attribute based encryption with direct efficiency tradeoff. In Mark Manulis, Ahmad-Reza Sadeghi, and Steve Schneider, editors, *ACNS 2016*, volume 9696 of *LNCS*, pages 249–266. Springer, Cham, June 2016.

AKY24. Shweta Agrawal, Simran Kumari, and Shota Yamada. Compact pseudorandom functional encryption from evasive LWE. Cryptology ePrint Archive, Paper 2024/1719, 2024.

ALdP11. Nuttapong Attrapadung, Benoît Libert, and Elie de Panafieu. Expressive key-policy attribute-based encryption with constant-size ciphertexts. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *PKC 2011*, volume 6571 of *LNCS*, pages 90–108. Springer, Berlin, Heidelberg, March 2011.

AT20. Nuttapong Attrapadung and Junichi Tomida. Unbounded dynamic predicate compositions in ABE from standard assumptions. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part III*, volume 12493 of *LNCS*, pages 405–436. Springer, Cham, December 2020.

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

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

Att19. Nuttapong Attrapadung. Unbounded dynamic predicate compositions in attribute-based encryption. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part I*, volume 11476 of *LNCS*, pages 34–67. Springer, Cham, May 2019.

AWY20. Shweta Agrawal, Daniel Wichs, and Shota Yamada. Optimal broadcast encryption from LWE and pairings in the standard model. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part I*, volume 12550 of *LNCS*, pages 149–178. Springer, Cham, November 2020.

AY20. Shweta Agrawal and Shota Yamada. Optimal broadcast encryption from pairings and LWE. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 13–43. Springer, Cham, May 2020.

BB04. Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer, Berlin, Heidelberg, May 2004.

BGW05. Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 258–275. Springer, Berlin, Heidelberg, August 2005.

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

BV22. Zvika Brakerski and Vinod Vaikuntanathan. Lattice-inspired broadcast encryption and succinct ciphertext-policy ABE. In Mark Braverman, editor, *ITCS 2022*, volume 215, pages 28:1–28:20. LIPIcs, January / February 2022.

BW06. Dan Boneh and Brent Waters. A fully collusion resistant broadcast, trace, and revoke system. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 211–220. ACM Press, October / November 2006.

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

CGW15. Jie Chen, Romain Gay, and Hoeteck Wee. Improved dual system ABE in prime-order groups via predicate encodings. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 595–624. Springer, Berlin, Heidelberg, April 2015.

CMM16. Melissa Chase, Mary Maller, and Sarah Meiklejohn. Déjà Q all over again: Tighter and broader reductions of q-type assumptions. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 655–681. Springer, Berlin, Heidelberg, December 2016.

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

CW24. Valerio Cini and Hoeteck Wee. Unbounded ABE for circuits from LWE, revisited. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part IV*, volume 15487 of *LNCS*, pages 238–267. Springer, Singapore, December 2024.

Del07. Cécile Delerablée. Identity-based broadcast encryption with constant size ciphertexts and private keys. In Kaoru Kurosawa, editor, *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 200–215. Springer, Berlin, Heidelberg, December 2007.

EHK+17. Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Luis Villar. An algebraic framework for Diffie-Hellman assumptions. *Journal of Cryptology*, 30(1):242–288, January 2017.

FN94. Amos Fiat and Moni Naor. Broadcast encryption. In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 480–491. Springer, Berlin, Heidelberg, August 1994.

GDCC16. Junqing Gong, Xiaolei Dong, Jie Chen, and Zhenfu Cao. Efficient IBE with tight reduction to standard assumption in the multi-challenge setting. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 624–654. Springer, Berlin, Heidelberg, December 2016.

GIKM98. Yael Gertner, Yuval Ishai, Eyal Kushilevitz, and Tal Malkin. Protecting data privacy in private information retrieval schemes. In *30th ACM STOC*, pages 151–160. ACM Press, May 1998.

GKW15. Romain Gay, Iordanis Kerenidis, and Hoeteck Wee. Communication complexity of conditional disclosure of secrets and attribute-based encryption. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 485–502. Springer, Berlin, Heidelberg, August 2015.

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

GKW18. Romain Gay, Lucas Kowalczyk, and Hoeteck Wee. Tight adaptively secure broadcast encryption with short ciphertexts and keys. In Dario Catalano and Roberto De Prisco, editors, *SCN 18*, volume 11035 of *LNCS*, pages 123–139. Springer, Cham, September 2018.

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

GW09. Craig Gentry and Brent Waters. Adaptive security in broadcast encryption systems (with short ciphertexts). In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 171–188. Springer, Berlin, Heidelberg, April 2009.

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

KW19. Lucas Kowalczyk and Hoeteck Wee. Compact adaptively secure ABE for NC^1 from k-Lin. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part I*, volume 11476 of *LNCS*, pages 3–33. Springer, Cham, May 2019.

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

LVW17. Tianren Liu, Vinod Vaikuntanathan, and Hoeteck Wee. Conditional disclosure of secrets via non-linear reconstruction. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 758–790. Springer, Cham, August 2017.

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

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

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

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

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

Wee16. Hoeteck Wee. Déjà Q: Encore! Un petit IBE. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part II*, volume 9563 of *LNCS*, pages 237–258. Springer, Berlin, Heidelberg, January 2016.

Wee21. Hoeteck Wee. Broadcast encryption with size $N^{1/3}$ and more from k-lin. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 155–178, Virtual Event, August 2021. Springer, Cham.

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

Wee24. Hoeteck Wee. Circuit ABE with $\mathsf{poly}(\mathsf{depth}, \lambda)$-sized ciphertexts and keys from lattices. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part III*, volume 14922 of *LNCS*, pages 178–209. Springer, Cham, August 2024.

Wee25. Hoeteck Wee. Almost optimal KP and CP-ABE for circuits from succinct LWE. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part III*, volume 15603 of *LNCS*, pages 34–62. Springer, Cham, May 2025.

WW25. Hoeteck Wee and David J. Wu. Unbounded distributed broadcast encryption and registered ABE from succinct LWE. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part III*, volume 16002 of *LNCS*, pages 204–235. Springer, Cham, August 2025.