Improved Dual System ABE in Prime-Order Groups via Predicate Encodings

Jie Chen[1,⋆], Romain Gay[2,⋆⋆], and Hoeteck Wee[2,⋆⋆⋆]

[1] East China Normal University, Shanghai, China
s080001@e.ntu.edu.sg

[2] ENS, Paris, France
rgay,wee@di.ens.fr

Abstract. We present a modular framework for the design of efficient adaptively secure attribute-based encryption (ABE) schemes for a large class of predicates under the standard k-Lin assumption in prime-order groups; this is the first uniform treatment of dual system ABE across different predicates and across both composite and prime-order groups. Via this framework, we obtain concrete efficiency improvements for several ABE schemes. Our framework has three novel components over prior works: (i) new techniques for simulating composite-order groups in prime-order ones, (ii) a refinement of prior encodings framework for dual system ABE in composite-order groups, (iii) an extension to weakly attribute-hiding predicate encryption (which includes anonymous identity-based encryption as a special case).

1 Introduction

Attribute-based encryption (ABE) [39, 22] is a new paradigm for public-key encryption that enables fine-grained access control for encrypted data. In ABE, ciphertexts are associated with descriptive values x in addition to a plaintext, secret keys are associated with values y, and a secret key decrypts the ciphertext if and only if $\mathsf{P}(x, y) = 1$ for some boolean predicate P. Here, y together with P may express an arbitrarily complex access policy, which is in stark contrast to traditional public-key encryption, where access is all or nothing. The simplest example of ABE is that of identity-based encryption (IBE) [40, 7, 17] where P corresponds to equality. The security requirement for ABE enforces resilience to collusion attacks, namely any group of users holding secret keys for different values learns nothing about the plaintext if none of them is individually authorized to decrypt the ciphertext. This should hold even if the adversary *adaptively* decides which secret keys to ask for.

ABE in Prime-Order Groups. The goal of this work is to obtain efficient adaptively secure ABE for a large class of predicates. We now have a fairly good understanding of how to obtain such schemes in composite-order bilinear groups, thanks to Waters' powerful *dual system encryption methodology* [42] and recent unifying frameworks in [2, 43] for the design of dual system ABE schemes. However, these latter frameworks only work in composite-order bilinear groups, for which group operations and especially pairing computations are prohibitively slow. In practice, prime-order bilinear groups are preferable [23] as they admit not only more efficient but also more compact instantiations. To mitigate the gap between ease of theoretical design and practical efficiency, a series of works studied techniques for converting cryptosystems relying on composite-order groups to cryptosystems based on prime-order groups [34, 35, 20, 30, 15, 14], largely in the context of dual system ABE. In addition, we have direct constructions of dual system prime-order hierarchical identity-based encryption (HIBE) schemes in [26, 5] that bypass a conversion from composite-order groups, but the techniques in these constructions do not seem to

[⋆] Shanghai Key Laboratory of Multidimensional Information Processing and Shanghai Key Lab of Trustworthy Computing. Supported by the National Natural Science Foundation of China (Grant Nos. 61472142, 61321064, 61172085), Science and Technology Commission of Shanghai Municipality (Grant Nos. 14YF1404200, 13JC1403500).

[⋆⋆] Supported in part by ANR-14-CE28-0003 (Project EnBiD).

[⋆⋆⋆] CNRS, INRIA and Columbia University. Supported in part by ANR-14-CE28-0003 (Project EnBiD), NSF Award CNS-1445424, the Alexander von Humboldt Foundation and a Google Faculty Research Award.

	compact HIBE	boolean formula	k-Lin	anonymous IBE	wAH ZIPE
DPVS [34, 35, 30, 15]	no	yes	yes	yes	yes
sparse DPVS [37]	yes	?	?	yes	yes
QANIZK [26]	yes	?	yes	yes	?
dual system groups [14]	yes	?	yes	?	?
MAC-to-(H)IBE [5]	yes	?	yes	yes	?
this work	yes	yes	yes	yes	yes

Fig. 1. Summary of previous approaches for building efficient dual system (H)IBE and ABE in prime-order groups. The first column refers to HIBE with constant-size ciphertexts; the second refers to KP/CP-ABE for boolean formula. The third column refers to instantiations from the general k-Lin assumption. The last two columns address extensions to stronger notions of security like anonymity and weakly attribute-hiding inner product encryption (wAH ZIPE). Additional discussion is provided in Section 1.2.

naturally extend beyond (H)IBE. Furthermore, the prior constructions rely on fairly distinct techniques, and efficiency improvements in one construction do not necessarily translate to a different construction and a different predicate. In short, prior works fall short of providing a unifying and modular framework for the design of efficient dual system ABE schemes in prime-order groups that work for a large class of predicates (c.f. Fig. 1).

1.1 Our contributions

We present a modular framework for the design of efficient dual system ABE schemes for a large class of predicates under the standard k-Lin assumption in prime-order groups; this is the first uniform treatment of dual system ABE across different predicates *and* across both composite and prime-order groups. Via this framework, we obtain concrete efficiency improvements for several ABE schemes. Our framework has three novel components over prior works: (i) new techniques for simulating composite-order groups in prime-order ones, (ii) a refinement of the encodings framework for dual system ABE for composite-order groups in [2, 43], (iii) an extension to weakly attribute-hiding predicate encryption [28, 9] (which includes anonymous IBE as a special case). The last two components answer the open problems left in [2, 43].

New techniques for simulating composite-order groups. The starting point of our construction is simply a simpler choice of basis. Fix a bilinear group (G_1, G_2, G_T) with $e : G_1 \times G_2 \rightarrow G_T$ of prime order p. We pick random matrices $(\mathbf{A}, \mathbf{B}) \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(k+1)\times k}$, along with random vectors $\mathbf{a}^{\perp}, \mathbf{b}^{\perp} \in \mathbb{Z}_p^{k+1}$ so that $\mathbf{a}^{\perp\top}\mathbf{A} = \mathbf{b}^{\perp\top}\mathbf{B} = \mathbf{0}$, and we assume $\mathbf{a}^{\perp\top}\mathbf{b}^{\perp} \neq 0$. Observe that[1]

$$([\mathbf{A}]_1, [\mathbf{b}^{\perp}]_1) := (g_1^{\mathbf{A}}, g_1^{\mathbf{b}^{\perp}}) \in G_1^{(k+1)\times k} \times G_1^{k+1}$$

forms a basis for G_1^{k+1}. Similarly,

$$([\mathbf{B}]_2, [\mathbf{a}^{\perp}]_2) := (g_2^{\mathbf{B}}, g_2^{\mathbf{a}^{\perp}}) \in G_2^{(k+1)\times k} \times G_2^{k+1}$$

forms a basis for G_2^{k+1}. In the context of dual system encryption, we use $[\mathbf{A}]_1$ as a basis for normal components in the ciphertext space, and $[\mathbf{b}^{\perp}]_1$ as a basis for semi-functional components. Similarly, we use $[\mathbf{B}]_2$ as a basis for normal components in the secret key space, and $[\mathbf{a}^{\perp}]_2$ as a basis for semi-functional components. Indistinguishability for elements with and without random semi-functional components follow readily from the k-Lin assumption. Moreover, we have an orthogonality property given by $\mathbf{a}^{\perp\top}\mathbf{A} = \mathbf{b}^{\perp\top}\mathbf{B} = \mathbf{0}$, which tells us that the normal and semi-functional components in different spaces cancel out.

[1] Following [18], we use the implicit representation notation for group elements, as explained in Section 4.1.

We can then randomize this basis by choosing $\mathbf{W} \in \mathbb{Z}_p^{(k+1)\times(k+1)}$ uniformly at random and using $([\mathbf{W}^\top\mathbf{A}]_1, [\mathbf{W}^\top\mathbf{b}^\perp]_1)$ for G_1^{k+1} and $([\mathbf{W}\mathbf{B}]_2, [\mathbf{W}\mathbf{a}^\perp]_2)$ for G_2^{k+1}. For decryption correctness, we will exploit the following "associative" property when the new basis interacts with the original one, namely:

$$e([\mathbf{A}]_1, [\mathbf{W}\mathbf{B}]_2) = e([\mathbf{W}^\top\mathbf{A}]_1, [\mathbf{B}]_2) \tag{1}$$

where we define the pairing operation on matrices via $e([\mathbf{M}]_1, [\mathbf{M}']_2) := e(g_1, g_2)^{\mathbf{M}^\top\mathbf{M}'}$. Observe that $\mathbf{W}$ has one unit of residual entropy given $(\mathbf{W}^\top\mathbf{A}, \mathbf{W}\mathbf{B})$. This will be crucial for carrying out the information-theoretic argument in the proof of ABE security via the dual system encryption methodology [42, 2, 43].

We note that prior transformations in prime-order groups in [20, 34, 35, 30] try to simulate all of the structure in composite-order groups (e.g. orthogonality). We simulate less structure (associativity, c.f. Eqn. (1)), thus leading to better concrete efficiency. However, when combined with the existing encodings framework for dual system ABE schemes in composite-order groups, we cannot even guarantee ABE decryption correctness. We compensate for less structure while simulating composite-order groups by imposing more structure to the encodings, which we can achieve without increasing the size of the encodings. We will exploit the additional structure in the encodings for correctness and for security. We now proceed to describe our encodings framework for ABE.

Modular approach for ABE. We begin with the observation that the prior composite-order ABE schemes in [43, 2] (generalizing [32, 33]) may be modified so that master public key, secret key and ciphertext are of the form:

$$\begin{aligned} \mathsf{mpk} &:= \big(g_1,\, g_1^{\mathbf{w}},\, e(g_1,g_1)^{\alpha}\big) \\ \mathsf{sk}_y &:= \big(g_1^r,\, g_1^{\mathsf{kE}(y,\alpha)+r\cdot\mathsf{rE}(y,\mathbf{w})}\big) \\ \mathsf{ct}_x &:= \big(g_1^s,\, g_1^{s\cdot\mathsf{sE}(x,\mathbf{w})},\, e(g_1,g_1)^{\alpha s}\cdot m\big) \end{aligned} \tag{2}$$

Here, g_1 is a generator of order p_1 where the underlying composite group order is the product of three primes p_1, p_2, p_3 (for simplicity we consider the case of a symmetric bilinear group); $\mathbf{w}$ is a vector of length n; and $\mathsf{kE}, \mathsf{rE}, \mathsf{sE}$ are a triple of deterministic "encoding" functions that depend on the underlying predicate P (we refer to these functions as key encoding, receiver encoding and sender encoding respectively.) Syntactically, this is already a refinement of the prior frameworks in [43, 2] which associates a single function with sk_y given by

$$(y, \alpha, \mathbf{w}, r) \mapsto \big(r,\, \mathsf{kE}(y,\alpha) + r\cdot\mathsf{rE}(y,\mathbf{w})\big) \tag{3}$$

in the exponent. The prior frameworks allow for instance for kE to be randomized. With the refinement in place, we can now specify the restricted α-reconstruction property used for correctness:

> **(restricted α-reconstruction.)** For every x, y for which $\mathsf{P}(x,y) = 1$, there is a linear map L_{xy} such that for all α, r,
>
> $$L_{xy}\Big(\mathsf{kE}(y,\alpha) + r\cdot\mathsf{rE}(y,\mathbf{w}),\, r\cdot\mathsf{sE}(x,\mathbf{w})\Big) = \alpha$$

This means that we can recover $e(g_1,g_1)^{\alpha s}$ given

$$e(g_1^s, g_1^{\mathsf{kE}(y,\alpha)+r\cdot\mathsf{rE}(y,\mathbf{w})}) \quad \text{and} \quad e(g_1^{s\cdot\mathsf{sE}(x,\mathbf{w})}, g_1^r),$$

upon which we can decrypt the ciphertext. Observe that we only need to pair the first component g_1^s of ct_x with the second component of sk_y and the second component of ct_x with the first component g_1^r of sk_y. Correctness now relies on a so-called *associativity property* [14], namely that for all i and all w_i:

$$e(g_1^s, g_1^{w_i r}) = e(g_1^{w_i s}, g_1^r) \tag{4}$$

To translate the scheme to prime-order groups, we carry out the following substitution:

$$\begin{aligned} & && w_i \mapsto \mathbf{W}_i \in \mathbb{Z}_p^{(k+1)\times(k+1)} \\ s \mapsto \mathbf{s} \in \mathbb{Z}_p^k, & && r \mapsto \mathbf{r} \in \mathbb{Z}_p^k \\ g_1^s \mapsto [\mathbf{As}]_1, & && g_1^r \mapsto [\mathbf{Br}]_2 \\ g_1^{w_i s} \mapsto [\mathbf{W}_i^{\top}\mathbf{As}]_1, & && g_1^{w_i r} \mapsto [\mathbf{W}_i\mathbf{Br}]_2 \end{aligned}$$

Using (1), we have

$$e([\mathbf{As}]_1, [\mathbf{W}_i\mathbf{Br}]_2) = e([\mathbf{W}_i^{\top}\mathbf{As}]_1, [\mathbf{Br}]_2)$$

which is exactly what we used in composite-order groups in (4). In fact, a stronger "pairwise associativity" property holds in composite-order groups, namely for all i, j and all w_i, w_j:

$$e(g_1^{w_j s}, g_1^{w_i r}) = e(g_1^{w_i s}, g_1^{w_j r})$$

which is not satisfied by our prime-order techniques since $\mathbf{W}_i$ and $\mathbf{W}_j$ do not commute. Restricted α-reconstruction means that we do not need to pair $g_1^{w_j s}$ with $g_1^{w_i r}$ during decryption, and thus the associativity property already suffices for decryption correctness. For maximal modularity, we describe our compiler using the framework of dual system groups introduced in [14], which allows us to simultaneously capture prime-order and composite-order groups.

Next, we specify the privacy property which we use in the proof of ABE security:

(α-privacy.) For every x, y for which $\mathsf{P}(x, y) = 0$, α is perfectly hidden given

$$\mathsf{sE}(x, \mathbf{w}),\ \mathsf{kE}(y, \alpha) + \mathsf{rE}(y, \mathbf{w})$$

where $\mathbf{w} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^n$.

We stress that the privacy requirement only needs to hold in a private-key setting where the adversary does not see $\mathbf{w}$ and in a one-time setting where the adversary only gets a single copy of $\mathsf{sE}(x, \mathbf{w}), \mathsf{kE}(y, \alpha) + \mathsf{rE}(y, \mathbf{w})$. As pointed out in [43], the dual system encryption methodology can be used to boost security in a private-key, one-time, non-adaptive setting as given by α-privacy to a full-fledged public-key, many-time, adaptive setting as is required for ABE security. One novelty in this work over [43, 2] lies in carrying this out over prime-order bilinear groups. In the proof, we exploit the fact that the key sk_y leaks no information about $\mathbf{w}$ when $r = 0$ (c.f. Eqn. (2)). This way, we can ensure that in each step in the proof of security, at most one secret key leaks information about $\mathbf{w}$ in the semi-functional space. This is important since α-privacy only holds when $\mathbf{w}$ is used once. We also introduce new attribute-hiding privacy requirements for encodings in this work (c.f. Section 7.2).

New encodings. For many predicates, the prior encodings in [43, 2] satisfy the new refinement trivially. In addition, we introduce a number of new encodings:

- For KP-ABE for boolean formula, the prior encoding corresponding to the secret key in [43, 2] is given by

  $$(r, \alpha_1 + r w_1, \ldots, \alpha_\ell + r w_\ell)$$

  where $(\alpha_1, \ldots, \alpha_\ell)$ are random shares of α using a linear secret-sharing scheme and fresh randomness for each secret key. This does not satisfy the syntactic refinement captured in Eqn. (3). In our scheme, we use

  $$(r, \alpha_1' + r(w_1 + \delta_1), \ldots, \alpha_\ell' + r(w_\ell + \delta_\ell))$$

  where $(\alpha_1', \ldots, \alpha_\ell')$ are *deterministically* derived from α using the secret-sharing scheme with randomness fixed to 0 and $(\delta_1, \ldots, \delta_\ell)$ are *random* shares of 0. In the ensuing KP-ABE scheme, we use the same

functionality	improvements	reference
KP-ABE boolean formula	50% savings in secret key size, faster decryption	Fig. 3, §B.1
CP-ABE boolean formula	50% savings in ciphertext size, faster decryption	Fig. 4, §B.2
KP-ABE arithmetic formula	first adaptively secure scheme	§A.6
CP-ABE arithmetic formula	first adaptively secure scheme	§A.6
NIPE	25-50% savings in secret key and ciphertext size and in decryption time	Fig. 5
weakly attribute-hiding ZIPE	25% savings in secret key and ciphertext size and in decryption time	Fig. 6, §C

Fig. 2. Summary of efficiency improvements in our new ABE schemes.

$\delta_1, \ldots, \delta_\ell$ across all secret keys whereas prior constructions use fresh randomness for secret-sharing for each key. In addition, we obtain an analogous construction for CP-ABE. Here, we avoid having to consider randomized sender encodings as in [43, 2]. The final encodings have the same sizes as the prior ones, while satisfying the new refinement requirement. Moreover, by using associativity (c.f. Eqn. (4)), we reduce the number of pairings for the decryption to a constant and avoid exponentiations in the target group at the cost of cheaper exponentiations in the source groups.

- We extend the encodings for KP-ABE and CP-ABE to arithmetic branching programs, based on the selectively secure KP-ABE for arithmetic branching programs in [25](c.f. Section A.6). Combined with our generic framework, we obtain the first adaptively secure KP-ABE and CP-ABE for arithmetic branching programs.
- We also present a new encoding for broadcast encryption with n users where both the receiver and sender encoding have sublinear $O(\sqrt{n})$ length (c.f. Section A.7) and a simple encoding for large universe fuzzy IBE(c.f. Section A.8).

Achieving weak attribute-hiding. In a weakly attribute-hiding scheme, we need to guarantee the privacy of the ciphertext attribute x against collusions that are not authorized to decrypt the challenge ciphertext. To achieve this property, we require additional properties from the underlying encoding and the underlying group structure (extending ideas from [36, 1, 5]). We use the fact that for any vector $\mathbf{c} \in \mathbb{Z}_p^{k+1}$ outside the span of $\mathbf{A}$, the vector $\mathbf{W}^\top \mathbf{c}$ is uniformly random given $\mathbf{W}^\top \mathbf{A}$, where $\mathbf{W}$ is a uniformly random matrix. We can then use $\mathbf{W}^\top \mathbf{c}$ to information-theoretically blind the attribute in the challenge ciphertext. For this to work, we need to make sure that the semi-functional secret keys do not leak any *additional* information about $\mathbf{WB}$.

New ABE schemes. We describe several concrete new ABE schemes obtained via our new framework (c.f. Fig. 2). Specifically, we obtain:

- ABE schemes for the inner product and non-zero inner product predicates with a 25% improvement in secret key and ciphertext sizes and decryption time, improving upon previous constructions in [37];
- a key-policy ABE scheme for boolean formula with a 50% improvement in secret key size and faster decryption and an analogous result for ciphertext-policy ABE, improving upon previous constructions in [36, 30];
- the first adaptively secure key-policy and ciphertext-policy ABE schemes for arithmetic formula and branching programs without an exponential security loss, improving upon previous constructions in [25, 11].

Along the way, we also generalize several previous constructions for $k = 2$ to general k with $k = 1$ being particularly relevant for practical efficiency. More generally, the parameters of our schemes under k-Lin

reference	$\lvert\mathsf{mpk}\rvert$	$\lvert\mathsf{sk}\rvert$	$\lvert\mathsf{ct}\rvert$	T_{Dec}	assumption
LOS+10 [33]	$(\ell+1)\lvert G_N\rvert+\lvert G_T\rvert$	$2\ell\lvert G_N\rvert$	$(\ell+1)\lvert G_N\rvert+\lvert G_T\rvert$	$2\ell P+2\ell E_T$	composite
A14, W14 [2, 43]	$(\ell+2)\lvert G_N\rvert+\lvert G_T\rvert$	$(\ell+1)\lvert G_N\rvert$	$(\ell+1)\lvert G_N\rvert+\lvert G_T\rvert$	$2\ell P+\ell E_T$	composite
OT10 [36]	$(21\ell+15)\lvert G_1\rvert+\lvert G_T\rvert$	$(7\ell+5)\lvert G_2\rvert$	$(7\ell+5)\lvert G_1\rvert+\lvert G_T\rvert$	$(7\ell+5)P+\ell E_T$	DLIN
CLL+12 [15]	$(8\ell+6)\lvert G_1\rvert+\lvert G_T\rvert$	$(4\ell+3)\lvert G_2\rvert$	$(4\ell+3)\lvert G_1\rvert+\lvert G_T\rvert$	$(4\ell+1)P+\ell E_T$	SXDH
ours	$(2\ell+1)\lvert G_1\rvert+\lvert G_T\rvert$	$(2\ell+2)\lvert G_2\rvert$	$(2\ell+2)\lvert G_1\rvert+\lvert G_T\rvert$	$4P+4\ell E$	SXDH
	$(6\ell+2)\lvert G_1\rvert+2\lvert G_T\rvert$	$(3\ell+3)\lvert G_2\rvert$	$(3\ell+3)\lvert G_1\rvert+\lvert G_T\rvert$	$6P+6\ell E$	DLIN
	$(k(k+1)\ell+k)\lvert G_1\rvert+k\lvert G_T\rvert$	$(k+1)(\ell+1)\lvert G_2\rvert$	$(k+1)(\ell+1)\lvert G_1\rvert+\lvert G_T\rvert$	$2(k+1)P+2(k+1)\ell E$	k-Lin

Fig. 3. Summary of existing adaptively secure KP-ABE schemes based on symmetric bilinear groups of composite order N with pairing $e : G_N \times G_N \to G_T$, or asymmetric bilinear groups of prime order p with pairing $e : G_1 \times G_2 \to G_T$. Here, ℓ is both the universe size and the maximum number of rows in the matrix $\mathbf{M}$ of the access structure since all listed schemes have the one-use restriction. We use E to denote exponentiation in G_N, G_1 or G_2; E_T to denote exponentiation in G_T; and P to denote a pairing. For CT, we omit the additive overhead of $O(\ell)$ *bits* in order to transmit the attribute vector. The encryption time is $O(\ell E + E_T)$ and the key generation time is $O(\ell E)$.

reference	$\lvert\mathsf{mpk}\rvert$	$\lvert\mathsf{sk}\rvert$	$\lvert\mathsf{ct}\rvert$	T_{Dec}	assumption
LOS+10 [33]	$(\ell+2)\lvert G_N\rvert+\lvert G_T\rvert$	$(\ell+2)\lvert G_N\rvert$	$(2\ell+1)\lvert G_N\rvert+\lvert G_T\rvert$	$(2\ell+1)P+\ell E_T$	composite
A14, W14 [2, 43]	$(\ell+2)\lvert G_N\rvert+\lvert G_T\rvert$	$(\ell+2)\lvert G_N\rvert$	$(\ell+1)\lvert G_N\rvert+\lvert G_T\rvert$	$(2\ell+1)P+\ell E_T$	composite
OT10 [36]	$(21\ell+15)\lvert G_1\rvert+\lvert G_T\rvert$	$(7\ell+5)\lvert G_2\rvert$	$(7\ell+5)\lvert G_1\rvert+\lvert G_T\rvert$	$(7\ell+5)P+\ell E_T$	DLIN
Lewko12 [31]	$(24\ell+6)\lvert G_1\rvert+2\lvert G_T\rvert$	$(6\ell+3)\lvert G_2\rvert$	$(6\ell+3)\lvert G_1\rvert+\lvert G_T\rvert$	$(6\ell+1)P+\ell E_T$	DLIN
ours	$(2\ell+3)\lvert G_1\rvert+\lvert G_T\rvert$	$(2\ell+4)\lvert G_2\rvert$	$(2\ell+2)\lvert G_1\rvert+\lvert G_T\rvert$	$4P+4\ell E$	SXDH
	$(6\ell+8)\lvert G_1\rvert+2\lvert G_T\rvert$	$(3\ell+6)\lvert G_2\rvert$	$(3\ell+3)\lvert G_1\rvert+\lvert G_T\rvert$	$6P+6\ell E$	DLIN
	$(k(k+1)(\ell+1)+k)\lvert G_1\rvert+k\lvert G_T\rvert$	$(k+1)(\ell+2)\lvert G_2\rvert$	$(k+1)(\ell+1)\lvert G_1\rvert+\lvert G_T\rvert$	$2(k+1)P+2(k+1)\ell E$	k-Lin

Fig. 4. Summary of existing adaptively secure CP-ABE schemes. For CT, we omit the additive overhead of $O(\ell)$ *bits* in order to transmit the access structure in CP-ABE schemes. The encryption time is $O(\ell E + E_T)$ and the key generation time is $O(\ell E)$.

is $k+1$ times those of the best composite-order schemes based on subgroup assumptions: this achieves a "seemingly best-possible" composite-to-prime-order transformation where each composite element is simulated using $k+1$ prime-order elements.

Finally, our prime-order ABE schemes are simpler to describe than prior schemes as they share the same structure as existing composite-order schemes. In particular, we obtain the following anonymous IBE scheme:

$$\begin{aligned}
\mathsf{mpk} &= [\mathbf{A}, \mathbf{W}_0^{\top}\mathbf{A}, \mathbf{W}_1^{\top}\mathbf{A}]_1, [\mathbf{k}^{\top}\mathbf{A}]_T \\
\mathsf{sk}_{\mathsf{id}} &= [\mathbf{Br}, \mathbf{k} + (\mathbf{W}_0 + \mathsf{id}\cdot\mathbf{W}_1)\mathbf{Br}]_2 \in G_2^{2(k+1)} \\
\mathsf{ct}_{\mathsf{id}} &= [\mathbf{As}, (\mathbf{W}_0 + \mathsf{id}\cdot\mathbf{W}_1)^{\top}\mathbf{As}]_1, [\mathbf{k}^{\top}\mathbf{As}]_T \cdot m \in G_1^{2(k+1)} \times G_T
\end{aligned}$$

where $\mathbf{A}, \mathbf{B} \in \mathbb{Z}_p^{(k+1)\times k}, \mathbf{W}_0, \mathbf{W}_1 \in \mathbb{Z}_p^{(k+1)\times(k+1)}, \mathbf{s}, \mathbf{r} \in \mathbb{Z}_p^k, \mathbf{k} \in \mathbb{Z}_p^{k+1}$. This scheme extends naturally to a non-anonymous BBG-style compact HIBE [10] (this is not the case for the prime-order IBE schemes in [30, 15]).

1.2 Discussion

Comparison with prior works. A summary of the prior approaches for obtaining efficient adaptively secure efficient dual system (H)IBE and ABE is presented in Fig. 1. The most general technique we have for simulating composite-order groups in prime-order ones are those based on "dual pairing vector spaces" (DPVS) [34, 35, 30, 15]. However, these techniques do not preserve the asymptotic efficiency of the

reference	$\vert\mathsf{mpk}\vert$	$\vert\mathsf{sk}\vert$	$\vert\mathsf{ct}\vert$	T_{Dec}	assumption
OT11 [37]	$(8n+23)\vert G_1\vert+\vert G_T\vert$	$(4n+5)\vert G_2\vert$	$13\vert G_1\vert+\vert G_T\vert$	$13P+(4n-4)E$	DLIN
ours	$(6n+2)\vert G_1\vert+2\vert G_T\vert$	$(3n+3)\vert G_2\vert$	$6\vert G_1\vert+\vert G_T\vert$	$6P+3nE+E_T$	DLIN
	$(k(k+1)n+k)\vert G_1\vert+k\vert G_T\vert$	$(k+1)(n+1)\vert G_2\vert$	$2(k+1)\vert G_1\vert+\vert G_T\vert$	$2(k+1)P+(k+1)nE+E_T$	k-Lin
OT11 [37]	$(8n+23)\vert G_1\vert+\vert G_T\vert$	$13\vert G_2\vert$	$(4n+5)\vert G_1\vert+\vert G_T\vert$	$13P+(4n-4)E$	DLIN
ours	$(6n+8)\vert G_1\vert+2\vert G_T\vert$	$9\vert G_2\vert$	$(3n+3)\vert G_1\vert+\vert G_T\vert$	$6P+3nE+E_T$	DLIN
	$(k(k+1)(n+1)+k)\vert G_1\vert+k\vert G_T\vert$	$3(k+1)\vert G_2\vert$	$(k+1)(n+1)\vert G_1\vert+\vert G_T\vert$	$2(k+1)P+(k+1)nE+E_T$	k-Lin

Fig. 5. Comparison amongst adaptively secure public-index NIPE, based on our encodings in Section A.2. We omitted schemes with weaker security guarantees, such as the co-selective NIPE from [3].

reference	$\vert\mathsf{mpk}\vert$	$\vert\mathsf{sk}\vert$	$\vert\mathsf{ct}\vert$	T_{Dec}	assumption	attribute-hiding
KSW [28]	$(2n+2)\vert G_N\vert$	$(2n+1)\vert G_N\vert$	$(2n+1)\vert G_N\vert+\vert G_T\vert$	$(2n+1)P$	composite	fully
OT10 [36]	$(3n^2+8n+4)\vert G_1\vert+\vert G_T\vert$	$(3n+2)\vert G_2\vert$	$(3n+2)\vert G_1\vert+\vert G_T\vert$	$(3n+2)P$	DLIN	weakly
OT11 [37]	$(10n+13)\vert G_1\vert+\vert G_T\vert$	$9\vert G_2\vert$	$(4n+1)\vert G_1\vert+\vert G_T\vert$	$9P+(4n-4)E$	DLIN	weakly
OT12 [38]	$(12n+16)\vert G_1\vert+\vert G_T\vert$	$11\vert G_2\vert$	$(5n+1)\vert G_1\vert+\vert G_T\vert$	$11P+(5n-5)E$	DLIN	fully
ours	$(2n+4)\vert G_1\vert+\vert G_T\vert$	$4\vert G_2\vert$	$(2n+2)\vert G_1\vert+\vert G_T\vert$	$4P+2nE$	SXDH	weakly
	$(6n+8)\vert G_1\vert+2\vert G_T\vert$	$6\vert G_2\vert$	$(3n+3)\vert G_1\vert+\vert G_T\vert$	$6P+3nE$	DLIN	weakly
	$(k(k+1)(n+1)+k)\vert G_1\vert+k\vert G_T\vert$	$2(k+1)\vert G_2\vert$	$(k+1)(n+1)\vert G_1\vert+\vert G_T\vert$	$2(k+1)P+(k+1)nE$	k-Lin	weakly

Fig. 6. Comparison amongst adaptively secure attribute-hiding ZIPE, based on our encodings in Section A.1. We omitted the schemes in [35, 33] based on parameterized assumptions. Note that the scheme in [28] only achieves selective security.

underlying schemes; in particular, applying them to the composite-order compact HIBE schemes in [32] blows up the ciphertext size from constant to linear. The sparse DPVS technique [37, 41] uses subgroups of sparse matrices with mostly zero entries to overcome this limitation; however, they substantially limit the generality of the DPVS technique: the structure of these matrices now depend on the predicate and the composite-order scheme (to preserve efficiency), and the analysis for correctness, efficiency and security are more involved. The constructions in [14] fail to extend to boolean formula due to the need for additional randomness for secret-sharing, and also do not extend to yield anonymous IBE. The direct constructions in [26, 5] that bypass a conversion from composite-order groups do not seem to naturally extend beyond (H)IBE: the former uses tag-based languages where tags correspond to identities, and the latter relies on the notion of message authentication codes where messages correspond to identities. In particular, we do not know analogues of these constructions for either the inner product predicate or CP/KP-ABE for boolean formula.

As noted earlier, another novel contribution in this work over prior unifying frameworks in [2, 43] (generalizing [32, 33]) for composite-order groups lies in realizing the weakly-attribute guarantee. This is particularly challenging in composite-order groups for two reasons: (i) there is an explicit anonymity attack on the Lewko-Waters IBE [32] in composite-order group and (ii) the attribute in the semi-functional ciphertext is leaked in the G_{p_1}-component. Interestingly, we are still able to show that our prime-order analog of the Lewko-Waters IBE is anonymous.

Organization. We recall the definition of an attribute-based encryption scheme in Section 2. We recall the notion of dual system groups in Section 3 and describe our instantiations in Section 4. We describe our notion of predicate encodings in Section 5 and present the instantiations in Section A. We present our generic ABE construction in Section 6. We handle weakly attribute-hiding predicate encryption in Section 7. We present self-contained description of our ABE schemes for boolean span programs and weakly attribute-hiding ZIPE scheme in Sections B and C.

2 Preliminaries

Notation. We denote by $s \leftarrow_{\mathrm{R}} S$ the fact that s is picked uniformly at random from a finite set S. By PPT, we denote a probabilistic polynomial-time algorithm. Throughout this paper, we use 1^λ as the security parameter. We use $\cdot$ to denote multiplication as well as component-wise multiplication.

2.1 Attribute-Based Encryption

An attribute-based encryption (ABE) scheme for a predicate $\mathsf{P}(\cdot,\cdot)$ consists of four algorithms $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$:

$\mathsf{Setup}(1^\lambda, \mathcal{X}, \mathcal{Y}, \mathcal{M}) \rightarrow (\mathsf{mpk}, \mathsf{msk})$. The setup algorithm gets as input the security parameter λ, the attribute universe $\mathcal{X}$, the predicate universe $\mathcal{Y}$, the message space $\mathcal{M}$ and outputs the public parameter mpk, and the master key msk.

$\mathsf{Enc}(\mathsf{mpk}, x, m) \rightarrow \mathsf{ct}_x$. The encryption algorithm gets as input mpk, an attribute $x \in \mathcal{X}$ and a message $m \in \mathcal{M}$. It outputs a ciphertext ct_x. Note that x is public given ct_x.

$\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, y) \rightarrow \mathsf{sk}_y$. The key generation algorithm gets as input msk and a value $y \in \mathcal{Y}$. It outputs a secret key sk_y. Note that y is public given sk_y.

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_y, \mathsf{ct}_x) \rightarrow m$. The decryption algorithm gets as input sk_y and ct_x such that $\mathsf{P}(x, y) = 1$. It outputs a message m.

Correctness. We require that for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$ such that $\mathsf{P}(x, y) = 1$ and all $m \in \mathcal{M}$,

$$\Pr[\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_y, \mathsf{Enc}(\mathsf{mpk}, x, m)) = m] = 1,$$

where the probability is taken over $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{X}, \mathcal{Y}, \mathcal{M})$, $\mathsf{sk}_y \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, y)$, and the coins of Enc.

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

$$\mathsf{Adv}^{\mathrm{ABE}}_{\mathcal{A}}(\lambda) := \Pr\left[b = b' : \begin{array}{c} (\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{X}, \mathcal{Y}, \mathcal{M}); \\ (x^*, m_0, m_1) \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{msk},\cdot)}(\mathsf{mpk}); \\ b \leftarrow_{\mathrm{R}} \{0,1\}; \mathsf{ct}_{x^*} \leftarrow \mathsf{Enc}(\mathsf{mpk}, x^*, m_b); \\ b' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{msk},\cdot)}(\mathsf{ct}_{x^*}) \end{array} \right] - \frac{1}{2}$$

with the restriction that all queries y that $\mathcal{A}$ makes to $\mathsf{KeyGen}(\mathsf{msk}, \cdot)$ satisfies $\mathsf{P}(x^*, y) = 0$ (that is, sk_y does not decrypt ct_{x^*}). An ABE scheme is *adaptively secure* if for all PPT adversaries $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathrm{ABE}}_{\mathcal{A}}(\lambda)$ is a negligible function in λ.

3 Dual System Groups

This section is largely adapted from [14].

3.1 Overview

Dual system groups contain a triple of abelian groups $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T)$ and a non-degenerate bilinear map $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$. For concreteness, we may think of $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T)$ as composite-order bilinear groups. Dual system groups take as input a parameter 1^n (think of n as the universe size in KP-ABE) and satisfy the following properties:

(subgroup indistinguishability.) There are two computationally indistinguishable ways to sample correlated $(n+1)$-tuples from $\mathbb{G}^{n+1}$: the "normal" distribution, and a higher-entropy distribution with "semi-functional components". We sample the normal distribution using SampG and the semi-functional components using $\widehat{\mathsf{SampG}}$. An analogous property holds for $\mathbb{H}^{n+1}$, with algorithms SampH and $\widehat{\mathsf{SampH}}$ respectively, with an important distinction in the auxiliary input provided to the distinguisher. For concreteness, think in terms of symmetric bilinear groups of composite order N where

$$\begin{array}{lll} \mathsf{SampG} \to (g_1^s, g_1^{s\mathbf{w}}) \in G_N^{n+1} & \text{and} & \mathsf{SampH} \to (g_1^r, g_1^{r\mathbf{w}}) \in G_N^{n+1} \\ \widehat{\mathsf{SampG}} \to (g_2^s, g_2^{s\mathbf{w}}) \in G_N^{n+1} & \text{and} & \widehat{\mathsf{SampH}} \to (g_2^r, g_2^{r\mathbf{w}}) \in G_N^{n+1} \end{array}$$

Here, N is the product of three primes p_1, p_2, p_3; g_1, g_2 are generators of order p_1, p_2; and $g_1^{\mathbf{w}} \in G_N^n$ is part of the public parameters.

(associativity.) For all $(g_0, g_1, \ldots, g_n) \in \mathbb{G}^{n+1}$ and all $(h_0, h_1, \ldots, h_n) \in \mathbb{H}^{n+1}$ drawn from the respective normal distributions according to SampG and SampH, we have that for all $i = 1, \ldots, n$,

$$e(g_0, h_i) = e(g_i, h_0).$$

We require this property for correctness (c.f. Eqn. (4)).

(right subgroup $\mathbb{H}$.) There is some distinguished element $h^* \in \mathbb{H}$, which generates the semi-functional components in $\mathbb{H}$. It is convenient to think of h^* as being orthogonal to the normal distribution over $\mathbb{G}$ (c.f. orthogonality). On the other hand, we require that h^* is *not* orthogonal to the semi-functional components in $\mathbb{G}$ (c.f. non-degeneracy), so that we get a random value when we decrypt a semi-functional ciphertext with a semi-functional key.

(parameter-hiding.) Both normal distributions can be efficiently sampled given the public parameters; on the other hand, given only the public parameters, the higher-entropy distributions contain n "units" of information-theoretic entropy (in the semi-functional component), one unit for each of the n elements in the $(n+1)$-tuple apart from the first. In the formal statement, the hidden entropy is captured by n random exponents $(\hat{u}_1, \ldots, \hat{u}_n)$ shared across $\mathbb{G}$ and $\mathbb{H}$. It is crucial here that we use the same $\hat{u}_i$ in $\mathbb{G}$ and in $\mathbb{H}$, so that decryption succeeds with nominally semi-functional objects.

3.2 Definitions

Syntax. Dual system groups consist of six randomized algorithms given by (SampP, SampGT, SampG, SampH) along with $(\widehat{\mathsf{SampG}}, \widehat{\mathsf{SampH}})$:

$\mathsf{SampP}(1^\lambda, 1^n)$: On input $(1^\lambda, 1^n)$, output public and secret parameters $(\mathsf{pp}, \mathsf{sp})$, where:

- pp contains a prime p of length $\Omega(\lambda)$, a triple of abelian groups $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T)$, a non-degenerate bilinear map $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$, a linear map μ defined on $\mathbb{H}$, along with some additional parameters used by $\mathsf{SampG}, \mathsf{SampH}$;
- the groups $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T)$ are $\mathbb{Z}_p$-modules where $\mathbb{Z}_p$ acts on $\mathbb{G}, \mathbb{H}, \mathbb{G}_T$ via exponentiation;
- given pp, we can uniformly sample from $\mathbb{H}$;
- sp contains $h^* \in \mathbb{H}$ (where $h^* \neq 1$), along with some additional parameters used by $\widehat{\mathsf{SampG}}, \widehat{\mathsf{SampH}}$;

$\mathsf{SampGT}: \mathrm{Im}(\mu) \to \mathbb{G}_\mathrm{T}$. (As a concrete example, suppose $\mu : \mathbb{H} \to \mathbb{G}_T$ and $\mathrm{Im}(\mu) = \mathbb{G}_\mathrm{T}$.)

$\mathsf{SampG}(\mathsf{pp})$: Output $\mathbf{g} \in \mathbb{G}^{n+1}$.

$\mathsf{SampH}(\mathsf{pp})$: Output $\mathbf{h} \in \mathbb{H}^{n+1}$.

$\widehat{\mathsf{SampG}}(\mathsf{pp},\mathsf{sp})$: Output $\hat{\mathbf{g}} \in \mathbb{G}^{n+1}$.

$\widehat{\mathsf{SampH}}(\mathsf{pp},\mathsf{sp})$: Output $\hat{\mathbf{h}} \in \mathbb{H}^{n+1}$.

The first four algorithms are used in the actual scheme, whereas the last two algorithms are used only in the proofs of security. We define SampG_0 to denote the first group element in the output of SampG, and we define $\widehat{\mathsf{SampG}}_0, \widehat{\mathsf{SampH}}_0$ analogously.

Remark 1. Given a $\mathbb{Z}_p$-linear function $L : \mathbb{Z}_p^n \to \mathbb{Z}_p$ given by $(w_1, \ldots, w_n) \mapsto a_1 w_1 + \cdots + a_n w_n$ (where $a_1, \ldots, a_n \in \mathbb{Z}_p$ are fixed constants), L acts on $\mathbb{Z}_p$-modules $\mathbb{G}^n, \mathbb{H}^n, \mathbb{G}_T^n$ in the natural way. For instance, $L : \mathbb{G}^n \to \mathbb{G}$ is given by $(g_1, \ldots, g_n) \mapsto g_1^{a_1} \cdots g_n^{a_n}$. This extends also to general $\mathbb{Z}_p$-linear functions $L : \mathbb{Z}_p^n \to \mathbb{Z}_p^m$ coordinate-wise.

Correctness. The requirements for correctness are as follows:

(projective.) For all $h \in \mathbb{H}$ and all coin tosses s, we have $\mathsf{SampGT}(\mu(h); s) = e(\mathsf{SampG}_0(\mathsf{pp}; s), h)$.

(associative.) For all $(g_0, g_1, \ldots, g_n) \leftarrow \mathsf{SampG}(\mathsf{pp})$ and $(h_0, h_1, \ldots, h_n) \leftarrow \mathsf{SampH}(\mathsf{pp})$ and for all $i = 1, \ldots, n$, we have $e(g_0, h_i) = e(g_i, h_0)$.

($\mathbb{H}$-subgroup.) The output of $\mathsf{SampH}(\mathsf{pp})$ is the uniform distribution over a subgroup of $\mathbb{H}^{n+1}$.

Security. The requirements for security are as follows:

(orthogonality.) $\mu(h^*) = 1$.

(non-degeneracy.) For all $\hat{h}_0 \leftarrow \widehat{\mathsf{SampH}}_0(\mathsf{pp},\mathsf{sp})$, h^* lies in the group generated by $\hat{h}_0$. For all $\hat{g}_0 \leftarrow \widehat{\mathsf{SampG}}_0(\mathsf{pp},\mathsf{sp})$, we have $e(\hat{g}_0, h^*)^\alpha$ is identically distributed to the uniform distribution over $\mathbb{G}_T$, where $\alpha \leftarrow_\mathrm{R} \mathbb{Z}_p$.

(left subgroup indistinguishability.) For any adversary $\mathcal{A}$, we define the advantage function:

$$\mathsf{Adv}_{\mathcal{A}}^{\mathrm{LS}}(\lambda) := |\Pr[\mathcal{A}(\mathsf{pp}, \boxed{\mathbf{g}}) = 1] - \Pr[\mathcal{A}(\mathsf{pp}, \boxed{\mathbf{g} \cdot \hat{\mathbf{g}}}) = 1]|$$

where

$$(\mathsf{pp},\mathsf{sp}) \leftarrow \mathsf{SampP}(1^\lambda, 1^n);\ \mathbf{g} \leftarrow \mathsf{SampG}(\mathsf{pp});\ \hat{\mathbf{g}} \leftarrow \widehat{\mathsf{SampG}}(\mathsf{pp},\mathsf{sp}).$$

(right subgroup indistinguishability.) For any adversary $\mathcal{A}$, we define the advantage function:

$$\mathsf{Adv}_{\mathcal{A}}^{\mathrm{RS}}(\lambda) := |\Pr[\mathcal{A}(\mathsf{pp}, h^*, \mathbf{g} \cdot \hat{\mathbf{g}}, \boxed{\mathbf{h}}) = 1] - \Pr[\mathcal{A}(\mathsf{pp}, h^*, \mathbf{g} \cdot \hat{\mathbf{g}}, \boxed{\mathbf{h} \cdot \hat{\mathbf{h}}}) = 1]|$$

where

$$(\mathsf{pp},\mathsf{sp}) \leftarrow \mathsf{SampP}(1^\lambda, 1^n);\ \mathbf{g} \leftarrow \mathsf{SampG}(\mathsf{pp});\ \hat{\mathbf{g}} \leftarrow \widehat{\mathsf{SampG}}(\mathsf{pp},\mathsf{sp});\ \mathbf{h} \leftarrow \mathsf{SampH}(\mathsf{pp});\ \hat{\mathbf{h}} \leftarrow \widehat{\mathsf{SampH}}(\mathsf{pp},\mathsf{sp}).$$

(parameter-hiding.) The following distributions are identically distributed

$$\{\mathsf{pp}, h^*, \boxed{\hat{\mathbf{g}}, \hat{\mathbf{h}}}\} \quad \text{and} \quad \{\mathsf{pp}, h^*, \boxed{\hat{\mathbf{g}} \cdot \hat{\mathbf{g}}', \hat{\mathbf{h}} \cdot \hat{\mathbf{h}}'}\}$$

where

$$
\begin{aligned}
&(\mathsf{pp},\mathsf{sp}) \leftarrow \mathsf{SampP}(1^{\lambda}, 1^{n});\\
&\hat{\mathbf{g}} = (\hat{g}_0,\ldots) \leftarrow \widehat{\mathsf{SampG}}(\mathsf{pp},\mathsf{sp});\quad \hat{\mathbf{h}} = (\hat{h}_0,\ldots) \leftarrow \widehat{\mathsf{SampH}}(\mathsf{pp},\mathsf{sp});\\
&\hat{\mathbf{g}}' := (1, \hat{g}_0^{\hat{u}_1},\ldots,\hat{g}_0^{\hat{u}_n}) \in \mathbb{G}^{n+1};\quad \hat{\mathbf{h}}' := (1, \hat{h}_0^{\hat{u}_1},\ldots,\hat{h}_0^{\hat{u}_n}) \in \mathbb{H}^{n+1};\quad \hat{u}_1,\ldots,\hat{u}_n \leftarrow_{\mathrm{R}} \mathbb{Z}_p.
\end{aligned}
$$

4 Instantiations of DSG from k-Lin

We present a new instantiation of dual system groups under the k-Lin assumption, inspired by the constructions in [5, 14].

Overview. The prior construction of DSG [14] (building upon [35, 36, 30, 15]) starts with a random $\mathbf{B} \leftarrow_{\mathrm{R}} \mathrm{GL}_{k+1}(\mathbb{Z}_p)$ and defines $\mathbf{B}^* := (\mathbf{B}^{\top})^{-1}$ so that $\mathbf{B}^{\top}\mathbf{B}^*$ is the identity matrix; then uses $\mathbf{B}$ for $\mathsf{SampG}, \widehat{\mathsf{SampG}}$ and $\mathbf{B}^*$ for $\mathsf{SampH}, \widehat{\mathsf{SampH}}$. In our construction, we may start with any pair of matrices $\mathbf{A}, \mathbf{B}$ in $\mathbb{Z}_p^{(k+1)\times k}$ of full rank:

- In addition, we pick $\mathbf{a}^{\perp}, \mathbf{b}^{\perp}$ so that ${\mathbf{a}^{\perp}}^{\top}\mathbf{A} = {\mathbf{b}^{\perp}}^{\top}\mathbf{B} = \mathbf{0}$ and ${\mathbf{a}^{\perp}}^{\top}\mathbf{b}^{\perp} \neq 0$; we then use $(\mathbf{A}, \mathbf{b}^{\perp})$ for $\mathsf{SampG}, \widehat{\mathsf{SampG}}$ and $(\mathbf{B}, \mathbf{a}^{\perp})$ for $\mathsf{SampH}, \widehat{\mathsf{SampH}}$.
- We achieve randomization as follows: again, pick a random $\mathbf{W} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(k+1)\times(k+1)}$ and replace $(\mathbf{A}, \mathbf{B})$ with $(\mathbf{W}^{\top}\mathbf{A}, \mathbf{W}\mathbf{B})$. The associativity property follows from the equation:

$$(\mathbf{W}^{\top}\mathbf{A})^{\top}\mathbf{B} = \mathbf{A}^{\top}(\mathbf{W}\mathbf{B})$$

Interestingly, the prior construction in [14] randomizes by multiplying a random $\mathbf{W}$ on the right, whereas our construction multiplies a random $\mathbf{W}$ on the left. Together with the fact that we no longer require the fact that $\mathbf{B}^{\top}\mathbf{B}^*$ is the identity, we substantially simplify the proof of subgroup indistinguishability.

4.1 Cryptographic assumptions

We follow the notation and algebraic framework for Diffie-Hellman-like assumptions in [18].

Prime-order bilinear groups. A generator $\mathcal{G}$ takes as input a security parameter λ and outputs a description $(p, G_1, G_2, G_T, g_1, g_2, e)$, where p is a prime of $\Theta(\lambda)$ bits; G_1, G_2 and G_T are cyclic groups of order p; g_1, g_2 are generators of G_1 and G_2 respectively; and $e : G_1 \times G_2 \to G_T$ is a non-degenerate bilinear map. Given $a \in \mathbb{Z}_p$, we use $[a]_1$ to denote g_1^a, $[a]_2$ to denote g_2^a, $[a]_T$ to denote $e(g_1, g_2)^a$. This extends to vectors and matrices in the obvious way. We define $e([\mathbf{A}]_1, [\mathbf{B}]_2) := [\mathbf{A}^{\top}\mathbf{B}]_T$.

Linear assumption. Let $\mathcal{D}_k$ be an efficiently samplable distribution of matrices $(\mathbf{A}, \mathbf{a}^{\perp})$ over $\mathbb{Z}_p^{(k+1)\times k} \times \mathbb{Z}_p^{k+1}$ so that $\mathbf{A}^{\top}\mathbf{a}^{\perp} = \mathbf{0}$ and $\mathbf{a}^{\perp} \neq \mathbf{0}$. In particular, we consider the distribution generated as follows: pick $a_1, \ldots, a_k \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$ and set

$$\mathbf{A} := \begin{pmatrix} a_1 & & & \\ & a_2 & & \\ & & \ddots & \\ & & & a_k \\ 1 & 1 & \ldots & 1 \end{pmatrix} \in \mathbb{Z}_p^{(k+1)\times k} \quad \text{and} \quad \mathbf{a}^{\perp} := \begin{pmatrix} a_1^{-1} \\ a_2^{-1} \\ \vdots \\ a_k^{-1} \\ -1 \end{pmatrix} \in \mathbb{Z}_p^{(k+1)}.$$

This distribution captures the k-linear assumption, which stipulates that

$$([\mathbf{A}], [\mathbf{As}]) \approx_c ([\mathbf{A}], [\mathbf{z}])$$

where $\mathbf{s} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k, \mathbf{z} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{k+1}$ in both G_1 and G_2.

Assumption 1 (k-Lin: the k-linear assumption in G_1) *For any adversary $\mathcal{A}$, we define the advantage function:*

$$\mathsf{Adv}_{\mathcal{A}}^{k\text{-Lin}} := \left|\Pr[\mathcal{A}((p, G_1, G_2, G_T, g_1, g_2, e); [\mathbf{A}]_1, [\mathbf{As}]_1) = 1] - \Pr[\mathcal{A}((p, G_1, G_2, G_T, g_1, g_2, e); [\mathbf{A}]_1, [\mathbf{z}]_1) = 1]\right|$$

where

$$(p, G_1, G_2, G_T, g_1, g_2, e) \leftarrow \mathcal{G}(1^\lambda);\ (\mathbf{A}, \mathbf{a}^\perp) \leftarrow \mathcal{D}_k;\ \mathbf{s} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k;\ \mathbf{z} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{k+1}.$$

We will slightly abuse notation and also use $\mathsf{Adv}_{\mathcal{A}}^{k\text{-Lin}}$ to denote the corresponding advantage function for G_2.

Basis lemma. The following structural lemma tells us that if we pick random $(\mathbf{A}, \mathbf{a}^\perp), (\mathbf{B}, \mathbf{b}^\perp) \leftarrow \mathcal{D}_k$, then with overwhelming probability, both $(\mathbf{A}, \mathbf{b}^\perp)$ and $(\mathbf{B}, \mathbf{a}^\perp)$ form a basis for $\mathbb{Z}_p^{k+1}$ and $\mathbf{a}^\perp, \mathbf{b}^\perp$ are not orthogonal. We will assume henceforth that this property always holds.

Lemma 1 (basis lemma). *With probability $1 - 1/p$ over $(\mathbf{A}, \mathbf{a}^\perp), (\mathbf{B}, \mathbf{b}^\perp) \leftarrow \mathcal{D}_k$, it holds that:*

$$\left(\mathbf{a}^\perp \not\in \mathit{span}(\mathbf{B})\right) \wedge \left(\mathbf{b}^\perp \not\in \mathit{span}(\mathbf{A})\right) \wedge \left({\mathbf{a}^\perp}^\top \mathbf{b}^\perp \neq 0\right).$$

Proof. It is easy to see that if ${\mathbf{a}^\perp}^\top \mathbf{b}^\perp \neq 0$, then

$$\left(\mathbf{a}^\perp \not\in \mathrm{span}(\mathbf{B})\right) \quad \text{and} \quad \left(\mathbf{b}^\perp \not\in \mathrm{span}(\mathbf{A})\right)$$

since every vector in span($\mathbf{A}$) is orthogonal to $\mathbf{a}^\perp$ and every vector in span($\mathbf{B}$) is orthogonal to $\mathbf{b}^\perp$. Observe that

$${\mathbf{a}^\perp}^\top \mathbf{b}^\perp = 1 + \sum_{i=1}^d (a_i b_i)^{-1}$$

and

$$\Pr\left[1 + \sum_{i=1}^d (a_i b_i)^{-1} \neq 0 : a_1, b_1, \ldots, a_k, b_k \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*\right] = 1 - 1/p.$$

The lemma then follows readily. □

Remark 2. Observe that Lemma 1 is not particular to the k-Lin distribution, since a similar proof works for any example of matrix distribution $\mathcal{D}_k$ presented in [18], namely $\mathcal{U}_{k+1,k}$, k-Casc, k-SCasc and k-ILin [18, Section 3.4].

4.2 Construction

Our construction is as follows:

$\mathsf{SampP}(1^\lambda, 1^n)$: On input $(1^\lambda, 1^n)$, do:

- run $(p, G_1, G_2, G_T, g_1, g_2, e) \leftarrow \mathcal{G}(1^\lambda)$, where $\mathcal{G}(1^\lambda)$ is an asymmetric prime-order group generator;
- define $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T, e) := (G_1^{k+1}, G_2^{k+1}, G_T, e)$;
- sample $(\mathbf{A}, \mathbf{a}^\perp), (\mathbf{B}, \mathbf{b}^\perp) \leftarrow \mathcal{D}_k$, along with $\mathbf{W}_1, \ldots, \mathbf{W}_n \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(k+1)\times(k+1)}$;
- define $\mu : G_2^{k+1} \to G_T^k$ by $\mu([\mathbf{k}]_2) = \left[\mathbf{A}^\top \mathbf{k}\right]_T$;
- set $h^* := \left[\mathbf{a}^\perp\right]_2$.

Output

$$\mathsf{pp} := \left((p, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e); \begin{array}{c} [\mathbf{A}]_1, [\mathbf{W}_1^\top \mathbf{A}]_1, \ldots, [\mathbf{W}_n^\top \mathbf{A}]_1 \\ [\mathbf{B}]_2, [\mathbf{W}_1 \mathbf{B}]_2, \ldots, [\mathbf{W}_n \mathbf{B}]_2 \end{array} \right) \quad \text{and} \quad \mathsf{sp} := \left(\mathbf{a}^\perp, \mathbf{b}^\perp, \mathbf{W}_1, \ldots, \mathbf{W}_n \right).$$

$\mathsf{SampGT}([\mathbf{p}]_T)$: Pick $\mathbf{s} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k$ and output $[\mathbf{s}^\top \mathbf{p}]_T \in G_T$.

$\mathsf{SampG}(\mathsf{pp})$: Pick $\mathbf{s} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k$ and output

$$\left([\mathbf{As}]_1, [\mathbf{W}_1^\top \mathbf{As}]_1, \ldots, [\mathbf{W}_n^\top \mathbf{As}]_1 \right) \in (G_1^{k+1})^{n+1}.$$

$\mathsf{SampH}(\mathsf{pp})$: Pick $\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k$ and output

$$\left([\mathbf{Br}]_2, [\mathbf{W}_1 \mathbf{Br}]_2, \ldots, [\mathbf{W}_n \mathbf{Br}]_2 \right) \in (G_2^{k+1})^{n+1}.$$

$\widehat{\mathsf{SampG}}(\mathsf{pp}, \mathsf{sp})$: Pick $\hat{s} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$ and output

$$\left([\mathbf{b}^\perp \hat{s}]_1, [\mathbf{W}_1^\top \mathbf{b}^\perp \hat{s}]_1, \ldots, [\mathbf{W}_n^\top \mathbf{b}^\perp \hat{s}]_1 \right) \in (G_1^{k+1})^{n+1}.$$

$\widehat{\mathsf{SampH}}(\mathsf{pp}, \mathsf{sp})$: Pick $\hat{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$ and output

$$\left([\mathbf{a}^\perp \hat{r}]_2, [\mathbf{W}_1 \mathbf{a}^\perp \hat{r}]_2, \ldots, [\mathbf{W}_n \mathbf{a}^\perp \hat{r}]_2 \right) \in (G_2^{k+1})^{n+1}.$$

Correctness. We check correctness properties as follows:

(projective.) This follows readily from the fact that for all $\mathbf{k} \in \mathbb{Z}_p^{k+1}, \mathbf{s} \in \mathbb{Z}_p^k$:

$$(\mathbf{As})^\top \mathbf{k} = (\mathbf{A}^\top \mathbf{k})^\top \mathbf{s}.$$

(associative.) This follows readily from the fact that for all $\mathbf{s} \in \mathbb{Z}_p^k, \mathbf{r} \in \mathbb{Z}_p^k, \mathbf{W}_i \in \mathbb{Z}_p^{(k+1)\times(k+1)}$:

$$(\mathbf{W}_i^\top \mathbf{As})^\top (\mathbf{Br}) = (\mathbf{As})^\top (\mathbf{W}_i \mathbf{Br}).$$

($\mathbb{H}$-subgroup.) This follows readily from the fact that $\mathbb{Z}_p^k$ is an additive group.

Security. We check security properties as follows:

(orthogonality.) This follows readily from $\mathbf{A}^\top \mathbf{a}^\perp = \mathbf{0}$.

(non-degeneracy.) This follows readily from $\mathbf{b}^{\perp\top} \mathbf{a}^\perp \neq 0$.

We establish left subgroup indistinguishability, right subgroup indistinguishability, and parameter-hiding in the next three lemmas. The left and right subgroup indistinguishability relies on the k-Lin assumption in prime-order groups, whereas parameter-hiding is unconditional.

Lemma 2 (left subgroup indistinguishability from k-Lin). *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that:*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathrm{LS}}(\lambda) \leq \mathsf{Adv}_{\mathcal{B}}^{k\text{-Lin}} + 2/p$$

and $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + k^2 \cdot \mathrm{poly}(\lambda, n)$ *where* $\mathrm{poly}(\lambda, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

The proof is a simpler case of the proof of Lemma 3, we omit it here.

Lemma 3 (right subgroup indistinguishability from k-Lin). *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that:*

$$\mathsf{Adv}^{\mathrm{RS}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{k\text{-Lin}}_{\mathcal{B}} + 2/p$$

and $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + k^2 \cdot \mathrm{poly}(\lambda, n)$ *where* $\mathrm{poly}(\lambda, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

We may rewrite the corresponding advantage function as:

$$\mathsf{Adv}^{\mathrm{RS}}_{\mathcal{A}}(\lambda) := |\Pr[\,\mathcal{A}(\mathsf{pp}, h^*, \mathbf{g}\cdot\hat{\mathbf{g}}, \mathbf{h}) = 1\,] - \Pr[\,\mathcal{A}(\mathsf{pp}, h^*, \mathbf{g}\cdot\hat{\mathbf{g}}, \mathbf{h}\cdot\hat{\mathbf{h}}) = 1\,]|$$

where

$$\begin{aligned}
&(\mathsf{pp},\mathsf{sp}) \leftarrow \mathsf{SampP}(1^\lambda, 1^n);\ \mathbf{s}, \mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k;\ \hat{s}, \hat{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*;\\
h^* &:= [\mathbf{a}^\perp]_2;\\
\mathbf{g}\cdot\hat{\mathbf{g}} &:= \Big([\mathbf{A}\mathbf{s} + \mathbf{b}^\perp\hat{s}]_1, [\mathbf{W}_1^{\top}(\mathbf{A}\mathbf{s} + \mathbf{b}^\perp\hat{s})]_1, \ldots, [\mathbf{W}_n^{\top}(\mathbf{A}\mathbf{s} + \mathbf{b}^\perp\hat{s})]_1\Big);\\
\mathbf{h} &:= \Big([\mathbf{B}\mathbf{r}]_2, [\mathbf{W}_1\mathbf{B}\mathbf{r}]_2, \ldots, [\mathbf{W}_n\mathbf{B}\mathbf{r}]_2\Big);\\
\mathbf{h}\cdot\hat{\mathbf{h}} &:= \Big([\mathbf{B}\mathbf{r} + \mathbf{a}^\perp\hat{r}]_2, [\mathbf{W}_1(\mathbf{B}\mathbf{r} + \mathbf{a}^\perp\hat{r})]_2, \ldots, [\mathbf{W}_n(\mathbf{B}\mathbf{r} + \mathbf{a}^\perp\hat{r})]_2\Big).
\end{aligned}$$

Proof. The adversary $\mathcal{B}$ samples $(\mathbf{A}, \mathbf{a}^\perp) \leftarrow \mathcal{D}_k$ along with $\mathbf{W}_1, \ldots, \mathbf{W}_n \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(k+1)\times(k+1)}$. Recall that $(\mathbf{B}, \mathbf{a}^\perp)$ is a basis for $\mathbb{Z}_p^{k+1}$, so $\{\mathbf{B}\mathbf{r} + \mathbf{a}^\perp\hat{r} : \mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k, \hat{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*\}$ is statistically close to the uniform distribution. Adversary $\mathcal{B}$ then gets as input

$$\big(\,(p, G_1, G_2, G_T, g_1, g_2, e), [\mathbf{B}]_2, [\mathbf{B}\mathbf{r} + \mathbf{a}^\perp\hat{r}]_2\,\big)$$

where either $\hat{r} = 0$ or $\hat{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$, and proceeds as follows:

Simulating pp, h^*. Output

$$\begin{array}{ll} [\mathbf{A}]_1,\ [\mathbf{W}_1^{\top}\mathbf{A}]_1,\ \ldots,\ [\mathbf{W}_n^{\top}\mathbf{A}]_1 & \\ & \text{and} \quad [\mathbf{a}^\perp]_2 \\ [\mathbf{B}]_2,\ [\mathbf{W}_1\mathbf{B}]_2,\ \ldots,\ [\mathbf{W}_n\mathbf{B}]_2 & \end{array}$$

Simulating $[\mathbf{A}\mathbf{s} + \mathbf{b}^\perp\hat{s}]_1, [\mathbf{W}_i^{\top}(\mathbf{A}\mathbf{s} + \mathbf{b}^\perp\hat{s})]_1$. Note that $\mathcal{B}$ does not know $\mathbf{b}^\perp$. Instead, $\mathcal{B}$ samples $\tilde{\mathbf{s}} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{k+1}$ and outputs

$$[\tilde{\mathbf{s}}]_1,\ [\mathbf{W}_1^{\top}\tilde{\mathbf{s}}]_1,\ \ldots,\ [\mathbf{W}_n^{\top}\tilde{\mathbf{s}}]_1.$$

Observe that $\mathbf{A}\mathbf{s} + \mathbf{b}^\perp\hat{s}$ is statistically close to the uniform vector $\tilde{\mathbf{s}}$ as long as $\mathbf{b}^\perp \notin \mathrm{span}(\mathbf{A})$ and $\hat{s} \leftarrow_{\mathrm{R}} \mathbb{Z}_p$.

Simulating the challenge. Upon receiving a k-Lin challenge, $\mathcal{B}$ outputs

$$[\mathbf{B}\mathbf{r} + \mathbf{a}^\perp\hat{r}]_2,\ [\mathbf{W}_1(\mathbf{B}\mathbf{r} + \mathbf{a}^\perp\hat{r})]_2,\ \ldots,\ [\mathbf{W}_n(\mathbf{B}\mathbf{r} + \mathbf{a}^\perp\hat{r})]_2$$

where either $\hat{r} = 0$ or $\hat{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p$.

Observe that:

- if $\hat{r} = 0$, then we can write the output challenge as

  $$[\mathbf{B}\mathbf{r}]_2,\ [\mathbf{W}_1\mathbf{B}\mathbf{r}]_2,\ \ldots,\ [\mathbf{W}_n\mathbf{B}\mathbf{r}]_2.$$

  which equals $\mathbf{h}$; we obtain the left distribution in the statement of the lemma;
- if $\hat{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p$, then we can write the output challenge as

  $$[\mathbf{B}\mathbf{r} + \mathbf{a}^\perp\hat{r}]_2,\ [\mathbf{W}_1(\mathbf{B}\mathbf{r} + \mathbf{a}^\perp\hat{r})]_2,\ \ldots,\ [\mathbf{W}_n(\mathbf{B}\mathbf{r} + \mathbf{a}^\perp\hat{r})]_2.$$

  which equals $\mathbf{h}\cdot\hat{\mathbf{h}}$; we obtain the right distribution in the statement of the lemma.

Typically, we sample $\hat{s}, \hat{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$ for $\widehat{\mathsf{SampG}}(\mathsf{pp}, \mathsf{sp})$ and $\widehat{\mathsf{SampH}}(\mathsf{pp}, \mathsf{sp})$; this yields a $2/p$ negligible difference in the advantage. The lemma then follows readily. □

Lemma 4 (parameter-hiding). *The following distributions are identically distributed*

$$\left\{ pp, [\mathbf{a}^\perp]_2, \begin{array}{l} [\mathbf{b}^\perp \hat{s}]_1, [\mathbf{W}_1^\top \mathbf{b}^\perp \hat{s}]_1, \ldots, [\mathbf{W}_n^\top \mathbf{b}^\perp \hat{s}]_1 \\ [\mathbf{a}^\perp \hat{r}]_2, [\mathbf{W}_1 \mathbf{a}^\perp \hat{r}]_2, \ldots, [\mathbf{W}_n \mathbf{a}^\perp \hat{r}]_2 \end{array} \right\} \quad \textit{and}$$

$$\left\{ pp, [\mathbf{a}^\perp]_2, \begin{array}{l} [\mathbf{b}^\perp \hat{s}]_1, [(\mathbf{W}_1^\top \mathbf{b}^\perp + \hat{u}_1 \mathbf{b}^\perp)\hat{s}]_1, \ldots, [(\mathbf{W}_n^\top \mathbf{b}^\perp + \hat{u}_n \mathbf{b}^\perp)\hat{s}]_1 \\ [\mathbf{a}^\perp \hat{r}]_2, [(\mathbf{W}_1 \mathbf{a}^\perp + \hat{u}_1 \mathbf{a}^\perp)\hat{r}]_2, \ldots, [(\mathbf{W}_n \mathbf{a}^\perp + \hat{u}_n \mathbf{a}^\perp)\hat{r}]_2 \end{array} \right\}$$

where $(pp, sp) \leftarrow \mathsf{SampP}(1^\lambda, 1^n)$*,* $\hat{s}, \hat{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$ *and* $\hat{u}_1, \ldots, \hat{u}_n \leftarrow_{\mathrm{R}} \mathbb{Z}_p$*.*

Proof. Fix $g_1, g_2, (\mathbf{A}, \mathbf{a}^\perp), (\mathbf{B}, \mathbf{b}^\perp), \hat{s}, \hat{r}$; that is, we prove that the statement holds for all $g_1, g_2, (\mathbf{A}, \mathbf{a}^\perp), (\mathbf{B}, \mathbf{b}^\perp), \hat{s}, \hat{r}$. Set $\mathbf{V} := \mathbf{a}^\perp \mathbf{b}^{\perp\top} \in \mathbb{Z}_p^{(k+1)\times(k+1)}$ which satisfies the following properties:

$$\mathbf{V}^\top \mathbf{A} = \mathbf{0} \quad \text{and} \quad \mathbf{V}\mathbf{B} = \mathbf{0} \tag{5}$$

$$\mathbf{V}\mathbf{a}^\perp = (\mathbf{a}^{\perp\top}\mathbf{b}^\perp)\mathbf{a}^\perp \quad \text{and} \quad \mathbf{V}^\top \mathbf{b}^\perp = (\mathbf{a}^{\perp\top}\mathbf{b}^\perp)\mathbf{b}^\perp \tag{6}$$

Eqn. (6) basically says that $\mathbf{a}^\perp$ and $\mathbf{b}^\perp$ are the respective eigenvectors of $\mathbf{V}$ and $\mathbf{V}^\top$. Now, consider the following "change of variables" in the first distribution, namely, replace

$$\mathbf{W}_i \text{ with } \mathbf{W}_i + \hat{u}_i (\mathbf{a}^{\perp\top}\mathbf{b}^\perp)^{-1}\mathbf{V}, \; i = 1, \ldots, n.$$

Clearly, this does not change the first distribution. Now, observe that

$$\left[(\mathbf{W}_i + \hat{u}_i(\mathbf{a}^{\perp\top}\mathbf{b}^\perp)^{-1}\mathbf{V})^\top \mathbf{A}\right]_1 = [\mathbf{W}_i^\top \mathbf{A}]_1;$$
$$\left[(\mathbf{W}_i + \hat{u}_i(\mathbf{a}^{\perp\top}\mathbf{b}^\perp)^{-1}\mathbf{V})\mathbf{B}\right]_2 = [\mathbf{W}_i \mathbf{B}]_2$$

where we use (5) in the last equalities. That is, pp remains unchanged. In addition, we have

$$\left[(\mathbf{W}_i + \hat{u}_i(\mathbf{a}^{\perp\top}\mathbf{b}^\perp)^{-1}\mathbf{V})^\top \mathbf{b}^\perp\right]_1 = [\mathbf{W}_i^\top \mathbf{b}^\perp + \hat{u}_i \mathbf{b}^\perp]_1;$$
$$\left[(\mathbf{W}_i + \hat{u}_i(\mathbf{a}^{\perp\top}\mathbf{b}^\perp)^{-1}\mathbf{V})\mathbf{a}^\perp\right]_2 = [\mathbf{W}_i \mathbf{a}^\perp + \hat{u}_i \mathbf{a}^\perp]_2$$

where we use (6) in the last equalities. Indeed, this is exactly the second distribution. □

5 Predicate Encodings

In this section, we describe a refinement of the predicate encodings from [43, 2] which we use in this work. We refer to Section 1.1 for an overview of the refinement.

Predicate encodings. Fix a predicate $\mathsf{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$. A $\mathbb{Z}_p$*-bilinear predicate encoding* for P is a tuple of deterministic algorithms $(\mathsf{sE}, \mathsf{rE}, \mathsf{kE}, \mathsf{sD}, \mathsf{rD})$ satisfying the following properties:

(linearity.) For all $(x, y) \in \mathcal{X} \times \mathcal{Y}$, the functions $\mathsf{sE}(x, \cdot)$, $\mathsf{rE}(y, \cdot)$, $\mathsf{kE}(y, \cdot)$, $\mathsf{sD}(x, y, \cdot)$, $\mathsf{rD}(x, y, \cdot)$ are $\mathbb{Z}_p$-linear.

(restricted α-reconstruction.) For all $(x, y) \in \mathcal{X} \times \mathcal{Y}$ such that $\mathsf{P}(x, y) = 1$ and for all $\mathbf{w} \in \mathcal{W}$:

$$\mathsf{sD}(x, y, \mathsf{sE}(x, \mathbf{w})) = \mathsf{rD}(x, y, \mathsf{rE}(y, \mathbf{w})) \quad \text{and} \quad \mathsf{rD}(x, y, \mathsf{kE}(y, \alpha)) = \alpha$$

(α-privacy.) For all $(x, y) \in \mathcal{X} \times \mathcal{Y}$ such that $\mathsf{P}(x, y) = 0$, and for all $\alpha \in \mathbb{Z}_p$, the joint distribution $\{\mathsf{sE}(x, \mathbf{w}), \mathsf{kE}(y, \alpha) + \mathsf{rE}(y, \mathbf{w})\}$ *perfectly* hides α. That is, for all $\alpha \in \mathbb{Z}_p$, the following joint distributions[2] are *identically*

[2] Note that since $\mathsf{kE}(y, \cdot)$ is $\mathbb{Z}_p$-linear, we have $\mathsf{kE}(y, 0) + \mathsf{rE}(y, \mathbf{w}) = \mathsf{rE}(y, \mathbf{w})$.

distributed:

$$\{x, y, \alpha, \mathsf{sE}(x, \mathbf{w}), \mathsf{kE}(y, \alpha) + \mathsf{rE}(y, \mathbf{w})\} \quad \text{and} \quad \{x, y, \alpha, \mathsf{sE}(x, \mathbf{w}), \mathsf{rE}(y, \mathbf{w})\}$$

where the randomness is taken over $\mathbf{w} \leftarrow_{\mathrm{R}} \mathcal{W}$.

Remark 3. Given a predicate encoding as defined above, we can construct an encoding $(\mathsf{rE}', \mathsf{sE}')$ which achieves the notion in [43, 2] by considering:

$$\mathsf{sE}' = \mathsf{sE} \quad \text{and} \quad \mathsf{rE}'(y, \alpha, \mathbf{w}, r) = \big(r, \mathsf{kE}(y, \alpha) + r \cdot \mathsf{rE}(y, \mathbf{w})\big)$$

Note that rE' leaks no information about $\mathbf{w}$ when $r = 0$ which trivially yields the $\mathbf{w}$-hiding property in [43] (aka parameter-hiding in [2]). Here, we use the fact that kE does not depend on $\mathbf{w}$.

Example: equality. Fix a prime integer p. Consider the equality predicate where $\mathcal{X} = \mathcal{Y} = \mathbb{Z}_p$ and $\mathsf{P}(x, y) = 1$ iff $x = y$. The following is a predicate encoding for equality used in [6, 32]:

$$\mathsf{sE}(x, (w_1, w_2)) := w_1 + w_2 x \qquad \mathsf{rE}(y, (w_1, w_2)) := w_1 + w_2 y \qquad \mathsf{kE}(y, \alpha) := \alpha$$

$$\mathsf{sD}(x, y, c) = c \qquad \mathsf{rD}(x, y, k) = k$$

When $x = y$, $w_1 + w_2 x = w_1 + w_2 y$ and we can reconstruct α. For α-privacy, we exploit the fact that $(w_1 + w_2 x, w_1 + w_2 y)$ are pairwise independent when $x \neq y$.

6 ABE from Dual System Groups and Predicate Encodings

Starting from a predicate encoding for P, we construct an ABE for P using dual system groups. We refer to Section 1.1 for an overview of the scheme, which is of the form:

$$\mathsf{mpk} := \big(g_1,\, g_1^{\mathbf{w}},\, e(g_1, g_1)^{\alpha}\big) \qquad \mathsf{sk}_y := \big(g_1^r,\, g_1^{\mathsf{kE}(y,\alpha)+r\cdot\mathsf{rE}(y,\mathbf{w})}\big) \qquad \mathsf{ct}_x := \big(g_1^s,\, g_1^{s\cdot\mathsf{sE}(x,\mathbf{w})},\, e(g_1, g_1)^{\alpha s} \cdot m\big)$$

We will generate mpk using $\mathsf{SampP}(1^\lambda, 1^n)$, where $\mathbf{w} \in \mathbb{Z}_p^n$. We will use $\mathsf{SampG}(\mathsf{pp})$ to generate the terms $(g_1^s, g_1^{s\mathbf{w}})$ in the ciphertext, from which we can compute $(g_1^s, g_1^{s\cdot\mathsf{sE}(x,\mathbf{w})})$ by linearity of $\mathsf{sE}(x, \cdot)$. Similarly, we use $\mathsf{SampH}(\mathsf{pp})$ to generate the terms $(g_1^r, g_1^{r\mathbf{w}})$ in the secret key, from which we can compute $(g_1^r, g_1^{r\cdot\mathsf{rE}(y,\mathbf{w})})$. We replace g_1^α with $\mathsf{msk} \leftarrow_{\mathrm{R}} \mathbb{H}$.

6.1 Construction

$\mathsf{Setup}(1^\lambda, 1^n)$: On input $(1^\lambda, 1^n)$, first sample

$$(\mathsf{pp}, \mathsf{sp}) \leftarrow \mathsf{SampP}(1^\lambda, 1^n).$$

Pick $\mathsf{msk} \leftarrow_{\mathrm{R}} \mathbb{H}$ and output the master public and secret key pair

$$\mathsf{mpk} := \big(\,\mathsf{pp},\, \mu(\mathsf{msk})\,\big) \quad \text{and} \quad \mathsf{msk}.$$

$\mathsf{Enc}(\mathsf{mpk}, x, m)$: On input $x \in \mathcal{X}$ and $m \in \mathbb{G}_T$, sample

$$(g_0, g_1, \ldots, g_n) \leftarrow \mathsf{SampG}(\mathsf{pp}; s),\; g_T' \leftarrow \mathsf{SampGT}(\mu(\mathsf{msk}); s)$$

and output[3]

$$\mathsf{ct}_x := \big(\,C_0 := g_0,\, \mathbf{C}_1 := \mathsf{sE}(x, (g_1, \ldots, g_n)),\, C' := g_T' \cdot m\,\big).$$

[3] See Remark 1 for an explanation of the function $\mathsf{sE}(x, (g_1, \ldots, g_n))$.

$\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, y)$: On input $y \in \mathcal{Y}$, sample

$$(h_0, h_1, \ldots, h_n) \leftarrow \mathsf{SampH}(\mathsf{pp})$$

and output

$$\mathsf{sk}_y := \big(\, K_0 := h_0,\ \mathbf{K}_1 := \mathsf{kE}(y, \mathsf{msk}) \cdot \mathsf{rE}(y, (h_1, \ldots, h_n))\,\big)\,.$$

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_y, \mathsf{ct}_x)$: Compute

$$e(g_0, \mathsf{msk}) \leftarrow e(C_0, \mathsf{rD}(x, y, \mathbf{K}_1)) / e(\mathsf{sD}(x, y, \mathbf{C}_1), K_0)$$

and recover the message as

$$m \leftarrow C' \cdot e(g_0, \mathsf{msk})^{-1} \in \mathbb{G}_T.$$

Correctness. For all $(x, y) \in \mathcal{X} \times \mathcal{Y}$ such that $\mathsf{P}(x, y) = 1$, we have

$$\begin{aligned}
& e(C_0, \mathsf{rD}(x, y, \mathbf{K}_1)) \\
&= e(g_0, \mathsf{rD}(x, y, \mathsf{rE}(y, (h_1, \ldots, h_n)))) \cdot e(g_0, \mathsf{rD}(x, y, \mathsf{kE}(y, \mathsf{msk}))) && \text{linearity of } \mathsf{rD}(x, y, \cdot) \text{ and } e(g_0, \cdot) \\
&= e(g_0, \mathsf{rD}(x, y, \mathsf{rE}(y, (h_1, \ldots, h_n)))) \cdot e(g_0, \mathsf{msk}) && \alpha\text{-reconstruction} \\
&= \mathsf{rD}(x, y, \mathsf{rE}(y, (e(g_0, h_1), \ldots, e(g_0, h_n)))) \cdot e(g_0, \mathsf{msk}) && e(g_0, \cdot) \text{ and } \mathsf{rD}(x, y, \mathsf{rE}(y, \cdot)) \text{ commute} \\
&= \mathsf{rD}(x, y, \mathsf{rE}(y, (e(g_1, h_0), \ldots, e(g_n, h_0)))) \cdot e(g_0, \mathsf{msk}) && \text{associativity in DSG} \\
&= \mathsf{sD}(x, y, \mathsf{sE}(x, (e(g_1, h_0), \ldots, e(g_n, h_0)))) \cdot e(g_0, \mathsf{msk}) && \alpha\text{-reconstruction} \\
&= e(\mathsf{sD}(x, y, \mathsf{sE}(x, (g_1, \ldots, g_n))), h_0) \cdot e(g_0, \mathsf{msk}) && e(\cdot, h_0) \text{ and } \mathsf{sD}(x, y, \mathsf{sE}(y, \cdot)) \text{ commute} \\
&= e(\mathsf{sD}(x, y, \mathbf{C}_1), K_0) \cdot e(g_0, \mathsf{msk})
\end{aligned}$$

In line 4 and line 7, we use the fact that the functions $e(g_0, \cdot)$, $e(\cdot, h_0)$ and $\mathsf{sD}(x, y, \mathsf{sE}(y, \cdot))$ commute with linear functions. That is, given a $\mathbb{Z}_p$-linear function $L : \mathbb{Z}_p^n \to \mathbb{Z}_p$ given by $(w_1, \ldots, w_n) \mapsto a_1 w_1 + \cdots + a_n w_n$, we have:

$$e(g_0, L(h_1, \ldots, h_n)) = e(g_0, h_1^{a_1} \cdots h_n^{a_n}) = e(g_0, h_1)^{a_1} \cdots e(g_0, h_n)^{a_n} = L(e(g_0, h_1), \ldots, e(g_0, h_n))$$

Finally, by *projective*, $g_T' = e(g_0, \mathsf{msk})$. Correctness follows readily.

6.2 Proof of Security

We prove the following theorem:

Theorem 1. *Under the left and right subgroup indistinguishability (described in Section 3), the ABE scheme described in Section 6.1 is adaptively secure (in the sense of Definition 2.1). More precisely, for any adversary $\mathcal{A}$ that makes at most q key queries against the ABE scheme, there exist adversaries $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ such that:*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathrm{ABE}}(\lambda) \le \mathsf{Adv}_{\mathcal{B}_1}^{\mathrm{LS}}(\lambda) + q \cdot \mathsf{Adv}_{\mathcal{B}_2}^{\mathrm{RS}}(\lambda) + q \cdot \mathsf{Adv}_{\mathcal{B}_3}^{\mathrm{RS}}(\lambda)$$

and

$$\max\{\mathsf{Time}(\mathcal{B}_1), \mathsf{Time}(\mathcal{B}_2), \mathsf{Time}(\mathcal{B}_3)\} \approx \mathsf{Time}(\mathcal{A}) + q \cdot \mathrm{poly}(\lambda, n)$$

where poly(λ, n) *is independent of*$\mathsf{Time}(\mathcal{A})$.

The proof follows via a series of games, analogous to that in [14, 43, 42, 32], and outlined in Fig. 7. We first define two auxiliary algorithms and then the semi-functional distributions, upon which we can describe the games.

Auxiliary algorithms. We consider the following algorithms:

$\widehat{\mathsf{Enc}}(\mathsf{pp}, x, m; \mathsf{msk}, \mathbf{t})$: On input $x \in \mathcal{X}$, $m \in \mathbb{G}_T$, and $\mathbf{t} := (T_0, T_1, \ldots, T_n) \in \mathbb{G}^{n+1}$, output

$$\mathsf{ct}_x := \big(\, T_0,\ \mathsf{sE}(x, (T_1, \ldots, T_n)),\ e(T_0, \mathsf{msk}) \cdot m \,\big)$$

$\widehat{\mathsf{KeyGen}}(\mathsf{pp}, \mathsf{msk}', y; \mathbf{t})$: On input $\mathsf{msk}' \in \mathbb{H}$, $y \in \mathcal{Y}$, and $\mathbf{t} := (T_0, T_1, \ldots, T_n) \in \mathbb{H}^{n+1}$, output

$$\mathsf{sk}_y := \big(\, T_0,\ \mathsf{kE}(y, \mathsf{msk}') \cdot \mathsf{rE}(y, (T_1, \ldots, T_n)) \,\big)\,.$$

In all the proofs and figures that follow, we denote $\mathsf{sE}(x, (T_1, \ldots, T_n))$ by $\mathsf{sE}(x, \mathbf{t})$ for notational convenience, and we define $\mathsf{rE}(y, \mathbf{t})$ analogously.

Auxiliary distributions.

Semi-functional master secret key.

$$\widehat{\mathsf{msk}} := \mathsf{msk} \cdot (h^*)^{\alpha},$$

where $\boxed{\alpha \leftarrow_{\mathrm{R}} \mathbb{Z}_p}$.

Semi-functional ciphertext.

$$\widehat{\mathsf{Enc}}(\mathsf{pp}, x, m; \mathsf{msk}, \boxed{\mathbf{g} \cdot \hat{\mathbf{g}}}),$$

where $\boxed{\mathbf{g} \leftarrow \mathsf{SampG}(\mathsf{pp}) \text{ and } \hat{\mathbf{g}} \leftarrow \widehat{\mathsf{SampG}}(\mathsf{pp}, \mathsf{sp})}$.

Pseudo-normal secret key.

$$\widehat{\mathsf{KeyGen}}(\mathsf{pp}, \mathsf{msk}, y; \boxed{\mathbf{h} \cdot \hat{\mathbf{h}}}),$$

where fresh $\boxed{\mathbf{h} \leftarrow \mathsf{SampH}(\mathsf{pp}) \text{ and } \hat{\mathbf{h}} \leftarrow \widehat{\mathsf{SampH}}(\mathsf{pp}, \mathsf{sp})}$ are chosen for each secret key.

Pseudo-SF secret key.

$$\widehat{\mathsf{KeyGen}}(\mathsf{pp}, \boxed{\widehat{\mathsf{msk}}}, y; \mathbf{h} \cdot \hat{\mathbf{h}}),$$

where fresh $\mathbf{h} \leftarrow \mathsf{SampH}(\mathsf{pp})$ and $\hat{\mathbf{h}} \leftarrow \widehat{\mathsf{SampH}}(\mathsf{pp}, \mathsf{sp})$ are chosen for each secret key.

Semi-functional secret key.

$$\widehat{\mathsf{KeyGen}}(\mathsf{pp}, \widehat{\mathsf{msk}}, y; \boxed{\mathbf{h}}),$$

where a fresh $\boxed{\mathbf{h} \leftarrow \mathsf{SampH}(\mathsf{pp})}$ is chosen for each secret key. We note that the semi-functional key generation algorithm is identical to the normal key generation except that it replaces msk with $\widehat{\mathsf{msk}}$ as input.

Game sequence. We present a series of games. We write $\mathsf{Adv}_{\mathrm{xxx}}(\lambda)$ to denote the advantage of $\mathcal{A}$ in $\mathsf{Game}_{\mathrm{xxx}}$.

- Game_0: is the real security game (c.f. Section 2.1).
- Game_1: is the same as Game_0 except that the challenge ciphertext is semi-functional.
- $\mathsf{Game}_{2,i,1}$ for i from 1 to q, $\mathsf{Game}_{2,i,1}$ is the same as Game_1 except that the first $i-1$ keys are semi-functional, the last $q-i$ keys are normal while the i'th key is pseudo-normal.
- $\mathsf{Game}_{2,i,2}$ for i from 1 to q, $\mathsf{Game}_{2,i,2}$ is the same as Game_1 except that the first $i-1$ keys are semi-functional, the last $q-i$ keys are normal while the i'th key is pseudo-SF.

game	ciphertext $(C_0, \mathbf{C}_1, C')$	secret key $(K_0, \mathbf{K}_1)$	justification	remark
0	$(1, \mathbf{1}, 1)$	$(1, (h^*)^{\mathsf{kE}(y,0)} \cdot \mathbf{1})$	$\mathbf{1} = (h^*)^{\mathsf{kE}(y,0)}$	actual scheme
1	$\boxed{(\hat{g}_0, \mathsf{sE}(x, \hat{\mathbf{g}}), e(\hat{g}_0, \mathsf{msk}))}$	$(1, (h^*)^{\mathsf{kE}(y,0)} \cdot \mathbf{1})$	left subgroup indist.	normal to SF ct_x
2.i.1	$(\hat{g}_0, \mathsf{sE}(x, \hat{\mathbf{g}}), e(\hat{g}_0, \mathsf{msk}))$	$(\boxed{\hat{h}_0}, (h^*)^{\mathsf{kE}(y,0)} \cdot \boxed{\mathsf{rE}(y, \hat{\mathbf{h}})})$	right subgroup indist.	normal to pseudo-normal sk_y
2.i.2	$(\hat{g}_0, \mathsf{sE}(x, \hat{\mathbf{g}}), e(\hat{g}_0, \mathsf{msk}))$	$(\hat{h}_0, \boxed{(h^*)^{\mathsf{kE}(y,\alpha)}} \cdot \mathsf{rE}(y, \hat{\mathbf{h}}))$	α-privacy	pseudo-normal to pseudo-SF sk_y
2.i.3	$(\hat{g}_0, \mathsf{sE}(x, \hat{\mathbf{g}}), e(\hat{g}_0, \mathsf{msk}))$	$(\boxed{1}, (h^*)^{\mathsf{kE}(y,\alpha)} \cdot \boxed{\mathbf{1}})$	right subgroup indist.	pseudo-SF to SF sk_y
3	$(\hat{g}_0, \mathsf{sE}(x, \hat{\mathbf{g}}), \boxed{\text{random}})$	$(1, (h^*)^{\mathsf{kE}(y,\alpha)} \cdot \mathbf{1})$		

Fig. 7. Sequence of games in the "semi-functional" space. We omitted the normal components: those sampled using $\mathsf{SampG}, \mathsf{SampH}$, and we omitted $e(g_0, \mathsf{msk}) \cdot m$ in C' and $\mathsf{kE}(y, \mathsf{msk})$ in sk_y. We drew a box to highlight the differences between each game and the preceding one, and games $2.i.$x refer to the i'th secret key. The semi-functional components of the keys transition from $(h^*)^{\mathsf{kE}(y,0)}$ to $(h^*)^{\mathsf{kE}(y,\alpha)}$. For the final transition, we use the fact that $e(\hat{g}_0, \mathsf{msk})$ is statistically random given $\mathsf{msk} \cdot (h^*)^{\alpha}$.

- $\mathsf{Game}_{2,i,3}$ for i from 1 to q, $\mathsf{Game}_{2,i,3}$ is the same as Game_1 except that the first i keys are semi-functional, the last $q - i$ keys are normal.
- Game_3: is the same as $\mathsf{Game}_{2,q,3}$, except that the challenge ciphertext is a semi-functional encryption of a random message in $\mathbb{G}_T$.

In Game_3, the view of the adversary is statistically independent of the challenge bit b. Hence, $\mathsf{Adv}_3(\lambda) = 0$. We complete the proof by establishing the following sequence of lemmas. In particular, the proofs of lemmas 5, 6, 8, 9 are the same as those of lemmas 1, 2, 5, 6 in [14, Section 4].

Lemma 5 (normal to SF ciphertext: Game_0 to Game_1). *For any adversary $\mathcal{A}$ that makes at most q key queries, there exists an adversary $\mathcal{B}_1$ such that*

$$|\mathsf{Adv}_0(\lambda) - \mathsf{Adv}_1(\lambda)| \leq \mathsf{Adv}^{\mathrm{LS}}_{\mathcal{B}_1}(\lambda)$$

and $\mathsf{Time}(\mathcal{B}_1) \approx \mathsf{Time}(\mathcal{A}) + q \cdot \mathrm{poly}(\lambda, n)$ *where* $\mathrm{poly}(\lambda, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. The adversary $\mathcal{B}_1$ gets as input

$$(\mathsf{pp}, \mathbf{t}),$$

where $\mathbf{t}$ is either $\mathbf{g}$ or $\mathbf{g} \cdot \hat{\mathbf{g}}$ and

$$\mathbf{g} \leftarrow \mathsf{SampG}(\mathsf{pp}),\ \hat{\mathbf{g}} \leftarrow \widehat{\mathsf{SampG}}(\mathsf{pp}, \mathsf{sp}),$$

and proceeds as follows:

Setup. Pick $\mathsf{msk} \leftarrow_{\mathrm{R}} \mathbb{H}$ and output

$$\mathsf{mpk} := \big(\, \mathsf{pp},\ \mu(\mathsf{msk}) \,\big).$$

Key Queries. On input the j'th secret key query y, output

$$\mathsf{sk}_y \leftarrow \widehat{\mathsf{KeyGen}}(\mathsf{pp}, \mathsf{msk}, y; \mathsf{SampH}(\mathsf{pp})).$$

Ciphertext. Upon receiving a challenge attribute x^* and two equal-length messages m_0, m_1, pick $\beta \leftarrow_{\mathrm{R}} \{0,1\}$ and output

$$\mathsf{ct}_{x^*} \leftarrow \widehat{\mathsf{Enc}}(\mathsf{pp}, x^*, m_\beta; \mathsf{msk}, \mathbf{t}).$$

Guess. When $\mathcal{A}$ halts with output β', $\mathcal{B}_1$ outputs 1 if $\beta = \beta'$ and 0 otherwise.

Observe that when $\mathbf{t} = \mathbf{g}$, ct_{x^*} is properly distributed as $\mathsf{Enc}(\mathsf{mpk}, x^*, m_\beta)$ from *projective*, the output is identical to that in Game_0; and when $\mathbf{t} = \mathbf{g}\cdot\hat{\mathbf{g}}$, the output is identical to that in Game_1. We may therefore conclude that: $|\mathsf{Adv}_0(\lambda) - \mathsf{Adv}_1(\lambda)| \le \mathsf{Adv}^{\mathrm{LS}}_{\mathcal{B}_1}(\lambda)$. □

Lemma 6 (normal to pseudo-normal keys: $\mathsf{Game}_{2,i-1,3}$ to $\mathsf{Game}_{2,i,1}$). *For $i = 1,\ldots,q$, for any adversary $\mathcal{A}$ that makes at most q key queries, there exists an adversary $\mathcal{B}_2$ such that*

$$|\mathsf{Adv}_{2,i-1,3}(\lambda) - \mathsf{Adv}_{2,i,1}(\lambda)| \le \mathsf{Adv}^{\mathrm{RS}}_{\mathcal{B}_2}(\lambda)$$

and $\mathsf{Time}(\mathcal{B}_2) \approx \mathsf{Time}(\mathcal{A}) + q\cdot\mathrm{poly}(\lambda, n)$ *where* $\mathrm{poly}(\lambda, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$. *(We note that* $\mathsf{Game}_{2,0,3}$ *is identical to* Game_1*.)*

Proof. The adversary $\mathcal{B}_2$ gets as input

$$(\mathsf{pp}, h^*, \mathbf{g}\cdot\hat{\mathbf{g}}, \mathbf{t}),$$

where $\mathbf{t}$ is either $\mathbf{h}$ or $\mathbf{h}\cdot\hat{\mathbf{h}}$ and

$$\mathbf{h} \leftarrow \mathsf{SampH}(\mathsf{pp}),\ \hat{\mathbf{h}} \leftarrow \widehat{\mathsf{SampH}}(\mathsf{pp}, \mathsf{sp}),$$

and proceeds as follows:

Setup. Pick $\mathsf{msk} \leftarrow_{\mathrm{R}} \mathbb{H}$, $\alpha \leftarrow_{\mathrm{R}} \mathbb{Z}_p$ and set $\widehat{\mathsf{msk}} := \mathsf{msk}\cdot(h^*)^\alpha$. Output

$$\mathsf{mpk} := \big(\,\mathsf{pp},\ \mu(\mathsf{msk})\,\big).$$

Key Queries. On input the j'th secret key query y, output

$$\mathsf{sk}_y \leftarrow \begin{cases} \widehat{\mathsf{KeyGen}}(\mathsf{pp}, \widehat{\mathsf{msk}}, y; \mathsf{SampH}(\mathsf{pp})) & \text{if } j < i \\ \widehat{\mathsf{KeyGen}}(\mathsf{pp}, \mathsf{msk}, y; \mathbf{t}) & \text{if } j = i \\ \widehat{\mathsf{KeyGen}}(\mathsf{pp}, \mathsf{msk}, y; \mathsf{SampH}(\mathsf{pp})) & \text{if } j > i \end{cases}.$$

Ciphertext. Upon receiving a challenge identity x^* and two equal-length messages m_0, m_1, pick $\beta \leftarrow_{\mathrm{R}} \{0,1\}$ and output

$$\mathsf{ct}_{x^*} \leftarrow \widehat{\mathsf{Enc}}(\mathsf{pp}, x^*, m_\beta; \mathsf{msk}, \mathbf{g}\cdot\hat{\mathbf{g}}).$$

Guess. When $\mathcal{A}$ halts with output β', $\mathcal{B}_2$ outputs 1 if $\beta = \beta'$ and 0 otherwise.

Observe that when $\mathbf{t} = \mathbf{h}$, the output is identical to that in $\mathsf{Game}_{2,i-1,3}$; and when $\mathbf{t} = \mathbf{h}\cdot\hat{\mathbf{h}}$, the output is identical to that in $\mathsf{Game}_{2,i,1}$. We may therefore conclude that: $|\mathsf{Adv}_{2,i-1,3}(\lambda) - \mathsf{Adv}_{2,i,1}(\lambda)| \le \mathsf{Adv}^{\mathrm{RS}}_{\mathcal{B}_2}(\lambda)$. □

Lemma 7 (pseudo-normal to pseudo-SF keys: $\mathsf{Game}_{2,i,1}$ to $\mathsf{Game}_{2,i,2}$). *For $i = 1,\ldots,q$, we have*

$$|\mathsf{Adv}_{2,i,1}(\lambda) - \mathsf{Adv}_{2,i,2}(\lambda)| = 0.$$

Proof. Observe that the only difference between $\mathsf{Game}_{2,i,1}$ and $\mathsf{Game}_{2,i,2}$ lies in that we replace msk in $\mathsf{Game}_{2,i,1}$ with $\widehat{\mathsf{msk}}$ in $\mathsf{Game}_{2,i,2}$ as input for the i'th secret key query, where $\mathsf{msk} \leftarrow_{\mathrm{R}} \mathbb{H}$, $\alpha \leftarrow_{\mathrm{R}} \mathbb{Z}_p$ and $\widehat{\mathsf{msk}} := \mathsf{msk}\cdot(h^*)^\alpha$. Thus, it suffices to establish the following:

> *Claim.* For all α, all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, where $\mathsf{P}(x, y) = 0$, the following distributions are identically distributed:
>
> $$\{\mathsf{pp}, \mathsf{msk}, (h^*)^\alpha, \widehat{\mathsf{Enc}}(\mathsf{pp}, x, m_\beta; \mathsf{msk}, \mathbf{g}\cdot\hat{\mathbf{g}}), \widehat{\mathsf{KeyGen}}(\mathsf{pp}, \boxed{\mathsf{msk}}, y; \mathbf{h}\cdot\hat{\mathbf{h}})\} \quad \text{and}$$
> $$\{\mathsf{pp}, \mathsf{msk}, (h^*)^\alpha, \widehat{\mathsf{Enc}}(\mathsf{pp}, x, m_\beta; \mathsf{msk}, \mathbf{g}\cdot\hat{\mathbf{g}}), \widehat{\mathsf{KeyGen}}(\mathsf{pp}, \boxed{\mathsf{msk}\cdot(h^*)^\alpha}, y; \mathbf{h}\cdot\hat{\mathbf{h}})\}.$$

We defer the proof of the claim for now, and first explain how the lemma follows from the claim. Given $(\mathsf{pp}, \mathsf{msk}, (h^*)^\alpha)$, we can output $\mathsf{mpk} := (\mathsf{pp}, \mu(\mathsf{msk}))$ and generate the first $i-1$ semi-functional secret keys, and the remaining $q-i$ normal secret keys using

$$\widehat{\mathsf{KeyGen}}(\mathsf{pp}, \mathsf{msk}\cdot(h^*)^\alpha, y; \mathsf{SampH}(\mathsf{pp})) \quad \text{and} \quad \widehat{\mathsf{KeyGen}}(\mathsf{pp}, \mathsf{msk}, y; \mathsf{SampH}(\mathsf{pp}))$$

respectively.

This would in turn imply that $\mathsf{Game}_{2,i,1}$ and $\mathsf{Game}_{2,i,2}$ are statistically indistinguishable. We note that this holds even if the adversary chooses y adaptively after seeing the challenge ciphertext ct_{x^*}, or if the challenge x^* is chosen after the adversary sees sk_y. □

Proof (of claim). By linearity, we have:

$$\widehat{\mathsf{Enc}}(\mathsf{pp}, x, m_\beta; \mathsf{msk}, \mathbf{g}\cdot\hat{\mathbf{g}}) = \widehat{\mathsf{Enc}}(\mathsf{pp}, x, m_\beta; \mathsf{msk}, \mathbf{g}) \cdot \widehat{\mathsf{Enc}}(\mathsf{pp}, x, 1; \mathsf{msk}, \hat{\mathbf{g}})$$
$$\widehat{\mathsf{KeyGen}}(\mathsf{pp}, \mathsf{msk}, y; \mathbf{h}\cdot\hat{\mathbf{h}}) = \widehat{\mathsf{KeyGen}}(\mathsf{pp}, \mathsf{msk}, y; \mathbf{h}) \cdot \widehat{\mathsf{KeyGen}}(\mathsf{pp}, 1, y; \hat{\mathbf{h}})$$
$$\widehat{\mathsf{KeyGen}}(\mathsf{pp}, \mathsf{msk}\cdot(h^*)^\alpha, y; \mathbf{h}\cdot\hat{\mathbf{h}}) = \widehat{\mathsf{KeyGen}}(\mathsf{pp}, \mathsf{msk}, y; \mathbf{h}) \cdot \widehat{\mathsf{KeyGen}}(\mathsf{pp}, (h^*)^\alpha, y; \hat{\mathbf{h}})$$

Therefore, it suffices to show that:

$$\{\mathsf{pp}, \mathsf{msk}, (h^*)^\alpha, \widehat{\mathsf{Enc}}(\mathsf{pp}, x, 1; \mathsf{msk}, \hat{\mathbf{g}}), \widehat{\mathsf{KeyGen}}(\mathsf{pp}, \boxed{1}, y; \hat{\mathbf{h}})\} \quad \text{and}$$
$$\{\mathsf{pp}, \mathsf{msk}, (h^*)^\alpha, \widehat{\mathsf{Enc}}(\mathsf{pp}, x, 1; \mathsf{msk}, \hat{\mathbf{g}}), \widehat{\mathsf{KeyGen}}(\mathsf{pp}, \boxed{(h^*)^\alpha}, y; \hat{\mathbf{h}})\}$$

are identically distributed.

By parameter-hiding, we may replace $(\mathsf{pp}, h^*, \boxed{\hat{\mathbf{g}}, \hat{\mathbf{h}}})$ with $(\mathsf{pp}, h^*, \boxed{\hat{\mathbf{g}}\cdot\hat{\mathbf{g}}', \hat{\mathbf{h}}\cdot\hat{\mathbf{h}}'})$, which means it suffices to show that:

$$\{\mathsf{pp}, \mathsf{msk}, (h^*)^\alpha, \widehat{\mathsf{Enc}}(\mathsf{pp}, x, 1; \mathsf{msk}, \hat{\mathbf{g}}\cdot\hat{\mathbf{g}}'), \widehat{\mathsf{KeyGen}}(\mathsf{pp}, \boxed{1}, y; \hat{\mathbf{h}}\cdot\hat{\mathbf{h}}')\} \quad \text{and}$$
$$\{\mathsf{pp}, \mathsf{msk}, (h^*)^\alpha, \widehat{\mathsf{Enc}}(\mathsf{pp}, x, 1; \mathsf{msk}, \hat{\mathbf{g}}\cdot\hat{\mathbf{g}}'), \widehat{\mathsf{KeyGen}}(\mathsf{pp}, \boxed{(h^*)^\alpha}, y; \hat{\mathbf{h}}\cdot\hat{\mathbf{h}}')\}$$

are identically distributed. At this point, we expand the expressions for $\widehat{\mathsf{Enc}}$ and $\widehat{\mathsf{KeyGen}}$:

$$\begin{aligned}\widehat{\mathsf{Enc}}(\mathsf{pp}, x, 1; \mathsf{msk}, \hat{\mathbf{g}}\cdot\hat{\mathbf{g}}') &= (\hat{g}_0, \mathsf{sE}(x, \hat{\mathbf{g}}) \cdot \mathsf{sE}(x, \hat{\mathbf{g}}'),\ e(\hat{g}_0, \mathsf{msk}))\\ &= (\hat{g}_0, \mathsf{sE}(x, \hat{\mathbf{g}}) \cdot \hat{g}_0^{\mathsf{sE}(x,\hat{\mathbf{u}})},\ e(\hat{g}_0, \mathsf{msk}))\end{aligned}$$

where $\hat{\mathbf{u}}$ denotes the vector $\hat{\mathbf{u}} := (\hat{u}_1, \ldots, \hat{u}_n)$ and thus $\mathsf{sE}(x, \hat{\mathbf{g}}') = \mathsf{sE}(x, \hat{g}_0^{\hat{\mathbf{u}}}) = \hat{g}_0^{\mathsf{sE}(x,\hat{\mathbf{u}})}$.

$$\widehat{\mathsf{KeyGen}}(\mathsf{pp}, 1, y; \hat{\mathbf{h}}\cdot\hat{\mathbf{h}}') = (\hat{h}_0, \mathsf{rE}(y, \hat{\mathbf{h}}) \cdot \hat{h}_0^{\mathsf{rE}(y,\hat{\mathbf{u}})})$$
$$\widehat{\mathsf{KeyGen}}(\mathsf{pp}, (h^*)^\alpha, y; \hat{\mathbf{h}}\cdot\hat{\mathbf{h}}') = (\hat{h}_0, \mathsf{kE}(y, (h^*)^\alpha) \cdot \mathsf{rE}(y, \hat{\mathbf{h}}) \cdot \hat{h}_0^{\mathsf{rE}(y,\hat{\mathbf{u}})})$$

Since h^* lies in the group generated by $\hat{h}_0$, we have $\mathsf{kE}(y, (h^*)^\alpha) = \mathsf{kE}(y, (h_0)^{\alpha'}) = \hat{h}_0^{\mathsf{kE}(y,\alpha')}$ for some $\alpha' \in \mathbb{Z}_p$; the claim then follows readily from α'-privacy, that is, $\mathsf{rE}(y, \hat{\mathbf{u}})$ and $\mathsf{kE}(y, \alpha') + \mathsf{rE}(y, \hat{\mathbf{u}})$ are identically distributed. □

Lemma 8 (pseudo-SF to SF keys: $\mathsf{Game}_{2,i,2}$ to $\mathsf{Game}_{2,i,3}$). *For $i = 1, \ldots, q$, for any adversary $\mathcal{A}$ that makes at most q key queries, there exists an adversary $\mathcal{B}_3$ such that*

$$|\mathsf{Adv}_{2,i,2}(\lambda) - \mathsf{Adv}_{2,i,3}(\lambda)| \leq \mathsf{Adv}^{\mathrm{RS}}_{\mathcal{B}_3}(\lambda)$$

and $\mathsf{Time}(\mathcal{B}_3) \approx \mathsf{Time}(\mathcal{A}) + q \cdot \mathrm{poly}(\lambda, n)$ *where* $\mathrm{poly}(\lambda, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. The proof is completely analogous to Lemma 6, except we use $\widehat{\mathsf{msk}}$ instead of msk to generate the i'th key query. That is, $\mathcal{B}_3$ is exactly the same as $\mathcal{B}_2$, with the following change:

Key Queries. On input the j'th secret key query y, output

$$\mathsf{sk}_y \leftarrow \begin{cases} \widehat{\mathsf{KeyGen}}(\mathsf{pp}, \widehat{\mathsf{msk}}, y; \mathsf{SampH}(\mathsf{pp})) & \text{if } j < i \\ \widehat{\mathsf{KeyGen}}(\mathsf{pp}, \boxed{\widehat{\mathsf{msk}}}, y; \mathbf{t}) & \text{if } j = i \\ \widehat{\mathsf{KeyGen}}(\mathsf{pp}, \mathsf{msk}, y; \mathsf{SampH}(\mathsf{pp})) & \text{if } j > i \end{cases}.$$

Observe that when $\mathbf{t} = \mathbf{h}$, the output is identical to that in $\mathsf{Game}_{2,i,3}$; and when $\mathbf{t} = \mathbf{h} \cdot \hat{\mathbf{h}}$, the output is identical to that in $\mathsf{Game}_{2,i,2}$. We may therefore conclude that: $|\mathsf{Adv}_{2,i,2}(\lambda) - \mathsf{Adv}_{2,i,3}(\lambda)| \leq \mathsf{Adv}^{\mathrm{RS}}_{\mathcal{B}_3}(\lambda)$. □

Lemma 9 (final transition: $\mathsf{Game}_{2,q,3}$ to Game_3). *For any adversary $\mathcal{A}$, we have*

$$|\mathsf{Adv}_{2,q,3}(\lambda) - \mathsf{Adv}_3(\lambda)| = 0.$$

Proof. First, we sample $(\mathsf{msk}, \widehat{\mathsf{msk}})$ in both games as follows: pick $\widehat{\mathsf{msk}} \leftarrow_{\mathrm{R}} \mathbb{H}$, $\alpha \leftarrow_{\mathrm{R}} \mathbb{Z}_p$ and set $\mathsf{msk} := \widehat{\mathsf{msk}} \cdot (h^*)^{-\alpha}$. We may then simulate key set-up and answer key queries given just $(\mathsf{pp}, \widehat{\mathsf{msk}})$ as follows:

Setup. Observe that

$$\mu(\mathsf{msk}) = \mu(\widehat{\mathsf{msk}}) \cdot \mu((h^*)^{-\alpha}) = \mu(\widehat{\mathsf{msk}})$$

where in the last equality, we use *orthogonality* $\mu(h^*) = 1$. Output

$$\mathsf{mpk} := \Big(\mathsf{pp},\ \mu(\widehat{\mathsf{msk}}) \Big).$$

Key Queries. On input the j'th secret key query y, output

$$\mathsf{sk}_y \leftarrow \widehat{\mathsf{KeyGen}}(\mathsf{pp}, \widehat{\mathsf{msk}}, y; \mathsf{SampH}(\mathsf{pp})).$$

Now, observe that the challenge ciphertext in $\mathsf{Game}_{2,q,3}$ is given by:

$$\widehat{\mathsf{Enc}}(\mathsf{pp}, x^*, m_\beta; \mathsf{msk}, \mathbf{g} \cdot \hat{\mathbf{g}}) = (C_0, \mathbf{C}_1, C_2' \cdot m_\beta),$$

where $(C_0, \mathbf{C}_1)$ depend only on $\mathbf{g} \cdot \hat{\mathbf{g}} = (g_0 \cdot \hat{g}_0, \ldots)$, and C_2' is given by:

$$C_2' = e(g_0 \cdot \hat{g}_0, \mathsf{msk}) = e(g_0 \cdot \hat{g}_0, \widehat{\mathsf{msk}} \cdot (h^*)^{-\alpha}) = e(g_0 \cdot \hat{g}_0, \widehat{\mathsf{msk}}) \cdot \boxed{e(\hat{g}_0, h^*)^{-\alpha}},$$

where in the last equality, we use linearity and the fact that $e(g_0, (h^*)^{-\alpha}) = 1$. Recall that $(\mathsf{pp}, \widehat{\mathsf{msk}}, \mathbf{g} \cdot \hat{\mathbf{g}})$ are all statistically independent of $\alpha \leftarrow_{\mathrm{R}} \mathbb{Z}_p$. Then, by *non-degeneracy*, given $\widehat{\mathsf{msk}}$, all of the secret keys, along with $(C_0, \mathbf{C}_1)$ in the challenge ciphertext, the quantity

$$e(\hat{g}_0, h^*)^{-\alpha}$$

is uniformly distributed over $\mathbb{G}_T$. This implies the challenge ciphertext is identically distributed to a semi-functional encryption of a random message in $\mathbb{G}_T$, as in Game_3. We may then conclude that: $|\mathsf{Adv}_{2,q,3}(\lambda) - \mathsf{Adv}_3(\lambda)| = 0$. □

7 Extension to Weakly Attribute-Hiding

We present an extension of our framework to weakly attribute-hiding predicate encryption [28, 9]. A predicate encryption scheme has the same syntax as an ABE in Section 2.1 except the attribute x on the ciphertext is not public; for security, we require in addition that x remains hidden from the adversary.

7.1 Security definition

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

$$\mathsf{Adv}^{\text{PE}}_{\mathcal{A}}(\lambda) := \Pr\left[b = b' : \begin{array}{c} (\mathsf{mpk},\mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{X}, \mathcal{Y}, \mathcal{M}); \\ (x_0^*, x_1^*, m_0, m_1) \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{msk},\cdot)}(\mathsf{mpk}); \\ b \leftarrow_{\mathrm{R}} \{0,1\}; \mathsf{ct}_{x_b^*} \leftarrow \mathsf{Enc}(\mathsf{mpk}, x_b^*, m_b); \\ b' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{msk},\cdot)}(\mathsf{ct}_{x_b^*}) \end{array} \right] - \frac{1}{2}$$

with the restriction that all queries y that $\mathcal{A}$ makes to $\mathsf{KeyGen}(\mathsf{msk},\cdot)$ satisfies $\mathsf{P}(x_0^*, y) = \mathsf{P}(x_1^*, y) = 0$ (that is, sk_y does not decrypt the challenge ciphertext). A predicate encryption scheme is *adaptively secure and weakly attribute-hiding* if for all PPT adversaries $\mathcal{A}$, the advantage $\mathsf{Adv}^{\text{PE}}_{\mathcal{A}}(\lambda)$ is a negligible function in λ.[4]

7.2 Attribute-Hiding Encodings

We say that a $\mathbb{Z}_p$-bilinear predicate encoding (c.f. Section 5) for $\mathsf{P} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ is *attribute-hiding* if it satisfies the following additional properties:

(x-oblivious α-reconstruction.) $\mathsf{sD}(x, y, \cdot)$ and $\mathsf{rD}(x, y, \cdot)$ are independent of x.

(attribute-hiding.) For all $(x, y) \in \mathcal{X} \times \mathcal{Y}$ such that $\mathsf{P}(x, y) = 0$, the joint distribution of $\{\mathsf{sE}(x, \mathbf{w}), \mathsf{rE}(y, \mathbf{w})\}$ is uniformly random. That is, the following distributions are *identically* distributed:

$$\big\{x, y, \mathsf{sE}(x, \mathbf{w}), \mathsf{rE}(y, \mathbf{w})\big\} \quad \text{and} \quad \big\{x, y, \mathbf{r}\big\}$$

where the randomness is taken over $\mathbf{w} \leftarrow_{\mathrm{R}} \mathcal{W}$ and $\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{|\mathsf{sE}(\cdot)|+|\mathsf{rE}(\cdot)|}$.

7.3 Attribute-Hiding Dual System Groups

Recall from the introduction in Section 1.1 that to realize weakly attribute-hiding predicate encryption, we will use the fact that for any vector $\mathbf{c} \in \mathbb{Z}_p^{k+1}$ outside the span of $\mathbf{A}$, the vector $\mathbf{W}^{\top}\mathbf{c} \in \mathbb{Z}_p^{k+1}$ is uniformly random given $\mathbf{W}^{\top}\mathbf{A} \in \mathbb{Z}_p^{(k+1)\times k}$, provided $\mathbf{WB}$ remains hidden. We can then use $\mathbf{W}^{\top}\mathbf{c}$ to completely blind the attribute in the challenge ciphertext (c.f. Lemma 17). We also need to make sure that the semi-functional secret keys do not leak any *additional* information about $\mathbf{WB}$ (c.f. Lemma 17). The former is captured by $\mathbb{G}$-uniformity, and the latter by $\mathbb{H}$-hiding. In particular, the secret keys in the predicate encryption scheme satisfy the following properties:

- the distribution of normal secret keys is completely determined given $\mathbf{B}, \mathbf{W}_1\mathbf{B}, \ldots, \mathbf{W}_n\mathbf{B}$ and leaks no *additional* information about $\mathbf{W}_1, \ldots, \mathbf{W}_n$;
- the distribution of semi-functional secret keys is completely determined given $\mathbf{A}, \mathbf{W}_1^{\top}\mathbf{A}, \ldots, \mathbf{W}_n^{\top}\mathbf{A}$ and leaks no *additional* information about $\mathbf{W}_1, \ldots, \mathbf{W}_n$.

Additional properties. We assume that pp in dual system groups has a $\mathsf{pp}_{\mathbb{G}}$-component which is sufficient to run SampG. We then require dual system groups to satisfy the following additional properties.

($\mathbb{H}$-hiding) There is an (inefficient) randomized procedure SampH^* that given $\mathsf{pp}_{\mathbb{G}}$ and h^*, outputs a distribution identical to that of

$$\mathbf{h} \cdot (h^*)^{(0,\hat{\mathbf{v}})}$$

where $\mathbf{h} \leftarrow \mathsf{SampH}(\mathsf{pp})$, $\hat{\mathbf{v}} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^n$.

[4] In a fully attribute-hiding scheme, the adversary is also allowed key queries y for which $\mathsf{P}(x_0^*, y) = \mathsf{P}(x_1^*, y) = 1$, in which case the challenge messages m_0, m_1 must be equal.

($\mathbb{G}$-uniformity) The following distributions are identically distributed

$$\left\{ \mathsf{pp}_{\mathbb{G}}, h^*, \boxed{\mathbf{g}\cdot\hat{\mathbf{g}}} \right\} \quad \text{and} \quad \left\{ \mathsf{pp}_{\mathbb{G}}, h^*, \boxed{\mathbf{g}'} \right\}$$

where

$$(\mathsf{pp},\mathsf{sp}) \leftarrow \mathsf{SampP}(1^\lambda, 1^n);\ \mathbf{g} = (g_0,\ldots) \leftarrow \mathsf{SampG}(\mathsf{pp});\ \hat{\mathbf{g}} = (\hat g_0,\ldots) \leftarrow \widehat{\mathsf{SampG}}(\mathsf{pp},\mathsf{sp});\ \mathbf{g}' \leftarrow_{\mathrm{R}} \{g_0\hat g_0\} \times \mathbb{G}^n.$$

7.4 Attribute-Hiding Requirements

We show that our instantiation satisfies the additional attribute-hiding requirements in Section 7.3 when $\mathsf{pp}_{\mathbb{G}}$ is defined to be:

$$\mathsf{pp}_{\mathbb{G}} := \big(\, (p, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, e); [\mathbf{A}]_1, \left[\mathbf{W}_1^\top\mathbf{A}\right]_1, \ldots, \left[\mathbf{W}_n^\top\mathbf{A}\right]_1,\ [\mathbf{B}]_2\,\big).$$

Recall that $\mathbf{A}^\top\mathbf{a}^\perp = \mathbf{B}^\top\mathbf{b}^\perp = \mathbf{0}$, and that by Lemma 1, we may assume that both $(\mathbf{A}, \mathbf{b}^\perp)$ and $(\mathbf{B}, \mathbf{a}^\perp)$ form a basis for $\mathbb{Z}_p^{k+1}$ and that $\mathbf{b}^{\perp\top}\mathbf{a}^\perp \neq 0$.

Lemma 10. *The instantiation satisfies* $\mathbb{H}$*-hiding.*

Proof. We may rewrite $\mathbf{h}\cdot(h^*)^{(0,\hat{\mathbf{v}})}$ as:

$$\mathbf{h}\cdot(h^*)^{(0,\hat{\mathbf{v}})} := \Big([\mathbf{Br}]_2, \left[\mathbf{W}_1\mathbf{Br} + \mathbf{a}^\perp \hat v_1\right]_2, \ldots, \left[\mathbf{W}_n\mathbf{Br} + \mathbf{a}^\perp \hat v_n\right]_2 \Big)$$

where $\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k$, $\hat{\mathbf{v}} := (\hat v_1, \ldots, \hat v_n) \leftarrow_{\mathrm{R}} \mathbb{Z}_p^n$. For $i = 1, \ldots, n$, consider the following change of basis to $\mathbf{W}_i\mathbf{Br} + \mathbf{a}^\perp \hat v_i$:

$$\begin{pmatrix}\mathbf{A}^\top \\ \mathbf{b}^{\perp\top}\end{pmatrix}(\mathbf{W}_i\mathbf{Br}+\mathbf{a}^\perp\hat v_i) = \begin{pmatrix}\mathbf{A}\mathbf{W}_i^\top\mathbf{Br} + \mathbf{A}^\top\mathbf{a}^\perp \hat v_i \\ \mathbf{b}^{\perp\top}(\mathbf{W}_i\mathbf{Br}+\mathbf{a}^\perp\hat v_i)\end{pmatrix} = \begin{pmatrix}(\mathbf{W}_i^\top\mathbf{A})^\top\mathbf{Br} \\ \mathbf{b}^{\perp\top}\mathbf{W}_i\mathbf{Br}+\mathbf{b}^{\perp\top}\mathbf{a}^\perp\hat v_i\end{pmatrix}.$$

Note that $\mathbf{b}^{\perp\top}\mathbf{W}_i\mathbf{Br} + \mathbf{b}^{\perp\top}\mathbf{a}^\perp \hat v_i$ is uniformly random over $\mathbb{Z}_p$. This means that $\mathbf{W}_i\mathbf{Br} + \mathbf{a}^\perp \hat v_i$ is identically distributed to a random vector $\mathbf{u}_i$ subject to the constraint $\mathbf{A}^\top\mathbf{u}_i = (\mathbf{W}_i^\top\mathbf{A})^\top\mathbf{Br}$. Given $\mathsf{pp}_{\mathbb{G}}$, we may then (inefficiently) sample from the distribution $\mathbf{h}\cdot(h^*)^{(0,\hat{\mathbf{v}})}$ as follows:

- pick $\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k$ to compute $\mathbf{Br}$;
- for $i = 1, \ldots, n$, sample a uniformly random vector $\mathbf{u}_i$ subject to the constraint $\mathbf{A}^\top\mathbf{u}_i = (\mathbf{W}_i^\top\mathbf{A})^\top\mathbf{Br}$.
- output $([\mathbf{Br}]_2, [\mathbf{u}_1]_2, \ldots, [\mathbf{u}_n]_2)$.

And the lemma follows. □

Lemma 11 ($\mathbb{G}$-uniformity). *The following distributions are identically distributed:*

$$\left\{[\mathbf{A}]_1, \left[\mathbf{W}_1^\top\mathbf{A}\right]_1, \ldots, \left[\mathbf{W}_n^\top\mathbf{A}\right]_1,\ [\mathbf{B}]_2; \left[\mathbf{a}^\perp\right]_2; \left[\mathbf{As} + \mathbf{b}^\perp\hat s\right]_1, \left[\mathbf{W}_1^\top(\mathbf{As} + \mathbf{b}^\perp \hat s)\right]_1, \ldots, \left[\mathbf{W}_n^\top(\mathbf{As} + \mathbf{b}^\perp \hat s)\right]_1\right\} \quad \textit{and}$$

$$\left\{[\mathbf{A}]_1, \left[\mathbf{W}_1^\top\mathbf{A}\right]_1, \ldots, \left[\mathbf{W}_n^\top\mathbf{A}\right]_1,\ [\mathbf{B}]_2; \left[\mathbf{a}^\perp\right]_2; \left[\mathbf{As} + \mathbf{b}^\perp\hat s\right]_1, [\mathbf{z}_1]_1, \ldots, [\mathbf{z}_n]_1\right\}$$

where $\mathbf{W}_1, \ldots, \mathbf{W}_n \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(k+1)\times(k+1)}$; $\mathbf{s} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k$, $\hat s \leftarrow_{\mathrm{R}} \mathbb{Z}_p^*$; $\mathbf{z}_1, \ldots, \mathbf{z}_n \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{k+1}$.

Proof. Observe that $\mathbf{As} + \mathbf{b}^\perp \hat s \notin \mathrm{span}(\mathbf{A})$, and therefore for each $i = 1, \ldots, n$, $\mathbf{W}_i^\top(\mathbf{As} + \mathbf{b}^\perp \hat s)$ is uniformly distributed, even given $\mathbf{W}_i^\top\mathbf{A}$. □

7.5 Weakly Attribute-Hiding PE

Starting from an attribute-hiding encoding and an attribute-hiding dual system group, we can construct a predicate encryption scheme as described in Section 6.1, with the following modification: we put $\mathsf{pp}_{\mathbb{G}}$ instead of pp in mpk (which suffices for SampG and Enc). We show that the ensuing scheme is weakly attribute-hiding:

game	ciphertext $(C_0, \mathbf{C}_1, C')$	secret key $(K_0, \mathbf{K}_1)$	justification	remark
0	$(1, \mathbf{1}, 1)$	$(1, (h^*)^{\mathsf{kE}(y,0)} \cdot \mathbf{1})$	$\mathbf{1} = (h^*)^{\mathsf{kE}(y,0)}$	actual scheme
1	$\boxed{(\hat{g}_0, \mathsf{sE}(x, \hat{\mathbf{g}}), e(\hat{g}_0, \mathsf{msk}))}$	$(1, (h^*)^{\mathsf{kE}(y,0)} \cdot \mathbf{1})$	left subgroup indist.	normal to SF ct_x
2.i.1	$(\hat{g}_0, \mathsf{sE}(x, \hat{\mathbf{g}}), e(\hat{g}_0, \mathsf{msk}))$	$(\boxed{\hat{h}_0}, (h^*)^{\mathsf{kE}(y,0)} \cdot \boxed{\mathsf{rE}(y, \hat{\mathbf{h}})})$	right subgroup indist.	normal to pseudo-normal sk_y
2.i.2	$(\hat{g}_0, \mathsf{sE}(x, \hat{\mathbf{g}}), e(\hat{g}_0, \mathsf{msk}))$	$(\hat{h}_0, \boxed{(h^*)^{\mathsf{kE}(y,\alpha)+\mathsf{rE}(y,\hat{\mathbf{v}}^i)}} \cdot \mathsf{rE}(y, \hat{\mathbf{h}}))$	attribute-hiding encoding	pseudo-normal to pseudo-SF sk_y
2.i.3	$(\hat{g}_0, \mathsf{sE}(x, \hat{\mathbf{g}}), e(\hat{g}_0, \mathsf{msk}))$	$(\boxed{1}, (h^*)^{\mathsf{kE}(y,\alpha)+\mathsf{rE}(y,\hat{\mathbf{v}}^i)} \cdot \boxed{\mathbf{1}})$	right subgroup indist.	pseudo-SF to SF sk_y
3	$(\hat{g}_0, \mathsf{sE}(x, \hat{\mathbf{g}}), \boxed{\text{random}})$	$(1, (h^*)^{\mathsf{kE}(y,\alpha)+\mathsf{rE}(y,\hat{\mathbf{v}}^i)} \cdot \mathbf{1})$		
4	$(\hat{g}_0, \boxed{\text{random}}, \text{random})$	$(1, (h^*)^{\mathsf{kE}(y,\alpha)+\mathsf{rE}(y,\hat{\mathbf{v}}^i)} \cdot \mathbf{1})$	$\mathbb{G}$-uniformity, $\mathbb{H}$-hiding attribute-hiding encoding	

Fig. 8. Sequence of games in the "semi-functional" space for weakly attribute-hiding PE. We omitted the normal components: those sampled using $\mathsf{SampG}, \mathsf{SampH}$, and we omitted $e(g_0, \mathsf{msk}) \cdot m$ in C' and $\mathsf{kE}(y, \mathsf{msk})$ in sk_y. We drew a box to highlight the differences between each game and the preceding one, and games $2.i.$x refer to the i'th secret key. The semi-functional components of the keys transition from $(h^*)^{\mathsf{kE}(y,0)}$ to $(h^*)^{\mathsf{kE}(y,\alpha)+\mathsf{rE}(y,\hat{\mathbf{v}}^i)}$, with a fresh $\hat{\mathbf{v}}^i \leftarrow_{\mathrm{R}} \mathbb{Z}_p^n$ for the i'th key. For the last second transition, we use the fact that $e(\hat{g}_0, \mathsf{msk})$ is statistically random given $\mathsf{msk} \cdot (h^*)^{\alpha}$. In the final transition, we use the fact that $\mathbf{C}_1$ (including normal components) is statistically random.

Theorem 2. *Under the left and right subgroup indistinguishability (described in Section 3), the predicate encryption scheme described above is adaptively secure and weakly attribute-hiding (in the sense of Definition 7.1). More precisely, for any adversary $\mathcal{A}$ that makes at most q key queries against the predicate encryption scheme, there exist adversaries $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ such that:*

$$\mathsf{Adv}^{\mathrm{PE}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathrm{LS}}_{\mathcal{B}_1}(\lambda) + q \cdot \mathsf{Adv}^{\mathrm{RS}}_{\mathcal{B}_2}(\lambda) + q \cdot \mathsf{Adv}^{\mathrm{RS}}_{\mathcal{B}_3}(\lambda) + \mathrm{negl}(\lambda)$$

and

$$\max\{\mathsf{Time}(\mathcal{B}_1), \mathsf{Time}(\mathcal{B}_2), \mathsf{Time}(\mathcal{B}_3)\} \approx \mathsf{Time}(\mathcal{A}) + q \cdot \mathrm{poly}(\lambda, n)$$

where $\mathrm{poly}(\lambda, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

The proof follows via a series of games, outlined in Fig. 8.

Auxiliary distributions. The auxiliary algorithms and distributions are the same as in Section 6.2 with the following modifications: (1) pseudo-SF and semi-functional secret keys have additional h^*-components, (2) $\widehat{\mathsf{Enc}}$ and $\widehat{\mathsf{KeyGen}}$ get as input $\mathsf{pp}_{\mathbb{G}}$ instead of pp (neither algorithm needs to run SampH).

Pseudo-SF secret key.

$$\widehat{\mathsf{KeyGen}}(\mathsf{pp}_{\mathbb{G}}, \boxed{\widehat{\mathsf{msk}}}, y; \boxed{\mathbf{h} \cdot \hat{\mathbf{h}} \cdot (h^*)^{(0, \hat{\mathbf{v}})}}),$$

where fresh $\boxed{\mathbf{h} \leftarrow \mathsf{SampH}(\mathsf{pp}),\ \hat{\mathbf{h}} \leftarrow \widehat{\mathsf{SampH}}(\mathsf{pp}, \mathsf{sp}), \text{ and } \hat{\mathbf{v}} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^n}$ are chosen for each secret key.

Semi-functional secret key.

$$\widehat{\mathsf{KeyGen}}(\mathsf{pp}_{\mathbb{G}}, \widehat{\mathsf{msk}}, y; \boxed{\mathbf{h} \cdot (h^*)^{(0, \hat{\mathbf{v}})}}),$$

where a fresh $\boxed{\mathbf{h} \leftarrow \mathsf{SampH}(\mathsf{pp}) \text{ and } \hat{\mathbf{v}} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^n}$ are chosen for each secret key.

Game sequence. We proceed exactly as in Section 6.2 with the same auxiliary algorithms but with the following modifications: (1) the distributions of pseudo-SF and semi-functional secret keys have additional

h^*-components, (2) the challenge ciphertext uses the attribute x_b^* as defined in the security experiment, and (3) we append an extra game Game_4 where we switch x_b^* to random at the end:

- Game_0: is the real security game (c.f. Section 7.1).
- Game_1: is the same as Game_0 except that the challenge ciphertext is semi-functional.
- $\mathsf{Game}_{2,i,1}$ for i from 1 to q, $\mathsf{Game}_{2,i,1}$ is the same as Game_1 except that the first $i-1$ keys are semi-functional, the last $q-i$ keys are normal while the i'th key is pseudo-normal.
- $\mathsf{Game}_{2,i,2}$ for i from 1 to q, $\mathsf{Game}_{2,i,2}$ is the same as Game_1 except that the first $i-1$ keys are semi-functional, the last $q-i$ keys are normal while the i'th key is pseudo-SF.
- $\mathsf{Game}_{2,i,3}$ for i from 1 to q, $\mathsf{Game}_{2,i,3}$ is the same as Game_1 except that the first i keys are semi-functional, the last $q-i$ keys are normal.
- Game_3: is the same as $\mathsf{Game}_{2,q,3}$, except that the challenge ciphertext is a semi-functional encryption of a random message in $\mathbb{G}_T$.
- Game_4: is the same as Game_3, except we replace x_b^* in the challenge ciphertext with a random attribute $x^* \leftarrow_{\mathrm{R}} \mathcal{X}$.

In Game_4, the view of the adversary is statistically independent of the challenge bit b. Hence, $\mathsf{Adv}_4(\lambda) = 0$. We complete the proof by establishing the following sequence of lemmas.

Lemma 12 (normal to SF ciphertext: Game_0 **to** Game_1**).** *For any adversary $\mathcal{A}$ that makes at most q key queries, there exists an adversary $\mathcal{B}_1$ such that*

$$|\mathsf{Adv}_0(\lambda) - \mathsf{Adv}_1(\lambda)| \le \mathsf{Adv}_{\mathcal{B}_1}^{\mathrm{LS}}(\lambda)$$

and $\mathsf{Time}(\mathcal{B}_1) \approx \mathsf{Time}(\mathcal{A}) + q \cdot \mathrm{poly}(\lambda, n)$ *where* $\mathrm{poly}(\lambda, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

The proof is exactly as Lemma 5.

Lemma 13 (normal to pseudo-normal keys: $\mathsf{Game}_{2,i-1,3}$ **to** $\mathsf{Game}_{2,i,1}$**).** *For $i = 1, \ldots, q$, for any adversary $\mathcal{A}$ that makes at most q key queries, there exists an adversary $\mathcal{B}_2$ such that*

$$|\mathsf{Adv}_{2,i-1,3}(\lambda) - \mathsf{Adv}_{2,i,1}(\lambda)| \le \mathsf{Adv}_{\mathcal{B}_2}^{\mathrm{RS}}(\lambda)$$

and $\mathsf{Time}(\mathcal{B}_2) \approx \mathsf{Time}(\mathcal{A}) + q \cdot \mathrm{poly}(\lambda, n)$ *where* $\mathrm{poly}(\lambda, n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$. *(We note that* $\mathsf{Game}_{2,0,3}$ *is identical to* Game_1*.)*

Proof. The proof is completely analogous to Lemma 6, except we will additional use h^* to generate the first $i-1$ semi-functional keys:

Key Queries. On input the j'th secret key query y, output

$$\mathsf{sk}_y \leftarrow \begin{cases} \widehat{\mathsf{KeyGen}}(\mathsf{pp}_{\mathbb{G}}, \widehat{\mathsf{msk}}, y; \mathsf{SampH}(\mathsf{pp}) \cdot \boxed{(h^*)^{(0,\hat{\mathbf{v}}^j)}}) & \text{if } j < i \\ \widehat{\mathsf{KeyGen}}(\mathsf{pp}_{\mathbb{G}}, \mathsf{msk}, y; \mathbf{t}) & \text{if } j = i \\ \widehat{\mathsf{KeyGen}}(\mathsf{pp}_{\mathbb{G}}, \mathsf{msk}, y; \mathsf{SampH}(\mathsf{pp})) & \text{if } j > i \end{cases}$$

Observe that when $\mathbf{t} = \mathbf{h}$, the output is identical to that in $\mathsf{Game}_{2,i-1,3}$; and when $\mathbf{t} = \mathbf{h} \cdot \hat{\mathbf{h}}$, the output is identical to that in $\mathsf{Game}_{2,i,1}$. We may therefore conclude that: $|\mathsf{Adv}_{2,i-1,3}(\lambda) - \mathsf{Adv}_{2,i,1}(\lambda)| \le \mathsf{Adv}_{\mathcal{B}_2}^{\mathrm{RS}}(\lambda)$. □

Lemma 14 (pseudo-normal to pseudo-SF keys: $\mathsf{Game}_{2,i,1}$ **to** $\mathsf{Game}_{2,i,2}$**).** *For $i = 1, \ldots, q$, we have*

$$|\mathsf{Adv}_{2,i,1}(\lambda) - \mathsf{Adv}_{2,i,2}(\lambda)| = 0.$$

Proof. It suffices to establish the following:

Claim. For all α, all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, where $\mathsf{P}(x,y) = 0$, the following distributions are identically distributed:

$$\{\mathsf{pp}, \mathsf{msk}, (h^*)^\alpha, \widehat{\mathsf{Enc}}(\mathsf{pp}_{\mathbb{G}}, x, m_\beta; \mathsf{msk}, \mathbf{g}\cdot\hat{\mathbf{g}}), \widehat{\mathsf{KeyGen}}(\mathsf{pp}_{\mathbb{G}}, \boxed{\mathsf{msk}}, y; \boxed{\mathbf{h}\cdot\hat{\mathbf{h}})}\} \quad \text{and}$$

$$\{\mathsf{pp}, \mathsf{msk}, (h^*)^\alpha, \widehat{\mathsf{Enc}}(\mathsf{pp}_{\mathbb{G}}, x, m_\beta; \mathsf{msk}, \mathbf{g}\cdot\hat{\mathbf{g}}), \widehat{\mathsf{KeyGen}}(\mathsf{pp}_{\mathbb{G}}, \boxed{\mathsf{msk}\cdot(h^*)^\alpha}, y; \boxed{\mathbf{h}\cdot\hat{\mathbf{h}}\cdot(h^*)^{(0,\hat{\mathbf{v}})}})\}.$$

The remaining proof is identical to the one for Lemma 7 except that we will use the knowledge of h^* to generate the first $i-1$ semi-functional secret keys. □

Proof (of claim). The process is similar to the proof of Lemma 7 until we expand the expressions for $\widehat{\mathsf{Enc}}$ and $\widehat{\mathsf{KeyGen}}$:

$$\begin{aligned}\widehat{\mathsf{Enc}}(\mathsf{pp}_{\mathbb{G}}, x, 1; \mathsf{msk}, \hat{\mathbf{g}}\cdot\hat{\mathbf{g}}') &= (\hat{g}_0, \mathsf{sE}(x,\hat{\mathbf{g}})\cdot\mathsf{sE}(x,\hat{\mathbf{g}}'), e(\hat{g}_0, \mathsf{msk}))\\ &= (\hat{g}_0, \mathsf{sE}(x,\hat{\mathbf{g}})\cdot\hat{g}_0^{\mathsf{sE}(x,\hat{\mathbf{u}})}, e(\hat{g}_0, \mathsf{msk}))\end{aligned}$$

where $\hat{\mathbf{u}}$ denotes the vector $\hat{\mathbf{u}} := (\hat{u}_1, \ldots, \hat{u}_n)$ and thus $\mathsf{sE}(x,\hat{\mathbf{g}}') = \mathsf{sE}(x, \hat{g}_0^{\hat{\mathbf{u}}}) = \hat{g}_0^{\mathsf{sE}(x,\hat{\mathbf{u}})}$;

$$\widehat{\mathsf{KeyGen}}(\mathsf{pp}_{\mathbb{G}}, 1, y; \hat{\mathbf{h}}\cdot\hat{\mathbf{h}}') = (\hat{h}_0, \mathsf{rE}(y,\hat{\mathbf{h}})\cdot\hat{h}_0^{\mathsf{rE}(y,\hat{\mathbf{u}})})$$

$$\widehat{\mathsf{KeyGen}}(\mathsf{pp}_{\mathbb{G}}, (h^*)^\alpha, y; \hat{\mathbf{h}}\cdot\hat{\mathbf{h}}'\cdot(h^*)^{(0,\hat{\mathbf{v}})}) = (\hat{h}_0, \mathsf{kE}(y,(h^*)^\alpha)\cdot\mathsf{rE}(y,\hat{\mathbf{h}})\cdot\hat{h}_0^{\mathsf{rE}(y,\hat{\mathbf{u}})}\cdot(h^*)^{\mathsf{rE}(y,\hat{\mathbf{v}})}).$$

Since h^* lies in the group generated by $\hat{h}_0$, we may replace h^* with $\hat{h}_0^\eta$ for some $\eta \in \mathbb{Z}_p$, then

$$\widehat{\mathsf{KeyGen}}(\mathsf{pp}_{\mathbb{G}}, (h^*)^\alpha, y; \hat{\mathbf{h}}\cdot\hat{\mathbf{h}}'\cdot(h^*)^{(0,\hat{\mathbf{v}})}) = (\hat{h}_0, \mathsf{rE}(y,\hat{\mathbf{h}})\cdot(\hat{h}_0)^{\mathsf{kE}(y,\eta\alpha)+\mathsf{rE}(y,\hat{\mathbf{u}})+\mathsf{rE}(y,\eta\hat{\mathbf{v}})})$$

The claim then follows readily from *attribute-hiding* encoding, that is, $\mathsf{sE}(x,\hat{\mathbf{u}})$ and $\mathsf{rE}(y,\hat{\mathbf{u}})\}$ are uniformly distributed. □

Lemma 15 (pseudo-SF to SF keys: $\mathsf{Game}_{2,i,2}$ to $\mathsf{Game}_{2,i,3}$). *For $i = 1, \ldots, q$, for any adversary $\mathcal{A}$ that makes at most q key queries, there exists an adversary $\mathcal{B}_3$ such that*

$$|\mathsf{Adv}_{2,i,2}(\lambda) - \mathsf{Adv}_{2,i,3}(\lambda)| \le \mathsf{Adv}_{\mathcal{B}_3}^{\mathrm{RS}}(\lambda)$$

and $\mathsf{Time}(\mathcal{B}_3) \approx \mathsf{Time}(\mathcal{A}) + q\cdot\mathrm{poly}(\lambda, n)$ *where* $\mathrm{poly}(\lambda,n)$ *is independent of* $\mathsf{Time}(\mathcal{A})$.

Proof. The proof is completely analogous to Lemma 13, except we use $\widehat{\mathsf{msk}}$ instead of msk along with additional h^* to generate the i'th key query. That is, $\mathcal{B}_3$ is exactly the same as $\mathcal{B}_2$, with the following change:

Key Queries. On input the j'th secret key query y, output

$$\mathsf{sk}_y \leftarrow \begin{cases}\widehat{\mathsf{KeyGen}}(\mathsf{pp}_{\mathbb{G}}, \widehat{\mathsf{msk}}, y; \mathsf{SampH}(\mathsf{pp})\cdot(h^*)^{(0,\hat{\mathbf{v}}^j)}) & \text{if } j < i\\ \widehat{\mathsf{KeyGen}}(\mathsf{pp}_{\mathbb{G}}, \boxed{\widehat{\mathsf{msk}}}, y; \mathbf{t}\cdot\boxed{(h^*)^{(0,\hat{\mathbf{v}}^i)}}) & \text{if } j = i\\ \widehat{\mathsf{KeyGen}}(\mathsf{pp}_{\mathbb{G}}, \mathsf{msk}, y; \mathsf{SampH}(\mathsf{pp})) & \text{if } j > i\end{cases}.$$

Observe that when $\mathbf{t} = \mathbf{h}$, the output is identical to that in $\mathsf{Game}_{2,i,3}$; and when $\mathbf{t} = \mathbf{h}\cdot\hat{\mathbf{h}}$, the output is identical to that in $\mathsf{Game}_{2,i,2}$. We may therefore conclude that: $|\mathsf{Adv}_{2,i,2}(\lambda) - \mathsf{Adv}_{2,i,3}(\lambda)| \le \mathsf{Adv}_{\mathcal{B}_3}^{\mathrm{RS}}(\lambda)$. □

Lemma 16 (second last transition: $\mathsf{Game}_{2,q,3}$ to Game_3). *For any adversary $\mathcal{A}$, we have*

$$|\mathsf{Adv}_{2,q,3}(\lambda) - \mathsf{Adv}_3(\lambda)| = 0.$$

Proof. The proof is completely analogous to Lemma 9, except we will additional use h^* to generate the all the semi-functional keys:

Key Queries. On input the j'th secret key query y, output

$$\mathsf{sk}_y \leftarrow \widehat{\mathsf{KeyGen}}(\mathsf{pp}_{\mathbb{G}}, \widehat{\mathsf{msk}}, y; \mathsf{SampH}(\mathsf{pp})\cdot\boxed{(h^*)^{(0,\hat{\mathbf{v}}^j)}}).$$

We may then conclude that: $|\mathsf{Adv}_{2,q,3}(\lambda) - \mathsf{Adv}_3(\lambda)| = 0$. $\square$

Lemma 17 (final transition: Game_3 **to** Game_4**).** *For any adversary $\mathcal{A}$, we have*

$$|\mathsf{Adv}_3(\lambda) - \mathsf{Adv}_4(\lambda)| = 0.$$

Proof. First, we sample $(\mathsf{msk}, \widehat{\mathsf{msk}})$ in both games as follows: pick $\widehat{\mathsf{msk}} \leftarrow_{\mathrm{R}} \mathbb{H}$, $\alpha \leftarrow_{\mathrm{R}} \mathbb{Z}_p$ and set $\mathsf{msk} := \widehat{\mathsf{msk}} \cdot (h^*)^{-\alpha}$.

We may then simulate key set-up and answer key queries given just

$$\{\mathsf{pp}_{\mathbb{G}}, h^*, \widehat{\mathsf{msk}}\} \quad \text{and} \quad \mathsf{SampH}^* \ \text{(from } \mathbb{H}\text{-}\textit{hiding}\text{)}$$

as follows:

Setup. Observe that

$$\mu(\mathsf{msk}) = \mu(\widehat{\mathsf{msk}}) \cdot \mu((h^*)^{-\alpha}) = \mu(\widehat{\mathsf{msk}})$$

where in the last equality, we use *orthogonality* $\mu(h^*) = 1$. Output

$$\mathsf{mpk} := \Big(\mathsf{pp}_{\mathbb{G}},\ \mu(\widehat{\mathsf{msk}})\Big).$$

Key Queries. On input the j'th secret key query y, output

$$\mathsf{sk}_y \leftarrow \widehat{\mathsf{KeyGen}}(\mathsf{pp}_{\mathbb{G}}, \widehat{\mathsf{msk}}, y; \mathsf{SampH}^*(\mathsf{pp}_{\mathbb{G}}, h^*)).$$

Now, observe that the challenge ciphertext in Game_3 is given by:

$$\widehat{\mathsf{Enc}}(\mathsf{pp}_{\mathbb{G}}, x_b^*, m; \widehat{\mathsf{msk}}, \mathbf{g} \cdot \hat{\mathbf{g}}) = (C_0, \mathbf{C}_1, C_2' \cdot m) = (g_0 \hat{g}_0, \mathsf{sE}(x_b^*, \mathbf{g} \cdot \hat{\mathbf{g}}), C_2' \cdot m),$$

where $(C_0, \mathbf{C}_1)$ depend only on $\mathbf{g} \cdot \hat{\mathbf{g}} = (g_0 \cdot \hat{g}_0, \ldots)$, and $C_2' \cdot m$ is uniformly distributed over $\mathbb{G}_T$. By $\mathbb{G}$-*uniformity*, we may replace $\{\mathsf{pp}_{\mathbb{G}}, h^*, \boxed{\mathbf{g} \cdot \hat{\mathbf{g}}}\}$ with $\{\mathsf{pp}_{\mathbb{G}}, h^*, \boxed{\mathbf{g}'}\}$ and obtain

$$C_0 = g_0 \hat{g}_0,\ \mathbf{C}_1 = \mathsf{sE}(x_b^*, \mathbf{g}')$$

where $\mathbf{g}' \leftarrow_{\mathrm{R}} \{g_0 \hat{g}_0\} \times \mathbb{G}^n$. The lemma then follows readily from *attribute-hiding* encoding, that is, $\mathbf{C}_1$ is uniformly distributed over $\mathbb{G}^{|\mathsf{sE}(\cdot)|}$. We may then conclude that: $|\mathsf{Adv}_3(\lambda) - \mathsf{Adv}_4(\lambda)| = 0$. $\square$

Acknowledgments. We thank Eike Kiltz and Jiaxin Pan for insightful discussions, and the anonymous reviewers for helpful feedback on the write-up.

References

[1] S. Agrawal, D. M. Freeman, and V. Vaikuntanathan. Functional encryption for inner product predicates from learning with errors. In *ASIACRYPT*, pages 21–40, 2011.

[2] N. Attrapadung. Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In *EUROCRYPT*, pages 557–577, 2014.

[3] N. Attrapadung and B. Libert. Functional encryption for inner product: Achieving constant-size ciphertexts with adaptive security or support for negation. In *Public Key Cryptography*, pages 384–402, 2010.

[4] A. Beimel. *Secure Schemes for Secret Sharing and Key Distribution.* Ph.D., Technion - Israel Institute of Technology, 1996.

[5] O. Blazy, E. Kiltz, and J. Pan. (Hierarchical) identity-based encryption from affine message authentication. In *CRYPTO*, pages 408–425, 2014.

[6] D. Boneh and X. Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In *EUROCRYPT*, pages 223–238, 2004.

[7] D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003.

[8] D. Boneh and M. Hamburg. Generalized identity based and broadcast encryption schemes. In *ASIACRYPT*, pages 455–470, 2008.

[9] D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. In *TCC*, pages 535–554, 2007. Also Cryptology ePrint Archive, Report 2006/287.

[10] D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *EUROCRYPT*, pages 440–456, 2005.

[11] D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, and D. Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In *EUROCRYPT*, pages 533–556, 2014.

[12] C. Chen, Z. Zhang, and D. Feng. Fully secure doubly-spatial encryption under simple assumptions. In *ProvSec*, pages 253–263, 2012.

[13] J. Chen and H. Wee. Fully, (almost) tightly secure IBE and dual system groups. In *CRYPTO (2)*, pages 435–460, 2013.

[14] J. Chen and H. Wee. Dual system groups and its applications — compact HIBE and more. IACR Cryptology ePrint Archive, Report 2014/265, 2014. Preliminary version in [13].

[15] J. Chen, H. W. Lim, S. Ling, H. Wang, and H. Wee. Shorter IBE and signatures via asymmetric pairings. In *Pairing*, pages 122–140, 2012.

[16] L. Cheung and C. C. Newport. Provably secure ciphertext policy ABE. In *ACM Conference on Computer and Communications Security*, pages 456–465, 2007.

[17] C. Cocks. An identity based encryption scheme based on quadratic residues. In *IMA Int. Conf.*, pages 360–363, 2001.

[18] A. Escala, G. Herold, E. Kiltz, C. Ràfols, and J. L. Villar. An algebraic framework for diffie-hellman assumptions. In *CRYPTO (2)*, pages 129–147, 2013.

[19] A. Fiat and M. Naor. Broadcast encryption. In *CRYPTO*, pages 480–491, 1993.

[20] D. M. Freeman. Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In *EUROCRYPT*, pages 44–61, 2010.

[21] S. Garg, A. Kumarasubramanian, A. Sahai, and B. Waters. Building efficient fully collusion-resilient traitor tracing and revocation schemes. In *ACM Conference on Computer and Communications Security*, pages 121–130, 2010.

[22] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM Conference on Computer and Communications Security*, pages 89–98, 2006.

[23] A. Guillevic. Comparing the pairing efficiency over composite-order and prime-order elliptic curves. In *ACNS*, pages 357–372, 2013.

[24] M. Hamburg. Spatial encryption. *IACR Cryptology ePrint Archive*, 2011:389, 2011.

[25] Y. Ishai and H. Wee. Partial garbling schemes and their applications. In *ICALP*, pages 650–662, 2014.

[26] C. S. Jutla and A. Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In *ASIACRYPT (1)*, pages 1–20, 2013.

[27] M. Karchmer and A. Wigderson. On span programs. In *Structure in Complexity Theory Conference*, pages 102–111, 1993.

[28] J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT*, pages 146–162, 2008.

[29] K. Lee and D. H. Lee. Adaptively secure broadcast encryption under standard assumptions with better efficiency. *IACR Cryptology ePrint Archive*, 2013:488, 2013.

[30] A. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In *EUROCRYPT*, pages 318–335, 2012.

[31] A. B. Lewko. *Functional Encryption: New Proof Techniques and Advancing Capabilities.* Ph.D., The University of Texas at Austin, 2012.

[32] A. B. Lewko and B. Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In *TCC*, pages 455–479, 2010.

[33] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In *EUROCRYPT*, pages 62–91, 2010.

[34] T. Okamoto and K. Takashima. Homomorphic encryption and signatures from vector decomposition. In *Pairing*, pages 57–74, 2008.

[35] T. Okamoto and K. Takashima. Hierarchical predicate encryption for inner-products. In *ASIACRYPT*, pages 214–231, 2009.

[36] T. Okamoto and K. Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In *CRYPTO*, pages 191–208, 2010.

[37] T. Okamoto and K. Takashima. Achieving short ciphertexts or short secret-keys for adaptively secure general inner-product encryption. In *CANS*, pages 138–159, 2011. Also, Cryptology ePrint Archive, Report 2011/648.

[38] T. Okamoto and K. Takashima. Adaptively attribute-hiding (hierarchical) inner product encryption. In *EUROCRYPT*, pages 591–608, 2012. Also, Cryptology ePrint Archive, Report 2011/543.

[39] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, pages 457–473, 2005.

[40] A. Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, 1984.

[41] K. Takashima. Expressive attribute-based encryption with constant-size ciphertexts from the decisional linear assumption. In *Security and Cryptography for Networks (SCN)*, pages 298–317, 2014.

[42] B. Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In *CRYPTO*, pages 619–636, 2009.

[43] H. Wee. Dual system encryption via predicate encodings. In *TCC*, pages 616–637, 2014.

A Instantiations of Predicate Encodings

We present $\mathbb{Z}_p$-bilinear predicate encodings for a large class of predicates that have been considered in the literature. For any vector $\mathbf{u}$, we denote by u_i the ith coordinate of $\mathbf{u}$. We denote by $\mathbf{0}$ the vector whose all coordinates are 0. The encodings in Sections A.1, A.2, A.3, A.4 are the same as those in [43].

A.1 Inner Product (ZIPE)

Predicate [28]. Here, $\mathcal{X} = \mathcal{Y} := \mathbb{Z}_p^n$ and

$$\mathsf{P}(\mathbf{x}, \mathbf{y}) = 1 \text{ iff } \mathbf{x}^\top \mathbf{y} = 0$$

First encoding (short secret keys) [6].

- $\mathcal{W} := \mathbb{Z}_p \times \mathbb{Z}_p^n$;
- $\mathsf{sE}(\mathbf{x}, (u, \mathbf{w})) := u\mathbf{x} + \mathbf{w} \in \mathbb{Z}_p^n$
- $\mathsf{rE}(\mathbf{y}, (u, \mathbf{w})) := \mathbf{w}^\top \mathbf{y} \in \mathbb{Z}_p$
- $\mathsf{kE}(\mathbf{y}, \alpha) := \alpha \in \mathbb{Z}_p$
- $\mathsf{sD}(\mathbf{x}, \mathbf{y}, \mathbf{c}) := \mathbf{c}^\top \mathbf{y}$
- $\mathsf{rD}(\mathbf{x}, \mathbf{y}, d) := d$

  Note that this encoding is actually an attribute-hiding encoding, as defined in Section 7.2.

(x-oblivious α-reconstruction.) $\mathsf{sD}(\mathbf{x}, \mathbf{y}, \cdot)$ and $\mathsf{rD}(\mathbf{x}, \mathbf{y}, \cdot)$ are independent of $\mathbf{x}$.

(attribute-hiding.) For all $\mathbf{x}, \mathbf{y}$ such that $\mathbf{x}^\top \mathbf{y} \neq 0$, $(\mathsf{sE}(\mathbf{x}, (u, \mathbf{w})), \mathsf{rE}(\mathbf{y}, (u, \mathbf{w}))) := (u\mathbf{x} + \mathbf{w}, \mathbf{w}^\top \mathbf{y})$ is uniformly distributed, when the randomness is taken over $(u, \mathbf{w}) \leftarrow_{\mathrm{R}} \mathcal{W}$.

Second encoding (short ciphertext) [10].

- $\mathcal{W} := \mathbb{Z}_p \times \mathbb{Z}_p \times \mathbb{Z}_p^n$;
- $\mathsf{sE}(\mathbf{x}, (u_0, u_1, \mathbf{w})) := u_1 + \mathbf{x}^\top \mathbf{w} \in \mathbb{Z}_p$
- $\mathsf{rE}(\mathbf{y}, (u_0, u_1, \mathbf{w})) := (u_0 \mathbf{y} + \mathbf{w}, u_1) \in \mathbb{Z}_p^{n+1}$
- $\mathsf{kE}(\mathbf{y}, \alpha) := (\mathbf{0}, \alpha) \in \mathbb{Z}_p^{n+1}$
- $\mathsf{sD}(\mathbf{x}, \mathbf{y}, c) := c$
- $\mathsf{rD}(\mathbf{x}, \mathbf{y}, (\mathbf{d}, d')) := \mathbf{x}^\top \mathbf{d} + d'$

A.2 Non-Zero Inner Product (NIPE)

Predicate [3]. Here, $\mathcal{X} = \mathcal{Y} := \mathbb{Z}_p^n$ and

$$\mathsf{P}(\mathbf{x}, \mathbf{y}) = 1 \text{ iff } \mathbf{x}^\top \mathbf{y} \neq 0$$

The constructions exploit the following simple algebraic fact: given $\mathbf{x}, \mathbf{y}, u\mathbf{x} + \mathbf{w}, \mathbf{y}^\top \mathbf{w}$,

- if $\mathbf{x}^\top \mathbf{y} \neq 0$, then we can recover u.
- if $\mathbf{x}^\top \mathbf{y} = 0$, then u is perfectly random.

First encoding (short ciphertext).

- $\mathcal{W} := \mathbb{Z}_p^n$;
- $\mathsf{sE}(\mathbf{x}, \mathbf{w}) := \mathbf{x}^\top \mathbf{w} \in \mathbb{Z}_p$
- $\mathsf{rE}(\mathbf{y}, \mathbf{w}) := \mathbf{w} \in \mathbb{Z}_p^n$
- $\mathsf{kE}(\mathbf{y}, \alpha) := \alpha \mathbf{y} \in \mathbb{Z}_p^n$
- $\mathsf{sD}(\mathbf{x}, \mathbf{y}, \mathbf{c}) := \mathbf{c} \cdot (\mathbf{x}^\top \mathbf{y})^{-1}$
- $\mathsf{rD}(\mathbf{x}, \mathbf{y}, \mathbf{d}) := \mathbf{x}^\top \mathbf{d} \cdot (\mathbf{x}^\top \mathbf{y})^{-1}$

Second encoding (short secret keys).

- $\mathcal{W} := \mathbb{Z}_p \times \mathbb{Z}_p^n$;
- $\mathsf{sE}(\mathbf{x}, (u, \mathbf{w})) := u\mathbf{x} + \mathbf{w} \in \mathbb{Z}_p^n$
- $\mathsf{rE}(\mathbf{y}, (u, \mathbf{w})) := (\mathbf{w}^\top \mathbf{y}, u) \in \mathbb{Z}_p^2$
- $\mathsf{kE}(\mathbf{y}, \alpha) := (0, \alpha) \in \mathbb{Z}_p^2$
- $\mathsf{sD}(\mathbf{x}, \mathbf{y}, \mathbf{c}) := \mathbf{c}^\top \mathbf{y} \cdot (\mathbf{x}^\top \mathbf{y})^{-1}$
- $\mathsf{rD}(\mathbf{x}, \mathbf{y}, (d, d')) := d \cdot (\mathbf{x}^\top \mathbf{y})^{-1} + d'$

A.3 Spatial Encryption

Predicate [8]. Here, $\mathcal{X} := \mathbb{Z}_p^n, \mathcal{Y} := \mathbb{Z}_p^{n \times \ell}$ and

$$\mathsf{P}(\mathbf{x}, \mathbf{Y}) = 1 \text{ iff } \mathbf{x} \in \text{span}(\mathbf{Y})$$

Here, span refers to the linear span of a collection of column vectors. Recall from [8] that spatial encryption generalizes HIBE.

Supporting delegation. Consider a predicate P that supports delegation, namely, there is a partial ordering $\leq$ on $\mathcal{Y}$ such that for all $x \in \mathcal{X}$, the predicate $\mathsf{P}(x, \cdot)$ is monotone, i.e.

$$(y \leq y') \wedge \mathsf{P}(x, y) = 1 \implies \mathsf{P}(x, y') = 1.$$

For instance, in HIBE, $y \leq y'$ iff y' is a prefix of y. A bilinear encoding $(\mathsf{sE}, \mathsf{rE})$ for such a predicate supports delegation if given y, y' such that $y \leq y'$, we can efficiently compute a linear map L such that for all $\alpha, \mathbf{w}, r$, L maps $(\mathbf{w}, \mathsf{rE}(y', \mathbf{w}) + r \cdot \mathsf{kE}(\alpha, \mathbf{w}))$ to $\mathsf{rE}(y, \mathbf{w}) + r \cdot \mathsf{kE}(\alpha, \mathbf{w})$. Note that we can always rerandomize the output due to linearity of receiver encoding.

Encoding (short ciphertext) [8, 10, 32, 12].

- $\mathcal{W} = \mathbb{Z}_p \times \mathbb{Z}_p^n$;
- $\mathsf{sE}(\mathbf{x}, (u, \mathbf{w})) := u + \mathbf{w}^\top \mathbf{x} \in \mathbb{Z}_p$
- $\mathsf{rE}(\alpha, \mathbf{Y}, (u, \mathbf{w})) := (\mathbf{w}^\top \mathbf{Y}, u) \in \mathbb{Z}_p^{\ell+1}$
- $\mathsf{kE}(\mathbf{Y}, \alpha) := (\mathbf{0}, \alpha) \in \mathbb{Z}_p^{\ell+1}$
- $\mathsf{sD}(\mathbf{x}, \mathbf{Y}, \mathbf{c}) := \mathbf{c}$
- $\mathsf{rD}(\mathbf{x}, \mathbf{Y}, (\mathbf{d}, d')) := \mathbf{d}^\top \mathbf{a} + d'$, where $\mathbf{a} \in \mathbb{Z}_p^\ell$ is such that $\mathbf{x} := \mathbf{Y}\mathbf{a}$.

α-privacy relies on the fact that if $\mathbf{x} \notin \text{span}(\mathbf{Y})$, then $\mathbf{w}^\top \mathbf{x}$ is statistically independent of $\mathbf{w}^\top \mathbf{Y}$ for a random $\mathbf{w} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^n$.

A.4 Doubly Spatial Encryption

Predicate [24]. Here, $\mathcal{X} := \mathbb{Z}_p^n \times \mathbb{Z}_p^{n\times\ell}, \mathcal{Y} := \mathbb{Z}_p^{n\times\ell'}$ and

$$\mathsf{P}((\mathbf{x}_0,\mathbf{X}),\mathbf{Y}) = 1 \text{ iff } (\mathbf{x}_0 + \mathrm{span}(\mathbf{X})) \cap \mathrm{span}(\mathbf{Y}) \neq \emptyset$$

Encoding [24].

- $\mathcal{W} = \mathbb{Z}_p \times \mathbb{Z}_p^n$;
- $\mathsf{sE}((\mathbf{x}_0,\mathbf{X}),(u,\mathbf{w})) := (\mathbf{w}^\top\mathbf{X}, u + \mathbf{w}^\top\mathbf{x}_0) \in \mathbb{Z}_p^{\ell+1}$
- $\mathsf{rE}(\mathbf{Y},(u,\mathbf{w})) := (\mathbf{w}^\top\mathbf{Y}, u) \in \mathbb{Z}_p^{\ell'+1}$
- $\mathsf{kE}(\mathbf{Y},\alpha) := (\mathbf{0},\alpha) \in \mathbb{Z}_p^{\ell'+1}$
- $\mathsf{sD}((\mathbf{x}_0,\mathbf{X}),\mathbf{Y},(\mathbf{c},c')) := \mathbf{c}^\top\mathbf{a} + c'$
- $\mathsf{rD}((\mathbf{x}_0,\mathbf{X}),\mathbf{Y},(\mathbf{d},d')) := \mathbf{d}^\top\mathbf{b} + d'$,
  where $\mathbf{a} \in \mathbb{Z}_p^{\ell}$ and $\mathbf{b} \in \mathbb{Z}_p^{\ell'}$ are such that $\mathbf{x}_0 + \mathbf{X}\mathbf{a} = \mathbf{Y}\mathbf{b}$.

α-privacy relies on the fact that if $(\mathbf{x}_0 + \mathrm{span}(\mathbf{X})) \cap \mathrm{span}(\mathbf{Y}) = \emptyset$ then $\mathbf{w}^\top\mathbf{x}_0$ is statistically independent of $\mathbf{w}^\top\mathbf{X}, \mathbf{w}^\top\mathbf{Y}$ for a random $\mathbf{w} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^n$.

A.5 Attribute-Based Encryption (ABE) for boolean span programs

We define (monotone) access structures using the language of (monotone) span programs [27].

Definition 1 (access structure [4, 27]). *A* (monotone) access structure *for attribute universe* $[n]$ *is a pair* $(\mathbf{M},\rho)$ *where* $\mathbf{M}$ *is a* $\ell \times \ell'$ *matrix over* $\mathbb{Z}_p$ *and* $\rho : [\ell] \to [n]$. *Given* $\mathbf{x} = (x_1,\ldots,x_n) \in \{0,1\}^n$, *we say that*

$$\mathbf{x} \textit{ satisfies } (\mathbf{M},\rho) \textit{ iff } \mathbf{1}^\top \in \textit{span}\langle\mathbf{M}_{\mathbf{x}}\rangle,$$

Here, $\mathbf{1}^\top := (1,0,\ldots,0) \in \mathbb{Z}^{1\times\ell'}$ *is a row vector;* $\mathbf{M}_{\mathbf{x}}$ *denotes the collection of vectors* $\{\mathbf{M}_j : x_{\rho(j)} = 1\}$ *where* $\mathbf{M}_j$ *denotes the j'th row of* $\mathbf{M}$*; and span refers to linear span of collection of (row) vectors over* $\mathbb{Z}_p$.

That is, $\mathbf{x}$ satisfies $(\mathbf{M},\rho)$ iff there exists constants $\omega_1,\ldots,\omega_\ell \in \mathbb{Z}_p$ such that

$$\sum_{j:x_{\rho(j)}=1} \omega_j \mathbf{M}_j = \mathbf{1}^\top \tag{7}$$

Observe that the constants $\{\omega_j\}$ can be computed in time polynomial in the size of the matrix $\mathbf{M}$ via Gaussian elimination. In order to achieve perfect α-privacy, we need to impose a one-use restriction, that is, ρ is a permutation and $\ell = n$. By re-ordering the rows of $\mathbf{M}$, we may assume WLOG that ρ is the identity map, which we omit in the rest of this section.

KP-ABE Predicate [22, 39]. Here, $\mathcal{X} := \{0,1\}^\ell, \mathcal{Y} := \mathbb{Z}_p^{\ell\times\ell'}$ and

$$\mathsf{P}(\mathbf{x},\mathbf{M}) = 1 \text{ iff } \mathbf{x} \text{ satisfies } \mathbf{M}$$

Encoding.

- $\mathcal{W} := \mathbb{Z}_p^\ell \times \mathbb{Z}_p^{\ell'-1}$.
- $\mathsf{sE}(\mathbf{x},(\mathbf{w},\mathbf{u})) := (x_1 w_1,\ldots,x_\ell w_\ell) \in \mathbb{Z}_p^\ell$
- $\mathsf{rE}(\mathbf{M},(\mathbf{w},\mathbf{u})) := \left(\mathbf{M}_1\binom{0}{\mathbf{u}} + w_1,\ldots,\mathbf{M}_\ell\binom{0}{\mathbf{u}} + w_\ell\right) \in \mathbb{Z}_p^\ell$.
- $\mathsf{kE}(\mathbf{M},\alpha) := \left(\mathbf{M}_1\binom{\alpha}{\mathbf{0}},\ldots,\mathbf{M}_\ell\binom{\alpha}{\mathbf{0}}\right) \in \mathbb{Z}_p^\ell$
- $\mathsf{sD}(\mathbf{x},\mathbf{M},\mathbf{c}) := \sum_{j=1}^\ell \omega_j c_j$ where $\omega_1,\ldots,\omega_\ell$ are computed as in (7)
- $\mathsf{rD}(\mathbf{x},\mathbf{M},\mathbf{d}) := \sum_{j=1}^\ell x_j\omega_j d_j$

Correctness. Suppose $\mathsf{P}(\mathbf{x},\mathbf{M}) = 1$. Observe that

$$\begin{aligned}\mathsf{rD}(\mathbf{x},\mathbf{M},\mathsf{kE}(\mathbf{M},\alpha)) &= \mathbf{1}^{\top}\begin{pmatrix}\alpha\\ \mathbf{0}\end{pmatrix} = \alpha\\ \mathsf{rD}(\mathbf{x},\mathbf{M},\mathsf{rE}(\mathbf{M},(\mathbf{w},\mathbf{u}))) &= \mathbf{1}^{\top}\begin{pmatrix}0\\ \mathbf{u}\end{pmatrix} + \sum_{j=1}^{\ell} x_j\omega_j w_j = \sum_{j=1}^{\ell} x_j\omega_j w_j = \mathsf{sD}(\mathbf{x},\mathbf{M},\mathsf{sE}(\mathbf{x},(\mathbf{w},\mathbf{u})))\end{aligned}$$

Privacy. Suppose $\mathsf{P}(\mathbf{x},\mathbf{M}) = 0$. We have

$$\begin{aligned}\mathsf{sE}(\mathbf{x},(\mathbf{w},\mathbf{u})) &= (x_1 w_1,\ldots,x_\ell w_\ell)\\ \mathsf{rE}(\mathbf{M},(\mathbf{w},\mathbf{u})) + \mathsf{kE}(\mathbf{M},\alpha) &= \left(\mathbf{M}_1\begin{pmatrix}\alpha\\ \mathbf{u}\end{pmatrix} + w_1,\ldots\mathbf{M}_\ell\begin{pmatrix}\alpha\\ \mathbf{u}\end{pmatrix} + w_\ell\right)\end{aligned}$$

Privacy follows readily from the fact that:

- $\{\mathbf{M}_j\begin{pmatrix}\alpha\\ \mathbf{u}\end{pmatrix} : x_j = 1\}$ reveal no information about α since $\mathbf{1}^{\top} \notin \mathrm{span}(\mathbf{M}_{\mathbf{x}})$;
- $\{\mathbf{M}_j\begin{pmatrix}\alpha\\ \mathbf{u}\end{pmatrix} : x_j = 0\}$ are masked by w_j.

CP-ABE Predicate [22, 16]. As before with $\mathcal{X}$ and $\mathcal{Y}$ switched, so that

$$\mathsf{P}(\mathbf{M},\mathbf{x}) = 1 \text{ iff } \mathbf{x} \text{ satisfies } \mathbf{M}$$

Encoding.

- $\mathcal{W} := \mathbb{Z}_p^{\ell} \times \mathbb{Z}_p^{\ell'-1} \times \mathbb{Z}_p$.
- $\mathsf{sE}(\mathbf{M},(\mathbf{w},\mathbf{u},u_0')) := (w_1 + \mathbf{M}_1\begin{pmatrix}u_0'\\ \mathbf{u}\end{pmatrix},\ldots,w_\ell + \mathbf{M}_\ell\begin{pmatrix}u_0'\\ \mathbf{u}\end{pmatrix}) \in \mathbb{Z}_p^{\ell}$.
- $\mathsf{rE}(\mathbf{x},(\mathbf{w},\mathbf{u},u_0')) := (u_0', x_1 w_1,\ldots,x_\ell w_\ell) \in \mathbb{Z}_p^{\ell+1}$
- $\mathsf{kE}(\mathbf{x},\alpha) := (\alpha,\mathbf{0}) \in \mathbb{Z}_p^{\ell}$.
- $\mathsf{sD}(\mathbf{M},\mathbf{x},\mathbf{c}) := \sum_{j=1}^{\ell} x_j\omega_j c_j$ where $\omega_1,\ldots,\omega_\ell$ are computed as in (7)
- $\mathsf{rD}(\mathbf{M},\mathbf{x},(d',\mathbf{d})) := d' + \sum_{j=1}^{\ell}\omega_j d_j$

Correctness. Suppose $\mathsf{P}(\mathbf{M},\mathbf{x}) = 1$. Observe that

$$\begin{aligned}\mathsf{rD}(\mathbf{M},\mathbf{x},\mathsf{kE}(\mathbf{x},\alpha)) &= \alpha + \sum_{j=1}^{\ell}\omega_j \cdot 0 = \alpha\\ \mathsf{rD}(\mathbf{M},\mathbf{x},\mathsf{rE}(\mathbf{x},(\mathbf{w},\mathbf{u},u_0')) &= u_0' + \sum_{j=1}^{\ell} x_j\omega_j w_j = \mathbf{1}^{\top}\begin{pmatrix}u_0'\\ \mathbf{u}\end{pmatrix} + \sum_{j=1}^{\ell} x_j\omega_j w_j = \mathsf{sD}(\mathbf{M},\mathbf{x},\mathsf{sE}(\mathbf{M},(\mathbf{w},\mathbf{u},u_0'))\end{aligned}$$

Privacy. Suppose $\mathsf{P}(\mathbf{M},\mathbf{x}) = 0$. We have

$$\begin{aligned}\mathsf{sE}(\mathbf{M},(\mathbf{w},\mathbf{u},u_0')) &= \left(\mathbf{M}_1\begin{pmatrix}u_0'\\ \mathbf{u}\end{pmatrix} + w_1,\ldots\mathbf{M}_\ell\begin{pmatrix}u_0'\\ \mathbf{u}\end{pmatrix} + w_\ell\right)\\ \mathsf{rE}(\mathbf{x},(\mathbf{w},\mathbf{u},u_0')) + \mathsf{kE}(\mathbf{x},\alpha) &= (u_0' + \alpha, x_1 w_1,\ldots,x_\ell w_\ell)\end{aligned}$$

Privacy follows readily from the fact that:

- α is masked by u_0';
- $\{\mathbf{M}_j\begin{pmatrix}u_0'\\ \mathbf{u}\end{pmatrix} : x_j = 1\}$ reveal no information about v since $\mathbf{1}^{\top} \notin \mathrm{span}(\mathbf{M}_{\mathbf{x}})$;
- $\{\mathbf{M}_j\begin{pmatrix}u_0'\\ \mathbf{u}\end{pmatrix} : x_j = 0\}$ are masked by w_j.

A.6 Attribute-Based Encryption (ABE) for arithmetic span programs

We use the terminology of arithmetic span programs, which captures both boolean as well as arithmetic formula and branching programs [25].

Definition 2 (arithmetic span program [25]). *An arithmetic span program $(\mathcal{V},\rho)$ is a collection of row vectors $\mathcal{V} = \{(\mathbf{y}_j^\top, \mathbf{z}_j^\top) : j \in [\ell]\}$ in $\mathbb{Z}_p^{1\times\ell'}$ and $\rho : [\ell] \to [n]$. We say that*

$$\mathbf{x} \in \mathbb{Z}_p^n \textit{ satisfies } (\mathcal{V},\rho) \textit{ iff } \mathbf{1}^\top \in \textit{span}\langle x_{\rho(j)}\mathbf{y}_j^\top + \mathbf{z}_j^\top\rangle,$$

where $\mathbf{1}^\top := (1,0,\ldots,0) \in \mathbb{Z}_p^{\ell'}$ and span refers to linear span of a collection of row vectors.

That is, $\mathbf{x}$ satisfies $(\mathcal{V},\rho)$ iff there exists constants $\omega_1,\ldots,\omega_\ell \in \mathbb{Z}_p$ such that

$$\sum_{j=1}^{\ell} \omega_j (x_{\rho(j)}\mathbf{y}_j^\top + \mathbf{z}_j^\top) = \mathbf{1}^\top. \tag{8}$$

In order to achieve perfect α-privacy, we need to impose a one-use restriction, that is, ρ is a permutation and $\ell = n$. By re-ordering the coordinates in $\mathcal{V}$, we may assume WLOG that ρ is the identity map, which we omit in the rest of this section.

KP-ABE Predicate [25]. Here, $\mathcal{X} := \mathbb{Z}_p^\ell, \mathcal{Y} := \{(\mathbf{y}_j^\top, \mathbf{z}_j^\top) : j \in [\ell]\}$ and

$$\mathsf{P}(\mathbf{x},\mathcal{V}) = 1 \text{ iff } \mathbf{x} \text{ satisfies } \mathcal{V}$$

Encoding.

- $\mathcal{W} := \mathbb{Z}_p^\ell \times \mathbb{Z}_p^\ell \times \mathbb{Z}_p^{\ell'-1}$.
- $\mathsf{sE}(\mathbf{x},(\mathbf{w},\mathbf{v},\mathbf{u})) := (x_1 w_1 + v_1,\ldots,x_\ell w_\ell + v_\ell) \in \mathbb{Z}_p^\ell$.
- $\mathsf{rE}(\mathcal{V},(\mathbf{w},\mathbf{v},\mathbf{u})) := \left(\mathbf{y}_1^\top \begin{pmatrix}0\\ \mathbf{u}\end{pmatrix} + w_1,\ldots,\mathbf{y}_\ell^\top \begin{pmatrix}0\\ \mathbf{u}\end{pmatrix} + w_\ell,\quad \mathbf{z}_1^\top \begin{pmatrix}0\\ \mathbf{u}\end{pmatrix} + v_1,\ldots,\mathbf{z}_\ell^\top \begin{pmatrix}0\\ \mathbf{u}\end{pmatrix} + v_\ell\right) \in \mathbb{Z}_p^{2\ell}$
- $\mathsf{kE}(\mathcal{V},\alpha) := \left(\mathbf{y}_1^\top \begin{pmatrix}\alpha\\ \mathbf{0}\end{pmatrix},\ldots,\mathbf{y}_\ell^\top \begin{pmatrix}\alpha\\ \mathbf{0}\end{pmatrix},\quad \mathbf{z}_1 \begin{pmatrix}\alpha\\ \mathbf{0}\end{pmatrix},\ldots,\mathbf{z}_\ell \begin{pmatrix}\alpha\\ \mathbf{0}\end{pmatrix}\right) \in \mathbb{Z}_p^{2\ell}$.
- $\mathsf{sD}(\mathbf{x},\mathcal{V},\mathbf{c}) := \sum_{j=1}^{\ell} \omega_j c_j$, where $\omega_1,\ldots,\omega_\ell$ are computed as in (8)
- $\mathsf{rD}(\mathbf{x},\mathcal{V},(\mathbf{d},\mathbf{d}')) := \sum_{j=1}^{\ell} \omega_j (x_j d_j + d'_j)$

Correctness. Suppose $\mathsf{P}(\mathbf{x},\mathcal{V}) = 1$. Observe that

$$\begin{aligned} \mathsf{rD}(\mathbf{x},\mathcal{V},\mathsf{kE}(\mathcal{V},\alpha)) &= \mathbf{1}^\top \begin{pmatrix}\alpha\\ \mathbf{0}\end{pmatrix} = \alpha \\ \mathsf{rD}(\mathbf{x},\mathcal{V},\mathsf{rE}(\mathcal{V},(\mathbf{w},\mathbf{v},\mathbf{u}))) &= \mathbf{1}^\top \begin{pmatrix}0\\ \mathbf{u}\end{pmatrix} + \sum_{j=1}^{\ell} \omega_j \left(x_j w_j + v_j\right) = \mathsf{sD}(\mathbf{x},\mathcal{V},\mathsf{sE}(\mathbf{x},(\mathbf{w},\mathbf{v},\mathbf{u}))) \end{aligned}$$

Privacy. Suppose $\mathsf{P}(\mathbf{x},\mathcal{V}) = 0$. Write

$$\beta_j := \mathbf{y}_j^\top \begin{pmatrix}\alpha\\ \mathbf{u}\end{pmatrix} \quad \text{and} \quad \gamma_j := \mathbf{z}_j^\top \begin{pmatrix}\alpha\\ \mathbf{u}\end{pmatrix}$$

Then, we have

$$\begin{aligned} \mathsf{sE}(\mathbf{x},(\mathbf{w},\mathbf{v},\mathbf{u})) &= (x_1 w_1 + v_1,\ldots,x_\ell w_\ell + v_\ell) \\ \mathsf{rE}(\mathcal{V},(\mathbf{w},\mathbf{v},\mathbf{u})) + \mathsf{kE}(\mathcal{V},\alpha) &= (\beta_1 + w_1,\ldots,\beta_\ell + w_\ell,\gamma_1 + v_1,\ldots,\gamma_\ell + v_\ell) \end{aligned}$$

Privacy follows readily from the fact that:

- $\{x_j\beta_j + \gamma_j : j \in [\ell]\}$ reveal no information about α since $\mathbf{1}^\top \notin \text{span}\langle x_j\mathbf{y}_j^\top + \mathbf{z}_j^\top\rangle$;

- given $(x_j, x_j\beta_j + \gamma_j)$, we can simulate

  $$(x_j w_j + v_j, \beta_j + w_j, \gamma_j + v_j)$$

  using the relation $x_j w_j + v_j = x_j(\beta_j + w_j) + (\gamma_j + v_j) - (x_j\beta_j + \gamma_j)$.

CP-ABE Predicate. As before with $\mathcal{X}$ and $\mathcal{Y}$ switched, so that

$$\mathsf{P}(\mathcal{V}, \mathbf{x}) = 1 \text{ iff } \mathbf{x} \text{ satisfies } \mathcal{V}$$

Encoding.

- $\mathcal{W} := \mathbb{Z}_p^\ell \times \mathbb{Z}_p^\ell \times \mathbb{Z}_p^{\ell'-1} \times \mathbb{Z}_p$.
- $\mathsf{sE}(\mathcal{V}, (\mathbf{w}, \mathbf{v}, \mathbf{u}, u_0')) := (\beta_1 + w_1, \dots, \beta_\ell + w_\ell, \gamma_1 + v_1, \dots, \gamma_\ell + v_\ell) \in \mathbb{Z}_p^{2\ell}$, where $\beta_i := \mathbf{y}_i^\top \binom{u_0'}{\mathbf{u}}$ and $\gamma_i := \mathbf{z}_i^\top \binom{u_0'}{\mathbf{u}}$ are the i'th shares of u_0'.
- $\mathsf{rE}(\mathbf{x}, (\mathbf{w}, \mathbf{v}, \mathbf{u}, u_0')) := (u_0', x_1 w_1 + v_1, \dots, x_\ell w_\ell + v_\ell) \in \mathbb{Z}_p^{\ell+1}$.
- $\mathsf{kE}(\mathbf{x}, \alpha) := (\alpha, \mathbf{0}) \in \mathbb{Z}_p^{\ell+1}$.
- $\mathsf{sD}(\mathcal{V}, \mathbf{x}, (\mathbf{c}, \mathbf{c}')) := \sum_{j=1}^{\ell} \omega_j (x_j c_j + c_j')$ where $\omega_1, \dots, \omega_\ell$ are computed as in (8)
- $\mathsf{rD}(\mathcal{V}, \mathbf{x}, (d', \mathbf{d})) := d' + \sum_{j=1}^{\ell} \omega_j d_j$

Correctness. Suppose $\mathsf{P}(\mathcal{V}, \mathbf{x}) = 1$.

Observe that

$$\begin{aligned}
\mathsf{rD}(\mathcal{V}, \mathbf{x}, \mathsf{kE}(\mathbf{x}, \alpha)) &= \alpha + \sum_{j=1}^{\ell} \omega_j \cdot 0 = \alpha \\
\mathsf{rD}(\mathcal{V}, \mathbf{x}, \mathsf{rE}(\mathbf{x}, (\mathbf{w}, \mathbf{v}, \mathbf{u}, u_0'))) &= u_0' + \sum_{j=1}^{\ell} \omega_j \left(x_j w_j + v_j \right) = \mathbf{1}^\top \binom{u_0'}{\mathbf{u}} + \sum_{j=1}^{\ell} \omega_j \left(x_j w_j + v_j \right) = \mathsf{sD}(\mathcal{V}, \mathbf{x}, \mathsf{sE}(\mathcal{V}, (\mathbf{w}, \mathbf{v}, \mathbf{u}, u_0')))
\end{aligned}$$

Privacy. Suppose $\mathsf{P}(\mathcal{V}, \mathbf{x}) = 0$. We have

$$\begin{aligned}
\mathsf{sE}(\mathcal{V}, (\mathbf{w}, \mathbf{v}, \mathbf{u}, u_0')) &= (\beta_1 + w_1, \dots, \beta_\ell + w_\ell, \gamma_1 + v_1, \dots, \gamma_\ell + v_\ell) \\
\mathsf{rE}(\mathbf{x}, (\mathbf{w}, \mathbf{v}, \mathbf{u}, u_0')) + \mathsf{kE}(\mathcal{V}, \alpha) &= (u_0' + \alpha, x_1 w_1 + v_1, \dots, x_\ell w_\ell + v_\ell)
\end{aligned}$$

Privacy follows readily from the fact that:

- α is masked by u_0';
- $\{x_j\beta_j + \gamma_j : j \in [\ell]\}$ reveal no information about u_0' since $\mathbf{1}^\top \notin \text{span}\langle x_j \mathbf{y}_j^\top + \mathbf{z}_j^\top \rangle$;
- given $(x_j, x_j\beta_j + \gamma_j)$, we can simulate

  $$(\beta_j + w_j, \gamma_j + v_j, x_j w_j + v_j)$$

  using the relation $x_j w_j + v_j = x_j(\beta_j + w_j) + (\gamma_j + v_j) - (x_j\beta_j + \gamma_j)$.

A.7 Broadcast encryption

We obtain a family of encodings for broadcast encryption parameterized by $t \in [n]$ that yields public parameters size $O(t + n/t)$, ciphertext size $O(t)$ and secret key size $O(n/t)$, generalizing the ideas in [21, 29]. For simplicity of notation, we assume that n is a multiple of t.

Predicate [19]. Here, $\mathcal{X} := \{0,1\}^n, \mathcal{Y} := [n]$ and

$$\mathsf{P}(\mathbf{x}, i) = 1 \text{ iff } x_i = 1$$

That is, $\mathbf{x}$ is the characteristic vector of a subset of $[n]$. It is convenient to rewrite the predicate as follows: $\mathcal{X} := (\{0,1\}^{n/t})^t, \mathcal{Y} := [t] \times [n/t]$ and

$$\mathsf{P}((\mathbf{x}_1, \ldots, \mathbf{x}_t), (i_1, i_2)) = 1 \text{ iff } \mathbf{x}_{i_1}^\top \mathbf{e}_{i_2} = 1$$

where (i_1, i_2) is the unique pair of integers satisfying $i = (i_1 - 1) \cdot n/t + i_2$ and $0 < i_2 \le n/t$, and $(\mathbf{e}_1, \ldots, \mathbf{e}_{n/t})$ is the standard basis of $\mathbb{Z}_q^{n/t}$.

Encoding.

- $\mathcal{W} := \mathbb{Z}_p^t \times \mathbb{Z}_p^{n/t}$.
- $\mathsf{sE}(\mathbf{x}, (\mathbf{w}, \mathbf{u})) := (w_1 + \mathbf{x}_1^\top \mathbf{u}, \ldots, w_t + \mathbf{x}_t^\top \mathbf{u}) \in \mathbb{Z}_p^t$
- $\mathsf{rE}((i_1, i_2), (\mathbf{w}, \mathbf{u})) := w_{i_1} \cdot \mathbf{e}_{i_2} + \mathbf{u} \in \mathbb{Z}_p^{n/t}$
- $\mathsf{kE}((i_1, i_2), \alpha) := \alpha \cdot \mathbf{e}_{i_2} \in \mathbb{Z}_p^{n/t}$
- $\mathsf{sD}(\mathbf{x}, (i_1, i_2), \mathbf{c}) := c_{i_1}$
- $\mathsf{rD}(\mathbf{x}, (i_1, i_2), \mathbf{d}) := \mathbf{x}_{i_1}^\top \mathbf{d}$

Correctness. Suppose $\mathsf{P}(\mathbf{x}, (i_1, i_2)) = 1$. Observe that

$$\begin{aligned} \mathsf{rD}(\mathbf{x}, (i_1, i_2), \mathsf{kE}((i_1, i_2), \alpha)) &= \alpha \cdot \mathbf{x}_{i_1}^\top \mathbf{e}_{i_2} = \alpha \\ \mathsf{rD}(\mathbf{x}, (i_1, i_2), \mathsf{rE}((i_1, i_2), (\mathbf{w}, \mathbf{u}))) &= w_{i_1} + \mathbf{x}_{i_1}^\top \mathbf{u} = \mathsf{sD}(\mathbf{x}, (i_1, i_2), \mathsf{sE}(\mathbf{x}, \mathbf{w})) \end{aligned}$$

Privacy. Privacy follows readily from the fact that:

- For all $j \neq i_1$, $w_j + \mathbf{x}_j^\top \mathbf{u}$ reveals no information about $\mathbf{u}$;
- If $\mathbf{x}_{i_1}^\top \mathbf{e}_{i_2} = 0$, then α is perfectly hidden given $\mathbf{x}_{i_1}$, $(\alpha + w_{i_1}) \cdot \mathbf{e}_{i_2} + \mathbf{u}$, and $w_{i_1} + \mathbf{x}_{i_1}^\top \mathbf{u}$.

A.8 Fuzzy IBE (large universe)

Predicate. [39] Here, $\mathcal{X} := \mathcal{S}_p^n, \mathcal{Y} := \mathcal{S}_p^n \times [n]$ where $\mathcal{S}_p^n$ denote all n-element subsets of $\mathbb{Z}_p$ for some fixed n and

$$\mathsf{P}(X, (Y, k)) = 1 \text{ iff } |X \cap Y| \ge k$$

Note that this is more general than the standard notion of fuzzy IBE since we allow the threshold k to vary with the secret key.

Encoding. The idea is as follows: we apply Shamir's k-out-of-n secret-sharing to α to obtain n shares $(\alpha_1, \ldots, \alpha_n)$. In addition, we pick a random degree $2n - 1$ polynomial (or any $2n$-wise-independent hash function) $T(\cdot)$. Then we set

- $\mathsf{sE}(\{x_1, \ldots, x_n\}, \mathbf{w}) := (T(x_1), \ldots, T(x_n))$
- $\mathsf{rE}((\{y_1, \ldots, y_n\}, k), \mathbf{w}) + \mathsf{kE}((\{y_1, \ldots, y_n\}, k), \alpha) := (T(y_1) + \alpha_1, \ldots, T(y_n) + \alpha_n)$.

Now, whenever $|\{x_1, \ldots, x_n\} \cap \{y_1, \ldots, y_n\}| \ge k$, we recover k of $\alpha_1, \ldots, \alpha_n$, from which we could recover α. Otherwise, at least $n - k + 1$ of the values $T(y_1), \ldots, T(y_n)$ are completely hidden given $T(x_1), \ldots, T(x_n)$, which completely mask $n - k + 1$ of the shares $\alpha_1, \ldots, \alpha_n$, upon which α is completely hidden via the secret-sharing property.

To realize the secret sharing of α for any threshold k, we pick $\mathbf{u} := (u_1,\ldots,u_k) \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k$ and we define

$$Q_{\mathbf{u}}(z) := \sum_{j=1}^{k-1} u_j z^j \text{ and for all } i \in [n],\ \alpha_i := \alpha + Q_{\mathbf{u}}(y_i).$$

For the $2n$-wise-independent hash function, we pick $\mathbf{v} := (v_1,\ldots,v_{2n}) \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{2n}$ and define

$$T_{\mathbf{v}}(z) := \sum_{j=0}^{2n-1} v_j z^j.$$

For any k-element subset $S \subseteq X \cap Y$ and for any $x \in S$, the interpolation polynomials are defined by

$$\Delta_{S,x}(z) := \frac{\prod_{x' \in S, x' \neq x} (z - x')}{\prod_{x' \in S, x' \neq x} (x - x')}.$$

We use the fact that for any polynomial P of degree at most $k-1$, we have $P(z) = \sum_{x \in S} P(x)\Delta_{S,x}(z)$.

- $\mathcal{W} := \mathbb{Z}_p^k \times \mathbb{Z}_p^{2n}$.
- $\mathsf{sE}(X,(\mathbf{u},\mathbf{v})) := (T_{\mathbf{v}}(x_1),\ldots,T_{\mathbf{v}}(x_n)) \in \mathbb{Z}_p^n$.
- $\mathsf{rE}((Y,k),(\mathbf{u},\mathbf{v})) := (T_{\mathbf{v}}(y_1) + Q_{\mathbf{u}}(y_1),\ldots,T_{\mathbf{v}}(y_n) + Q_{\mathbf{u}}(y_n)) \in \mathbb{Z}_p^n$.
- $\mathsf{kE}((Y,k),\alpha) := (\alpha,\ldots,\alpha) \in \mathbb{Z}_p^n$
- $\mathsf{sD}(X,(Y,k),\mathbf{c}) := \sum_{i:x_i \in S} c_i \Delta_{S,x_i}(0)$, where S is a k-element subset s.t. $S \subseteq X \cap Y$.
- $\mathsf{rD}(X,(Y,k),\mathbf{d}) := \sum_{j:y_j \in S} d_j \Delta_{S,y_j}(0)$.

Correctness. Suppose $\mathsf{P}(X,(Y,k)) = 1$ and any k-element subset $S \subseteq X \cap Y$, observe that

$$\mathsf{rD}(X,(Y,k),\mathsf{kE}((Y,k),\alpha)) = \sum_{j:y_j \in S} \alpha \Delta_{S,y_j}(0) = \alpha$$

in the second equality, we use the fact that $\sum_{j:y_j \in S} \alpha \Delta_{S,y_j}(z) = \alpha$ is a constant polynomial, that is, $\sum_{j:y_j \in S} \Delta_{S,y_j}(0) = 1$; and

$$\begin{aligned}\mathsf{rD}(X,(Y,k),\mathsf{rE}((Y,k),(\mathbf{u},\mathbf{v}))) &= \sum_{j:y_j \in S} T_{\mathbf{v}}(y_j)\Delta_{S,y_j}(0) + \sum_{j:y_j \in S} Q_{\mathbf{u}}(y_j)\Delta_{S,y_j}(0)\\ &= \sum_{i:x_i \in S} T_{\mathbf{v}}(x_i)\Delta_{S,x_i}(0)\\ &= \mathsf{sD}(X,(Y,k),\mathsf{sE}(X,(\mathbf{u},\mathbf{v})))\end{aligned}$$

in the second equality, we use the fact that $\sum_{j:y_j \in S} Q_{\mathbf{u}}(y_j)\Delta_{S,y_j}(0) = Q_{\mathbf{u}}(0) = 0$.

Privacy. Privacy follows readily from the fact that $T_{\mathbf{v}}(\cdot)$ is a $2n$-wise-independent hash function and that the secret sharing has threshold k.

B Concrete ABE schemes

We provide a self-contained description of our KP-ABE and CP-ABE schemes for boolean span programs obtained by instantiating our main construction in Section 6 with our encoding in Section A.5; note that we removed terms from mpk that are not required for encryption.

B.1 KP-ABE scheme

$\mathsf{Setup}(1^\lambda, 1^\ell)$: On input $(1^\lambda, 1^\ell)$, sample

$$(\mathbf{A}, \mathbf{a}^\perp), (\mathbf{B}, \mathbf{b}^\perp) \leftarrow \mathcal{D}_k,\ \mathbf{W}_1, \ldots, \mathbf{W}_\ell, \mathbf{U}_2, \ldots, \mathbf{U}_{\ell'} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(k+1)\times(k+1)},\ \mathbf{k} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{k+1}$$

and output the master public and secret key pair

$$\mathsf{mpk} := \left([\mathbf{A}]_1,\ \left[\mathbf{W}_1^\top \mathbf{A}\right]_1,\ \ldots,\ \left[\mathbf{W}_\ell^\top \mathbf{A}\right]_1,\ e([\mathbf{A}]_1, [\mathbf{k}]_2) \right) \in (G_1^{(k+1)\times k})^{\ell+1} \times G_T^k$$

and

$$\mathsf{msk} := (\ \mathbf{k},\ \mathbf{B}, \mathbf{W}_1,\ \ldots,\ \mathbf{W}_\ell,\ \mathbf{U}_2,\ \ldots,\ \mathbf{U}_{\ell'}\ ) \in \mathbb{Z}_p^{(k+1)} \times \mathbb{Z}_p^{(k+1)\times k} \times (\mathbb{Z}_p^{(k+1)\times(k+1)})^{\ell+\ell'-1}.$$

$\mathsf{Enc}(\mathsf{mpk}, \mathbf{x}, m)$: On input an attribute vector $\mathbf{x} := (x_1, \ldots, x_\ell) \in \{0,1\}^\ell$ and $m \in G_T$, pick $\mathbf{s} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k$ and output

$$\mathsf{ct}_{\mathbf{x}} := \Big(C_0 := [\mathbf{A}\mathbf{s}]_1,\ C_1 := \left[x_1 \mathbf{W}_1^\top \mathbf{A}\mathbf{s}\right]_1,\ \ldots,\ C_\ell := \left[x_\ell \mathbf{W}_\ell^\top \mathbf{A}\mathbf{s}\right]_1,\ C' := e(g_1, g_2)^{\mathbf{k}^\top \mathbf{A}\mathbf{s}} \cdot m \Big) \in (G_1^{k+1})^{\ell+1} \times G_T.$$

$\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathbf{M})$: On input a boolean span program $\mathbf{M}$, where $\mathbf{M} \in \mathbb{Z}_p^{\ell \times \ell'}$, pick $\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k$ and output

$$\mathsf{sk}_{\mathbf{M}} := \left(\begin{array}{l} K_0 := [\mathbf{B}\mathbf{r}]_2 \\ K_1 := \left[(\mathbf{k} \| \mathbf{U}_2\mathbf{B}\mathbf{r} \| \cdots \| \mathbf{U}_{\ell'}\mathbf{B}\mathbf{r})\mathbf{M}_1^\top + \mathbf{W}_1\mathbf{B}\mathbf{r}\right]_2 \\ \qquad\qquad \vdots \\ K_\ell := \left[(\mathbf{k} \| \mathbf{U}_2\mathbf{B}\mathbf{r} \| \cdots \| \mathbf{U}_{\ell'}\mathbf{B}\mathbf{r})\mathbf{M}_\ell^\top + \mathbf{W}_\ell\mathbf{B}\mathbf{r}\right]_2 \end{array} \right) \in (G_2^{k+1})^{\ell+1}.$$

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_{\mathbf{M}}, \mathsf{ct}_{\mathbf{x}})$: If $\mathbf{x}$ satisfies $\mathbf{M}$, compute $\omega_1, \ldots, \omega_\ell \in \mathbb{Z}_p$ such that

$$\sum_{j:x_j=1} \omega_j \mathbf{M}_j = \mathbf{1}.$$

Then, compute

$$e(g_1, g_2)^{\mathbf{k}^\top \mathbf{A}\mathbf{s}} \leftarrow e(C_0, \prod_{j:x_j=1} K_j^{\omega_j}) \cdot e(\prod_{j:x_j=1} C_j^{-\omega_j}, K_0),$$

and recover the message as $m \leftarrow C' / e(g_1, g_2)^{\mathbf{k}^\top \mathbf{A}\mathbf{s}} \in G_T$.

B.2 CP-ABE scheme

$\mathsf{Setup}(1^\lambda, 1^\ell)$: On input $(1^\lambda, 1^\ell)$, sample

$$(\mathbf{A}, \mathbf{a}^\perp), (\mathbf{B}, \mathbf{b}^\perp) \leftarrow \mathcal{D}_k,\ \mathbf{W}_1, \ldots, \mathbf{W}_\ell, \mathbf{V} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(k+1)\times(k+1)},\ \mathbf{k} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{k+1}$$

and output the master public and secret key pair

$$\mathsf{mpk} := \left([\mathbf{A}]_1,\ \left[\mathbf{W}_1^\top \mathbf{A}\right]_1,\ \ldots,\ \left[\mathbf{W}_\ell^\top \mathbf{A}\right]_1,\ \left[\mathbf{V}^\top \mathbf{A}\right]_1,\ e([\mathbf{A}]_1, [\mathbf{k}]_2) \right) \in (G_1^{(k+1)\times k})^{\ell+2} \times G_T^k$$

and

$$\mathsf{msk} := (\ \mathbf{k},\ \mathbf{B},\ \mathbf{W}_1,\ \ldots,\ \mathbf{W}_\ell,\ \mathbf{V}\ ) \in \mathbb{Z}_p^{(k+1)} \times \mathbb{Z}_p^{(k+1)\times k} \times (\mathbb{Z}_p^{(k+1)\times(k+1)})^{\ell+1}.$$

$\mathsf{Enc}(\mathsf{mpk}, \mathbf{M}, m)$: On input a boolean span program $\mathbf{M}$, where $\mathbf{M} \in \mathbb{Z}_p^{\ell \times \ell'}$, and a message $m \in G_T$, pick[5] $\mathbf{U}_2, \ldots, \mathbf{U}_{\ell'} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(k+1)\times(k+1)}$, $\mathbf{s} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k$ and output

$$\mathsf{ct}_{\mathbf{M}} := \begin{pmatrix} C_0 := [\mathbf{As}]_1 \\ C_1 := \left[(\mathbf{V}^\top\mathbf{As} \| \mathbf{U}_2^\top\mathbf{As} \| \cdots \| \mathbf{U}_{\ell'}^\top\mathbf{As})\mathbf{M}_1^\top + \mathbf{W}_1^\top\mathbf{As}\right]_1 \\ \vdots \\ C_\ell := \left[(\mathbf{V}^\top\mathbf{As} \| \mathbf{U}_2^\top\mathbf{As} \| \cdots \| \mathbf{U}_{\ell'}^\top\mathbf{As})\mathbf{M}_\ell^\top + \mathbf{W}_\ell^\top\mathbf{As}\right]_1 \\ C' := e(g_1, g_2)^{\mathbf{k}^\top\mathbf{As}} \cdot m \end{pmatrix} \in (G_1^{k+1})^{\ell+1} \times G_T.$$

$\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathbf{x})$: On input an attribute vector $\mathbf{x} := (x_1, \ldots, x_\ell) \in \{0,1\}^\ell$, pick $\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k$ and output

$$\mathsf{sk}_{\mathbf{x}} := \Big(\, K_0 := [\mathbf{Br}]_2\,,\; K_1 := [x_1\mathbf{W}_1\mathbf{Br}]_2\,,\; \ldots,\; K_\ell := [x_\ell\mathbf{W}_\ell\mathbf{Br}]_2\,,\; K_{\ell+1} := [\mathbf{k} + \mathbf{VBr}]_2\,,\; \Big) \in (G_2^{k+1})^{\ell+2}.$$

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_{\mathbf{x}}, \mathsf{ct}_{\mathbf{M}})$: If $\mathbf{x}$ satisfies $\mathbf{M}$, compute $\omega_1, \ldots, \omega_\ell \in \mathbb{Z}_p$ such that

$$\sum_{j : x_j = 1} \omega_j \mathbf{M}_j = \mathbf{1}.$$

Then, compute

$$e(g_1, g_2)^{\mathbf{k}^\top\mathbf{As}} \leftarrow e(C_0, K_{\ell+1} \cdot \prod_{j : x_j = 1} K_j^{\omega_j}) \cdot e(\prod_{j : x_j = 1} C_j^{-\omega_j}, K_0),$$

and recover the message as $m \leftarrow C' / e(g_1, g_2)^{\mathbf{k}^\top\mathbf{As}} \in G_T$.

C Concrete weakly attribute-hiding ZIPE scheme

We provide a self-contained description of our weakly attribute-hiding ZIPE scheme, obtained by instantiating our main construction in Section 7 with our encoding in Section A.1.

$\mathsf{Setup}(1^\lambda, 1^n)$: On input $(1^\lambda, 1^n)$, sample

$$(\mathbf{A}, \mathbf{a}^\perp), (\mathbf{B}, \mathbf{b}^\perp) \leftarrow \mathcal{D}_k,\; \mathbf{U}, \mathbf{W}_1, \ldots, \mathbf{W}_n \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{(k+1)\times(k+1)},\; \mathbf{k} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^{k+1}$$

and output the master public and secret key pair

$$\mathsf{mpk} := \big(\, [\mathbf{A}]_1\,,\; \left[\mathbf{U}^\top\mathbf{A}\right]_1,\; \left[\mathbf{W}_1^\top\mathbf{A}\right]_1,\; \ldots,\; \left[\mathbf{W}_n^\top\mathbf{A}\right]_1,\; e([\mathbf{A}]_1, [\mathbf{k}]_2)\,\big) \in (G_1^{(k+1)\times k})^{n+2} \times G_T^k$$

and

$$\mathsf{msk} := (\,\mathbf{k},\, \mathbf{B},\, \mathbf{U},\, \mathbf{W}_1,\, \ldots,\, \mathbf{W}_n\,) \in \mathbb{Z}_p^{(k+1)} \times \mathbb{Z}_p^{(k+1)\times k} \times (\mathbb{Z}_p^{(k+1)\times(k+1)})^{n+1}.$$

$\mathsf{Enc}(\mathsf{mpk}, \mathbf{x}, m)$: On input an attribute vector $\mathbf{x} := (x_1, \ldots, x_n) \in \mathbb{Z}_p^n$ and $m \in G_T$, pick $\mathbf{s} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k$ and output

$$\mathsf{ct}_{\mathbf{x}} := \Big(\, C_0 := [\mathbf{As}]_1\,,\; C_1 := \left[(x_1\mathbf{U}^\top + \mathbf{W}_1^\top)\mathbf{As}\right]_1,\; \ldots,\; C_n := \left[(x_n\mathbf{U}^\top + \mathbf{W}_n^\top)\mathbf{As}\right]_1,\; C' := e(g_1, g_2)^{\mathbf{k}^\top\mathbf{As}} \cdot m \Big) \in (G_1^{k+1})^{n+1} \times G_T.$$

$\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \mathbf{y})$: On input an attribute vector $\mathbf{y} := (y_1, \ldots, y_n) \in \mathbb{Z}_p^n$, pick $\mathbf{r} \leftarrow_{\mathrm{R}} \mathbb{Z}_p^k$ and output

$$\mathsf{sk}_{\mathbf{y}} := \big(\, K_0 := [\mathbf{Br}]_2\,,\; K_1 := \left[\mathbf{k} + (y_1\mathbf{W}_1 + \cdots + y_n\mathbf{W}_n)\mathbf{Br}\right]_2 \big) \in (G_2^{k+1})^2.$$

[5] Since there is only one ciphertext in the security experiment, having Enc instead of Setup pick $\mathbf{U}_2, \ldots, \mathbf{U}_{\ell'}$ yields exactly the same distribution.

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_{\mathbf{y}}, \mathsf{ct}_{\mathbf{x}})$: Compute

$$e(g_1, g_2)^{\mathbf{k}^\top \mathbf{A}\mathbf{s}} \leftarrow e(C_0, K_1) \cdot e(\prod_{j=1}^{n} C_j^{-y_j}, K_0),$$

and recover the message as $m \leftarrow C' / e(g_1, g_2)^{\mathbf{k}^\top \mathbf{A}\mathbf{s}} \in G_T$.